



ระบบตรวจสอบการเข้าออกอาคารผ่านเครือข่าย

BUILDING ACCESS SYSTEM VIA LAN



นายชินบุรณ เอี่ยมฉิม รหัส 44370120

นายณัฐพงษ์ ลำเนียงงาม รหัส 44370153

ห้องสมุดคณะวิศวกรรมศาสตร์  
วันที่รับ..... 7 / มี.ย. 2553 / .....

เลขทะเบียน..... 14942991

เลขเรียกหนังสือ..... ฟร.

มหาวิทยาลัยนเรศวร 8566 6

2547

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2547



## ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	ระบบตรวจสอบการเข้าออกอาคารผ่านเครื่องถ่าย
ผู้เสนอโครงการ	นายชินบุรณ เอี่ยมฉิม รหัส 44370120 นายณัฐพงษ์ สำเนียงงาม รหัส 44370153
อาจารย์ที่ปรึกษา	ผศ. ดร.ชงยุทธ ชนบดีเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2547

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครสวรรค์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์  
คณะกรรมการสอบ โครงการวิศวกรรม

..... ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ชงยุทธ ชนบดีเฉลิมรุ่ง)

..... กรรมการ

(ดร.สุรเชษฐ์ กานต์ประชา)

..... กรรมการ

(นายพนัส นัถฤทธิ)

หัวข้อโครงการ	ระบบตรวจสอบการเข้าออกอาคารผ่านเครือข่าย
ผู้ดำเนินโครงการ	นายชินบุรณ เอี่ยมฉิม รหัส 44370120
	นายณัฐพงษ์ สำเนียงงาม รหัส 44370153
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ศร.รอยบุทร ชนบทติเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2547

### บทคัดย่อ

โครงการนี้จัดทำขึ้นเพื่อพัฒนาระบบตรวจสอบการเข้า-ออกอาคาร ให้สามารถเข้า-ออกได้ในสถานที่ต่างๆภายในมหาวิทยาลัย รวมทั้งมีเครื่องที่ใช้เป็น Database Server ที่ใช้ในการบันทึกเก็บข้อมูลของบุคคลที่เข้าใช้อาคาร โดยผ่านระบบเครือข่ายของมหาวิทยาลัยและโครงการนี้ยังได้พัฒนาแก้ไขเมื่อเกิดการชนกันของข้อมูลที่ตัวแปลง RS485 ไปเป็น RS232 ไม่ให้เกิดการ error ขึ้นที่โปรแกรม โดยเมื่อเกิดการชนของข้อมูลแล้วโปรแกรมจะทำการ restart ตัวเองขึ้นมาใหม่ทำให้ใช้งานได้อย่างต่อเนื่อง

ในส่วน Database ได้ใช้ SQL Server 7.0 ในการเก็บข้อมูลของบุคคลที่เข้าใช้อาคารโดย SQL Server 7.0 เป็นโปรแกรมแบบ Multiuser relational database management system ที่ใช้ในการสร้างระบบฐานข้อมูลแบบ Client-Server ที่มีประสิทธิภาพและเสถียรภาพสูง

ผลที่ได้จากการทดลองนี้ สามารถแก้ปัญหาการชนกันของข้อมูลได้ และรับ-ส่งข้อมูลได้ในระยะไกล ภายใต้ ระบบ Network ของมหาวิทยาลัยนเรศวร

<b>Project Title</b>	BUILDING ACCESS SYSTEM VIA LAN		
<b>Name</b>	Mr.Chinaboon Iamchim	ID	44370120
	Mr.Natthapong Samneangngam	ID	44370153
<b>Project Advisor</b>	Yongyut Chonbodeechalermsroong, Assist. Prof., Ph. D.		
<b>Major</b>	Computer Engineering		
<b>Department</b>	Electrical and Computer Engineering		
<b>Academic Year</b>	2547		

---

### ABSTRACT

This project is to improve the previous project, "room and building access system" in order to be able to access many buildings in larger area such as in Naresuan University. It includes a database server for recording data about each person entering each building. This project also resolves the problem in data collision in the RS485-to-RS232 converter by restarting the program itself for resuming the work.

In database system, we use SQL Server 7.0 to store data each person. SQL Server 7.0 is a "Multiuser relational database management system" for constructing Client-Server database with high efficiency and reliability.

The result from experiment shows that it can resolve the collision problem and can receive and transmit data in Naresuan University Local Area Network.

---

---



## กิตติกรรมประกาศ

โครงการวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่  
ปรึกษาโครงการคือ ผู้ช่วยศาสตราจารย์ ดร.ขงยุทธ ชนบดีเฉลิมรุ่ง ที่คอยให้ความช่วยเหลือในทุก  
ๆ ด้าน ในโอกาสนี้ทางคณะผู้จัดทำโครงการจึงขอขอบคุณทุก ๆ ท่านที่มีส่วนช่วยทำให้โครงการนี้  
ประสบความสำเร็จได้ด้วยดี

นายชินบูรณ์ เอี่ยมฉิม  
นายรัฐพงษ์ สำเนียงงาม



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
<b>บทที่ 1 บทนำ</b>	
1.1 ความเป็นมาของโครงการ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
1.6 งบประมาณ.....	2
<b>บทที่ 2 หลักการและทฤษฎี</b>	
2.1 บาร์โค้ด.....	3
2.2 พอร์ตอนุกรม (Serial Port).....	7
2.3 เน็ตเวิร์ก (Network).....	15
2.4 ระบบฐานข้อมูล (Database System).....	27
2.5 เทคโนโลยีการติดต่อกับฐานข้อมูลของ Microsoft.....	38
2.6 ไมโครคอนโทรลเลอร์ MCS-51.....	39

## สารบัญ (ต่อ)

	หน้า
บทที่ 3 ขั้นตอนการดำเนินโครงการ	
3.1 ขั้นตอนการดำเนินงาน.....	41
3.2 ภาพรวมของระบบตรวจสอบการเข้า-ออกอาคารผ่านเครือข่าย.....	41
3.3 การทำงานของระบบตรวจสอบการเข้า-ออกอาคารผ่านเครือข่าย.....	44
3.4 การออกแบบ Database.....	51
3.5 การติดต่อ Database SQL Server ของ Visual Basic.....	54
3.6 การติดต่อกับพอร์ตต่าง ๆ ของ MCS-51 AT89C2051.....	56
3.7 การทำงานของชุดควบคุม.....	60
บทที่ 4 ผลการทดลอง	
4.1 ผลการทดลอง.....	62
บทที่ 5 สรุปผลการทดลองโครงการ ปัญหา และข้อเสนอแนะ	
5.1 สรุปผลการทดลองโครงการ.....	71
5.2 ปัญหาและข้อเสนอแนะ.....	71
เอกสารอ้างอิง.....	72
ภาคผนวก	
ภาคผนวก ก.....	74
ภาคผนวก ข.....	95
ประวัติผู้เขียนโครงการ.....	126

## สารบัญตาราง

ตารางที่	หน้า
2.1 อักขระของรหัส Code 39.....	4
2.2 ลักษณะของ D – Type.....	9
2.3 เปรียบเทียบค่าต่างๆของสายเน็ตเวิร์กโดยทั่วไป.....	25
2.4 แสดงคีย์หลัก (Primary Key) และคีย์สำรอง (Alternate Key).....	31
2.5 แสดงคีย์คีย์นอก (Foreign Key).....	32



## สารบัญรูป

รูปที่	หน้า
2.1 Code39 เป็นบาร์โค้ดที่ใช้กันอย่างกว้างขวาง.....	6
2.2 EAN8 ใช้สำหรับรหัสที่มีความยาว 5 หลัก.....	6
2.3 EAN 13 ใช้กับข้อความที่มีความยาว 10 หลัก.....	6
2.4 แผนผังของสายสัญญาณในพอร์ตอนุกรม.....	11
2.5 แสดงวิธีการรับ – ส่ง ข้อมูลแบบ RS 422 ในอุดมคติ.....	12
2.6 ลักษณะการต่อสาย RS 422 แบบ Full Duplex .....	12
2.7 ลักษณะการต่อสาย RS 422 แบบ Simplex.....	13
2.8 แสดงตัวอย่าง Specification ของสายสัญญาณที่ใช้กับระบบ RS 422/485.....	14
2.9 เน็ตเวิร์ก.....	15
2.10 Peer to Peer Network.....	16
2.11 Client – Server Network.....	17
2.12 Bus Network.....	18
2.13 Ring Network.....	20
2.14 Star Network.....	20
2.15 ภายในสาย โคอแฉีก.....	21
2.16 หัวแบบ BNC และสายโคแฉีก.....	22
2.17 อุปกรณ์เชื่อมต่อสาย โคอแฉีก BNC barrel connector ใช้ในกรณีที่ต้องการต่อสายให้ยาวขึ้น.....	22
2.18 BNCT Connector เชื่อมต่อระหว่างสายเคเบิลกับเน็ตเวิร์กการ์ด.....	22
2.19 BNC Terminat อุปกรณ์สำหรับปิดหัวและท้ายสายเคเบิล.....	22
2.20 UTP.....	23
2.21 สาย UTP.....	24
2.22 RJ45 คอนเน็กเตอร์.....	24
2.23 Fiber Optic.....	24
2.24 บริดจ์.....	27
3.1 การทำงานของระบบเข้า-ออกอาคารผ่านเครือข่าย.....	42
3.2 Flowchart ทำงานการเข้าอาคาร.....	44
3.3 Flowchart การทำงานการออกอาคาร.....	46
3.4 Flowchart การทำงานเมื่อเกิดการชนกันของข้อมูล (Error).....	47

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.5 Flowchart ฟังก์ชันการทำงานผ่าน network.....	49
3.6 การออกแบบตาราง Admins.....	51
3.7 การออกแบบตาราง Users.....	51
3.8 การออกแบบตาราง Door.....	52
3.9 การออกแบบตาราง Event.....	52
3.10 ER-Diagram ของตาราง Admins, Users, Event.....	53
3.11 ER-Diagram ของตารางDoor.....	53
3.12 การติดต่อ Database SQL Severของ Visual Basic.....	54
3.13 แสดงวงจรการติดต่อกับพอร์ทต่างๆ ของ MCS-51.....	56
3.14 วงจรพิมพ์ของบอร์ดควบคุม.....	57
3.15 บอร์ดชุดแปลงสัญญาณ.....	58
3.16 วงจรของชุดแปลงสัญญาณ RS 232-422/485.....	59
3.17 การทำงานของชุดควบคุม.....	60
4.1 Monitor.....	62
4.2 Login.....	63
4.3 การออกจากโปรแกรม.....	63
4.4 Administrator.....	64
4.5 Add Admin.....	64
4.6 Add User.....	65
4.7 Report.....	65
4.8 Search.....	66
4.9 การ Restart ตัวเองของโปรแกรม.....	66
4.10 ตาราง Event.....	67
4.11 ตาราง Admins.....	68
4.12 ตาราง Users.....	68
4.13 ตาราง Doors.....	69
4.14 การทดลองผ่าน network.....	70

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาของโครงการ

เนื่องจากในปัจจุบันคนจำนวนมากได้มีการเข้า-ออกอาคารและห้องทำงานต่างๆ ไม่เว้นแต่  
ละวัน จนบางครั้งทำให้เกิดความเสียหายของข้อมูล ข่าวสารและทรัพย์สินที่จำเป็น จึงทำให้เกิด  
แนวความคิดที่จะนำระบบควบคุมการเข้า-ออกอาคารและห้องทำงาน ด้วยวิธีการใช้บาร์โค้ดโดย  
ผ่านระบบ LAN (Local Area Network) มาป้องกันข้อมูล ข่าวสารและทรัพย์สิน รวมทั้งอำนวยความสะดวก  
ความสะดวกให้แก่ผู้ใช้งาน โดยการใส่บัตรในการเข้า-ออก และป้อนรหัสลับอีกทีหนึ่ง ข้อมูลจะถูก  
เก็บไว้ในฐานข้อมูล ซึ่งสามารถตรวจสอบข้อมูลได้ตลอดเวลาว่าได้มีการเข้า-ออก จากอาคารและ  
ห้องทำงาน วันไหน เวลาเท่าใด

### 1.2 วัตถุประสงค์

1. เพื่อช่วยในการรักษาความปลอดภัยของข้อมูลข่าวสารทรัพย์สินในอาคารและห้อง ต่างๆ
2. สามารถเข้า-ออกอาคารทั่วทั้งมหาวิทยาลัยผ่านเครือข่ายได้
3. ช่วยให้สามารถสืบค้นข้อมูลของผู้ที่เข้ามาใช้อาคารและห้อง ได้
4. สามารถแก้ปัญหาการชนกันของข้อมูลเมื่อมีการรูดบัตรพร้อมหลายตำแหน่ง

### 1.3 ขอบเขตของโครงการ

โครงการนี้มีเป้าหมายในการที่จะสร้างระบบตรวจสอบการเข้า-ออกอาคาร โดยมีหัวข้อที่จะ  
ศึกษาดังนี้

1. แก้ปัญหาการชนกันของข้อมูลเมื่อเวลารูดบัตรพร้อมกันหลายตำแหน่งในเครื่อง  
คอมพิวเตอร์เครื่องเดียว
2. ใช้บาร์โค้ดโดยผ่านระบบเครือข่าย-Lan(Local-Area Network)
3. การออกแบบระบบฐานข้อมูล และการทำงานของระบบฐานข้อมูล

## 1.4 ขั้นตอนการดำเนินงาน

กิจกรรม	2546		2547										
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	
1. เขียน โครงร่างการทำงาน	↔												
2. ศึกษาและรวบรวมข้อมูล			↔										
3. พัฒนาโปรแกรมและระบบฐานข้อมูล						↔							
4. ออกแบบและติดตั้งระบบ LAN								↔					
5. ทดลองและวิเคราะห์งาน									↔				
6. ปรับปรุงและแก้ไข โครงการให้เสร็จสมบูรณ์											↔		
7. ทำปริญญานิพนธ์													↔

## 1.5 ผลที่คาดว่าจะได้รับ

- เข้าใจการทำงานของระบบ LAN และติดตั้งระบบ LAN ได้
- แก้ไขปัญหาเฉพาะหน้าได้
- รู้จักการใช้โปรแกรมเพื่อควบคุมการทำงานของคอมพิวเตอร์และ

ไมโครคอนโทรลเลอร์

- ได้รับความรู้ในด้านการทำงานระบบฐานข้อมูล และออกแบบระบบฐานข้อมูล
- สามารถนำไปประยุกต์ใช้งานได้จริง

## 1.6 งบประมาณ

1. ค่าวัสดุอุปกรณ์

2,000 บาท

รวมค่าใช้จ่าย 2,000 บาท (สองพันบาทถ้วน)

หมายเหตุ: ทุกรายการเฉลี่ย



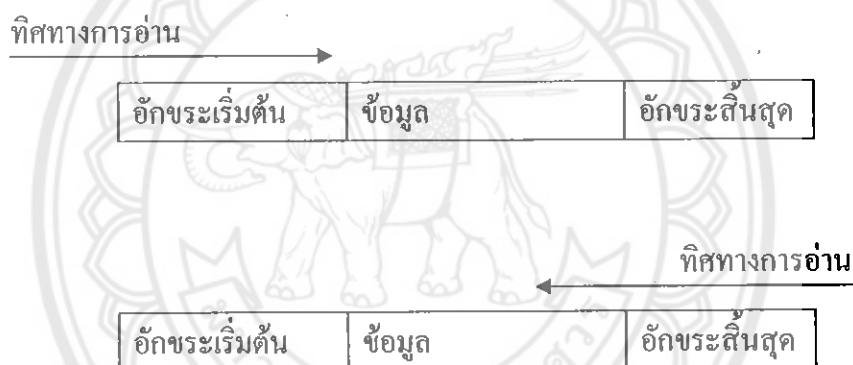
## บทที่ 2

### หลักการและทฤษฎี

#### 2.1 บาร์โค้ด

บาร์โค้ด เป็นการแสดงของข้อมูลในลักษณะแทนความหมายต่างๆ เช่น ตามกล่องสินค้า แต่ละชนิด บัตรนิสิต นักศึกษา เป็นต้น

บาร์โค้ดในปัจจุบันมีอยู่มากมายหลายมาตรฐาน แต่จะต้องมีสิ่งหนึ่งที่เหมือนกันคือ จะต้องเริ่มต้นด้วยอักขระเริ่มต้น และลงท้ายด้วยอักขระสิ้นสุด อักขระเริ่มต้นจะมีไว้ให้เครื่องอ่าน บาร์โค้ด รู้ว่าอักขระต่อจากนี้คือ ข้อมูล เช่น รหัสนิสิต โดยจะเป็นข้อมูลไปเรื่อยๆจนกว่าจะถึงอักขระสิ้นสุด และอักขระเริ่มต้นจะเป็นอักขระสิ้นสุดได้ เพราะการอ่านสามารถอ่านได้ 2 ทิศทาง ดังนี้



บาร์โค้ดแบ่งเป็น 2 ประเภท

1. แบบ 1Dimension หรือ Linear Barcode

เป็นแบบปกติทั่วไปของบาร์โค้ด เป็นการนำเอาตัวอักษร หรือตัวเลข มาแสดงในรูปแบบแท่ง

เช่น

- UPC/EAN/JAN

- Code 128

- Code 39

- Code 93

2. แบบ 2Dimension มีใช้กันไม่มากนัก เพราะต้องลงทุนสูง ทั้ง เครื่องพิมพ์และเครื่องอ่าน

แต่สามารถที่จะประหยัดเนื้อที่ในการแสดงแทนรหัส เช่น

- PDF 417

- Code 49

## Code 39

ตารางที่ 2.1 อักขระของรหัส Code 39

ตัวอักษร	เลขฐานสอง	Check-sum
0	000110100	0
1	100100001	1
2	001100001	2
3	101100000	3
4	000110001	4
5	100110000	5
6	001110000	6
7	000100101	7
8	100100100	8
9	001100100	9
A	100001001	10
B	001001001	11
C	101001000	12
D	000011001	13
E	100011000	14
F	001011000	15
G	000001101	16
H	100001100	17
I	001001100	18
J	000011100	19
K	100000011	20
L	001000011	21
M	101000010	22
N	000010011	23
O	100010010	24
P	001010010	25
Q	000000111	26

ตารางที่ 2.1 (ต่อ)

R	100000110	27
S	001000110	28
T	000010110	29
U	110000001	30
V	011000001	31
W	111000000	32
X	010010001	33
Y	110010000	34
Z	011010000	35
-	010000101	36
.	110000100	37
SPACE	011000100	38
*	010010100	-
\$	010101000	39
/	010100010	40
+	010001010	41
%	000101010	42

## ตัวอย่าง การเข้ารหัส Code 39

กำหนดให้รหัสของข้อความ 98PQ

- ขั้นแรกหาผลรวมของค่า check-sum ของอักขระทุกตัวในข้อความนั้น คือ

$$98PQ (9+8+25+26 = 68)$$

- หารผลรวมที่ได้ด้วย 43 ( $68/43 = 1$  เศษ 25)

- ต่อมาให้ไปดูตาราง ว่าอักขระใดที่มีค่า check-sum เท่ากับ 25 ซึ่งอักขระที่ได้ คือ ตัว P

- ดังนั้นข้อความก็จะถูกแปลงเป็นรหัส Code39 รวมทั้ง check-sum และอักขระเริ่มต้นและสิ้นสุดด้วย ดังนี้ \*98PQP\*

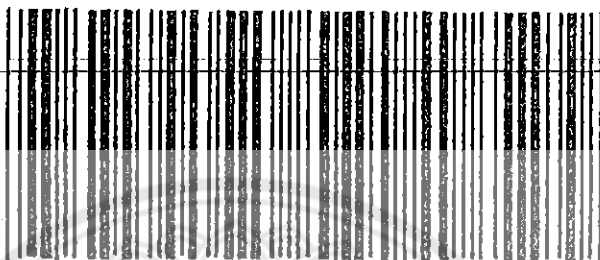
- จากนั้นนำข้อความที่ได้ มาแปลงเป็นรหัสไบนารีจะได้ดังนี้

$$*98PQP* \implies 010010100 / 0 / 001100100 / 0 / 100100100 / 0 / 001010010 / 0 /$$

$$000000111 / 0 / 010010100$$

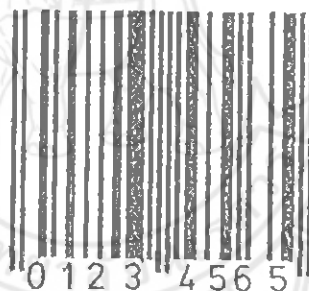
6. แล้วนำเลขไบนารีที่ได้มาแทนด้วยแถบเส้นหรือช่องว่าง โดยให้ไบนารี “0” แทน ด้วยแถบเส้นหรือช่องว่างที่แคบ (narrow) และไบนารี “1” แทนด้วยส่วนที่กว้าง (wide) จะได้บาร์โค้ดตามความต้องการ

หมายเหตุ อักขระแต่ละตัวในข้อความหนึ่งๆ จะถูกแยกจากกันด้วยช่องว่างแคบๆ (narrow space) ซึ่งมีค่าไบนารีเป็น “0”



012 3456789

รูปที่ 2.1 Code39 เป็นบาร์โค้ดที่ใช้กันอย่างกว้างขวาง



0 123 456 5

รูปที่ 2.2 EAN8 ใช้สำหรับรหัสที่มีความยาว 5 หลัก



0 12 3456 7 890 1 2

รูปที่ 2.3 EAN 13 ใช้กับข้อความที่มีความยาว 10 หลัก

## 2.2 พอร์ตอนุกรม (Serial Port)

### 2.2.1 พื้นฐานการสื่อสารแบบอนุกรม

หลังจากที่ได้ศึกษาการทำงานและสร้างแอปพลิเคชันที่ทำงานผ่านพอร์ตขนานไปแล้วนั้น ต่อไปก็ถึงคิวของการติดต่อและควบคุมผ่านพอร์ตอนุกรม หรือ Serial Port โดยจะนำเสนอผ่านการเขียนโปรแกรมควบคุมด้วย Visual Basic เพื่อควบคุมพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ โดยจะอธิบายในรูปแบบที่ง่ายต่อการทำความเข้าใจ โดยหลีกเลี่ยงรายละเอียดด้านเทคนิคใดๆ ซึ่งน่าจะช่วยให้ผู้อ่านเกิดความเข้าใจในหลักการการทำงาน และสามารถประยุกต์ใช้งานได้เอง ส่วนท่านผู้อ่านที่มีพื้นฐานคืออยู่แล้วก็จะสามารถประยุกต์ใช้งานได้หลากหลายยิ่งขึ้น

ถึงแม้ว่าการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ครั้งละหลายๆ บิตพร้อมกันส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน

แต่ว่าการส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนานคือ การสามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรมสามารถแบ่งออกเป็น 3 รูปแบบดังนี้

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

นอกจากนี้แล้วยังสามารถแบ่งประเภทของการสื่อสารแบบอนุกรมตามลักษณะสัญญาณในการส่งได้อีก 2 แบบคือ

**การสื่อสารแบบซิงโครนัส (Synchronous)** สำหรับการสื่อสารแบบซิงโครนัสนี้จะใช้สัญญาณนาฬิกาควบคุมการรับส่งสัญญาณ เช่น สายเคเบิลคอมพิวเตอร์ โดยจะมีสายสัญญาณเส้นหนึ่งเป็นสายสัญญาณนาฬิกา ส่วนอีกเส้นหนึ่งเป็นสายของข้อมูล (และมักจะมีสายกราวด์ด้วย)

สำหรับการสื่อสารแบบซิงโครนัสนี้เหมาะสำหรับการทำงานในระยะใกล้ ข้อมูลที่จะส่งมีไม่มากนัก เพราะถ้าระยะทางไกลขึ้นจะทำให้สัญญาณนาฬิกามีปัญหา อีกทั้งต้องมีสายหลายเส้นทำให้สิ้นเปลืองมาก

**การสื่อสารแบบอะซิงโครนัส (Asynchronous)** สำหรับการสื่อสารแบบอะซิงโครนัส กำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล, ส่วนไหนเป็นตัวข้อมูล, ส่วนไหนจะเป็นส่วนตรวจสอบความถูกต้องข้อมูล และส่วนไหนเป็นส่วนบิตท้ายของข้อมูล โดยต้องกำหนดให้สัญญาณนาฬิกา

เท่ากันทั้งภาคส่ง และภาครับ ซึ่งจะมีอุปกรณ์พิเศษที่ชื่อว่า UART หรือ Universal Asynchronous Receiver / Transmitter คอยควบคุมการรับและส่งข้อมูล

สำหรับมาตรฐานของการส่งข้อมูลแบบอนุกรมอีกแบบที่ได้รับความนิยมอย่างสูงตั้งแต่อดีตถึงปัจจุบันโดยใช้งานกันอย่างแพร่หลายทั้งการสื่อสารและการควบคุมทางอุตสาหกรรมนั่นก็คือ มาตรฐาน RS – 232C

### 2.2.2 รู้จักกับมาตรฐาน RS – 232C

มาตรฐาน RS – 232C เป็นมาตรฐานที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกรณ์ต่อพ่วงจากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมา แต่มาตรฐานที่ได้รับความนิยมและใช้กันกว้างขวางมากที่สุดคือมาตรฐาน RS-232C ซึ่งถูกประกาศใช้ในปี 1969 โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association: EIA) ในยุคแรกๆการอินเตอร์เฟสแบบ RS-132C ถูกออกแบบสำหรับเชื่อมต่อเทอร์มินอล(DTE : Data Terminal Equipment) กับ โมเด็ม(DCE : Data Communication Equipment) ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเส้นเดียวกัน

มาตรฐาน RS – 232C ได้แบ่งอุปกรณ์ออกเป็น 2 ประเภท ซึ่งอุปกรณ์ทั้งสองประเภทนี้ก็คือ

1. อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (เอาต์พุต)
2. อุปกรณ์ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (อินพุต)

ตามมาตรฐาน RS-232C แล้วคอนเนกเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเนกเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งคอนเนกเตอร์ที่นิยมใช้กันอยู่จะมีชนิด D-Type แบบ 9 ขา และแบบ 25 ขา โดยจะติดตั้งอยู่หลังเครื่องคอมพิวเตอร์ ระดับแรงดันจะมีค่าระหว่าง  $-3\text{ V}$  ถึง  $-15\text{ V}$

สำหรับลอจิก High และลอจิก Low จะมีระดับแรงดันระหว่าง  $+3\text{ V}$  ถึง  $+15\text{ V}$  สามารถส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตร แต่ถ้าเราต้องการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมากๆ เราจำเป็นต้องใช้อุปกรณ์อื่น ๆ เข้าช่วย เช่น การใช้โมเด็ม เป็นต้น

### 2.2.3 ลักษณะของคอนเนกเตอร์แบบ D – Type

หัวต่อแบบ D-Type ที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้นจะมีอยู่ 2 ลักษณะคือ แบบ 9 ขา และแบบ 25 ขา บางครั้งเราจะเรียกว่า DB9 และ DB 25 ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน

ตารางที่ 2.2 ลักษณะของ D – Type

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญญาณ
Pin 2	Pin 3	TD	Transit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

#### 2.2.4 รายละเอียดของสายสัญญาณ

สำหรับรายละเอียดของสายสัญญาณนั้นประกอบไปด้วย

Transmit Data : TD	ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์
Receive Data : RD	ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์
Request To Send : RTS	ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทางเพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมา
Clear To Send : CTS	ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD
Data Set Ready : DSR	ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่กับขา DTR
Signal-Ground : SG	เป็นกราวด์ของระบบ
Carrier Detect : CD	ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
Data Terminal Ready : DTR	ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อกับขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง
Ring Indicator : RI	ขานี้จะ Active เมื่อ โมเด็ม ได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

### 2.2.5 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส นั่นคือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ทั้งส่งส่วนที่เป็นข้อมูล และส่วนที่ใช้ควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากเครื่องส่งแบบอนุกรมจึงต้องถูกแยกแยะว่าใช้สำหรับวัตถุประสงค์ใด โดยเราสามารถแบ่งได้เป็น 4 ส่วน คือ

1. Start Bit	ขนาด 1 บิต
2. บิตข้อมูล (Data Character)	ขนาด 1 หรือ 8 บิต
3. Parity Bit	ขนาด 1 บิต
4. Stop Bit	ขนาด 1 หรือ 2 บิต

แต่ละตัวอักษรที่ถูกส่งออกไปเป็นกลุ่มจะประกอบไปด้วยบิตเริ่มต้น บิตข้อมูล บิตพาริตี (จะมีหรือไม่มีก็ได้) และบิตจบ โดยเราพอจะสรุปหน้าที่ของแต่ละส่วนได้ดังนี้

**Start Bit** หรือบิตเริ่มต้น จะใส่ที่จุดเริ่มต้นเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง

**Data Character** หรือบิตข้อมูล การส่งบิตข้อมูลจะส่งเป็นกลุ่มๆ โดยทั่วไปจะส่งเป็น 7 หรือ 8 บิต ซึ่งเพียงพอสำหรับการส่ง Ascii Word

**Parity Bit** หรือบิตพาริตี ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง เราจะใส่บิตพาริตีเข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้กันว่าใช้พาริตีแบบไหนในการส่งข้อมูลซึ่งหลักการในการกำหนดบิตพาริตีมีหลายแบบดังนี้

-พาริตีคู่ (Even Parity) ค่าของบิตพาริตีนี้เมื่อรวมกับทุกๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคู่ ตัวอย่างเช่น ข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้น บิตพาริตีจะเป็น 0

-พาริตีคี่ (Odd Parity) ค่าของบิตพาริตีนี้เมื่อรวมกับทุกๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่ ตัวอย่างเช่น ข้อมูล 1000101 มีเลข 1 ทั้งหมด 3 ตัว ดังนั้นบิตพาริตีจะเป็น 1

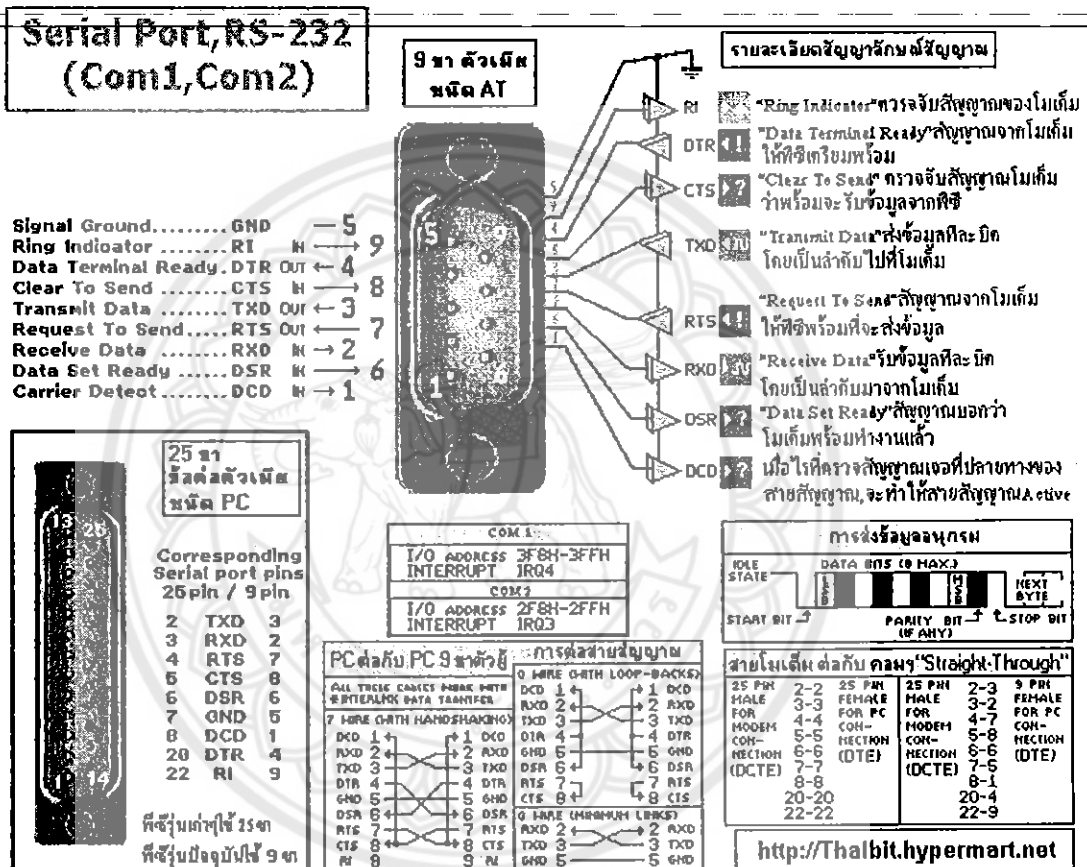
-ไม่มีพาริตี (None) ถ้าตั้งบิตพาริตีเป็น None ทั้งภาครับและภาคส่งจะไม่มีตรวจสอบบิตพาริตี

**Stop Bit** หรือบิตจบ เป็นบิตที่ส่งมาปิดท้ายข้อมูล



### 2.2.6 อัตราเร็วในการรับส่งข้อมูลแบบอนุกรม

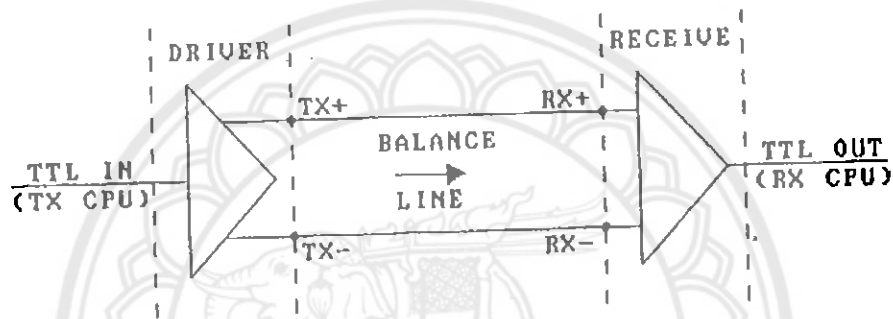
การที่อุปกรณ์ 2 อย่างจะติดต่อกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งค่าอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐาน RS-232C นั้นมีใช้ดังนี้ 110, 150, 300, 600, 1200, 2400, 4800 , 9600 และ 19200 บิตต่อวินาที



รูปที่ 2.4 แผนผังของสายสัญญาณ ในพอร์ตอนุกรม

2.2.7 มาตรฐาน RS - 422

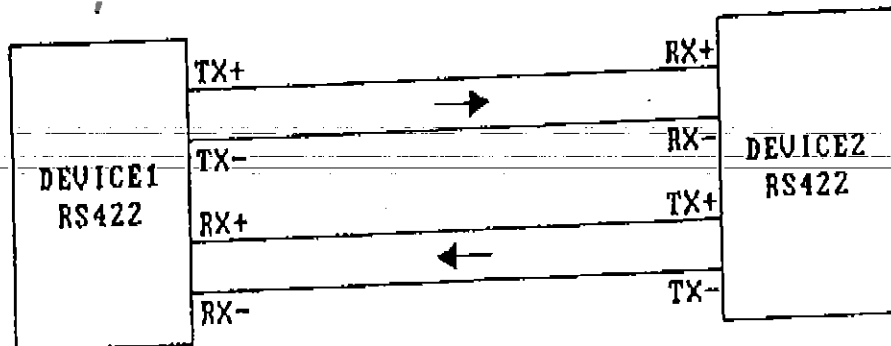
การสื่อสารแบบอนุกรม ด้วย RS - 422 จะใช้เทคนิคการส่งแบบ Balance Line ซึ่งวงจร Driver แบบนี้จะสามารถส่งสัญญาณที่มีค่าระหว่าง  $\pm 2$  volt ถึง  $\pm 6$  volt ได้ดี และในส่วนของ วงจรภาครับเองก็ยังสามารถตรวจจับสัญญาณที่มีขนาดต่ำถึง 200 mv. ได้ ถ้าใช้อุปกรณ์ได้ตรงตาม มาตรฐานที่กำหนดไว้ สามารถส่งได้ไกลถึง 1000 ฟุต แต่ถ้าความเร็วที่ใช้ในการรับส่งมีค่า ต่ำกว่า 10 Mbps. จะสามารถส่งได้ไกลถึง 4000 ฟุต (1,200 เมตร) แต่อย่างไรก็ตามต้องพิจารณา กับ องค์ประกอบอื่น ๆ ด้วย เช่น คุณภาพสายสัญญาณที่ใช้ในการรับส่ง, คุณภาพของ ขั้วต่อสัญญาณ, และ ระดับของสัญญาณรบกวนที่สายสัญญาณเดินผ่าน



รูปที่ 2.5 วิธีการรับ - ส่ง ข้อมูลแบบ RS 422 ในอุดมคติ

- การเชื่อมต่อ RS-422 แบบ Full duplex

เป็นการรับส่งข้อมูลแบบ 2 ทิศทาง ได้พร้อมกันซึ่งสามารถรับส่งได้พร้อมกันตลอดเวลา ตามต้องการ ในการเชื่อมต่อแบบนี้ จะต้องใช้วงจร Line Driver ถึง 2 ชุด คือรับ 1 ชุด และส่ง อีก 1 ชุด โดยจะมีสายสัญญาณชุดละ 1 คู่ (2 เส้น) การต่อแบบ Full Duplex นี้ มีลักษณะคล้ายกับการพูด โทรศัพท์ โดยไม่จำเป็นต้องให้อีกฝ่ายหนึ่งพูดจบก็ได้ แต่วิธีการแบบนี้จะใช้กับอุปกรณ์แบบ Point to Point

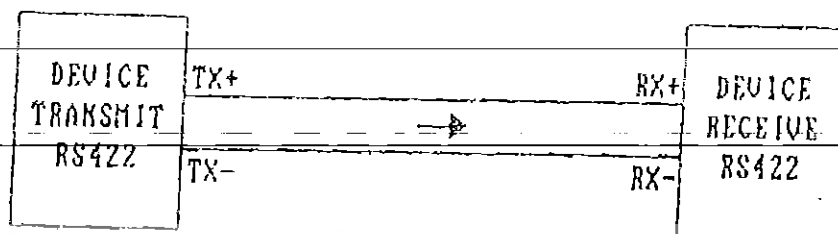


รูปที่ 2.6 ลักษณะการต่อสาย RS 422 แบบ Full Duplex

- การเชื่อมต่อ RS422แบบ simplex

เป็นการรับข้อมูลแบบทิศทางเดียว โดยกำหนดทิศทางไว้คงที่ ไม่สามารถเปลี่ยนได้ด้วย

~~โปรแกรม โดยทิศทางสามารถกำหนดเป็นแบบรับเข้าหรือส่งออกอย่างเดียวน~~



รูปที่ 2.7 ลักษณะการต่อสาย RS 422 แบบ Simplex

### 2.2.8 การรับส่ง ข้อมูลด้วย RS - 485

การรับส่งข้อมูลแบบ RS - 485 จะมีลักษณะคล้ายกับ RS - 422 ซึ่ง ใช้การส่งแบบสมดุล (Balanced Transmission) คือ เป็นการส่งข้อมูลที่ใช้ผลต่างของระดับแรงดันของสัญญาณ 2 เส้น สามารถส่งข้อมูลด้านความเร็วสูง (สูงสุดได้ถึง 2.5 เมกะบิตต่อวินาที และระยะทางได้ไกลมากขึ้น จาก RS - 422) คือ 1000 ม.

การส่งข้อมูลด้วย RS - 485 มีอยู่ 2 แบบ คือ ใช้สายสัญญาณ 4 เส้น และใช้สายสัญญาณ 2 เส้น

#### 1. การใช้สัญญาณแบบ 4 เส้น

สัญญาณข้อมูลที่รับและส่งจะใช้สัญญาณคนละคู่กัน คือ ส่ง 2 เส้น และรับ 2 เส้น ส่งสัญญาณสื่อสารแบบ Full Duplex ได้ คือสามารถส่งและรับข้อมูลได้ในเวลาเดียวกัน

#### 2. แบบใช้สายสัญญาณ 2 เส้น

สัญญาณที่รับและส่งจะใช้สายสัญญาณร่วมกัน โดยจะใช้การสื่อสารแบบ Half Duplex คือ ไม่สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน และในเครือข่ายหนึ่งจะมีผู้ส่ง ได้เพียงผู้เดียว เท่านั้น จะส่งพร้อมกันไม่ได้

### 2.2.9 สายสัญญาณที่ใช้กับ RS - 422 / RS - 485

สำหรับสายที่จะใช้ในระบอบ Line Driver แบบ RS - 422 / RS - 485 นั้นจะต้องเป็นสายที่ ออกแบบมาเพื่อใช้กับงานด้านการสื่อสารโดยเฉพาะ เช่น สายสัญญาณแบบ Unshiled Twis Pair (UTP) ซึ่งเป็นสายคู่ตีเกลียวแบบไม่มี Shiled ซึ่งเหมาะสำหรับใช้งานภายในอาคาร ซึ่งไม่มี แหล่งกำเนิดสัญญาณรบกวนแผ่กระจายออกมาในรัศมีที่พาดผ่านไปมากนัก แต่ถ้าเส้นทางที่

สายสัญญาณพาดผ่านไบนั้น มีระดับของสัญญาณรบกวนมากก็ต้องใช้สายแบบ Shiled Twis Pair (STP) ซึ่งเป็นสายคู่ตีเกลียวแบบมี Shiled นั้นเอง โดยในการนำไปต่อใช้งานนั้นต้องถูกคู่ด้วย

Application	Conductor		Insulation		Shield	Vct. of Prop. W.	Nominal Capacitance (pF/100m)	Nominal Attenuation		Max. End Cross-Talk	
	No. of pairs	AWG Size/ MATL D.C.F. structure	Material Thickness (mic)	Material NOM. OD. (mic)				Impedance (kOhm)	dB/100m	dB/100m	dB/100m



Category 5 / 100 Mbps.  
UTP & STP SOLID CONDUCTOR

EIA/TIA 568 10 Mbps IEEE 802.3 (10 Base-T) 4-18 Mbps IEEE 802.5 4 Mbps Arcnet 20 Mbps Arcnet Thick 100 Mbps ANSI X3T9.5 TPOOI Non Plenum NEC - CM FCC P11	4	24	None	None	PVC	0.084	125	65%	Maximum 17.0 (56.8)	0.007	Maximum	0.002	0.002	0.002
						0.205	107		17.0 (56.8)	0.004	0.004	0.004	0.004	0.004
						0.512	102		17.0 (56.8)	0.205	0.205	0.205	0.205	0.205
						0.772	100		17.0 (56.8)	0.512	0.512	0.512	0.512	0.512
						1	89		17.0 (56.8)	1	1	1	1	1
						1-100	100		17.0 (56.8)	1-100	1-100	1-100	1-100	1-100
									17.0 (56.8)	0.600 (2.07)	0.600 (2.07)	0.600 (2.07)	0.600 (2.07)	0.600 (2.07)
									17.0 (56.8)	1.300 (4.25)	1.300 (4.25)	1.300 (4.25)	1.300 (4.25)	1.300 (4.25)
									17.0 (56.8)	1.600 (5.00)	1.600 (5.00)	1.600 (5.00)	1.600 (5.00)	1.600 (5.00)
									17.0 (56.8)	2.000 (6.55)	2.000 (6.55)	2.000 (6.55)	2.000 (6.55)	2.000 (6.55)
									17.0 (56.8)	2.500 (8.20)	2.500 (8.20)	2.500 (8.20)	2.500 (8.20)	2.500 (8.20)
									17.0 (56.8)	2.600 (8.18)	2.600 (8.18)	2.600 (8.18)	2.600 (8.18)	2.600 (8.18)
									17.0 (56.8)	3.200 (10.3)	3.200 (10.3)	3.200 (10.3)	3.200 (10.3)	3.200 (10.3)
									17.0 (56.8)	3.000 (9.16)	3.000 (9.16)	3.000 (9.16)	3.000 (9.16)	3.000 (9.16)
									17.0 (56.8)	6.250 (19.1)	6.250 (19.1)	6.250 (19.1)	6.250 (19.1)	6.250 (19.1)
									17.0 (56.8)	6.700 (22.0)	6.700 (22.0)	6.700 (22.0)	6.700 (22.0)	6.700 (22.0)

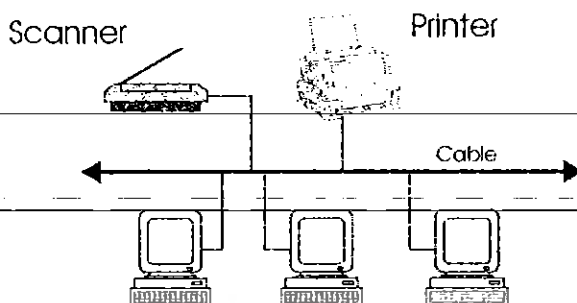
รูปที่ 2.8 แสดงตัวอย่าง Specification ของสายสัญญาณที่ใช้กับระบบ RS 422/485

2.2.10 มาตรฐานการรับส่งข้อมูลที่จะนำมาใช้

จากการศึกษาคุณสมบัติและรายละเอียดของแต่ละมาตรฐานการรับส่งข้อมูล แล้วแต่ละระบบก็จะมีข้อดี และข้อเสียแตกต่างกันไป ซึ่งก็จะขึ้นอยู่กับจุดประสงค์ของงานที่จะนำไปใช้ ในส่วนนี้มีจุดประสงค์ที่จะสื่อสารข้อมูล ในระยะทางที่ไกลออกไปและสามารถที่จะทำการเชื่อมต่ออุปกรณ์ในการทำงานหลายจุด เป็นแบบ Network เมื่อพิจารณามาตรฐาน RS-232 แล้ว ไม่สามารถจะนำมาประยุกต์ใช้ในส่วนนี้ได้เนื่องจากความจำกัดในระยะทาง ซึ่งเมื่อไกลออกไปสัญญาณจะถูกรบกวนและอ่อนกำลังลง มาตรฐาน RS-422 สามารถนำมาใช้ได้ เนื่องจากสามารถรับส่งข้อมูลออกไปในระยะไกลขึ้นมากกว่า RS-232 แต่ถ้าหากต้องการสื่อสารแบบ Full Duplex แล้ว RS-422 จะสามารถทำได้เพียงแค่ 2 จุด (แบบ Point to point) ไม่สามารถทำได้หลายจุด ยกเว้นจะสื่อสารแบบ Simplex RS-422 จะสามารถต่อออกไปได้ถึง 32 จุด แต่มีข้อจำกัดคือ สามารถส่งได้อย่างเดียว ฉะนั้นจึงมาพิจารณาที่มาตรฐาน RS-485 ซึ่งสามารถรับส่งสัญญาณได้ไกลกว่า RS-422 และสามารถสื่อสารได้ทั้งแบบ Half Duplex และแบบ Full Duplex

## 2.3 เน็ตเวิร์ก (Network)

### 2.3.1 ความหมายและประโยชน์ของเน็ตเวิร์ก (ขนาดเล็ก)



รูปที่ 2.9 เน็ตเวิร์ก

จากเมื่อก่อนเน็ตเวิร์กมีใช้กันแคในวงแคบๆ และใช้กับคอมพิวเตอร์ระบบใหญ่ ๆ เท่านั้น บัดนี้กลุ่มคอมพิวเตอร์ขนาดเล็กก็สามารถนำประโยชน์ของเน็ตเวิร์กที่เคยใช้กับเน็ตเวิร์กขนาดใหญ่ มาใช้ได้เช่นกันนอกจากนี้อินเทอร์เน็ตที่เราใช้กันอยู่ทุกวันนี้ก็เป็นระบบเน็ตเวิร์กรูปแบบหนึ่งด้วย

เน็ตเวิร์กตามความหมายที่เข้าใจง่าย ๆ ก็คือนำคอมพิวเตอร์ตั้งแต่ 2 เครื่องมาเชื่อมต่อกัน โดยใช้สายเคเบิล ( หรือไม่ก็ได้ เช่น เน็ตเวิร์กไร้สาย ( wireless)) เพื่อให้มีการใช้ประโยชน์ร่วมกัน เช่น มีการแชร์แลกเปลี่ยนข้อมูล มีการใช้ทรัพยากรร่วมกันทำให้ประหยัดค่าใช้จ่าย ถึงแม้ว่าจริง ๆ แล้วเน็ตเวิร์กจะมีอะไรซับซ้อนมากกว่าการนำคอมพิวเตอร์มากกว่า 1 เครื่องเข้ามาเชื่อมกัน แต่เน็ตเวิร์กก็ยังต้องมีหลักการของการใช้ประโยชน์ของการใช้ทรัพยากรร่วมกันอยู่ดี

### 2.3.2 ประเภทของเน็ตเวิร์ก

เราสามารถแบ่งรูปแบบของเน็ตเวิร์กได้หลายประเภท

#### ☞ ตามขนาด

คูพื้นที่ของระบบเน็ตเวิร์กเป็นสำคัญ ซึ่งเราสามารถแบ่งได้เป็น 3 ลักษณะดังนี้

#### 1. LAN (Local Area Network)

เป็นกลุ่มของคอมพิวเตอร์ที่เชื่อมต่อกันในพื้นที่จำกัด เช่น ภายในตึกสำนักงาน หรือภายในโรงงาน ส่วนมากจะใช้สายเคเบิลในการติดต่อสื่อสารกัน

#### 2. MAN (Metropolitan Area Network)

เป็นการนำระบบ LAN หลาย ๆ LAN ที่มีพื้นที่อยู่ใกล้เคียงกันมาเชื่อมต่อกันให้มีขนาดใหญ่ขึ้น เช่น เชื่อมต่อกันในเมือง หรือจังหวัด เป็นต้น

### 3. WAN ( Wide Area Network)

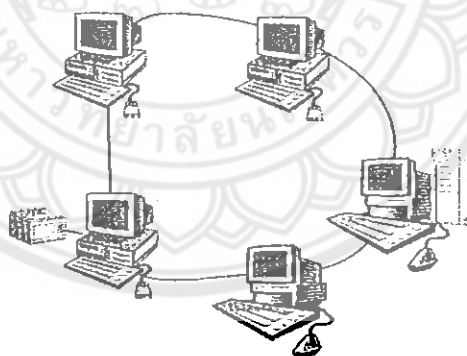
เป็นกลุ่มของคอมพิวเตอร์ที่เชื่อมต่อกันแบบกว้างขวางอาจจะเป็นภายในประเทศ หรือระหว่างประเทศ เป็นการใช้หลาย ๆ LAN และหลาย ๆ MAN ซึ่งอยู่กันคนละพื้นที่เชื่อมต่อเข้าหากัน เช่น สำนักงานที่ New York ติดต่อกับ tokyo การติดต่อสื่อสารกันอาจจะใช้ตั้งแต่สายโทรศัพท์ จนกระทั่งถึงดาวเทียม อาจจะมีอีกประเภทหนึ่งคือ SAN ( Small Area Network ) เป็นกลุ่มของคอมพิวเตอร์ที่เชื่อมต่อกันในที่พื้นที่ขนาดเล็กมาก อาจจะเป็นในบ้านหรือสำนักงานขนาดเล็กที่มีจำนวนของคอมพิวเตอร์ไม่ควรจะเกิน 10 เครื่อง

#### 👉 ตามหน้าที่

แต่ถ้าเรารู้ลักษณะการทำงาน การทำหน้าที่ของคอมพิวเตอร์แต่ละเครื่องของระบบเน็ตเวิร์กเป็นสำคัญอาจแบ่งได้เป็น 2 ประเภท

#### 1. เพียร์ทูเพียร์เน็ตเวิร์ก ( Peer to Peer Network )

เป็นลักษณะของกลุ่มคอมพิวเตอร์ที่คอมพิวเตอร์ทุกเครื่องมีสิทธิเท่าเทียมกันหมด ( Peer ) ไม่มีเครื่องไหนทำหน้าที่เป็นศูนย์กลางของเน็ตเวิร์ก เครื่องทุกเครื่องเป็นทั้งผู้ใช้และผู้ถูกใช้ สามารถเป็นได้ทั้ง Client และ Server ( ดูหัวข้อถัดไป ) ไม่มีเครื่องไหนมีหน้าดูแลจัดการระบบทั้งหมด เป็นต้นว่าการแชร์ทรัพยากรจะอย่างไร ใครเป็นผู้ดูแล ผู้ใช้แต่ละเครื่องจะเป็นผู้ดูแลข้อมูลและทรัพยากรของตัวเอง



รูปที่ 2.10 Peer to Peer Network

#### ลักษณะสำคัญ คือ

- เพียร์ทูเพียร์จะเป็นเน็ตเวิร์กขนาดเล็กอาจจะเรียกว่า Workgroup ก็ได้ คือมีคอมพิวเตอร์ไม่เกิน 10-15 เครื่อง
- ราคาไม่แพง เนื่องจากเป็นเน็ตเวิร์กขนาดเล็กจึงมีค่าใช้จ่ายไม่มาก ไม่มีความต้องการด้านฮาร์ดแวร์เป็นพิเศษ ( ไม่ต้องมีเซิร์ฟเวอร์ )
- การติดตั้งดูแลทำได้ง่าย ไม่ต้องมีผู้ทำหน้าที่ดูแลระบบเน็ตเวิร์กโดยตรง

- ส่วนมากใช้กับ Windows NT for Workstation เราสามารถใช้เน็ตเวิร์กเพียร์ทูเพียร์กับระบบการปฏิบัติการเหล่านี้ได้โดยไม่ต้องซื้อโปรแกรมอื่นเพิ่มเติม

จากข้อดีและข้อจำกัดดังกล่าว ระบบเพียร์ทูเพียร์จึงเหมาะกับ

- เครื่องคอมพิวเตอร์ไม่เกิน 10-15 เครื่อง

- ตำแหน่งของคอมพิวเตอร์อยู่ไม่ห่างกันมากนัก

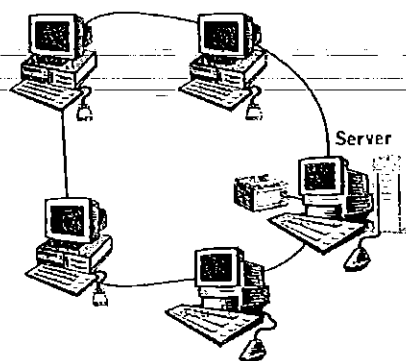
- ไม่ต้องการความสามารถในการรักษาความปลอดภัยมากนัก เพราะเป็นระบบรักษาความปลอดภัยในระบบเพียร์ทูเพียร์ไม่สูงมาก

- ไม่มีความต้องการที่จะขยายขนาดของเน็ตเวิร์กในอนาคต เช่น การเพิ่มคอมพิวเตอร์เข้าไปในระบบหลาย ๆ เครื่อง

ระบบเพียร์ทูเพียร์จะเป็นระบบเน็ตเวิร์กที่เราใช้กับเนื้อหาภายในเล่มนี้ เพราะว่าเป็นไปตามแนวคิดหลักของชื่อหนังสือ สร้างเน็ตเวิร์กขนาดเล็ก

## 2. ไคลเอ็นต์ – เซิร์ฟเวอร์ ( Client Server )

ในกรณีที่เป็นองค์กรของเรามีผู้ใช้มากกว่า 15-20 เครื่อง ระบบเน็ตเวิร์กแบบเพียร์ทูเพียร์ดูจะไม่ค่อยเหมาะสมเสียแล้ว ระบบไคลเอ็นต์ – เซิร์ฟเวอร์จะเป็นระบบที่เหมาะสมกว่าเพราะมีความสามารถในการดูแลควบคุมการทำงานของระบบเน็ตเวิร์กที่มีผู้ใช้เป็นจำนวนมากได้ดีกว่า ระบบเน็ตเวิร์กแบบนี้จะเป็นระบบที่มีศูนย์กลางมีคอมพิวเตอร์ที่ทำหน้าที่ดูแลระบบ อำนวยความสะดวก จัดเก็บข้อมูล รักษาความปลอดภัยให้กับคอมพิวเตอร์ทั้งระบบ มีหน้าที่คล้าย ๆ เป็นหัวหน้ากลุ่ม เราจะเรียกคอมพิวเตอร์ที่ทำหน้าที่นี้ว่า เซิร์ฟเวอร์ ( Server ) ส่วนเครื่องที่เหลือในระบบที่ไม่ได้ทำหน้าที่นี้จะเรียกว่า ไคลเอ็นต์ ( Client ) หรือ เวิร์กสเตชัน ( Workstation ) เป็นกลุ่มคอมพิวเตอร์ในระบบที่ทำหน้าที่รับบริการจากเครื่องเซิร์ฟเวอร์จะทำหน้าที่ควบคุมการใช้งานทุกอย่างของเน็ตเวิร์ก เช่น ข้อมูล เครื่องพิมพ์จะถูกดูแลและแชร์ โดยเครื่องเซิร์ฟเวอร์อุปกรณ์และทรัพยากรทุกอย่างจะเชื่อมต่อกับเครื่องเซิร์ฟเวอร์โดยตรง เครื่องไคลเอ็นต์ทุกเครื่องจะใช้งานทรัพยากรต่าง ๆ โดยผ่านทางเซิร์ฟเวอร์



รูปที่ 2.11 Client – Server Network

### ลักษณะสำคัญ

- ใช้กับเน็ตเวิร์กขนาดใหญ่ โดยอาจจะเริ่มตั้งแต่ 15 เครื่องจนถึงเป็นหมื่น ๆ เครื่อง
- มีการควบคุมระบบจากศูนย์กลาง เช่น การดูแลรักษาความปลอดภัย สำรองข้อมูล ดูแลการให้ใช้ทรัพยากรต่าง ๆ และอาจจะถูกดูแลจกจุด ๆ เดียว
- จะต้องมีเครื่องที่มีประสิทธิภาพสูงทำหน้าที่เป็นเซิร์ฟเวอร์
- ค่าใช้จ่ายสูง
- ใช้โปรแกรมเน็ตเวิร์กที่เป็นที่นิยม เช่น Windows NT Server , Network ทำหน้าที่เป็นเซิร์ฟเวอร์ เราไม่สามารถใช้ Windows 98/95 เป็นเซิร์ฟเวอร์ได้
- การใช้และการดูแลรักษาค่อนข้างยาก ต้องมีผู้ควบคุมดูแลระบบ  
ดังนั้นระบบเน็ตเวิร์กแบบไคลเอนต์ – เซิร์ฟเวอร์ จึงเหมาะกับ
- ในระบบมีจำนวนคอมพิวเตอร์เป็นจำนวนมาก
- ต้องการบริหารระบบจากจุด ๆ เดียว
- ต้องการระบบที่มีการรักษาความปลอดภัยสูง
- มีความต้องการขยายขนาดของเน็ตเวิร์กในอนาคต

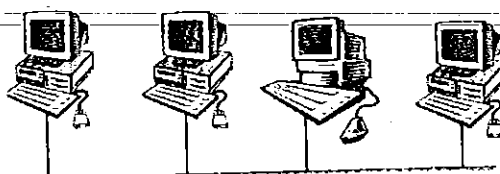
### 2.3.3 โครงสร้างของระบบเน็ตเวิร์ก ( Network Topology )

หมายถึงรูปแบบในการจัดวางตำแหน่งของคอมพิวเตอร์ สายเคเบิล ละอุปกรณ์อื่น ๆ เพื่อที่จะให้ข้อมูลได้ทำงานตามทิศทางที่เรากำหนดไว้ โครงสร้างเน็ตเวิร์กที่ต่างกันมีความต้องการด้านอุปกรณ์ต่าง ๆ เช่น ฮาร์ดแวร์ ซอฟต์แวร์ ที่แตกต่างกันไปด้วย โดยทั่วไปสามารถแบ่งออกได้เป็น

3 แบบ

1. แบบ Bus
- 2.แบบ Star
3. แบบ Ring

#### แบบบัส ( BUS )



รูปที่ 2.12 Bus Network



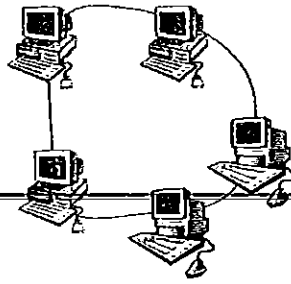
เป็นเน็ตเวิร์กที่ง่ายที่สุดและเป็นที่ยอมรับอย่างกว้างขวาง โดยการเชื่อมต่อคอมพิวเตอร์ทุกเครื่องด้วย การใช้สายเคเบิลเป็นสายหลัก เชื่อมต่อคอมพิวเตอร์ทุกเครื่อง โดยมีเน็ตเวิร์กการ์ดเป็นตัวเชื่อมระหว่างสายเคเบิลกับคอมพิวเตอร์ ข้อมูลจะถูกส่งออกไปตามสายไปยังคอมพิวเตอร์ทุก ๆ เครื่อง ไม่สนใจว่าเครื่องไหนคือผู้รับ คอมพิวเตอร์แต่ละเครื่องจะตรวจสอบเองว่าข้อมูลที่ถูกลำส่งออกมานั้นเป็นของตนหรือไม่ ถ้าไม่ จะปล่อยข้อมูลผ่านไป แต่ถ้าใช่ก็จะรับข้อมูลนำไปใช้ ณ เวลาใดเวลาหนึ่ง คอมพิวเตอร์เครื่องเดียวเท่านั้นที่สามารถส่งข้อมูลได้ ในระบบนี้คอมพิวเตอร์ไม่สามารถส่งข้อมูลได้พร้อม ๆ กันหลายเครื่องในเวลาเดียวกัน เป็นสาเหตุให้ประสิทธิภาพของเน็ตเวิร์กจะน้อยลงเมื่อมีจำนวนคอมพิวเตอร์มากขึ้น สายเคเบิลที่เป็นสายกลางหรือสายหลักที่คอมพิวเตอร์ต้องได้รับและส่งข้อมูลเรียกว่า Backbone สายที่ใช้ส่วนมากจะเป็นสายโคแอกเชียล (Coaxial) แบ่งเป็น 2 ประเภท คือ Thicknet และ Thinnet มีลักษณะคล้าย ๆ กับสายเคเบิลทีวี (UBC) การใช้จะต้องมีอุปกรณ์ที่ปิดหัวและท้ายของสายเคเบิลด้วยเรียกว่า เทอร์มิเนเตอร์ (Terminator) คอยรับสัญญาณไม่ให้สะท้อนกลับไปยังอาจจะเป็นการรบกวนสัญญาณได้ บัสนี้เป็นวิธีที่ง่ายและสะดวกที่สุดในการติดตั้งเน็ตเวิร์ก ไม่ต้องมีฮาร์ดแวร์มากมาย มีเพียงสายเคเบิลเน็ตเวิร์กการ์ดกับเทอร์มิเนเตอร์เท่านั้นก็พอแล้ว มักใช้กับเน็ตเวิร์กขนาดเล็กซึ่งมีคอมพิวเตอร์ไม่มากนัก

#### ข้อดีและข้อเสีย

- ประหยัด สะดวก ง่ายต่อการติดตั้ง
- ถ้าหากมีจุดใดของระบบ เช่น สายเคเบิล ไม่สามารถทำงานได้ ทั้งระบบก็จะเสียไม่สามารถใช้งานได้
- ถ้าหากมีจุดใดจุดหนึ่งเสียเป็นการยากที่ตรวจพบ
- จำนวนคอมพิวเตอร์มีผลต่อประสิทธิภาพของเน็ตเวิร์ก

#### แบบวงแหวน ( Ring )

เป็นการเชื่อมต่อคอมพิวเตอร์เข้าด้วยกันในลักษณะของรูปวงแหวน โดยใช้สายเคเบิลการต่อลักษณะนี้จะไม่มีจุดเริ่มต้นหรือจุดสุดท้าย การส่งข้อมูลจะวิ่งผ่านคอมพิวเตอร์ทุกเครื่องเป็นรูปวงแหวนในทิศทางเดียวกัน เมื่อคอมพิวเตอร์เครื่องหนึ่งต้องการส่งข้อมูล มันจะทำการส่งข้อมูลตำแหน่ง และที่อยู่ของเครื่องที่มันต้องการจะส่งข้อมูลไปให้ คอมพิวเตอร์ทุกเครื่องจะรับสัญญาณมาเช็กว่าเป็นข้อมูลของตนเองหรือเปล่า ถ้าไม่ จะส่งให้เครื่องต่อ ๆ ไป สัญญาณจะวิ่งไปจนกระทั่งเจอคอมพิวเตอร์เครื่องนั้นและรับข้อมูลนำไปใช้ เนื่องจากสัญญาณจะวิ่งไปเรื่อย ๆ เป็นวงกลม จึงไม่ต้องการอุปกรณ์ปิดหัวปิดท้ายมักจะใช้กับเน็ตเวิร์กที่มีคอมพิวเตอร์อยู่ไม่กี่โกลมากนัก ใช้โทเค็นเป็นสื่อในการส่งสัญญาณ โทเค็นจะถูกวิ่งผ่านไปทุก ๆ เครื่องเรื่อย ๆ จนกว่าเครื่องที่ต้องการส่งข้อมูลจะดึงโทเค็นไปใช้และส่งสัญญาณออกมาเครื่องที่มีโทเค็นเท่านั้นที่สามารถส่งข้อมูลได้



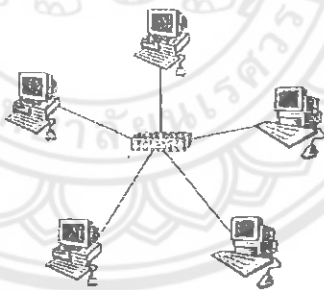
รูปที่ 2.13 Ring Network

**ข้อดีและข้อเสีย**

- คล้ายกับแบบบัส คือ ถ้าคอมพิวเตอร์เครื่องหนึ่งเครื่องใดเสีย หรือสายเคเบิลขาดเสียหาก็จะทำให้เน็ตเวิร์กทั้งระบบไม่สามารถทำงานได้
- การขยายขนาดของระบบหรือการเพิ่มคอมพิวเตอร์เข้าไปทำได้ยาก ถ้าจะต้องทำต้องมีการตัดและต่อสายเข้าไปใหม่

**แบบดาว (Star)**

เป็นลักษณะการเชื่อมต่อโดยเชื่อมต่อคอมพิวเตอร์ทุกเครื่อง เข้าสู่อุปกรณ์ส่วนกลางที่เรียกว่า ฮับ (Hub) ข้อมูลหรือสัญญาณจะเดินทางจากเครื่องส่ง ไปสู่ผู้รับ โดยผ่านฮับ



รูปที่ 2.14 Star Network

**ข้อดีและข้อเสีย**

- การขยายขนาดสามารถทำได้โดยง่าย โดยการต่อสายเคเบิลที่เชื่อมกับคอมพิวเตอร์เข้าสู่ฮับก็เรียบร้อย
- เมื่อเกิดปัญหากับสายเคเบิลหรือคอมพิวเตอร์ ไม่ทำให้คอมพิวเตอร์ทั้งระบบเสีย เครื่องที่ไม่เสียยังสามารถใช้งานได้ปกติและสามารถตรวจหาจุดที่เสียได้โดยง่ายด้วย
- ค่าใช้จ่ายค่อนข้างสูงถ้ามองในแง่เน็ตเวิร์กขนาดเล็กเพราะเราต้องซื้อฮับและต้องใช้สายเคเบิลเป็นจำนวนมากกว่าระบบแบบอื่น แต่ปัจจุบันราคาฮับถูกมาก ดังนั้นก็ไม่ถือว่าค่าใช้จ่ายสูงนัก

- แต่ถ้าฮับเสียบ เน็ตเวิร์กทั้งระบบก็จะเสียบตามไปด้วย

### 2.3.4 ประเภทของลื่อนำสัญญาณ

#### 1. ประเภทมีสาย

เช่น สายเคเบิลต่าง ๆ เป็นแบบใช้สายสัญญาณเชื่อมต่อในระบบเน็ตเวิร์ก เช่น สายเคเบิล สายเคเบิลที่เป็นที่นิยมกันส่วนมาก เช่น

- สาย Coaxial

- สาย Twisted pair

- สาย Fiber Optic

สายเคเบิลแต่ละประเภทมีข้อดีข้อเสียต่างกันออกไปขึ้นอยู่กับประเภทของการทำงาน ดังนั้นการเลือกใช้สายเคเบิลจึงมีความสำคัญเป็นอย่างมาก เพราะแต่ละชนิดมีความแตกต่างกัน ขึ้นอยู่กับความต้องการของเน็ตเวิร์กแต่ละประเภทด้วย เช่น

ความเร็วในการรับส่งข้อมูล สายแต่ละประเภทมีความเร็วในการรับส่งข้อมูลที่ต่างกัน เลือกให้เหมาะกับชนิดของงาน

ระยะทางและสัญญาณรบกวนยิ่งไกลสัญญาณที่ส่งออกมาก็จะอ่อนลงและสายแต่ละประเภทป้องกันสัญญาณรบกวนได้ต่างกัน ถ้าเราต้องใช้เน็ตเวิร์กที่มีระยะห่างระหว่างเครื่องมาก หรือที่มีสิ่งรบกวนมาก ควรเลือกสายชนิดใดนั้น ต้องเลือกให้เหมาะสมด้วย

ค่าใช้จ่าย ถ้าเป็นของดีก็ยิ่งแพง ควรคิดให้รอบคอบว่าประสิทธิภาพกับราคาคุ้มกันหรือไม่

☞ สายโคแอกเชียล (Coaxial)



รูปที่ 2.15 ภายในสายโคแอก

เคยเป็นที่นิยมอย่างมากเป็นสายเส้นกลมส่วนมากมีสีดำมีไส้กลางเป็นแกนทองแดงหุ้มด้วยพลาสติกชั้นถัดมาประกอบด้วยโลหะพอยด์ที่ทอมาหุ้มเพื่อป้องกันสัญญาณรบกวน ชั้นนอกหุ้มพลาสติกอีกชั้นสามารถทนต่อการรบกวนของสัญญาณภายนอกได้ดีกว่าแบบ Twisted Pair และเมื่อมีการส่งสัญญาณเป็นระยะทางไกลจะมีการสูญเสียสัญญาณไปบ้างแต่น้อยกว่าแบบ Twisted Pair ส่วนมากเรานิยมสายประเภทนี้กับเน็ตเวิร์กแบบบัสดังนั้นจึงต้องมีที่ปิดหัวปิดท้ายสาย โคแอกเชียลสามารถแบ่งเป็นประเภทใหญ่ ๆ ได้ 2 ประเภท

- Thinnet หรืออาจจะเรียก 10base2

ส่วนใหญ่สายโคแอกเชียลที่เป็นที่รู้จักกันมักจะเป็นสายประเภทนี้ สามารถเชื่อมต่อกันได้โดยตรงกับเน็ตเวิร์กการ์ด โดยมีอุปกรณ์ลักษณะคล้ายตัว T เรียกว่า “ทีคอนเน็กเตอร์” เป็นตัวเชื่อมต่อ ไม่ต้องการอุปกรณ์เพิ่มเติม ใช้งาน ถูกเหมาะกับเน็ตเวิร์กขนาดเล็ก มีความยาวได้ถึง 185 เมตร โดยไม่ต้องใช้รีพีตเตอร์ ( Repeater ) < การส่งสัญญาณหรือข้อมูลไปตามสายเคเบิลต่าง ๆ กำลังส่งของสัญญาณจะลดลงเรื่อย ๆ ยิ่งไกลมาก สัญญาณก็จะต่ำลงเรื่อย ๆ ดังนั้นต้องมีอุปกรณ์ขยายสัญญาณเข้าช่วยเมื่อต้องการส่งสัญญาณให้มีระยะไกลขึ้น อุปกรณ์ชิ้นนั้นก็คือ รีพีตเตอร์ ( Repeater ) > สามารถส่งข้อมูลได้สูงสุด 10 Mbps ( 10 เมกะบิต ต่อ 1 วินาที )

อุปกรณ์ต่าง ๆ ของสายโคแอก



รูปที่ 2.16 หัวแบบ BNC และสายโคแอก

รูปที่ 2.17 อุปกรณ์เชื่อมต่อสายโคแอก

BNC barrel connector ใช้ในกรณีที่ต้องการต่อสายให้ยาวขึ้น



รูปที่ 2.18 BNCT Connector เชื่อมต่อระหว่างสายเคเบิลกับเน็ตเวิร์กการ์ด



รูปที่ 2.19 BNC Terminat อุปกรณ์สำหรับปิดหัวและท้ายสายเคเบิล

- Thicknet หรือ 10 base5

จะมีขนาดใหญ่กว่าหนากว่าตามชื่อของมันและมีความยาวได้ถึง 500 เมตร โดยไม่ต้องมีรีพีตเตอร์มีความทนทานและต้านทานต่อสัญญาณรบกวนภายนอกได้ดีกว่าแบบ Thinnet ด้วยเหตุผลนี้ จึงนิยมติดตั้งไว้นอกอาคาร อาจจะเป็นลักษณะของ Backbone แต่ราคาค่อนข้างสูงและความยืดหยุ่นต่อการเชื่อมต่อกับคอมพิวเตอร์ที่ค่อนข้างจะซับซ้อนกว่าแบบ Thinnet มาก

#### ☞ สายตีเกลียวคู่ ( Twisted – Pair )

มีลักษณะคล้าย ๆ กับสายโทรศัพท์แต่ใหญ่กว่า มีลักษณะเป็นเส้นลวดทองแดงพันเป็นเกลียวเพื่อป้องกันสัญญาณรบกวน ( แต่ก็ยังป้องกันสัญญาณรบกวนได้น้อยกว่าแบบ โคแอกเชียล ) แล้วห่อหุ้มด้วยพลาสติกอีกชั้น สามารถติดตั้งได้โดยง่าย แบ่งเป็น 2 ประเภท



รูปที่ 2.20 UTP

#### - UTP ( Unshielded Twisted Pair )

เป็นสายเคเบิลที่เป็นที่นิยมมากที่สุดคืออีกชื่อหนึ่งก็คือ 10BaseT สามารถแบ่งได้เป็น 5 กลุ่ม แต่ที่ได้รับความนิยมมากมี 2 กลุ่ม ( Category ) คือ Cat 5 และ Cat 3 ใช้หัวคอนเน็กเตอร์แบบ RJ45 เชื่อมต่อกับเน็ตเวิร์กการ์ด ส่งสัญญาณได้ไม่เกิน 100 เมตร โดยไม่มีรีพีตเตอร์ ราคาถูกที่สุดในจำนวนสายเคเบิลทั้ง 3 ประเภท แต่มติดตั้งง่าย แต่รับสัญญาณรบกวนได้ง่าย ระยะทางจำกัด ไม่เหมาะกับสถานที่ที่มีสิ่งภายนอกรบกวนมาก เช่น ใกล้อุปกรณ์ไฟฟ้าหรือภายนอกตึกอาคาร แบ่งได้เป็น 5 กลุ่มตามความสามารถในการส่งข้อมูลสัญญาณ

กลุ่มที่ 1 สามารถส่งข้อมูลได้ 4 Mbps ( สาย โทรศัพท์ )

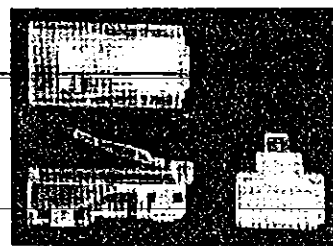
กลุ่มที่ 2 สามารถส่งข้อมูลได้ 4 Mbps ( Local talk และ Apple )

กลุ่มที่ 3 สามารถส่งข้อมูลได้ 16 Mbps ( นิยมใช้ Ethernet )

กลุ่มที่ 4 สามารถส่งข้อมูลได้ 20 Mbps ( 16 Mbps Token Ring )

กลุ่มที่ 5 สามารถส่งข้อมูลได้ 100 Mbps ( แพร่หลายมากที่สุดในปัจจุบัน )

## อุปกรณ์ของ UTP



รูปที่ 2.21 สาย UTP

รูปที่ 2.22 RJ45 คอนเน็กเตอร์

### - STP (Shield twisted pair)

คล้ายกับสาย UTP แต่จะมีโลหะที่ทอหุ้มล้อมรอบอีกที ทนต่อสัญญาณรบกวนได้ดีกว่า ในระยะทางประมาณ 100 เมตรเหมือนกัน แต่ไม่ค่อยยืดหยุ่นในการใช้สายเนื่องจากมีขนาดใหญ่ ติดตั้งยากพอสมควรและราคาแพงกว่าแบบ UTP จึงไม่ค่อยเป็นที่นิยม

☞ สายไฟเบอร์อปติก (Fiber Optic) หรือใยแก้วนำแสง



รูปที่ 2.23 Fiber Optic

จะนำส่งสัญญาณ โดยใช้แสงโดยจะเปลี่ยนจากสัญญาณ ไฟฟ้าให้เป็นแสง ในการส่งสัญญาณข้อมูลประกอบด้วยส่วนกลางทำด้วยแก้ว แท่งแก้ว หรือพลาสติก ซึ่งถูกล้อมรอบด้วยเส้นใยแล้วมีพลาสติกหุ้มชั้นนอกสุดเพื่อป้องกันการสูญเสียสูญหายของสัญญาณ สัญญาณต่าง ๆ ภายนอกไม่สามารถรบกวนได้เลยในขณะที่สายอื่น ๆ สัญญาณจะอ่อนลงเรื่อย ๆ เมื่อมีระยะทางไกลขึ้น แต่ใยแก้วนำแสงสัญญาณจะไม่อ่อนลงเลย ราคาจึงสูงมากและยากต่อการติดตั้งและดูแลรักษา ยาก ถ้าเกิดความเสียหายหรือหัก การซ่อมแซมทำได้ลำบาก สามารถส่งสัญญาณได้ในทิศทางเดียว ดังนั้นในการติดตั้งจึงต้องมี 2 เส้นคู่กัน รับ 1 เส้น ส่ง 1 เส้น สามารถส่งข้อมูลได้ 100 Mbps จนกระทั่งถึง Gbps ขึ้นอยู่กับชนิดของใยแก้วนำแสง ดังนั้นจึงเหมาะกับเน็ตเวิร์กที่ต้องการความปลอดภัยสูง เหมาะที่จะทำเป็น Backbone ที่มีการส่งข้อมูลเป็นจำนวนมากและเป็นระยะทางไกล เช่น ระหว่างตึก เป็นต้น

## ตาราง 2.3 เปรียบเทียบค่าต่างๆของสายเน็ตเวิร์กทั่วไป

## สรุปตารางเปรียบเทียบค่าต่างๆของสายเน็ตเวิร์กโดยทั่วไป

ป.ร.

85665

ชนิด	ราคา	การติดตั้ง	ความเร็ว	ระยะทาง	สัญญาณรบกวน
Thinnet	ถูกกว่า STP	ง่าย	10 Mbps	185 เมตร	น้อยกว่า UTP
Thicknet	แพงกว่า STP	ไม่ง่าย	10 Mbps	500 เมตร	น้อยกว่า UTP
STP	แพงกว่า UTP	ไม่ยาก	16-500 Mbps	100 เมตร	น้อยกว่า UTP
	ถูกกว่า Thicknet				
UTP	ถูกที่สุด	ง่ายมาก	10-100 Mbps	100 เมตร	มาก
Fiber Optic	แพงสุด	ยาก	100 Mbps	10 กิโลเมตร	ไม่มี

## 2. ประเภทไม่ใช้สาย

หรืออาจจะเรียกว่าสื่อไร้สาย เป็นสื่ออีกประเภทหนึ่งของระบบเน็ตเวิร์ก เหมาะสำหรับการต่อเน็ตเวิร์กที่มีระยะทางห่างไกลกันมาก ใช้สายเคเบิลอาจไม่เหมาะ หรือในที่ที่ไม่อาจใช้สายเคเบิลได้หรือใช้กับระบบเน็ตเวิร์กที่ตั้งของสำนักงานไม่อยู่กันที่ มีการเปลี่ยนสถานที่ตลอด หรือใช้กับผู้ใช้เน็ตเวิร์กที่ต้องเดินทางตลอดเวลา ตัวอย่างเช่น

- ระบบวิทยุ
- ไมโครเวฟ
- แสงอินฟราเรด
- ดาวเทียม

## เน็ตเวิร์กการ์ดหรือ NIC ( Network Interface Card)

เป็นอุปกรณ์ที่ถูกติดตั้งในคอมพิวเตอร์เพื่อเชื่อมกับเคเบิลในระบบเน็ตเวิร์ก ทำหน้าที่ส่งและรับข้อมูลจากคอมพิวเตอร์กับเน็ตเวิร์ก โดยผ่านสายเคเบิล จะแปลงสัญญาณที่ได้รับจากเคเบิลให้เป็นข้อมูลที่คอมพิวเตอร์สามารถเข้าใจได้ และแปลงสัญญาณจากคอมพิวเตอร์ให้อยู่ในรูปแบบที่สามารถเดินทางไปในสายเคเบิลได้ อาจแบ่งได้ตามประเภทของสล็อต หรือช่องที่เสียบในคอมพิวเตอร์ โดยทั่วไปมี 2 แบบ

- แบบ ISA – ราคาถูก
- แบบ PCI- ติดตั้งง่ายกว่า แต่ราคาแพงกว่า ISA นิดหน่อย

แต่ละการ์ดจะมีรูปแบบการเชื่อมต่อเน็ตเวิร์กได้หลายแบบขึ้นอยู่กับคอนเน็กเตอร์ที่อยู่ด้านหลังของเน็ตเวิร์กการ์ด บางการ์ดมีคอนเน็กเตอร์แบบ BNC สำหรับสายโคแอกซ์หรือ RJ45 สำหรับสาย UTP บางการ์ดอาจจะมีทั้ง 2 ชนิด ในเน็ตเวิร์กการ์ดอันเดียวกัน จะเลือกช่องคอนเน็กเตอร์แบบไหนก็ขึ้นอยู่กับเคเบิลที่เราเลือกใช้

### 2.3.5 ฮับ ( HUB)

เป็นอุปกรณ์ที่ถูกติดตั้งในคอมพิวเตอร์เพื่อเชื่อมกับเคเบิลในระบบเน็ตเวิร์กแบบดาว จะมีช่องเสียบที่เรียกว่าพอร์ตเป็นตัวกำหนดว่า เราสามารถต่อคอมพิวเตอร์ได้กี่เครื่องนอกจากนี้เรายังสามารถใช้อันต่อกับฮับ ซึ่งจะช่วยให้ขนาดของเน็ตเวิร์กใหญ่ขึ้นได้-การใช้วงง่ายและสะดวกมีช่องไฟแต่ละพอร์ตคอยบอกการทำงานของของพอร์ตว่ากำลังทำอะไรอยู่ ถ้าไม่ต้องการใช้คอมพิวเตอร์เครื่องใดในเน็ตเวิร์ก ก็แค่ดึงสายออกจากพอร์ต ฮับสามารถแบ่งเป็น 3 แบบ

#### 1. Active

เมื่อสัญญาณเข้ามาจะมีการปรับแต่งก่อนที่ส่งออกไป เช่น คัดสัญญาณรบกวนออก หรือ ทำหน้าที่คล้ายรีพีตเตอร์ด้วย ทำให้ส่งสัญญาณได้ระยะที่ไกลขึ้น มีราคาแพงกว่าแบบ Passive

#### 2. Passive

ไม่มีอุปกรณ์คอยกำจัดสัญญาณรบกวนเหมือน Active สัญญาณเพียงเข้ามาแล้วก็ออกไป

#### 3. Switching

เป็นฮับที่มีอุปกรณ์พิเศษอยู่ ทำให้สามารถรับส่งข้อมูลได้เร็วขึ้น แต่จะมีราคาแพงกว่าฮับแบบปกติ

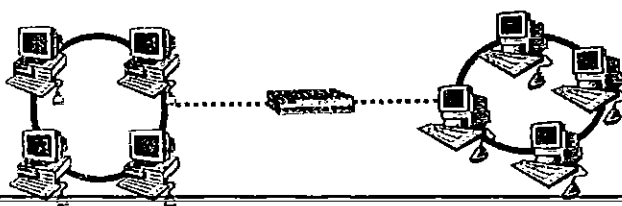
### 2.3.6 รีพีตเตอร์ ( Repeater)

ทำหน้าที่คล้าย ๆ กับเครื่องขยายสัญญาณ เมื่อมีการส่งสัญญาณเข้ามาจากระยะทางไกล สัญญาณจะอ่อนลง รีพีตเตอร์จะทำการขยายสัญญาณ เพื่อให้ได้ระยะทางที่เพิ่มมากขึ้น

### 2.3.7 บริดจ์ ( Bridge)

ทำหน้าที่เชื่อมต่อระบบเน็ตเวิร์กเข้าด้วยกันได้ นิยมใช้เชื่อมเน็ตเวิร์กย่อย ๆ หลายเน็ตเวิร์กเข้าด้วยกัน ทำให้กลายเป็นเน็ตเวิร์กที่ใหญ่ขึ้นเพียงเน็ตเวิร์กเดียว และเน็ตเวิร์กทั้งหมดสามารถสื่อสารกันได้ทำให้ลดปัญหาการคับคั่งของสัญญาณเน็ตเวิร์ก เมื่อมีการส่งสัญญาณมาที่บริดจ์ บริดจ์ จะทำการตรวจสอบว่าสัญญาณจะถูกส่งไปไหน ถ้าเป็นที่อยู่ที่อยู่ในเน็ตเวิร์กของเราเอง มันก็จะย้อนกลับมาหาในเน็ตเวิร์กของเรา แต่ถ้าบริดจ์ตรวจพบว่าเป็นเน็ตเวิร์กอื่น ก็จะส่งไปที่เน็ตเวิร์กนั้น มันจะคอยตรวจสอบและจะส่งสัญญาณไปที่ทิศทางที่ถูกต้อง ทำหน้าที่คล้ายกับเป็นตำรวจจราจรของระบบเน็ตเวิร์กคอยตรวจดูรถ ( สัญญาณ ) ว่าควรไปในทิศทางใด เหมาะสำหรับกลุ่มของเน็ตเวิร์กที่ไม่ซับซ้อนมากนัก





รูปที่ 2.24 บริดจ์

### 2.3.8 เราเตอร์ (Router)

เป็นลักษณะการเชื่อมต่อเน็ตเวิร์กกับเน็ตเวิร์กเข้าด้วยกัน จริง ๆ แล้วก็คล้ายกับบริดจ์ แต่ฉลาดกว่ามากอาจเรียกว่า ซูเปอร์บริดจ์ เราเตอร์จะเลือกเส้นทางส่งที่ดีที่สุดเพื่อทำการส่งข้อมูล โดยตรวจสอบที่อยู่ของผู้รับเป็นเกณฑ์ ถ้าหากว่ามีช่องทางใดเสียหาย เราเตอร์จะรู้ทันทีและจะหลีกเลี่ยงเส้นทางนั้น ดังนั้นจะส่งข้อมูลได้เร็วและถูกต้องยิ่งขึ้น คุณลองนึกถึงตำรวจจราจรอีกครั้งหนึ่ง ตำรวจคนนี้จะฉลาดกว่าคนเก่าหลายเท่า เมื่อมีรถ ( สัญญาณ ) ว่างเข้ามาตำรวจจะสอบถามรถว่าจะวิ่งไปที่ไหนและจะทำการบอกเส้นทางที่ดีที่สุดให้ อาจจะเป็นเส้นทางลัดเพื่อให้ไปถึงได้เร็วขึ้น ถ้ามีถนนใดชำรุดหรือเสีย จะบอกรถให้หลีกเลี่ยงและแนะนำทางใหม่ให้แทน ตำรวจจะไม่มีอาการปล่อยรถวิ่งหาทางเองอย่างเด็ดขาดเพราะตำรวจรู้ทุกเส้นทางแต่ตำรวจของบริดจ์ ถ้ารถบอกที่อยู่แล้ว ตำรวจไม่รู้จักก็ปล่อยให้รถไปหาทางเอาเอง ( บรอดคาสต์ ) ซึ่งรถอาจจะวิ่งมั่วทำให้เกะกะได้ เราเตอร์จึงเหมาะกับเน็ตเวิร์กขนาดใหญ่และซับซ้อนดังนั้นราคาจึงสูงมาก

## 2.4 ระบบฐานข้อมูล (Database System)

ฐานข้อมูล (Database) คือ กลุ่มของข้อมูลที่มีความสัมพันธ์เกี่ยวข้องเป็นเรื่องเดียวกัน เช่น กลุ่มข้อมูลเกี่ยวกับพนักงานบริษัท ประกอบด้วย รหัสพนักงาน ชื่อ นามสกุล เบอร์โทรศัพท์ และกลุ่มข้อมูลดังกล่าวถูกจัดเก็บอยู่รวมกันหลาย ๆ กลุ่ม ซึ่งอาจจะเก็บอยู่ในรูปแฟ้มเอกสารหรืออยู่ในคอมพิวเตอร์

กล่าวโดยสรุปแล้ว ฐานข้อมูลมีลักษณะสำคัญ ดังนี้

- เป็นเรื่องเกี่ยวกับการจัดเก็บข้อมูล
- ข้อมูลที่จัดเก็บมีความสัมพันธ์เกี่ยวข้องเป็นเรื่องเดียวกัน
- สามารถแสดงออกมาอยู่ในรูปแบบของตารางได้

### 2.4.1 ส่วนประกอบของตารางข้อมูลในฐานข้อมูล

โดยทั่วไปแล้วตารางข้อมูลที่ใช้งานกันจะประกอบด้วย แถว (Row) และคอลัมน์ (Column) ต่างๆ แต่ถ้ามองกันในรูปแบบของฐานข้อมูลแล้ว เราจะเรียกรายละเอียดในแถวว่า เร

คอร์ด (Record) และเรียกรายละเอียดในแนวคอลัมน์ว่า ฟิลด์ (Field)

ในฐานข้อมูล 1 ระบบ อาจประกอบด้วยตารางข้อมูลมากกว่า 1 ตาราง ฐานข้อมูลที่มี ตารางข้อมูลมากกว่า 1 ตาราง และมีตารางตั้งแต่ 1 คู่ขึ้นไปที่มีความสัมพันธ์กันด้วยฟิลด์ใดฟิลด์หนึ่ง เราเรียกฐานข้อมูลประเภทนี้ว่า “ฐานข้อมูลเชิงสัมพันธ์” หรือ Relational Database

#### 2.4.2 ประโยชน์ของระบบฐานข้อมูล

ฐานข้อมูลจะช่วยสร้างระบบการจัดการเก็บข้อมูลขององค์กรให้เป็นระเบียบ แยกแยกข้อมูลตามประเภท ทำให้ข้อมูลประเภทเดียวกันจัดเก็บอยู่ด้วยกัน สามารถค้นหาและเรียกใช้ได้ง่าย ไม่ว่าจะนำมาพิมพ์รายงาน นำมาคำนวณ หรือนำมาวิเคราะห์ ซึ่งทั้งนี้ขึ้นอยู่กับการใช้ประโยชน์ขององค์กรหรือหน่วยงานนั้น ๆ

จากประโยชน์ของระบบฐานข้อมูลด้วยคอมพิวเตอร์ข้างต้น อาจกล่าวได้ระบบฐานข้อมูลมีข้อดีมากกว่าการเก็บข้อมูลในระบบแฟ้มข้อมูล ดังนี้

1. หลีกเลี่ยงความขัดแย้งของข้อมูลได้
2. สามารถใช้ข้อมูลร่วมกัน
3. สามารถลดความซ้ำซ้อนของข้อมูล
4. การรักษาความถูกต้องเชื่อถือได้ของข้อมูล
5. สามารถกำหนดความเป็นมาตรฐานเดียวกันได้
6. สามารถกำหนดระบบรักษาความปลอดภัยให้กับข้อมูลได้
7. ความเป็นอิสระของข้อมูล

#### 2.4.3 ประโยชน์ของฐานข้อมูลเชิงสัมพันธ์

1. ช่วยลดความซ้ำซ้อนของการจัดเก็บข้อมูล
2. ช่วยให้สามารถเรียกใช้ข้อมูลได้ตรงกัน(ข้อมูลอัปเดตได้ทันเวลา) เนื่องจากข้อมูลดูแลรักษาจากที่เดียวกัน
3. ช่วยป้องกันการผิดพลาดจากการป้อนข้อมูลและแก้ไขข้อมูล(ป้อนข้อมูลที่ตารางหลัก)
4. ช่วยประหยัดเนื้อที่การจัดเก็บข้อมูลในคอมพิวเตอร์ และอื่น ๆ (ไม่เก็บข้อมูลซ้ำซ้อนเก็บข้อมูลเท่าที่จำเป็น)

#### 2.4.4 โครงสร้างของฐานข้อมูล

โครงสร้างของฐานข้อมูลประกอบด้วย

1. Character คือ ตัวอักษรแต่ละตัว / ตัวเลข / เครื่องหมาย
2. Field คือ เขตข้อมูล / ชุดข้อมูลที่ใช้แทนความหมายของสื่อโครงสร้าง เช่น ชื่อของบุคคล ชื่อของวัสดุสิ่งของ

3. Record คือ ระเบียบ หรือรายการข้อมูล เช่น ระเบียบของพนักงานแต่ละคน

4. Table /File คือ ตาราง หรือแฟ้มข้อมูล ประกอบขึ้นด้วยระเบียบต่างๆ เช่น ตารางข้อมูลของบุคคล ตารางข้อมูลของวัสดุถึงของ

~~5. Database คือ ฐานข้อมูล ประกอบด้วยตาราง และแฟ้มข้อมูลต่างๆ ที่เกี่ยวข้องหรือมีความสัมพันธ์กัน~~

#### 2.4.5 องค์ประกอบของระบบฐานข้อมูล

ระบบฐานข้อมูลโดยส่วนใหญ่แล้ว เป็นระบบที่มีการนำเอาคอมพิวเตอร์มาช่วยในกระบวนการจัดเก็บข้อมูล ค้นหาข้อมูล ประมวลผลข้อมูล เพื่อให้ได้สารสนเทศที่ต้องการแล้วนำไปใช้ในการปฏิบัติงานและบริหารงานของผู้บริหาร โดยอาศัยโปรแกรมเข้ามาช่วยจัดการข้อมูลจากกระบวนการดังกล่าวนี้

จากกระบวนการดังกล่าวนี้ระบบฐานข้อมูลจึงมีองค์ประกอบ 5 ประเภท คือ

1. ฮาร์ดแวร์(Hardware)
2. โปรแกรม ( Program หรือ Software) ซึ่งมีหน้าที่ควบคุมดูแลการสร้างฐานข้อมูล การเรียกใช้ข้อมูล และ การจัดทำรายงาน เรียกว่าโปรแกรมระบบจัดการฐานข้อมูล (Database Management System : DBMS)
3. ข้อมูล (Data)
4. บุคลากร (Peopleware) คือ ผู้ใช้งาน (User) พนักงานปฏิบัติการ(Operator) นักวิเคราะห์และออกแบบระบบ (System Analyst) ผู้เขียนโปรแกรมประยุกต์ใช้งาน (Programmer) และผู้บริหารฐานข้อมูล (Database Administrator : DBA)
5. ขั้นตอนการปฏิบัติงาน (Procedure) เป็นขั้นตอนและวิธีการต่างๆ ในการปฏิบัติงาน เพื่อการทำงานที่ถูกต้องและเป็นไปตามขั้นตอนที่ได้กำหนดไว้ จึงควรทำเอกสารที่ระบุขั้นตอนการทำงานของหน้าที่ต่าง ๆ ในระบบฐานข้อมูล ทั้งขั้นตอนปกติ และขั้นตอนในสภาวะที่ระบบเกิดปัญหา (Failure)

#### 2.4.6 ความสัมพันธ์ระหว่างแฟ้มข้อมูล

ในฐานข้อมูลหนึ่ง ๆ สามารถที่จะกำหนดชนิดของความสัมพันธ์ระหว่างแฟ้มข้อมูลที่เป็นไปได้ 3 ชนิด

1. ความสัมพันธ์แบบ 1 ต่อ 1 (One-To-One Relationship)
2. ความสัมพันธ์แบบ 1 ต่อ หลาย หรือ หลาย ต่อ 1 (One-To-Many or Many-To-One Relationship)
3. ความสัมพันธ์แบบหลายต่อหลาย (Many-To-Many Relationship)

### 2.4.7 คำศัพท์ที่ใช้ในระบบฐานข้อมูล

#### ☞ คำศัพท์พื้นฐานในระบบฐานข้อมูล

- เอนทิตี(Entity) หมายถึง ชื่อของสิ่งหนึ่งสิ่งใด เปรียบเสมือนคำนาม ได้แก่ บุคคล

สถานที่ สิ่งของ เช่น นักศึกษา อาจารย์ ภาควิชา ฯลฯ

- แอททริบิวต์ (Attribute) หมายถึง รายละเอียดของข้อมูลใน Entity หนึ่ง ซึ่งเป็นข้อมูลที่แสดงลักษณะและคุณสมบัติของ Entity เช่น Entity ของนักศึกษา ได้แก่ ชื่อ อายุ เพศ ฯลฯ

- ความสัมพันธ์(Relationship) หมายถึง คำกริยาที่แสดงความสัมพันธ์ระหว่าง Entity เช่น ความสัมพันธ์ระหว่างหลักสูตรวิชาและนักศึกษา ก็เป็นในลักษณะหลักสูตรวิชาที่นักศึกษานั้น ๆ เรียนอยู่ ส่วนความสัมพันธ์ระหว่างนักศึกษากับหลักสูตรวิชา ก็เป็นในลักษณะที่ว่า นักศึกษาเรียนในหลักสูตรวิชานั้น ๆ

#### ☞ คำศัพท์ที่ใช้ในระบบฐานข้อมูลเชิงสัมพันธ์

- รีเลชัน(Relation) มีคำเรียกทั่วไปว่า ตารางข้อมูล(Table) เนื่องจากเป็นคำนามที่แทนข้อมูลของเรื่องใดเรื่องหนึ่ง จึงใช้คำว่า รีเลชัน แทนความหมายของตารางในระบบฐานข้อมูลเชิงสัมพันธ์

- ทูเพิล (Tuple) ค่าของข้อมูลที่อยู่ในแต่ละแถว(Row) หรือที่เรียกกันว่า เรคอร์ด(Record)

- คาร์ดินาลิตี (Cardinality) คือ จำนวนแถวของข้อมูลในแต้รีเลชัน

- แอททริบิวต์(Attribute) คือ รายละเอียดของข้อมูลในแต่ละคอลัมน์(Column)หรือฟิลด์(Field)

- คีย์หลัก(Primary Key) คือ แอททริบิวต์ที่มีค่าของข้อมูลเป็นเอกลักษณ์หรือเฉพาะเจาะจง หรือเป็นค่าที่ไม่ซ้ำกัน ในแต่ละทูเพิล

- โดเมน(Domain) หมายถึง ขอบเขตของค่าของข้อมูล

### 2.4.8 ประเภทของคีย์

เค้าร่างของรีเลชัน(Relation Schema)/ตาราง จะประกอบด้วยรายละเอียดของแอททริบิวต์/คอลัมน์/ฟิลด์ต่าง ๆ ซึ่งแอททริบิวต์ใดแอททริบิวต์หนึ่งจะมีคุณสมบัติเป็นคีย์ โดยคีย์นั้น ๆ สามารถใช้ในการแสดงหรือบ่งบอกถึงค่าของแอททริบิวต์อื่น ๆ ในทูเพิล/เรคอร์ดใดทูเพิล/เรคอร์ดหนึ่งของรีเลชัน/ตารางข้อมูลนั้น ๆ หรือใช้ในการเชื่อมโยงข้อมูลกับข้อมูลอื่นในอีกรีเลชัน/ตารางข้อมูลหนึ่ง

#### ☞ ประเภทของคีย์ อาจแบ่งเป็น 2 ประเภทหลัก คือ

##### 1. คีย์หลัก(Primary Key)

เป็นแอททริบิวต์ที่มีคุณสมบัติของข้อมูลที่เป็นค่าเอกลักษณ์หรือมีค่าที่ไม่ซ้ำซ้อนกัน คุณสมบัติดังกล่าวจะสามารถระบุข้อมูลนั้นเป็นข้อมูลของทูเพิล/เรคอร์ดใด แอททริบิวต์ที่มี

คุณสมบัติเป็นคีย์หลักอาจประกอบด้วยหลายแอททริบิวต์/คอลัมน์/ฟิลด์รวมกัน เพื่อที่จะกำหนดค่าที่เป็นเอกลักษณ์ได้ คีย์หลักที่ประกอบด้วยหลายแอททริบิวต์นี้เรียกว่า คีย์ผสม(Composite Key) นั่นคือเมื่อแอททริบิวต์แต่ละตัวประกอบกันจึงจะให้ค่าที่เป็นเอกลักษณ์หรือไม่ซ้ำซ้อนกันได้

นอกจากนี้ ในรีเลชันหนึ่ง ๆ อาจจะมีแอททริบิวต์ที่มีคุณสมบัติเป็นคีย์หลักได้มากกว่าหนึ่ง

แอททริบิวต์ แอททริบิวต์เหล่านี้เรียกว่า คีย์คู่แข่ง(Candidate Key) ถ้าแอททริบิวต์หนึ่งถูกกำหนดให้เป็นคีย์หลัก อีกแอททริบิวต์หนึ่งที่มีคุณสมบัติเป็นคีย์หลัก แต่ไม่ได้ถูกเลือกให้เป็นคีย์หลักจะเรียกว่า คีย์สำรอง(Alternate Key) คูตารางประกอบ

ตาราง 2.4 แสดงคีย์หลัก(Primary Key)และคีย์สำรอง(Alternate Key)

รหัสหลักสูตร	รหัสนักศึกษา	เลขที่บัตรประชาชน	ชื่อ	นามสกุล	เพศ	วันเดือนปีเกิด
11500000	45510013	3650800711555	ศักดา	ศักดิ์ศรีพาณิชย์	1	12/11/53
11500000	45510047	3650800710456	วุฒิชัย	วิสุทธิพรต	1	10/5/73
11500000	45510062	3650800610789	กชพงศ์	ยณะศักดิ์	1	1/8/78

จากตาราง 2.4 กำหนดให้รหัสหลักสูตร และรหัสนักศึกษา เป็นคีย์หลัก ซึ่งเมื่อประกอบกันแล้วจะให้ค่าที่ไม่ซ้ำซ้อนกันในแต่ละทิวเพิล/เรคอร์ด ส่วนเลขที่บัตรประชาชนจะเป็นคีย์คู่แข่ง เนื่องจากสามารถจะเป็นคีย์หลักได้ เพราะเลขที่บัตรประชาชนเป็นคีย์อีกคีย์หนึ่งที่ไม่ซ้ำกันอยู่แล้ว เมื่อไม่ได้รับเลือกให้เป็นคีย์หลัก เลขที่บัตรประชาชน จึงเป็นคีย์สำรอง

## 2. คีย์นอก(Foreign Key)

เป็นแอททริบิวต์ในรีเลชันหนึ่งที่ใช้ในการอ้างอิงถึงแอททริบิวต์เดียวกันในอีกรีเลชันหนึ่ง โดยที่แอททริบิวต์นี้จะมีคุณสมบัติเป็นคีย์หลักในรีเลชันที่ถูกอ้างอิงถึง การที่มีแอททริบิวต์นี้ปรากฏอยู่ในรีเลชันทั้งสองก็เพื่อประโยชน์ในการเชื่อมโยงข้อมูลซึ่งกันและกันนั่นเอง

ตาราง 2.5 แสดงคีย์คีย์นอก(Foreign Key)

รหัสหลักสูตร	รหัสนักศึกษา	เลขที่บัตรประชาชน	ชื่อ	นามสกุล	เพศ	วันเดือนปีเกิด
11500000	45510013	3650800711555	ศักดา	ศักดิ์ศรีพาณิชย์	1	12/11/53
11500000	45510047	3650800710456	วุฒิชัย	วิสุทธิพรต	1	10/5/73
11500000	45510062	3650800610789	ก้องพงศ์	ยณะศักดิ์	1	1/8/78

รหัสหลักสูตร	รหัสนักศึกษา	เลขที่บัตรประชาชน	ชื่อ	นามสกุล	เพศ	วันเดือนปีเกิด
11500000	45510013	3650800711555	ศักดา	ศักดิ์ศรีพาณิชย์	1	12/11/53
11500000	45510047	3650800710456	วุฒิชัย	วิสุทธิพรต	1	10/5/73
11500000	45510062	3650800610789	ก้องพงศ์	ยณะศักดิ์	1	1/8/78

รหัสหลักสูตร	ชื่อหลักสูตร	จำนวนนักศึกษา
11500000	สาารารณสุขศาสตรบัณฑิต	250
11300000	เทคนิคเภสัชกรรม	160
11100000	สาารารณสุขชุมชน	400

จากตาราง กำหนดให้รหัสของข้อมูลของนักศึกษา และรหัสของข้อมูลของหลักสูตร ซึ่งจะเห็นว่าสองตารางใช้รหัสหลักสูตรเป็นคีย์นอก ซึ่งใช้เชื่อมโยงความสัมพันธ์ระหว่างรหัส โดยที่รหัสหลักสูตรและรหัสนักศึกษาเป็นคีย์หลักของรหัสของข้อมูลของนักศึกษา ส่วนรหัสหลักสูตรเป็นคีย์หลักของรหัสของข้อมูลของหลักสูตร

#### 2.4.9 รูปแบบบรรทัดฐาน(Normal Form)

ในการทำ Normalization นี้ เป็นหลักสำคัญของการออกแบบฐานข้อมูล (Database Design) หรือที่เรียกว่า Logical Database Design นั่นเอง ซึ่งการทำ Normalization ก็คือ วิธีการที่จะบอกให้ทราบว่า รหัสที่ออกแบบนั้น ๆ จะมีปัญหาเกิดขึ้นหรือไม่ และถ้ามีปัญหาเกิดขึ้นจะทำการแก้ไขที่เกิดขึ้นอย่างไร ซึ่ง Normalization เองเป็นเครื่องมือที่จะช่วยให้การออกแบบฐานข้อมูลในลักษณะเป็น Conceptual Schema Design ได้โดยที่จะจัดการปัญหาเกิดขึ้นน้อยที่สุด หรืออาจจะไม่มีปัญหาเลยก็ได้

รูปแบบบรรทัดฐาน สามารถแบ่งออกเป็นรูปแบบต่าง ๆ กันได้ 6 รูปแบบ คือ

1. รูปแบบบรรทัดฐานขั้นที่ 1 (First Normal Form : 1NF)
2. รูปแบบบรรทัดฐานขั้นที่ 2 (Second Normal Form : 2NF)
3. รูปแบบบรรทัดฐานขั้นที่ 3 (Third Normal Form : 3NF)
4. รูปแบบบรรทัดฐานบอยส์และคอดด์(Boyce/Codd Normal Form : BCNF)
5. รูปแบบบรรทัดฐานขั้นที่ 4 (Fourth Normal Form : 4NF)
6. รูปแบบบรรทัดฐานขั้นที่ 5 (Fifth Normal Form : 5NF)

โดยทั่วไปแล้วจะทำรูปแบบบรรทัดฐานขั้นที่ 1-3 เท่านั้น

รูปแบบบรรทัดฐานขั้นที่ 1 (First Normal Form : 1NF)

รีเลชันหนึ่ง ๆ จะอยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 ได้เมื่อค่าของแอททริบิวต์หนึ่งในแต่ละทิวเปิลจะมีค่าของข้อมูล ได้เพียงค่าเดียว หรือ ไม่มีค่าซ้ำกันนั่นเอง

เมื่อมีการกำหนดคีย์หลักเพื่อให้รีเลชันนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 แล้ว แต่อาจจะมีผลผิดพลาดบางอย่างที่จะเกิดขึ้นกับข้อมูลในรีเลชัน ได้แก่

- ความผิดพลาดที่เกิดจากการเพิ่มข้อมูล (Insert Anomaly) หมายถึง การกำหนดแอททริบิวต์ที่เป็นคีย์หลักยังไม่เหมาะสม จากกฎความบูรณาภาพของเอนทิตี สรุปว่า แอททริบิวต์ที่เป็นส่วนของคีย์หลักจะ ไม่มีค่าไม่ได้ (Not Null) เป็นค่าที่เป็นเอกลักษณ์หรือ ไม่มีค่าซ้ำ (Unique)

- ความผิดพลาดที่เกิดจากการลบข้อมูล (Delete Anomaly) หมายถึง รีเลชันที่ประกอบไปด้วยแอททริบิวต์ที่เกินเงื่อนไขในการใช้งานหรือมีแอททริบิวต์มากเกินไป โดยที่ข้อมูลของบางแอททริบิวต์อาจไม่จำเป็นต้องใช้งานแต่จะต้องมาติดกับแอททริบิวต์อื่น ๆ ที่ต้องใช้งาน

- ความผิดพลาดที่เกิดจากการแก้ไขข้อมูล (Update Anomaly) หมายถึง การแก้ไขข้อมูลที่มีอยู่ในรีเลชันจะทำให้เกิดความยุ่งยากและเสียเวลาหากมีข้อมูลซ้ำ ๆ กันอยู่ในรีเลชันเดียวกันหรือต่างรีเลชันกันก็ตาม ทั้งนี้อาจก่อให้เกิดความผิดพลาดขึ้นได้ หากข้อมูลที่ควรจะเหมือนกันแต่กลับไม่เหมือนกัน

ฉะนั้นวิธีการที่จะช่วยแก้ไขปัญหาทั้ง 3 ข้อ คือ การแตกรีเลชันที่มีอยู่เดิมอย่างเหมาะสม เพื่อให้การแก้ไขข้อมูลที่เก็บในรีเลชันสามารถทำการแก้ไขเพียงครั้งเดียว ซึ่งจะลดความซ้ำซ้อนของข้อมูลที่เกิดขึ้น

รูปแบบบรรทัดฐานขั้นที่ 2 (Second Normal Form : 2NF)

รีเลชันนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 1 และมีคุณสมบัติอีก 1 อย่างคือ ทุกแอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะต้องมีความสัมพันธ์ระหว่างค่าของแอททริบิวต์แบบฟังก์ชันกับคีย์หลัก (Full Functional Dependency) หรือก็คือ ค่าของแอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะระบุค่าโดยแอททริบิวต์ที่เป็นคีย์หลักได้ หรือ โดยแอททริบิวต์ทั้งหมดที่ประกอบกันเป็นคีย์หลักในกรณีที่มีคีย์หลัก

เป็นคีย์ผสม

### รูปแบบบรรทัดฐานขั้นที่ 3 (Third Normal Form : 3NF)

รีเลชันนั้น ๆ อยู่ในรูปแบบบรรทัดฐานขั้นที่ 2 และมีคุณสมบัติอีก 1 อย่างคือ แอททริบิวต์ที่ไม่ได้เป็นคีย์หลักจะไม่มีคุณสมบัติที่จะกำหนดค่าของแอททริบิวต์อื่นที่ไม่ใช่คีย์หลัก

ความสัมพันธ์ระหว่างค่าของแอททริบิวต์ที่เป็นแบบทรานซิทีฟ(Transitive Dependency) นั้น ถ้าหากรีเลชันใดมีคุณสมบัตินี้ จะทำให้เกิดความผิดพลาดที่มาจากการลบ และแก้ไขข้อมูลได้คือ

1. ความผิดพลาดที่เกิดจากการแก้ไขข้อมูล (Update Anomaly) ถ้ามีการแก้ไขข้อมูลจะต้องมีการแก้ไขหลาย ๆ ทูเพิล/เรคอร์ด การแก้ไขแบบนี้อาจทำให้เกิดปัญหาการแก้ไขข้อมูลไม่ครบทุกทูเพิล/เรคอร์ด อาจจะทำให้เกิดข้อมูลที่แก้ไขไม่ตรงกันได้

2. ความผิดพลาดที่เกิดจากการลบข้อมูล (Delete Anomaly) ถ้ามีการลบข้อมูลบางข้อมูลออกไปจากรีเลชัน/ตารางหรือทูเพิล/เรคอร์ดใด ๆ จะทำให้ข้อมูลที่มีอยู่ไม่ถูกต้องตามความเป็นจริง การแก้ไขปัญหาคือความผิดพลาดสามารถทำได้โดย วิธีการแตกรีเลชันที่มีอยู่เดิม (Decomposition) ซึ่งการแตกรีเลชัน มีขั้นตอนดังนี้

1. นำเอาแอททริบิวต์ที่เป็นตัวกำหนดค่ากับแอททริบิวต์ที่ถูกกำหนดค่า แยกออกมาสร้างเป็นรีเลชันใหม่
2. และให้แอททริบิวต์ที่เป็นตัวกำหนดค่าแอททริบิวต์อื่น ๆ เป็นคีย์หลักของรีเลชันใหม่ที่สร้างขึ้น

#### 2.4.10 ภาษา SQL (Standard Query Language)

ภาษา Standard relational database Query Language (SQL) เป็นภาษามาตรฐานสำหรับระบบฐานข้อมูล ซึ่งเป็นภาษาที่พัฒนาขึ้นมาโดยบริษัทไอบีเอ็ม(IBM) ภาษา SQL เป็นส่วนหนึ่งของระบบฐานข้อมูลแบบรีเลชันเนล (Relational Database) ที่ได้รับความนิยมมากเพราะง่ายต่อความเข้าใจ และอยู่ในรูปภาษาอังกฤษ

ภาษาSQL แบ่งออกเป็น 2 ส่วน คือ

- ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL)
- ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language: DML)
  - ภาษาควบคุม (Control Language)
  - ภาษาในการเลือกข้อมูล (Data Query Language)



รูปแบบการใช้คำสั่ง SQL สามารถใช้ได้เป็น 2 รูปแบบ ดังนี้ คือ

- คำสั่ง SQL ที่ใช้เรียกดูข้อมูลได้ทันที (Interactive SQL)เป็นการเรียกใช้คำสั่ง SQLสั่งงานบนจอภาพ เพื่อเรียกดูข้อมูลในขณะที่ทำงานได้ทันที เช่น

---

```
SELECT CITY
```

```
FROM SUPPLIER
```

```
WHERE SNO = 'SE';
```

---

- คำสั่ง SQL ที่ใช้เขียนร่วมกับโปรแกรมอื่นๆ (Embedded SQL)เป็นคำสั่ง SQL ที่ใช้ร่วมกับคำสั่งของโปรแกรมภาษาต่างๆ เช่น PL/I PASCAL ฯลฯ หรือแม้แต่ว่ากับคำสั่งในโปรแกรมที่ระบบจัดการฐานข้อมูลนั้นมีใช้เฉพาะ เช่น ORACLE มี PL/SQL (Procedural Language /SQL)ที่สามารถเขียนโปรแกรมและนำคำสั่ง SQL มาเขียนร่วมด้วย เป็นต้น

☞ ภาษาที่ใช้สำหรับนิยามข้อมูล (Data Definition Language-DDL )

Data Definition Language (DDL) เป็นภาษาที่ใช้นิยามโครงสร้างข้อมูล เพื่อเปลี่ยนแปลงหรือยกเลิกโครงสร้างฐานข้อมูลตามที่ออกแบบไว้ โครงสร้างดังกล่าวคือ สคีมา (Schema) นั้นเอง

ภาษา DDLประกอบด้วย 3คำสั่งคือ

1. คำสั่งการสร้าง (Create) ได้แก่ การสร้างตารางและอินเด็กซ์
2. คำสั่งเปลี่ยนแปลงโครงสร้าง
- 3.คำสั่งยกเลิก(Drop) ต่างๆ

การสร้างตารางในฐานข้อมูลแบบรีเลชันเนล โดยเฉพาะฐานข้อมูลขนาดใหญ่บนระบบ UNIX จะทำการป้อนคำสั่งในลักษณะเท็กซ์โหมด (Text Mode) เข้าไปในระบบฐานข้อมูล ดังรูปแบบต่อไปนี้

- คำสั่งการสร้างตาราง

```
CREATE TABLE <ชื่อตาราง>
```

```
(<ชื่อคอลัมน์ ประเภทของข้อมูล>[,<ชื่อคอลัมน์ ประเภทของข้อมูล>]....);
```

---

ประเภทของข้อมูลแบ่งเป็น 5 ประเภทใหญ่ๆ ขึ้นอยู่กับซอฟต์แวร์ระบบฐานข้อมูลที่ใช้ว่า

คืออะไร ตัวอย่างเช่น CHAR, INTEGER, DATE ฯลฯ

- คำสั่งการลบโครงสร้างตาราง

```
DROP TABLE <ชื่อตารางที่ต้องการลบ>
```

- คำสั่งการเปลี่ยนแปลงโครงสร้างตาราง

ALTER TABLE <ชื่อตารางที่ต้องการเปลี่ยนแปลง>

<คำสั่งการเปลี่ยนแปลง> [,<ชื่อคอลัมน์ประเภทของข้อมูล>]

- คำสั่งดัชนี

ดัชนี ( INDEX ) มีความสำคัญมากต่อฐานข้อมูลเชิงสัมพันธ์ เนื่องจากระบบฐานข้อมูลแบบรีเลชันเนล (RDBMS) จะใช้ดัชนีในการค้นหาระเบียบที่ค้นหาได้อย่างรวดเร็ว โดยดัชนีที่ถูกสร้างขึ้นจะเก็บไว้แยกจากตารางในพื้นที่ต่างหากของคอมพิวเตอร์ โดยปกติ ถ้าไม่มีการประกาศดัชนีไว้การค้นหาข้อมูลในตาราง นั้นจะต้องทำแบบเรียงลำดับจากแถวที่หนึ่งจนถึงแถวสุดท้าย การสร้างดัชนีสำหรับตารางใดๆ จะทำได้โดยการเลือกคอลัมน์ใดคอลัมน์หนึ่งจากตารางมาเป็นดัชนี และตารางหนึ่งๆ สามารถมีได้หลายดัชนี นอกจากเพิ่มความรวดเร็วในการดึงข้อมูลแล้วยังสามารถนำไปใช้ในการควบคุมคอลัมน์ ที่นำมาสร้างเป็นดัชนีให้มีการเก็บข้อมูลที่ไม่ซ้ำกัน (Unique) อีกด้วย

การสร้างดัชนีจะใช้คำสั่ง CREATE INDEX แล้วตามด้วยชื่อดัชนีที่เราตั้งขึ้น ดังรูปแบบต่อไปนี้

```
CREATE [UNIQUE] INDEX <ชื่อตารางที่ตั้งขึ้น _____>
ON (<ชื่อตารางที่สร้างดัชนี> (<ชื่อคอลัมน์_1> [,<ชื่อคอลัมน์_2>]...);
```

- การลบดัชนี

เมื่อต้องการลบดัชนีที่สร้างขึ้น ก็สามารถทำได้ด้วยคำสั่ง DROP INDEX แล้วตามด้วยชื่อดัชนีที่ต้องการลบ ดังรูปแบบดังนี้

```
DROP INDEX <ชื่อดัชนี>
```

### ☞ ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML)

หลังจากที่เราสร้างโครงสร้างฐานข้อมูลขึ้นแล้ว คำสั่งต่อไปในการป้อนข้อมูลลงในฐานข้อมูล และเปลี่ยนแปลงข้อมูลในฐานข้อมูล โดยการใช้ภาษาสำหรับการจัดการข้อมูล (Data Manipulation Language-DML) ใช้จัดการข้อมูลภายในตารางภายในฐานข้อมูล และภาษาแก้ไขเปลี่ยนแปลงตาราง แบ่งออกเป็น 4 Statement คือ

- Select Statement : การเรียกหา(Retrieve) ข้อมูลจากฐานข้อมูล
- Insert Statement : การเพิ่มเติมข้อมูลลงในตาราง(Table) จากฐานข้อมูล
- Delete Statement: การลบข้อมูลลงออกจากตาราง(Table) จากฐานข้อมูล

• Update Statement: การเปลี่ยนแปลงข้อมูลลงในตาราง (Table) จากฐานข้อมูล

- คำสั่ง ความหมาย

SELECT เรียกค้นข้อมูลในตาราง

~~INSERT~~ เพิ่มแถวข้อมูลลงในตาราง

DELETE ลบแถวข้อมูล

UPDATE ปรับปรุงแถวข้อมูลในตาราง

- คำสั่งค้นหาข้อมูล (Query Statement)

คำสั่ง SELECT เป็นคำสั่งการเรียกดูข้อมูลหรือ ค้นข้อมูลตามเงื่อนไขที่ระบุ เนื่องจากคำสั่ง SELECT เป็นคำสั่งที่มีรูปแบบการใช้งานที่ง่ายเพื่อช่วยในการค้นหาข้อมูลที่ซับซ้อน ดังมีรูปแบบดังนี้

SELECT <ชื่อคอลัมน์ที่ต้องการดูข้อมูล>

FROM <ชื่อตาราง>

WHERE <เงื่อนไขตามที่ระบุ>

SELECT --- เป็นคำสั่งให้ทำการเรียกดูข้อมูลในคอลัมน์ที่ระบุ ซึ่งอาจจะมากกว่า หนึ่งก็ได้ และถ้ามีมากกว่าหนึ่งคอลัมน์ต้องค้นด้วย คอมม่า (,) และนอกจากนี้ยังสามารถใช้เครื่องหมายดอกจัน (\*) เพื่อแสดงถึงการขอข้อมูลทั้งหมดได้อีกด้วย

FROM --- เป็นคำ ส่วนประกอบของคำสั่งที่บอกถึงตารางที่ต้องการดู ซึ่งอาจจะมีมากกว่า หนึ่งตารางก็ได้ ที่จะถูกเรียกใช้จากคำสั่ง SELECT

WHERE --- เป็นส่วนประกอบของคำสั่ง ที่ใช้บ่งบอกเงื่อนไขที่จะใช้ในการค้นหาข้อมูล ขึ้นมาจากตารางใดๆที่อยู่หลัง FROM นี้

- คำสั่งเติมข้อมูล (Insert Statement)

INSERT INTO <ชื่อตาราง >

VALUES (<ชื่อคอลัมน์ 1> [<ชื่อคอลัมน์ 2>]...);

- คำสั่งแก้ไขและลบแถว (Update Statement )

UPDATE <ชื่อตาราง >

SET <ค่าที่ต้องการ>

WHERE <เงื่อนไข>

#### 2.4.11 Microsoft SQL Server

Microsoft SQL Server เป็นโปรแกรมจัดการฐานข้อมูลระดับเซิร์ฟเวอร์ที่มีขีดความสามารถในการรองรับข้อมูลขนาดใหญ่ที่ช่วยให้การบริหารจัดการฐานข้อมูลมีประสิทธิภาพ รวดเร็ว สนองต่อความต้องการขององค์กรขนาดใหญ่

Microsoft SQL Server เป็นระบบฐานข้อมูลและโซลูชันการวิเคราะห์ที่สมบูรณ์แบบ ซึ่งนำเสนอความน่าเชื่อถือและประสิทธิภาพ ด้านการขยายระบบที่เว็บและองค์กรธุรกิจต้องการ ด้วยการรองรับ XML และ HTTP ทำให้การเข้าถึงและการแลกเปลี่ยนข้อมูลนั้นทำได้ง่ายขึ้น ในขณะที่ความสามารถในการวิเคราะห์อันทรงพลังยังช่วยเพิ่มคุณค่าของข้อมูล และด้วยความพร้อมของระบบที่ดีขึ้น ทำให้ความสามารถในการทำงานอย่างต่อเนื่อง ของระบบนั้นสูงขึ้น, จัดการกับงานในแต่ละวัน โดยอัตโนมัติได้ดียิ่งขึ้น รวมทั้งปรับปรุงเครื่องมือด้านการเขียน โปรแกรม และการพัฒนา ความเร็วด้านการบริการ ได้ดีขึ้นเช่นกัน

Microsoft SQL Server ประกอบไปด้วยเวอร์ชันต่างๆ ดังต่อไปนี้

- **Personal Edition** เป็นระบบจัดการฐานข้อมูลที่ใช้ได้เฉพาะเครื่องส่วนตัวเท่านั้น () ไม่สามารถใช้งานร่วมกับ Microsoft SQL Server เวอร์ชันอื่นได้โดยตรง ส่วนใหญ่นิยมนำมาใช้ในการทดสอบการเขียนแอปพลิเคชัน ใช้ติดตั้งบน Windows 9x, ME, XP

- **Standard Edition** เป็นระบบจัดการฐานข้อมูลที่สามารถรองรับเซิร์ฟเวอร์ได้สูงสุดถึง 4 CPU และอ้างหน่วยความจำได้ถึง 2 GB นิยมนำมาใช้งานกับองค์กรขนาดใหญ่ ต้องติดตั้งบน Windows NT 4, 2000 หรือ 2003 Server แต่ถ้าติดตั้งบน Windows 9x, ME, XP จะติดตั้งให้เฉพาะส่วนที่เป็น Client Tools หรือ Connectivity เท่านั้น

- **Enterprise Edition** เป็นระบบจัดการฐานข้อมูลที่สามารถรองรับเซิร์ฟเวอร์ได้สูงสุดถึง 32 CPU และอ้างหน่วยความจำได้ถึง 64 GB นิยมนำมาใช้งานกับองค์กรที่มีขนาดใหญ่และมีสาขามากมาย เพราะมีระบบสนับสนุนการทำงานมากมาย เช่น การทำ Data Mining, Data Warehouse เป็นต้น ต้องติดตั้งบน Windows NT 4, 2000 หรือ 2003 Server เท่านั้น

- **.NET Microsoft SQL Server** เป็นโครงสร้างหลักของ การจัดการและวิเคราะห์ข้อมูล สำหรับ แอปพลิเคชัน และบริการของ Microsoft .NET ที่จะมีขึ้นในอนาคต

#### 2.5 เทคโนโลยีการติดต่อกับฐานข้อมูลของ Microsoft

Microsoft ได้พัฒนาเทคโนโลยีการติดต่อกับระบบฐานข้อมูลสำหรับ โปรแกรมและเครื่องมือพัฒนาระบบต่างๆมาหลายอย่างด้วยกัน ได้แก่

- **ADO (ActiveX Data Object)** เป็นเทคโนโลยีการติดต่อกับระบบฐานข้อมูลที่ทางบริษัท Microsoft ได้พัฒนาขึ้นมาใหม่เพื่อให้ผู้ใช้สามารถเขียน โปรแกรมเพื่อติดต่อกับระบบฐานข้อมูลหรือระบบข้อมูลต่างๆไม่ว่าจะเป็นข้อมูล E-mail ข้อมูลภาพและอื่นๆได้โดยสะดวก รวดเร็วและดีขึ้น

กว่าเทคโนโลยีเดิมของ Microsoft ที่ใช้ติดต่อกับฐานข้อมูลเท่านั้น ซึ่งได้แก่ DAO, ODBC และ RDO

-DAO (Data Access Object) ถูกสร้างขึ้นเพื่อให้โปรแกรมสามารถติดต่อกับฐานข้อมูลของ Microsoft Access ได้ โดย DAO จะเป็นออบเจกต์ที่ใช้ติดต่อกับ Jet Database Engine ซึ่งเป็นเสมือนหัวใจหรือเครื่องยนต์ของ MS Access ด้วยความที่ MS Access มีผู้ใช้จำนวนมากจึงทำให้ DAO มีการใช้งานอย่างแพร่หลาย แต่ DAO มีความสามารถจำกัดอยู่ที่ Microsoft Access เท่านั้น

-ODBC (Open Database Connectivity) ถูกสร้างขึ้นสำหรับพัฒนาระบบ Client-Server โดยทำให้สามารถติดต่อกับฐานข้อมูลชนิดต่างๆ ได้หลากหลาย ซึ่ง ODBC จะทำตัวเป็นเสมือนตัวกลางเชื่อมโยงระหว่างโปรแกรมที่เราใช้งานกับระบบฐานข้อมูลที่อยู่ข้างหลังในปัจจุบัน ODBC เป็นเสมือนมาตรฐานกลางของการติดต่อกับฐานข้อมูลที่เป็น Relational database

-RDO (Remote Data Object) ถูกสร้างขึ้นเพื่อให้สามารถติดต่อกับฐานข้อมูลในระบบ Client-Server ได้อย่างสะดวกและง่ายกว่าการเขียนโปรแกรมติดต่อกับ ODBC โดยตรง กล่าวโดยสรุปก็คือ ADO เป็นออบเจกต์ใหม่และเทคโนโลยีใหม่ที่ทาง Microsoft พัฒนาขึ้น เพื่อให้โปรแกรมเมอร์เขียนโปรแกรมติดต่อกับฐานข้อมูลต่างๆ ได้สะดวกขึ้นง่ายขึ้น และมีประสิทธิภาพเพิ่มมากขึ้นกว่า DAO และ RDO ที่มีอยู่เดิม นอกจากนี้ ADO ยังเป็นเครื่องมือที่ไม่ขึ้นกับภาษา (Language Independence) นั่นหมายถึงว่าเราสามารถใช้คำสั่งหรือไวยากรณ์ของ ADO ที่เหมือนกันเสมอไม่ว่าจะเรียกใช้จากภาษา Visual Basic, Visual C++, Visual J++ หรือ VB Script ก็ตาม

## 2.6 ไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ (microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่มีบรรจุความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณออกทางเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกา ทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรรีเลย์อิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดีโดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลง ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ “ไมโคร” (micro) ซึ่งหมายถึงไมโครโพรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ซึ่งภายในประกอบด้วยหน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU : Arithmetic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรเชื่อมต่อสัญญาณนาฬิกาอีกคำหนึ่งก็คือคำว่า “คอนโทรลเลอร์” (controller) หมายถึงอุปกรณ์ควบคุม

ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

### 2.6.1 โครงสร้างของ MCS-51

~~ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายหมายเลขขึ้นอยู่กับโครงสร้าง~~  
ภายในบางหมายเลขจะมีหน่วยความจำภายในเป็นแบบ ROM บางเบอร์เป็นแบบ EPROM บางหมายเลขมี RAM ภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ เป็นต้น ซึ่งรายละเอียดจะศึกษาได้จากคู่มือโดยตรง และลักษณะของขาต่าง ๆ จะเหมือนกัน คุณสมบัติที่สำคัญของ MCS-51 มีดังนี้

- มีหน่วยความจำ ROM 4K bytes
- มีหน่วยความจำ RAM 128 bytes
- มีพอร์ต I/O ขนาด 8 บิต 4 พอร์ต
- มี Timer 16 บิต 2 ตัว
- สามารถอินเทอร์รัพท์ได้ 5 แหล่ง
- มีวงจรรอสซิงเกิลเตอร์และวงจรรนาฬิกาบนชิป
- มีพอร์ตอนุกรมที่สามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
- อ้างหน่วยความจำโปรแกรมภายนอกได้ 64K
- อ้างหน่วยความจำข้อมูลที่ละบิตได้ 64K
- สามารถประมวลผลที่ละบิตได้
- สามารถอ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
- หนึ่งวัฏจักรคำสั่งกินเวลาประมาณ 1 ไมโครวินาที ขณะทำงานด้วย Clock 12 MHz

## บทที่ 3

### ขั้นตอนการดำเนินโครงการ

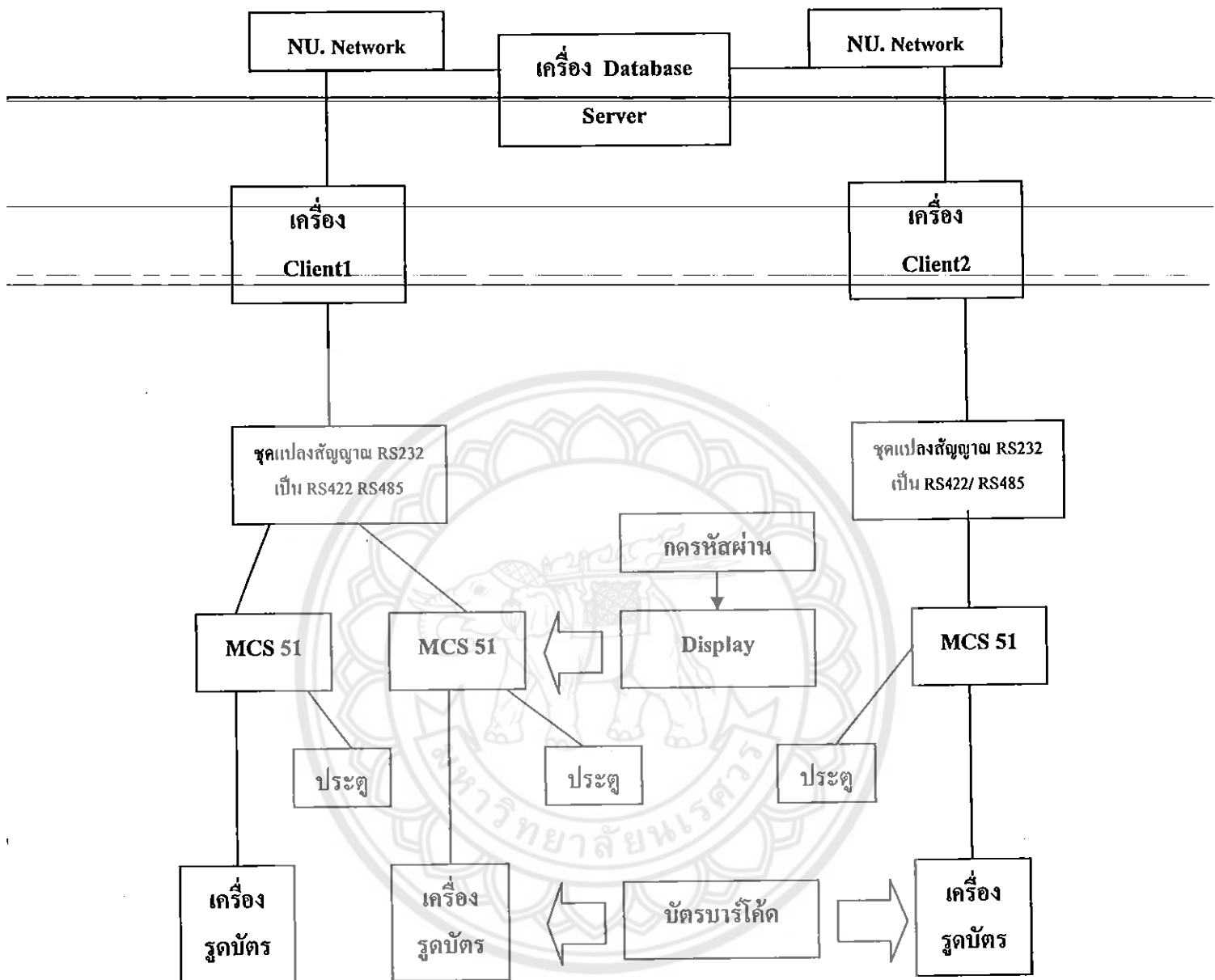
#### 3.1 ขั้นตอนการดำเนินงาน

1. ศึกษาและรวบรวมข้อมูลเกี่ยวกับโครงสร้างและการทำงานของ Database (Microsoft SQL Sever) ,ระบบ Network ,โปรแกรม Visual Basic ,ระบบ Client-Sever ,ไมโครคอนโทรลเลอร์ตระกูล MCS 51 การอินเตอร์เฟสแบบ RS232, RS422 /RS485
2. ออกแบบ Database โดยใช้ Microsoft SQL Sever 7.0 และออกแบบหน้าตาของโปรแกรม โดยใช้โปรแกรม Visual Basic 6.0
3. เชื่อมต่อระบบ Lan เข้าด้วยกัน
4. เชื่อมต่อตัวโปรแกรมเข้ากับ Database
5. ประกอบวงจรแปลงการรับ-ส่ง ข้อมูลจาก RS232 ไปเป็น RS422/RS485
6. เขียนโปรแกรมทดสอบการทำงานของบอร์ดควบคุม และทดลองวงจรแปลงการรับ-ส่ง ข้อมูลจาก RS232 ไปเป็น RS422/RS485

#### 3.2 ภาพรวมของระบบตรวจสอบการเข้า-ออกอาคารผ่านเครือข่าย

จากแผนภาพ เมื่อนำบัตรที่มีบาร์โค้ดไปให้เครื่องรูดบาร์โค้ดอ่านแล้วครหัสผ่านที่ตัวบอร์ดควบคุม ข้อมูลของรหัสบาร์โค้ดและข้อมูลของรหัสผ่านจะถูกส่ง โดยผ่าน ชุดแปลงสัญญาณ RS232 เป็น RS422/RS485 ไปยังเครื่องคอมพิวเตอร์ที่เป็นเครื่อง Client1 หรือ เครื่อง Client2 เพื่อนำข้อมูลไปตรวจสอบกับฐานข้อมูล โดยผ่านทาง Network ที่บนเครื่อง Database Sever ถ้าข้อมูลถูกต้อง ข้อมูลของผู้ใช้จะถูกเก็บไว้ที่ตารางของ Database ที่ชื่อ Event แล้วส่งสัญญาณกลับผ่านทาง Network ไปยังที่ตัวบอร์ดควบคุม เพื่อสั่งให้ประตูเปิด

กรณีการเช็คออกจากห้อง เพียงนำบัตรที่มีบาร์โค้ดเดิมไปให้เครื่องรูดบาร์โค้ดอ่าน แล้วเดินผ่าน Sensor บอร์ดจะส่งข้อมูล ไปให้คอมพิวเตอร์เพื่อเก็บเวลาออกไว้ใน Database แล้วส่งสัญญาณกลับ ไปยังที่ตัวบอร์ดควบคุม เพื่อสั่งให้ประตูเปิด



รูปที่ 3.1 การทำงานของระบบเข้า-ออกอาคารผ่านเครื่องจ่าย



จากรูปที่3.1 จะมีเครื่องคอมพิวเตอร์ที่เป็น Server 1เครื่องเอาไว้สำหรับเก็บDatabase และ  
เช็คข้อมูลต่างๆ เครื่องClientมี 2เครื่อง โดยเครื่อง Client ทั้ง2เครื่องจะติดต่อกับชุดแปลงสัญญาณ  
RS232 เป็น RS422/RS485 โดยต่อกับเครื่องรูดบัตร โดยผ่าน MCS 51

---

#### เช็คเข้า

เมื่อนำบัตรที่มีบาร์โค้ดมารูดผ่านเครื่องรูดบัตร บอร์ดควบคุมก็จะรวบรวมข้อมูลจาก Key Pad  
เมื่อมีการกดPassword ที่Key Pad บอร์ดควบคุมจะทำการส่งข้อมูลไปให้คอมพิวเตอร์ข้อมูลนี้ก็คือ

start bit + ID + Barcode + Password + stop bit

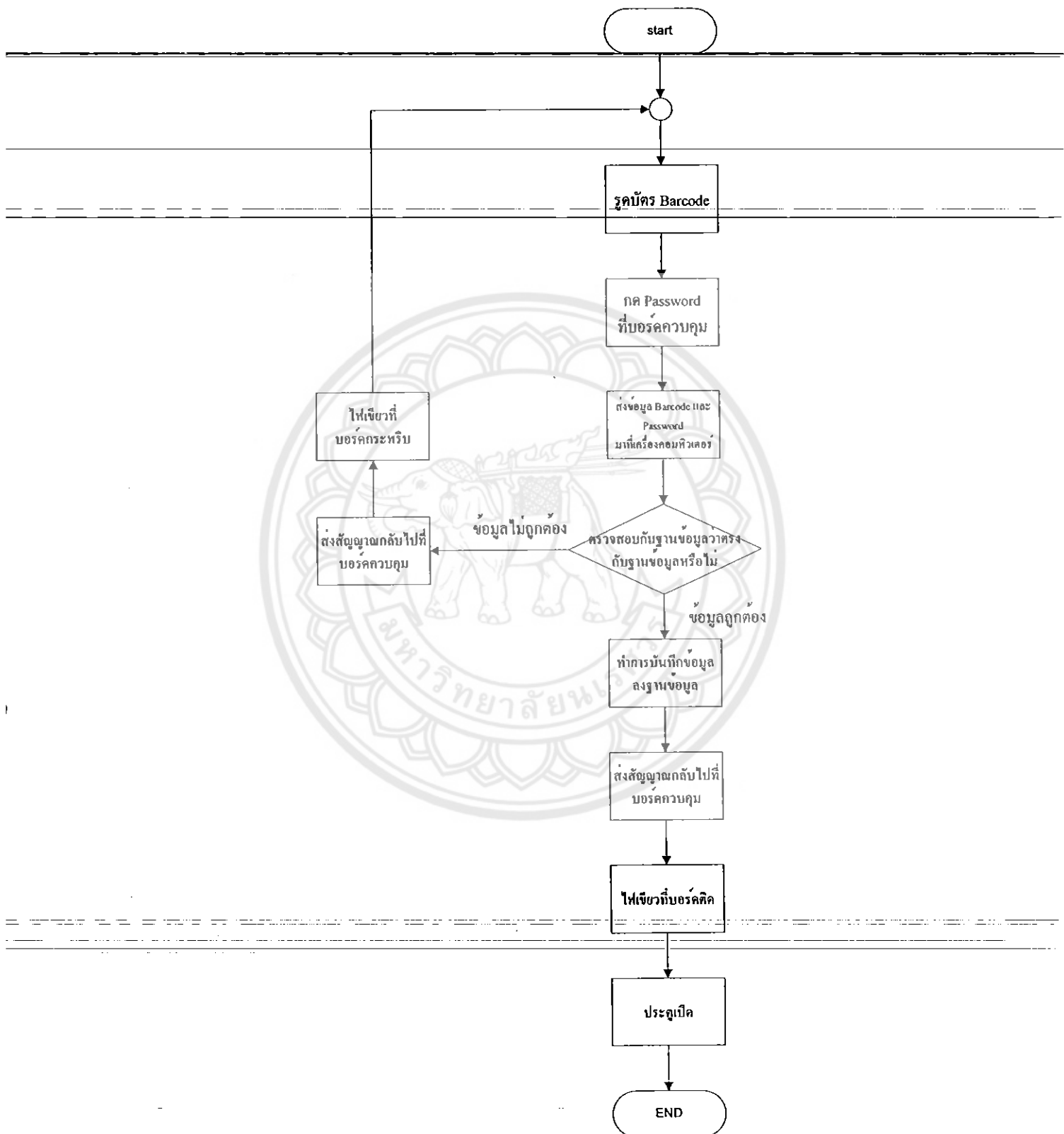
---

#### เช็คออก

เมื่อมีการอ่านบาร์โค้ดเข้ามาแล้ว sensor ทำงาน บอร์ดจะรู้ว่าเป็นการเช็คออก และใน  
ระหว่างนั้นบอร์ดจะส่งข้อมูลไปให้คอมพิวเตอร์ ข้อมูลก็คือ start bit + ID + Barcode ++ stop bit



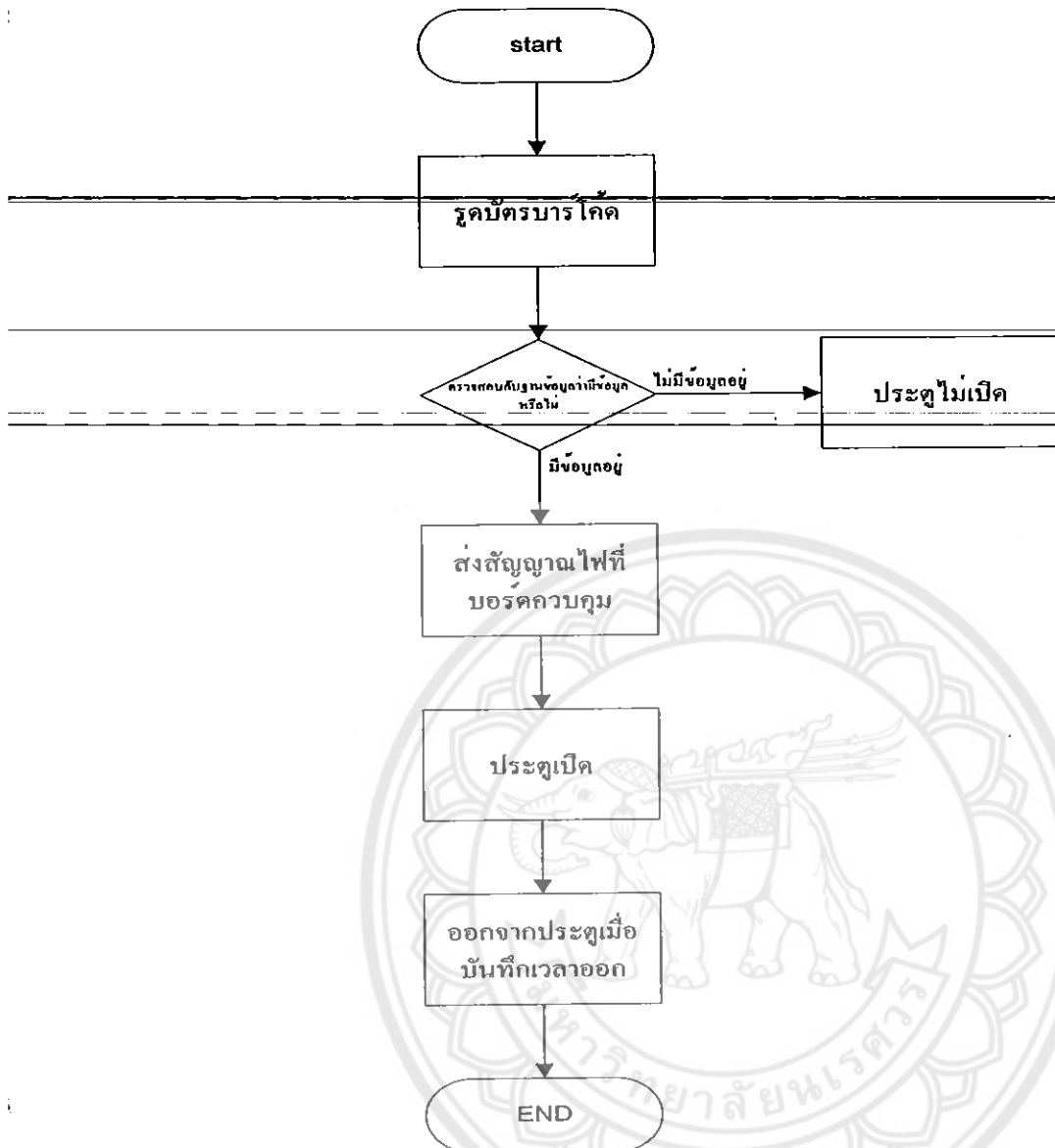
### 3.3 การทำงานของระบบตรวจสอบการเข้า-ออกอาคารผ่านเครือข่าย



รูปที่ 3.2 Flowchart ทำงานการเข้าอาคาร

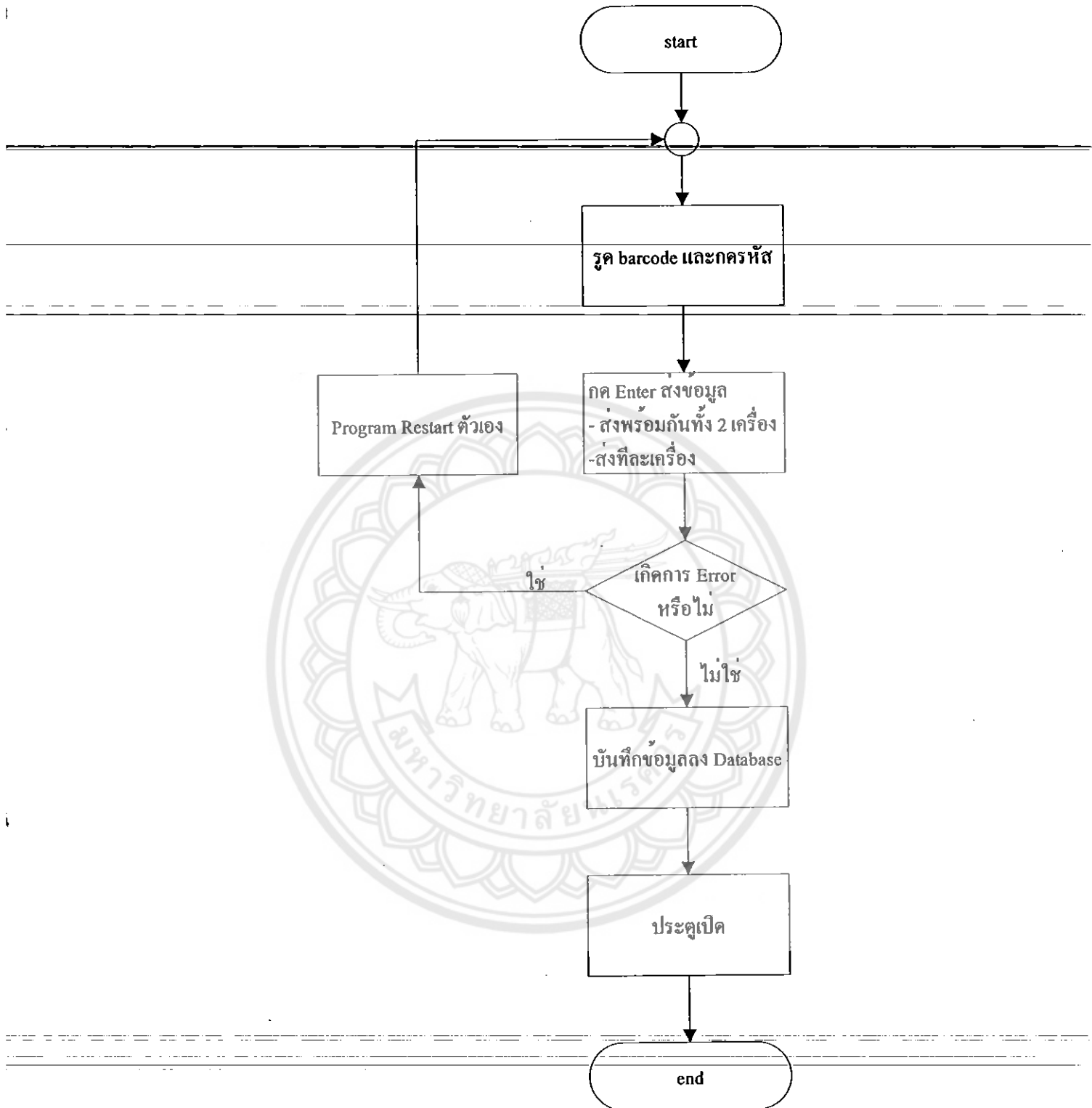
จากรูปที่ 3.2 เมื่อเราต้องการเข้ามาภายในอาคารจะต้องทำการรูดบัตรและกรหัสผ่านที่อยู่บนบอร์ดควบคุมซึ่งรหัสผ่านจะเป็นที่หลักก็ได้แล้วแต่เราจะตั้งไว้ที่ฐานข้อมูลแล้วกดเครื่องหมาย”#” บอร์ดควบคุมก็จะส่งข้อมูล Barcode และรหัสผ่านไปยังเครื่อง Server ที่เก็บฐานข้อมูลไว้ Server ก็จะทำการตรวจสอบว่าข้อมูล Barcode และรหัสผ่านถูกต้องหรือไม่ถ้าถูกต้องเครื่อง Server ก็จะส่งข้อมูลไปที่บอร์ดควบคุมและเครื่อง Client ทั้ง 2 เครื่องเพื่อสั่งให้ประตูเปิดและแสดงข้อมูลตามลำดับแต่ถ้าข้อมูล Barcode และรหัสผ่านไม่ถูกต้องเครื่อง Server ก็จะส่งข้อมูลมายังบอร์ดควบคุมและไฟสีเขียวบนบอร์ดก็จะกะพริบประตูก็จะไม่เปิดต้องทำการรูดบัตรและใส่รหัสผ่านใหม่





รูปที่ 3.3 Flowchart การทำงานการออกอาคาร

จากรูปที่ 3.3 เมื่อเราต้องการจะออกจากอาคารให้ทำการรูดบัตรข้อมูลก็จะผ่านไปที่บอร์ดควบคุมแล้วบอร์ดควบคุมก็จะส่งข้อมูลไปยังเครื่อง Server แล้วเครื่อง Server ก็จะทำการเช็คข้อมูล Barcode ที่อยู่ในฐานข้อมูลว่าถูกต้องหรือไม่ ถ้าข้อมูลถูกต้องตัว Sensor ก็จะทำงานเมื่อเราเดินผ่านตัว Sensor ประตูก็จะเปิดออก แต่ถ้าข้อมูลไม่ถูกต้องตัว Sensor ก็จะไม่ทำงานประตูก็จะไม่เปิดออกก็จะไม่สามารถออกมาภายนอกอาคารได้



รูปที่ 3.4 Flowchart การทำงานเมื่อเกิดการชนกันของข้อมูล (Error)

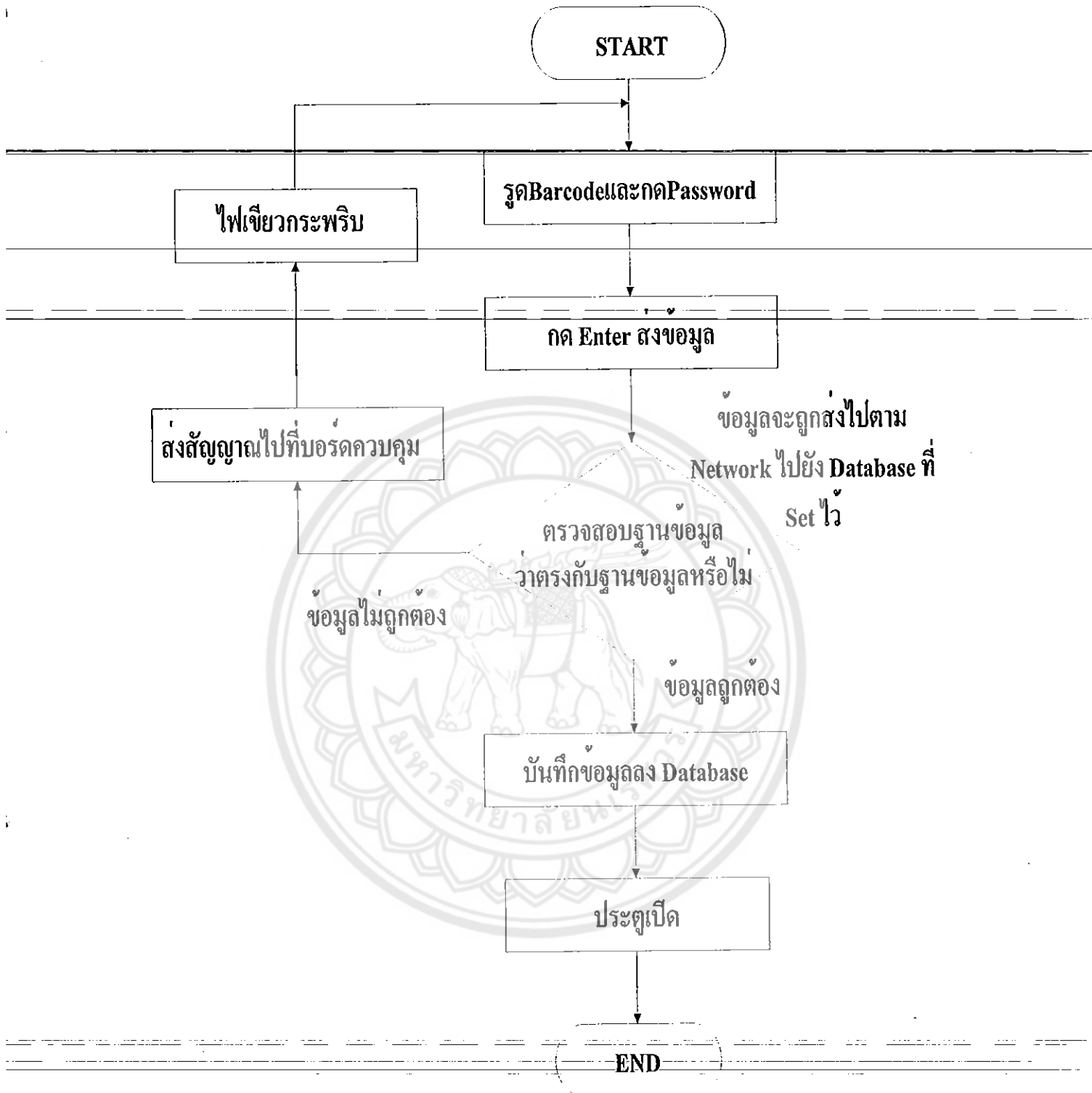
จากรูปที่ 3.4 เมื่อทำการรีบูตและกดรหัสผ่านในกรณีรีบูตพร้อมกัน 2 เครื่อง ถ้าส่งพร้อมกัน 2 เครื่องในเวลาเดียวกันถ้าบอร์ดควบคุมเกิดการ Error โปรแกรมก็จะ Restart ตัวเองต้องทำการรีบูตใหม่แต่ถ้ารีบูต 2 เครื่องในเวลาใกล้เคียงกัน โปรแกรมจะมีการหน่วงเวลาและทำการส่งข้อมูลออกไปทีละเครื่องไปที่เครื่อง Server เพื่อทำการเช็คข้อมูลต่อไป

---

---

---





รูปที่ 3.5 Flowchart การทำงานผ่าน network

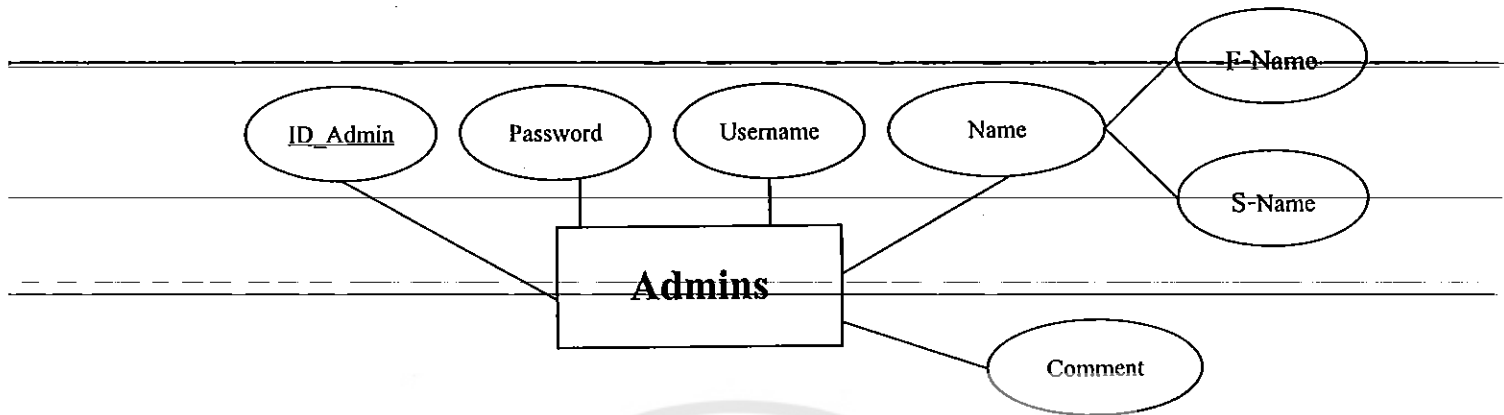
จากรูปที่ 3.5 ถ้าเครื่อง Server อยู่อีกที่และเครื่อง Client ทั้ง 2 เครื่องอยู่อีกที่เช่นถ้าเครื่อง Server อยู่ที่ตึก EN ห้อง EN 116 และเครื่อง Client ทั้ง 2 เครื่อง อยู่ที่ตึก EE เครื่อง Client 1 อยู่ที่ห้อง EE 513 และเครื่อง Client 2 อยู่ที่ห้อง EE 509 ก็จะสามารถติดต่อกันโดยผ่านระบบเครือข่ายของมหาวิทยาลัยหรือระบบ Lan เมื่อทำการรูดบัตรและครรหัสผ่านข้อมูลก็จะถูกส่งไปยังเครื่อง Server โดยผ่านระบบเครือข่ายของมหาวิทยาลัยหรือระบบ Lan เพื่อทำการเช็คข้อมูลว่าตรงกับในฐานข้อมูลหรือไม่ถ้าข้อมูลถูกต้องเครื่อง Server ก็จะส่งข้อมูลไปยังเครื่อง Client ทั้ง 2 เครื่อง โดยผ่านระบบเครือข่ายของมหาวิทยาลัยหรือระบบ Lan เช่นกันและเครื่อง Client ทั้ง 2 เครื่องก็จะส่งข้อมูลไปยังบอร์ควบคุมเพื่อทำการเปิดประตูและทำการแสดงข้อมูลผู้เข้าอาคารที่หน้าของเครื่อง Client ทั้ง 2 เครื่องแต่ถ้าข้อมูลไม่ถูกต้องเครื่อง Server ก็จะส่งข้อมูลกลับ ไปยังเครื่อง Client ทั้ง 2 เครื่อง โดยผ่านระบบเครือข่ายของมหาวิทยาลัยหรือระบบ Lan อีกเช่นกัน เครื่อง Client ทั้ง 2 เครื่องก็จะส่งข้อมูลไปยังบอร์ควบคุมประตูก็จะไม่เปิดออก





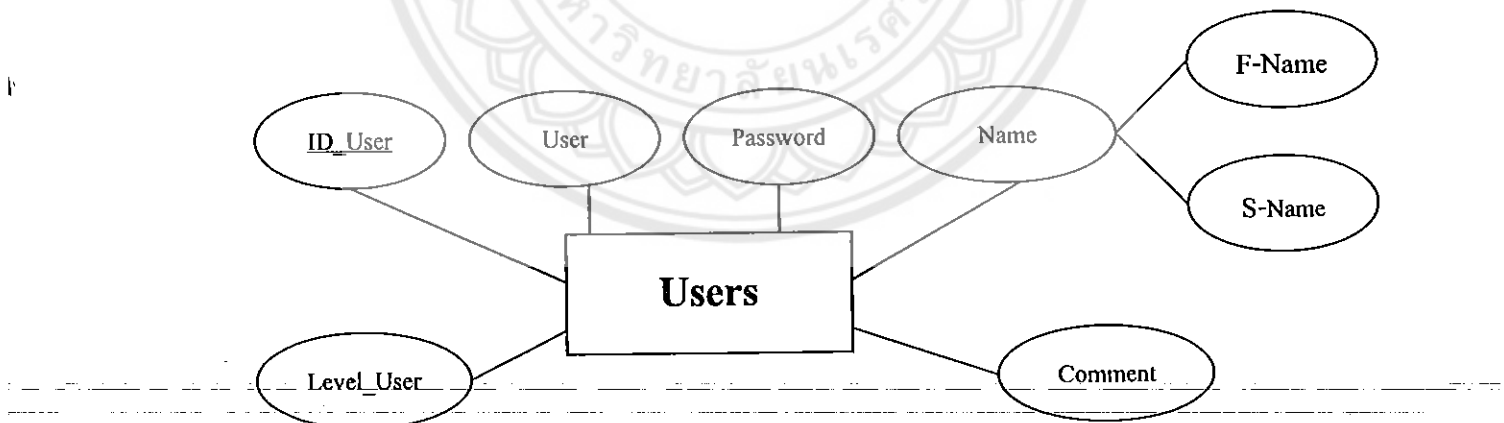
### 3.4 การออกแบบ Database

#### 3.4.1 การออกแบบ



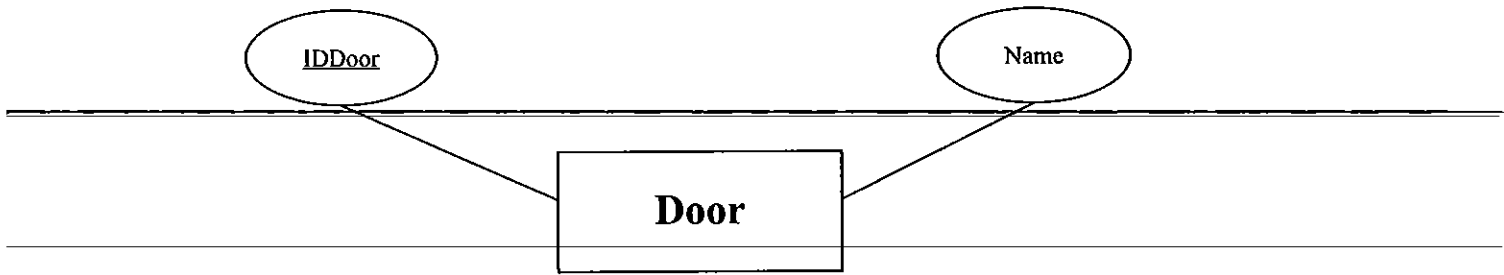
รูปที่ 3.6 การออกแบบตาราง Admins

จากรูปที่ 3.6 ในตาราง Admins จะประกอบไปด้วย Column ID\_Admin, Password, Username, Name และ Comment โดยที่ ID\_Admin เป็น Primary Key



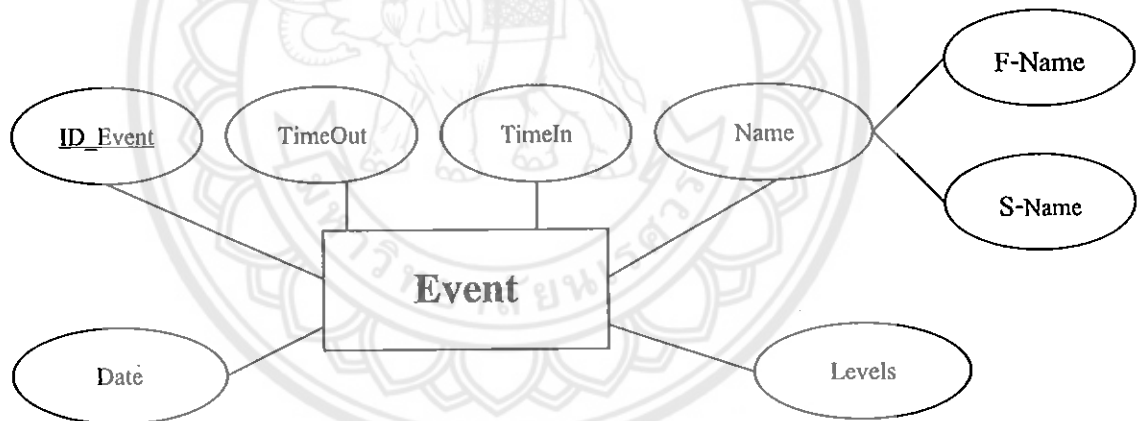
รูปที่ 3.7 การออกแบบตาราง Users

จากรูปที่ 3.7 ในตาราง User จะประกอบไปด้วย Column ID\_User, User, Password, Name, Level\_User และ Comment โดยที่ ID\_User เป็น Primary Key



รูปที่ 3.8 การออกแบบตาราง Door

จากรูปที่ 3.8 ในตาราง Door จะประกอบไปด้วย Column IDDoor และ Name โดยที่ IDDoor เป็น Primary Key

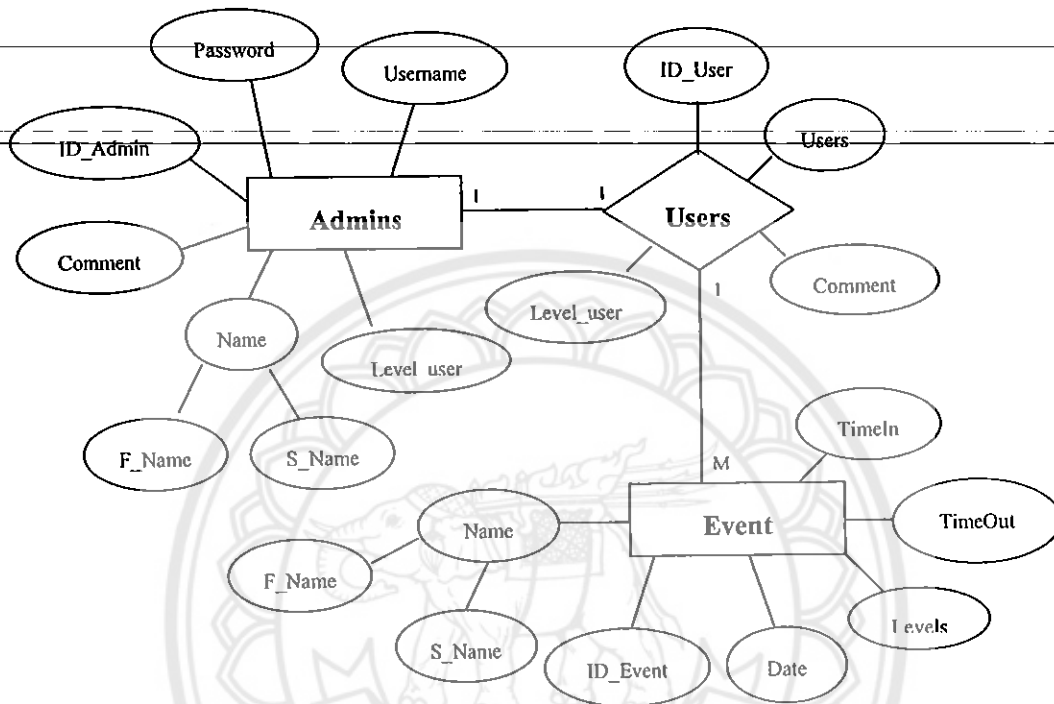


รูปที่ 3.9 การออกแบบตาราง Event

จากรูปที่ 3.9 ในตาราง Event จะประกอบไปด้วย Column ID\_Event, TimeOut, TimeIn, Name, Levels และ Date โดยที่ ID\_Event เป็น Primary Key

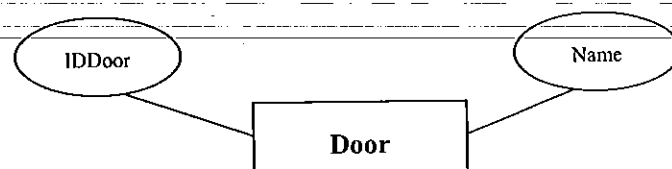
### 3.4.2 ความสัมพันธ์ของตาราง

Entity-Relationship Diagram (ER-Diagram) เป็นการแสดงความสัมพันธ์ของตาราง ทั้งนี้เพื่อให้ผู้ใช้งานสามารถมองภาพของข้อมูลในระบบได้ชัดเจนยิ่งขึ้น



รูปที่ 3.10 ER-Diagram ของตาราง Admins, Users, Event

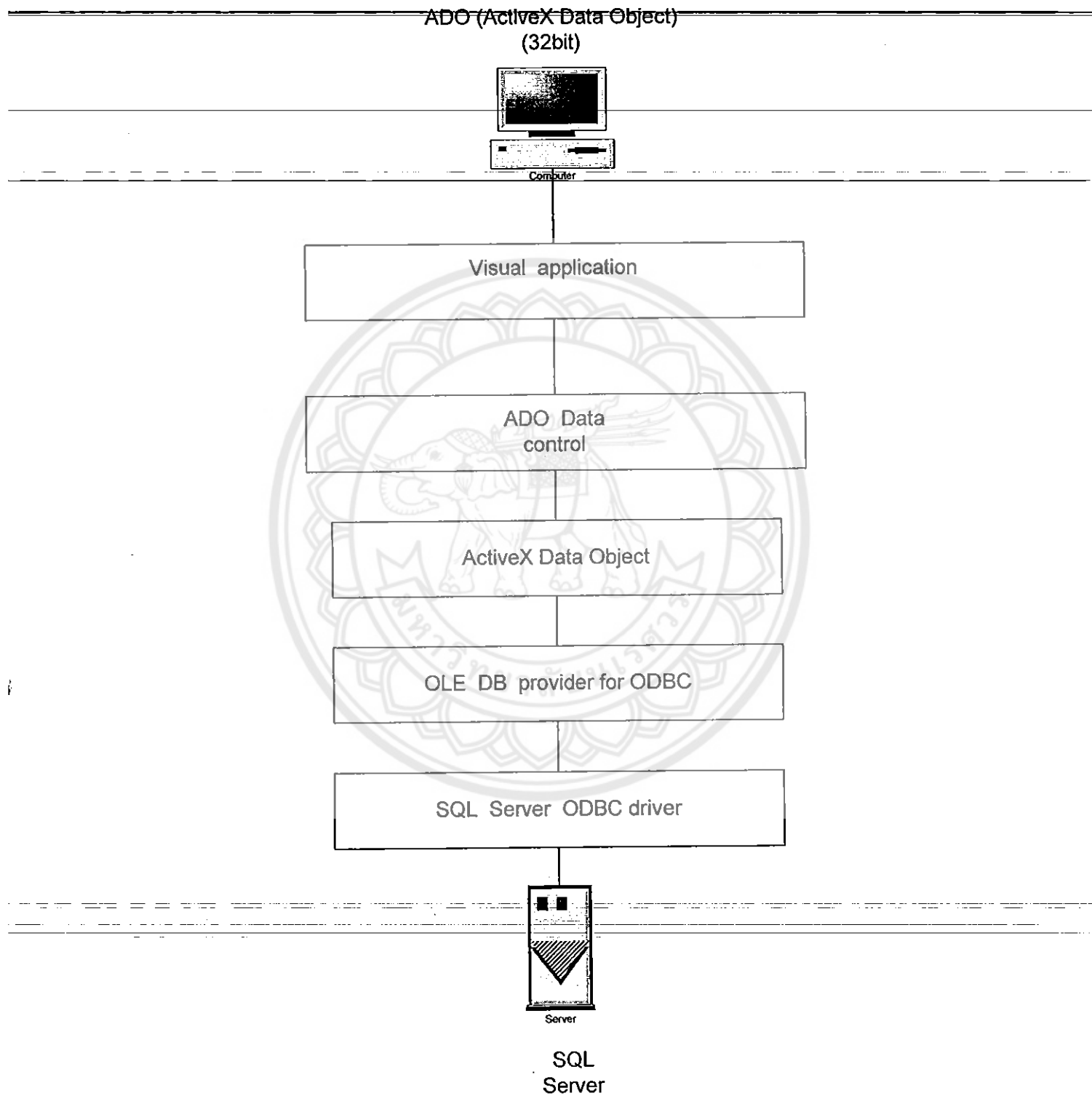
จากรูป 3.10 ตาราง Admins, Users และ Event มีความสัมพันธ์กันโดยตาราง Admins และ ตาราง Users มี field Password เชื่อมความสัมพันธ์กัน ส่วนตาราง Event และ ตาราง Users ก็มี field name เชื่อมความสัมพันธ์กัน ดังนั้นตารางทั้งสามจึงมีความสัมพันธ์กันดังรูปที่ 3.10



รูปที่ 3.11 ER-Diagram ของตาราง Door

ตาราง Door ไม่มีความสัมพันธ์กับตารางใดๆเลย

### 3.5 การติดต่อ Database SQL Severของ Visual Basic



รูปที่ 3.12 การติดต่อ Database SQL Serverของ Visual Basic

จากรูปที่ 3.12 คือแผนภาพการติดต่อ Database SQL Server ของ Visual Basic โดยโครงการนี้จะใช้วิธีการติดต่อกับฐานข้อมูลโดยใช้ ADO (ActiveX Data Object) โดยเป็นเทคโนโลยีการติดต่อกับระบบฐานข้อมูลที่ทาง Microsoft ได้พัฒนาขึ้นมาใหม่เพื่อให้ผู้ใช้สามารถเขียนโปรแกรมเพื่อติดต่อกับระบบฐานข้อมูลหรือระบบข้อมูลต่างๆไม่ว่าข้อมูล E-mail ข้อมูลภาพและอื่นๆได้โดยสะดวก รวดเร็วและดีขึ้นกว่าเทคโนโลยีเดิมของ Microsoft ที่ใช้ติดต่อกับระบบฐานข้อมูลเท่านั้น จากแผนภาพที่แสดงจะมีอยู่ 5 ส่วนใหญ่ๆดังนี้

-Visual Application คือ โปรแกรม Microsoft Visual Basic 6.0 ที่เราจะใช้เขียนโปรแกรมและติดต่อกับฐานข้อมูล

-ADO Data Control คือ เป็นเครื่องมือหรือ Tool ที่อยู่ใน Microsoft Visual Basic 6.0 จะประกอบไปด้วยเครื่องมือต่างๆที่ทำให้ง่ายต่อการใช้งาน

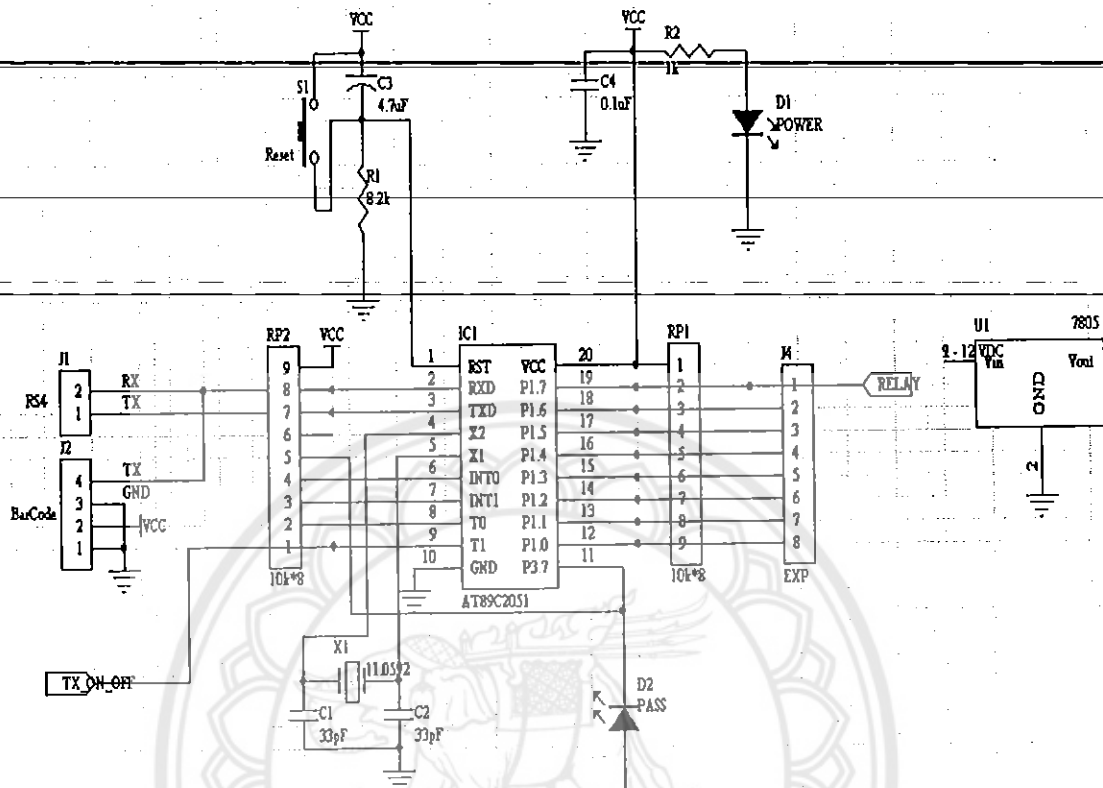
-ActiveX Data Object คือ เทคโนโลยีที่เราเลือกใช้ในการติดต่อกับฐานข้อมูล

-OLE DB provider for ODBC คือ เทคโนโลยีอีกตัวหนึ่งที่ใช้ติดต่อกับฐานข้อมูลทั้ง 4 ส่วนข้างต้นจะเป็นการตั้งค่าใน Microsoft Visual Basic 6.0

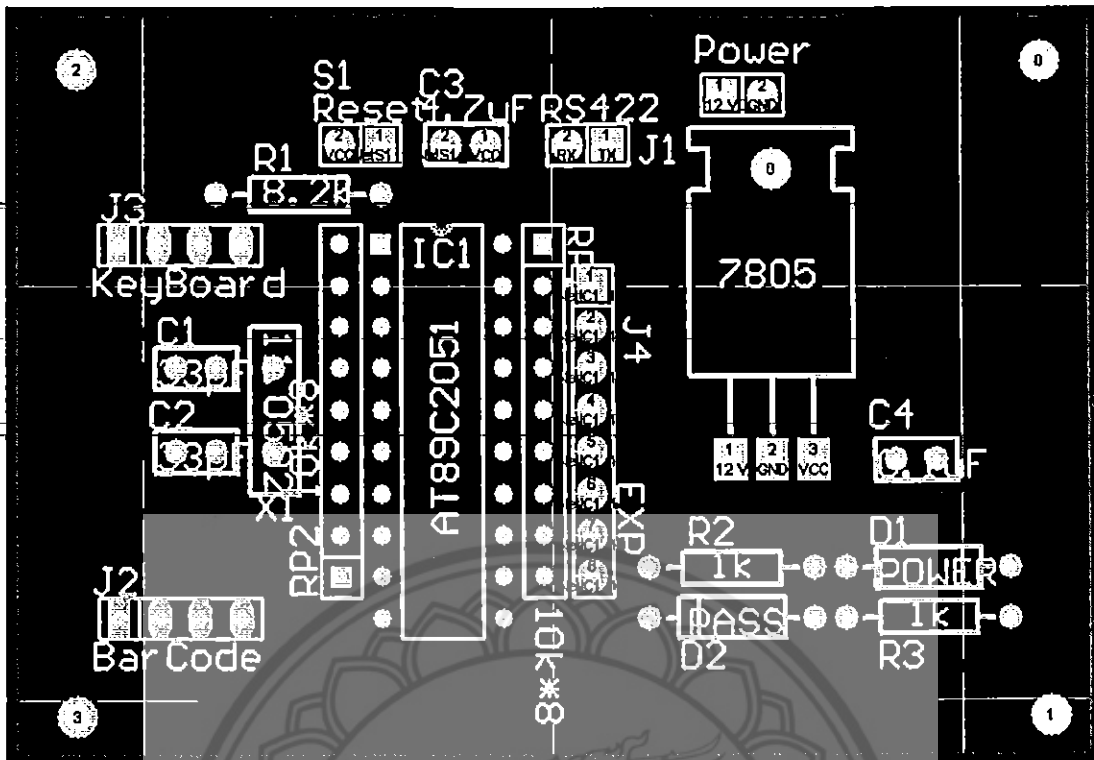
ส่วนอีกหนึ่งส่วนที่เหลือจะเป็นตั้งค่าในระบบปฏิบัติการ Microsoft Windows ในส่วนของ Control Panel ซึ่งได้แก่

-SQL Server ODBC driver คือ เป็นเสมือนตัวกลางที่ทำให้ติดต่อกับฐานข้อมูลได้ระหว่างโปรแกรม Microsoft Visual Basic 6.0 กับโปรแกรม Microsoft SQL Server 7.0

### 3.6 การติดต่อกับพอร์ทต่าง ๆ ของ MCS-51 AT89C2051



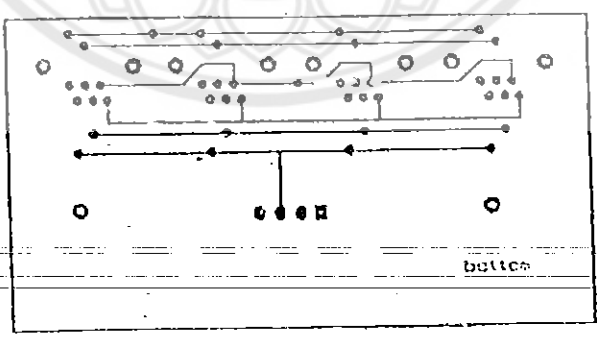
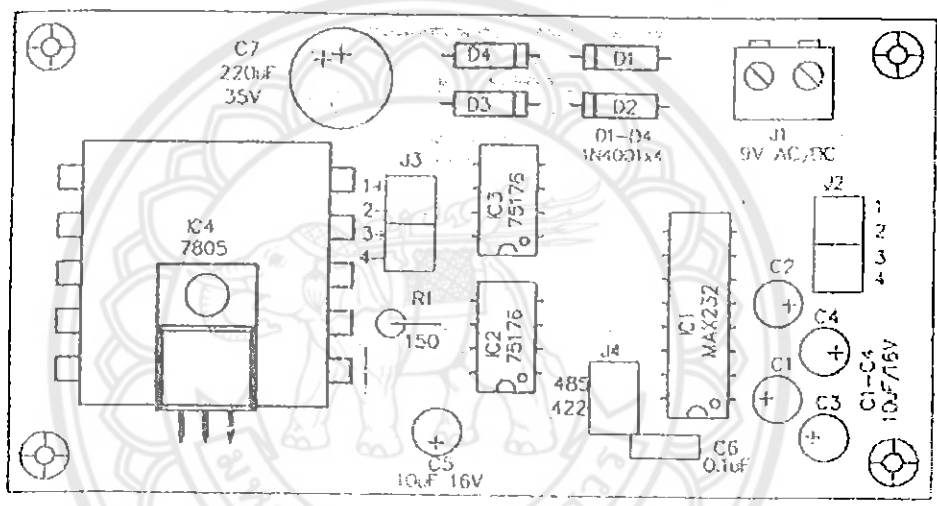
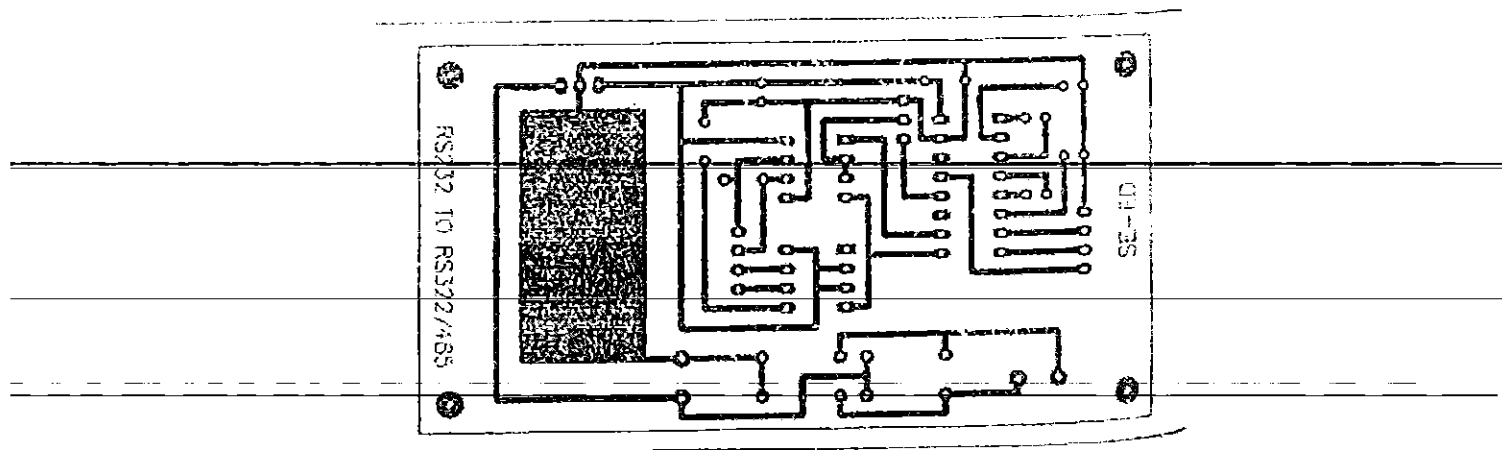
รูปที่ 3.13 แสดงวงจรการติดต่อกับพอร์ทต่างๆ ของ MCS-51



รูปที่ 3.14 วงจรพิมพ์ของบอร์ดควบคุม

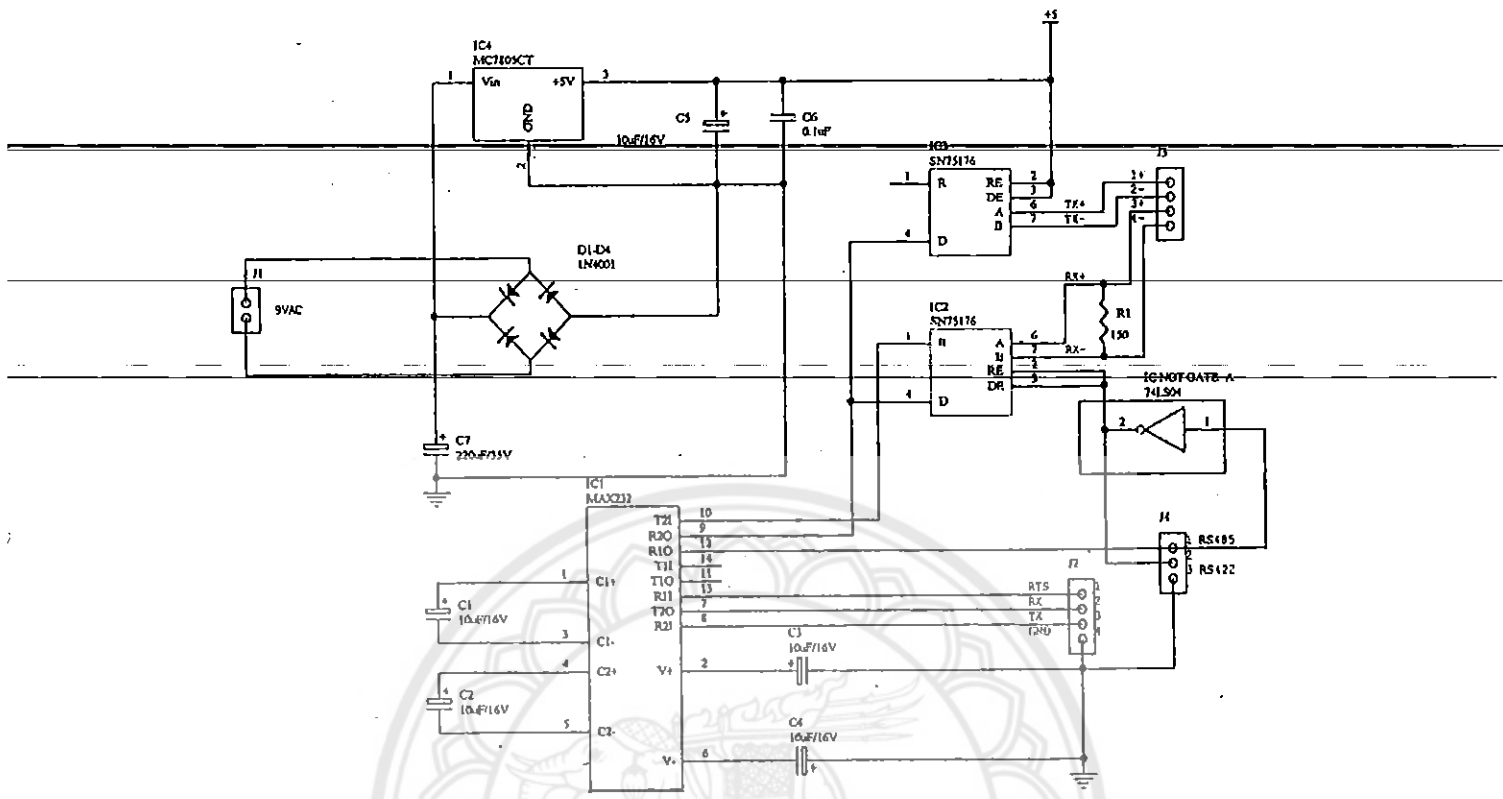
จากรูปที่ 3.13 และ 3.14 ในวงจรของบอร์ดควบคุมประกอบด้วย IC MCS 51 ของ ATMELE AT89C89C2051 ซึ่งเป็น CPU ที่มี 20 ขา ที่ติดต่อกับอุปกรณ์ต่อพ่วงดังนี้ ชุดแปลงสัญญาณ RS 422/485 โดยขา tx จะต่อกับ R-Pack ที่ขา 8 แล้วเข้าสู่ MCS-51 ที่ขา 3 (TXD) และสัญญาณ (RXD), และสัญญาณ Rx ต่อผ่าน R-Pack ขาที่ 7 เข้าสู่ AT89C2051 ขา 2 (RXD), อุปกรณ์ barcode reader จะติดต่อกับ MCS-51 โดยขา tx ของ barcode reader จะต่อเข้ากับขา RXD ของ MCS-51 โดยขา RXD ของ MCS-51 ก็ขาที่ 2 ทั้งสัญญาณ RS 422 จาก computer และ barcode จะเป็นส่วนของการรับ-ส่ง ข้อมูล ส่วนขาที่ใช้ควบคุม mosfet IRFZ 44n เพื่อ trick on-off relay

สำหรับแหล่งจ่ายไฟของวงจรนี้จะรับภาคจ่ายไฟมาจาก IC 7805 ซึ่งจะรักษาระดับแรงดันที่ +5.v โดยจ่ายเข้าที่ขา 20 และขา 10 ซึ่งเป็นขากราวด์



รูปที่3.15 บอร์ดชุดแปลงสัญญาณ





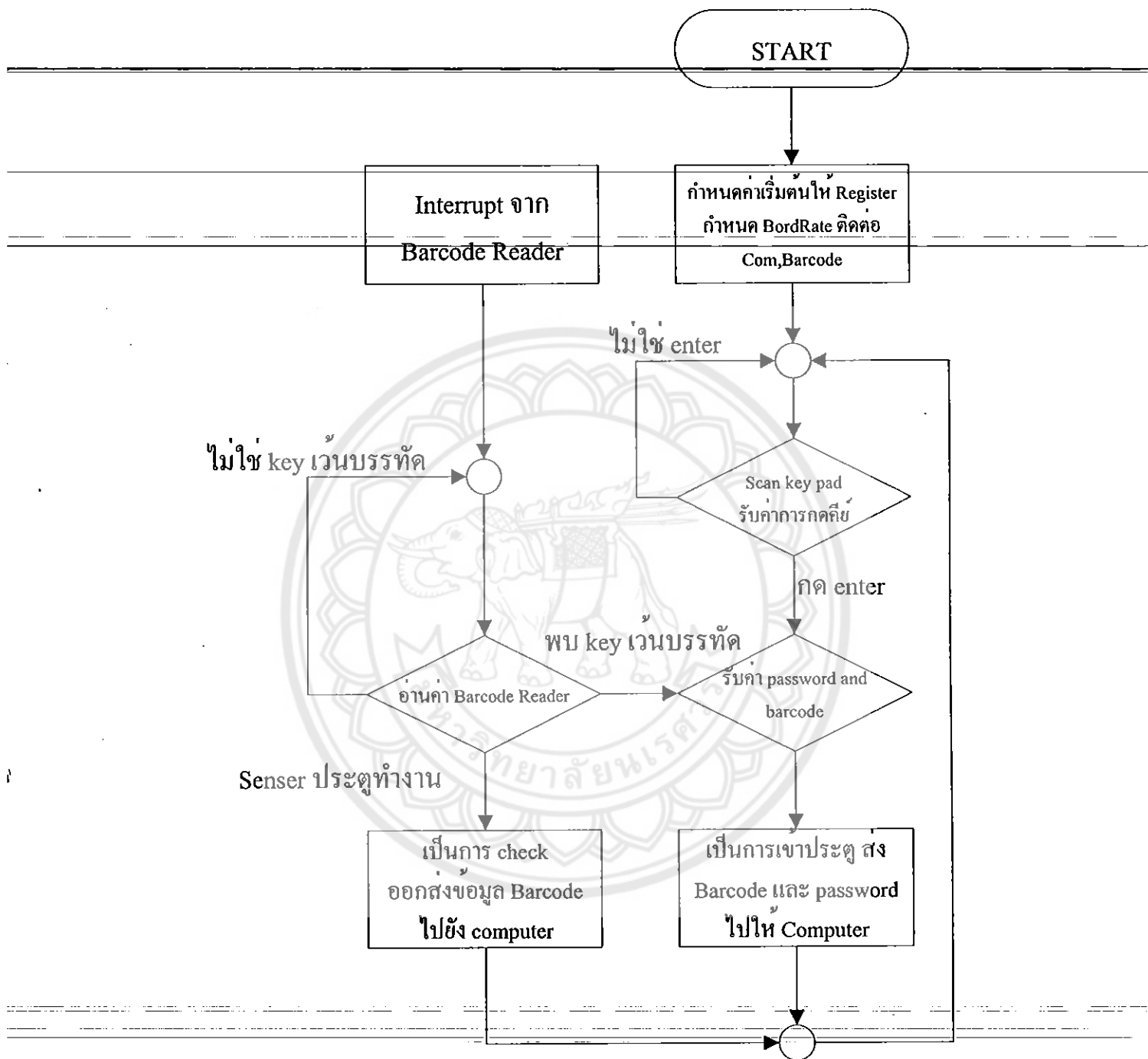
รูปที่ 3.16 วงจรของชุดแปลงสัญญาณ RS 232-422/485

จากรูปที่ 3.15 และ 3.16 ในชุดแปลงสัญญาณจาก RS 232 ไปเป็น RS 422/485 โดยปกตินี้มาตรฐานของ RS 232 จะเป็นสัญญาณแบบ TTL เมื่อผ่าน line drive ก็จะเปลี่ยนจากสัญญาณ TTL มาเป็นระดับแรงดันที่ต่างกัน  $\pm 12$  v เทียบกับกราวด์ มาตรฐาน RS232 นี้จะมีข้อจำกัดทางด้านสัญญาณรบกวน ซึ่งส่งได้จำกัดในระยะ 50 ฟุต (15 เมตร)

ในโครงการนี้ต้องการพัฒนาให้สามารถควบคุมได้ระยะไกล จึงต้องหามาตรฐานการส่งใหม่โดยใช้มาตรฐาน RS 422/485 ซึ่งทั้งคู่จะมีการสื่อสารในลักษณะเดียวกัน คือใช้แบบ balance line หมายถึงไม่ได้เทียบกราวด์ แต่เป็นการใช้ระดับสัญญาณที่ต่างกันคือ  $\pm 2$  v ถึง  $\pm 6$  v ฉะนั้น จึงทำการแปลงสัญญาณ RS 232 มาเป็น RS 422/485 เริ่มต้นจากที่รับสัญญาณ TTL มาจากคอมพิวเตอร์แล้วมาเข้า line driver MAX-232 ซึ่งเปลี่ยนสัญญาณ TTL มาเป็นลอจิกที่มีความต่างศักย์คือ  $\pm 12$  v แล้วส่งไปให้ IC 78176 ซึ่งมีหน้าที่ทำให้สัญญาณเปลี่ยนมาเป็นแบบ balance line คือมี Tx+, Rx+, Tx-, Rx- สามารถส่งให้ไกลขึ้นได้ 4,000 ฟุต

สำหรับภาคจ่ายไฟนั้นจะรับ AC 9 v มาเข้าที่ rectifier ผ่าน C 7 เพื่อกรองกระแสไฟให้เรียบแล้วส่งไปที่ IC 7805 เพื่อรักษาระดับแรงดัน 5 v เพื่อจ่ายไฟให้กับวงจร

## 3.7 การทำงานของชุดควบคุม



รูปที่ 3.17 การทำงานของชุดควบคุม

จากรูปที่ 3.16 เริ่มต้นที่การกำหนดค่าเริ่มต้น Register ต่าง ๆ เช่น การกำหนด Baud Rate, Port Com, ID, Key Pad และที่เกี่ยวข้อง

ในสถานการณ์ปกติ Board จะทำการ scan Key Pad อยู่ตลอดเวลาเพื่อรอการอินเตอร์รัพท์ จาก barcode reader เมื่อมีการอ่านค่าจาก barcode reader board ก็จะรอรับข้อมูลจาก Key Pad หรือ sensor ถ้าหากมีการอ่านค่าบาร์โค้ดเข้ามาแล้วมีการกด Key Pad บอร์ดก็จะรับรู้ว่าเป็นการเช็คเข้า และอีกกรณีที่มีการอ่านบาร์โค้ดแล้ว sensor ทำงานบอร์ดก็จะรับรู้ว่าเป็นการออก

กรณีการเช็คเข้า คือ มีการกด Key Pad หลังจากมีการอ่านข้อมูลจากบาร์โค้ด ขั้นตอนต่อไปคือ บอร์ดควบคุมจะทำการส่งข้อมูลไปให้คอมพิวเตอร์ ข้อมูลก็คือ start bit + ID + Barcode + Password + stop bit

กรณีการเช็คออก คือ เมื่อมีการอ่านบาร์โค้ดเข้ามาแล้ว sensor ทำงาน บอร์ดจะรู้ว่าเป็นการเช็คออก และในระหว่างนั้นบอร์ดจะส่งข้อมูลไปให้คอมพิวเตอร์ ข้อมูลก็คือ start bit + ID + Barcode ++ stop bit



## บทที่ 4

### ผลการทดลอง

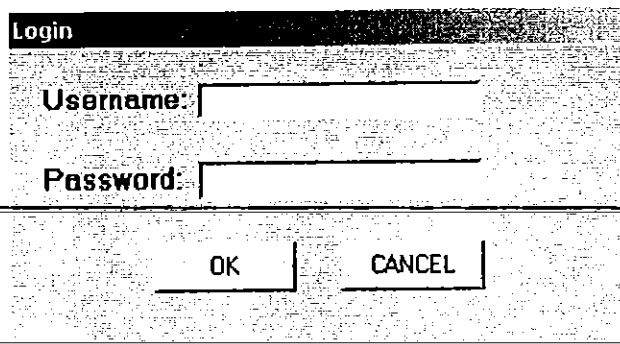
#### 4.1 ผลการทดลอง

##### 4.1.1 ส่วนของ โปรแกรม

ID	Event Name	TimeIn	TimeOut	Date	Levels
1	nathapong	13:33:48	13:34:24	20/5/2542	Door2
2	chekkopong	13:34:51	13:36:14	20/5/2542	Door2
3	chlnaboon	13:35:27	13:36:00	20/5/2542	Door2
4	nathapong	13:40:39	13:41:54	20/5/2542	Door1
5	chlnaboon	13:40:41	13:41:48	20/5/2542	Door2
6	nathapong	14:07:46	14:07:58	20/5/2548	Door2
7	sittichai	17:19:57	17:20:04	20/5/2548	Door2

รูปที่ 4.1 Monitor

เป็นส่วนของ Monitor ซึ่งจะเป็นส่วนที่แสดงข้อมูลการเข้า-ออกอาคาร และที่ Setting  
เป็นส่วนของผู้ดูแลระบบเอาไว้จัดการต่างๆเกี่ยวกับ Database



Login

Username:

Password:

OK CANCEL

รูปที่4.2 Login

เป็นส่วนที่เอาไว้ให้ผู้ดูแลระบบเข้าไปตั้งค่าหรือดูข้อมูลต่างๆของระบบ โดยใส่ข้อมูล

Username และ Password



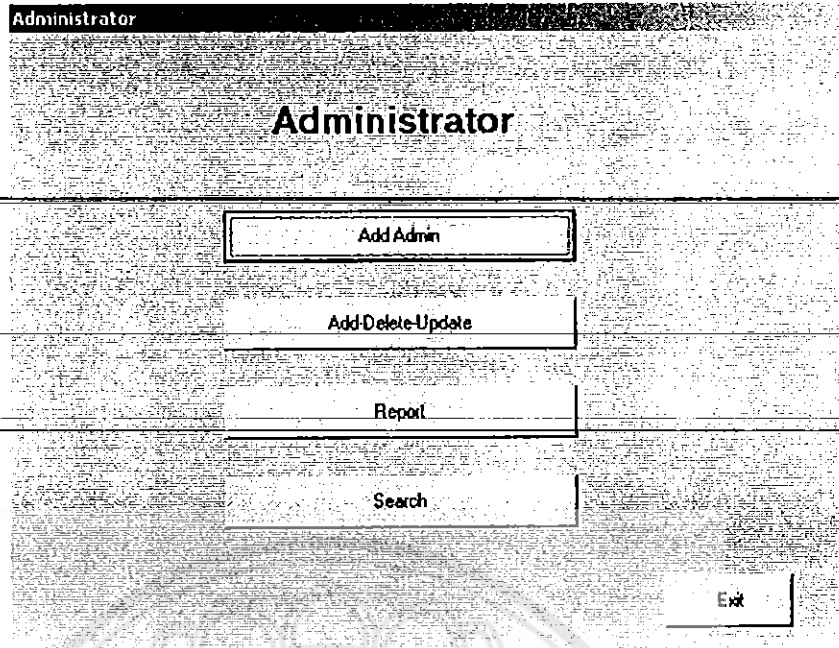
Exit

Password:

OK CANCEL

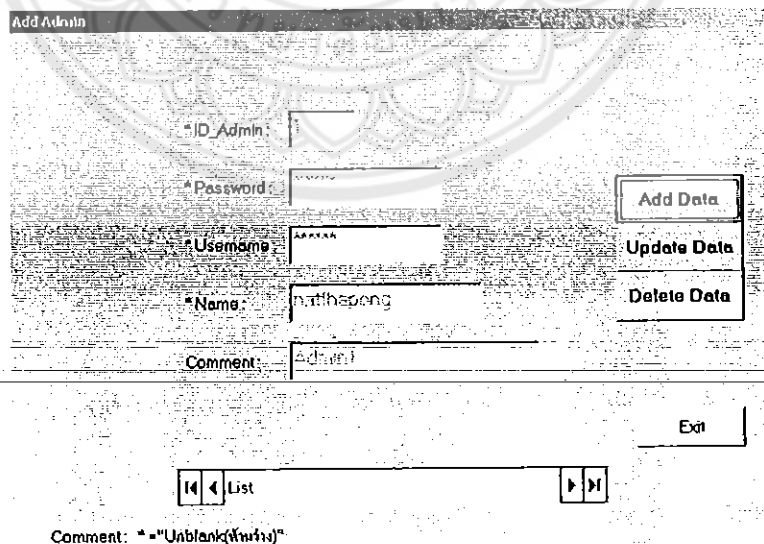
รูปที่4.3 การออกจากโปรแกรม

เป็นส่วนที่เอาไว้ให้ผู้ดูแลระบบใช้สำหรับปิดโปรแกรม โดยใส่ข้อมูล Password



รูปที่ 4.4 Administrator

เมื่อผู้ดูแลระบบทำการ Login เข้ามาแล้วจะพบกับหน้า Administrator เพื่อให้ผู้ดูแลระบบเลือกว่าจะเพิ่ม ลด หรือแก้ไขผู้ดูแลระบบ (Add Admin), เพิ่ม ลด หรือแก้ไขข้อมูลผู้ใช้ (Add-Delete-Update), รายงานการเข้า-ออกอาคาร (Report) และค้นหาข้อมูล (Search)



รูปที่ 4.5 Add Admin

เป็นส่วนที่มีไว้เพิ่มผู้ดูแลระบบ, แก้ไขข้อมูล และลบผู้ดูแลระบบ

Add-Delete-Update Users

ID User	User	Paramed	Name	Level User	Comment
1	44370153	0153	nathapong	1	Admin1
2	44370120	0120	chinsaboon	1	Admin2
3	44370089	0089	chekkapong	2	User1
6	44370492	0492	satchai	2	User2
*					

Exit

Comment :ID, User, Barcode, Password  
Name and Levels = !\Unblank!\(rubb)

รูปที่4.6 Add User

เป็นส่วนที่มีไว้เพิ่มผู้ใช้, แก้ไขข้อมูล และลบผู้ใช้ โดยผู้ดูแลระบบจะมี Level\_User=1,  
ผู้ใช้ Level\_User=2

Report

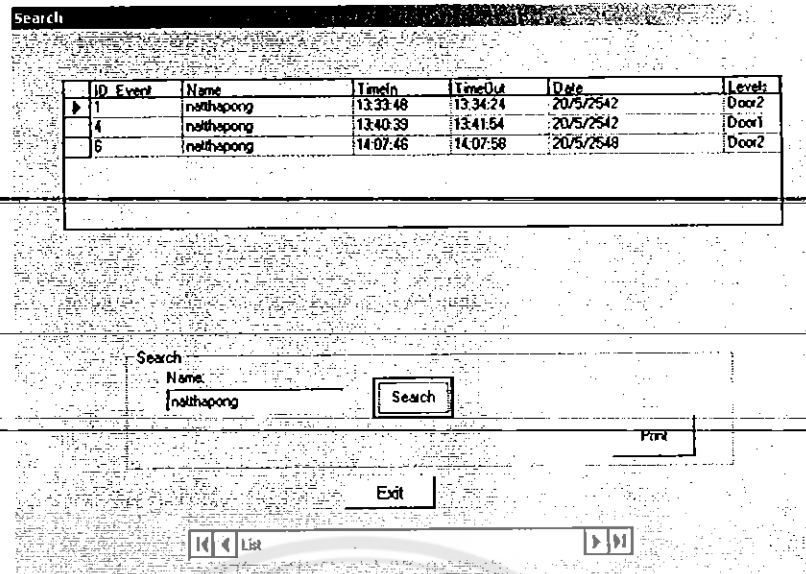
ID Event	Name	TimeIn	TimeOut	Date	Level
1	nathapong	13:33:48	13:34:24	20/5/2542	Door2
2	chekkapong	13:34:51	13:36:14	20/5/2542	Door2
3	chinsaboon	13:35:27	13:36:00	20/5/2542	Door2
4	nathapong	13:40:39	13:41:54	20/5/2542	Door1
5	chinsaboon	13:40:41	13:41:48	20/5/2542	Door2
6	nathapong	14:07:46	14:07:59	20/5/2548	Door2
7	satchai	17:19:57	17:20:04	20/5/2548	Door2

Print Clear Report Exit

1/4 List

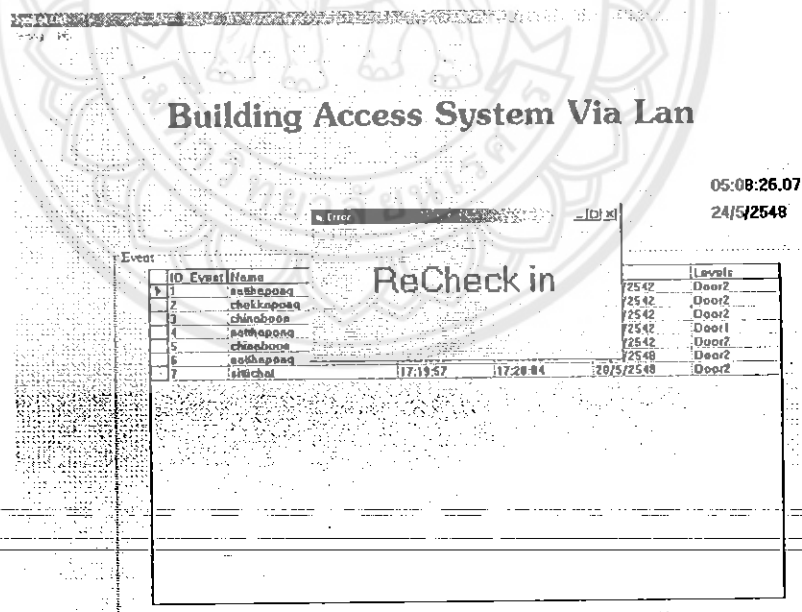
รูปที่4.7 Report

เป็นการแสดงส่วนของรายงานการเข้า-ออกอาคารทั้งหมด สามารถที่จะสั่งพิมพ์รายงาน  
ออกทางเครื่อง Printer ได้



รูปที่ 4.8 Search

เป็นการแสดงการค้นหาข้อมูล และสั่งพิมพ์รายงานออกทางเครื่อง Printer ได้



รูปที่ 4.9 การ Restart ตัวเองของโปรแกรม

เมื่อเกิดการชนกันของข้อมูล(ส่งข้อมูลเข้ามาพร้อมกันหลายๆเครื่อง) หรือเกิดการ Error ขึ้นในโปรแกรม โปรแกรมจะขึ้นหน้าจอ ReCheck in และโปรแกรมจะ Restart ตัวเอง และใช้งานได้ตามปกติ



#### 4.1.2 ส่วนของ Database

ID_Event	Name	TimeIn	TimeOut	Date	Level
1	natthapong	13:33:48	13:34:24	20/5/2542	Door2
2	chekkapong	13:34:51	13:36:14	20/5/2542	Door2
3	chinaboon	13:35:27	13:36:00	20/5/2542	Door2
4	natthapong	13:40:39	13:41:54	20/5/2542	Door1
5	chinaboon	13:40:41	13:41:48	20/5/2542	Door2
6	natthapong	14:07:46	14:07:58	20/5/2548	Door2
7	sritichat	17:19:57	17:20:04	20/5/2548	Door2

รูปที่ 4.10 ตาราง Event

ตาราง Event เป็นส่วนที่เก็บ ชื่อ, เวลาเข้า, เวลาออก, วันที่ และประตูที่เข้าใช้อาคาร เมื่อทำการรูดบัตรและกดรหัสผ่าน ถูกต้องแล้ว ชื่อ, เวลาเข้า, วันที่ และประตู จะถูกบันทึกลงในตาราง Event และเมื่อออกจากอาคาร เวลาออก ก็จะถูกบันทึกลงในตาราง Event เช่นกัน

ID Admin	Password	Username	Name	Comment
1	0153	admin1	natthapong	Admin1
2	0120	admin2	chhaboon	Admin2

รูปที่ 4.11 ตาราง Admins

ตาราง Admins เป็นส่วนที่เก็บ Username, Password และ ชื่อ ของผู้ดูแลระบบ ใช้สำหรับ Login ใน ส่วนของ โปรแกรม เพื่อจัดการต่างๆเกี่ยวกับ Database

ID User	User	Password	Name	Level User	Comment
1	44370153	0153	natthapong	1	Admin1
2	44370120	0120	chhaboon	1	Admin2
3	44370088	0088	chekkapong	2	User 1
6	44370492	0492	sitichat	2	User2

รูปที่ 4.12 ตาราง Users

เป็นส่วนที่เก็บ User (รหัส Barcode), Password, ชื่อ และ ระดับผู้ใช้ ของผู้ที่สามารถเข้า-ออกอาคารได้ โดย “1” คือ ระดับ ผู้ดูแลระบบ “2” คือ ระดับผู้ใช้ทั่วไป

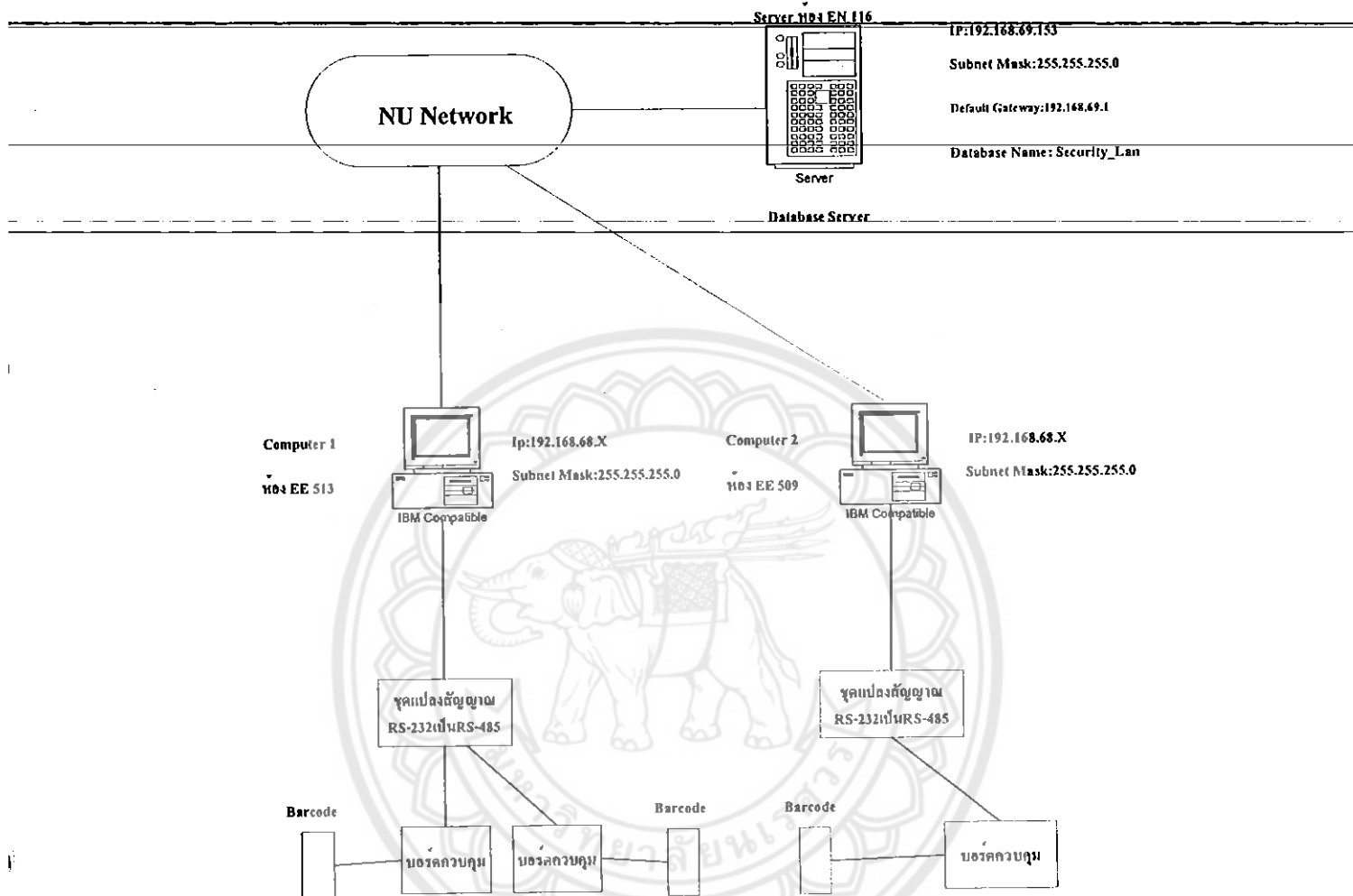
IDDoor	Name
1	Door1
2	Door2

รูปที่4.13 ตาราง Doors

เป็นส่วนที่เก็บ ชื่อประตู



### 4.1.3 การทำการทดลอง ระบบเข้า-ออกอาคารผ่านเครือข่าย



รูปที่ 4.14 การทดลองผ่าน network

จากรูป สามารถเชื่อมต่อได้เฉพาะ Network ของมหาวิทยาลัยเท่านั้น ถ้าอยู่ข้างนอกแต่ถ้าต่อ Internet ของมหาวิทยาลัยก็สามารถเชื่อมต่อได้ เพราะอยู่ในเครือข่ายเดียวกัน แต่ถ้าต่อ Internet ที่ไม่ใช่เครือข่ายของมหาวิทยาลัย จะไม่สามารถติดต่อกับ Database ได้ เพราะ Network ข้างนอกจะไม่สามารถมองเห็น Network ภายในมหาวิทยาลัยได้ ถ้าจะติดต่อกันได้ต้องไปติดต่อที่ตึก IT ให้ Set เครื่อง Server ของโครงการนี้ให้เป็น Sub Domain (เครื่อง Server ย่อย) ก็จะสามารถใช้ระบบตรวจสอบการเข้า-ออกอาคารผ่านเครือข่ายภายนอกมหาวิทยาลัยได้

## บทที่ 5

# สรุปผลการทดลองโครงการ ปัญหา และข้อเสนอแนะ

### 5.1 สรุปผลการทดลองโครงการ

จากการศึกษา ออกแบบ ทดลอง รวมทั้งการแก้ไขปัญหาต่างๆ โครงการนี้สามารถแก้ไขปัญหาการชนกันของข้อมูลได้ โดยเมื่อเกิดการชนกันของข้อมูลโปรแกรมจะทำการ Restart ตัวเองขึ้นมาใหม่ ทำให้ทำการใช้งานได้อย่างต่อเนื่อง และยังสามารถส่งข้อมูลได้ในระยะไกล ภายใต้ระบบ Network ของมหาวิทยาลัยนเรศวร

### 5.2 ปัญหาและข้อเสนอแนะ

โครงการนี้ยังมีข้อบกพร่องอยู่บ้าง ผู้จัดทำมีความประสงค์จะเสนอแนะแนวทางในการแก้ไขดังต่อไปนี้

- บอร์ดควบคุมทั้งสองบอร์ดเมื่อเปิดทำการทดลองได้สักระยะ บอร์ดควบคุมจะไม่ทำการส่งข้อมูลไปให้คอมพิวเตอร์ ทำให้ไม่สามารถเปรียบเทียบข้อมูลใน Database ได้ สันนิษฐานว่าบอร์ดควบคุมอาจจะร้อน ควรทำให้บอร์ดควบคุมไม่ร้อน โดยการใช้พัดลมเป่าบอร์ดควบคุม หรือทดลองในห้องปรับอากาศที่มีความเย็น

## เอกสารอ้างอิง

- [1] ~~Halvorson, Michael, Microsoft Visual Basic Professional 6.0 Step by Step, United StateOf American : a division of Thomson Canada Limited., 1998~~
- [2] กรรณิกา รุจิรวณิชเทพ, คำพล จันตะภาค, ระบบควบคุมการเข้าออกอาคารและห้องทำงาน. พิชญโลก : มหาวิทยาลัยนเรศวร, 2546
- [3] ~~จักรวาล แก้วสุวรรณ, ปรีชา ภาวะสิน, ระบบตรวจสอบการเข้าใช้ห้องคอมพิวเตอร์. พิชญโลก : มหาวิทยาลัยนเรศวร, 2545~~
- [4] ฉัททวุฒิ พิษผล, พิชิต สันติกุลานนท์, คู่มือเรียน Visual Basic 6. กรุงเทพมหานคร : ไบรวิชั่น, 2542
- [5] ดิเรก วงษ์วานิช. ติดตั้ง/ใช้งานโฮมเน็ตเวิร์กด้วยตนเอง. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2544
- [6] ธนพล ฉันทจรสวัสดิชัย, เรียนรู้เทคนิคการทำรายงานบนเครื่องพิมพ์ด้วย Visual Basic. กรุงเทพมหานคร : เอส.พี.ซี. บุ๊คส์, 2542
- [7] บัณฑิต จามรภูติ, การใช้งาน Windows Server2000 ภาคปฏิบัติ เล่ม 2. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2547
- [8] วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ -MCS-51. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์, 2544
- [9] ยุทธภูมิ เขียวแสง และคณะ. แบบจำลองเครื่องตรวจสอบบัตรจอแสดงผลและ เปิด-ปิด ด่าน. พิชญโลก : มหาวิทยาลัยนเรศวร, 2545
- [10] สุกชัย สมพานิช, เรียนรู้และฝึกฝนการเขียนโปรแกรมใช้งานฐานข้อมูลด้วย Visual Basic. กรุงเทพมหานคร : อินโฟเพรส, 2547
- [11] สมพร จิวรสกุล, คู่มือการติดตั้งและใช้งาน SQL SERVER7.0. นนทบุรี : อินโฟเพรส, 2543
- [12] สุขาย ธนเวสทีयर, นรินทร์ อัครพิเชษฐ, Fundamental of Visual Basic Client-Server Programming. กรุงเทพมหานคร : SUM Publishing, 1998
- [11] อภิชาติ ภูพลับ, เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic. นนทบุรี : Infopress Developer Book, 2546
- [13] อรรถพล บุญยะ โภคา, วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม. กรุงเทพมหานคร : อินโนเวตีฟ เอ็กเพอริเมนต์, 2544
- [14] "การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ Visual Basic 6.0" [Online]. Available : <http://www.thaiio.com/>



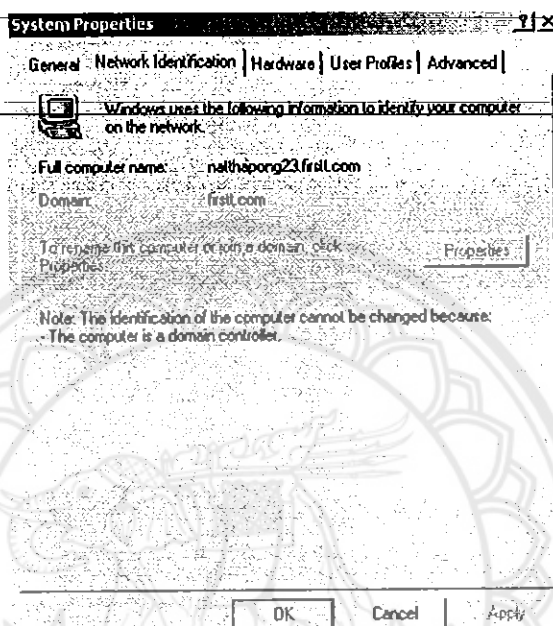




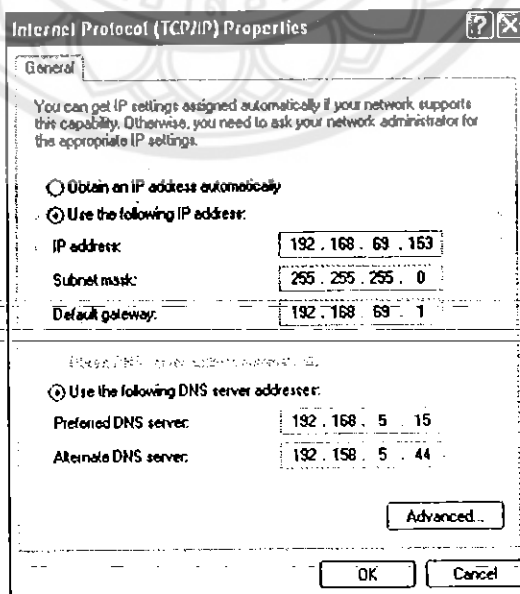
# การติดตั้งและการใช้งานโปรแกรม

## 1. การ Set ชื่อเครื่องของคอมพิวเตอร์

### เครื่อง Database Server (Windows 2000 Server)



รูปที่ 1 ชื่อเครื่อง และ Domain ของเครื่อง Server



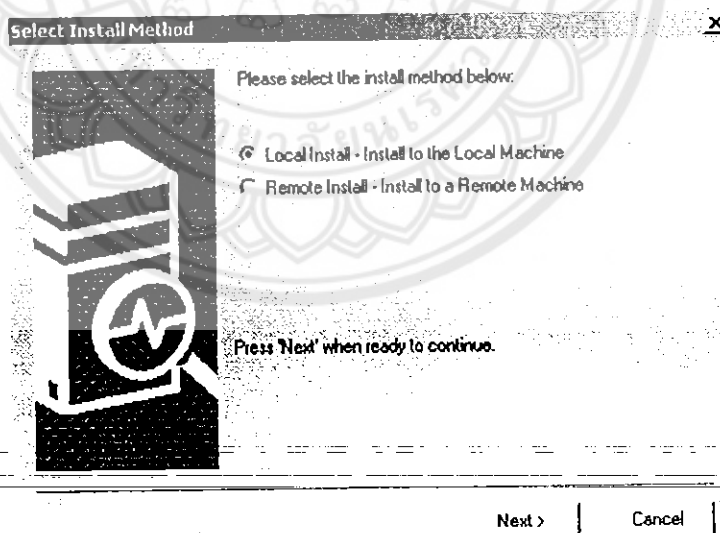
รูปที่ 2 IP address ของเครื่อง Server

## 2. การติดตั้ง Microsoft SQL Server 7.0



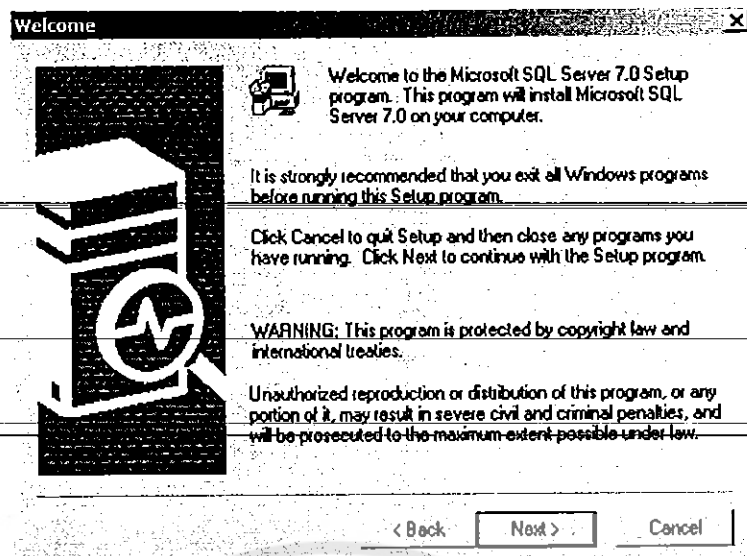
รูปที่3 รายการที่ต้องการติดตั้งระบบ

เลือกติดตั้ง Database Server ที่ Standard Edition



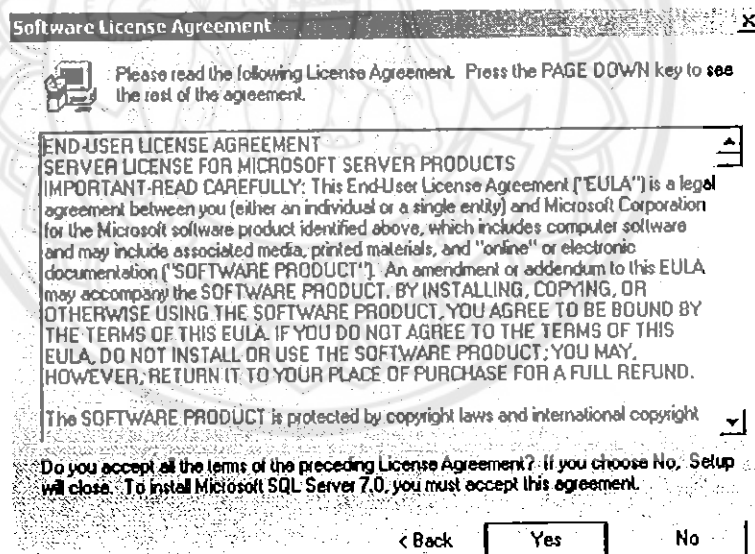
รูปที่4 ตัวเลือกรูปการติดตั้ง

เลือกวิธีการติดตั้งแบบ Local Install



รูปที่ 5 การต้อนรับเข้าสู่การติดตั้ง

คลิกปุ่ม Next



รูปที่ 6 รายละเอียดของการใช้ Software โปรแกรม

อ่านข้อตกลงของการใช้ Software โปรแกรมให้เข้าใจ จากนั้นคลิกปุ่ม Yes เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป

User Information x

Enter your name below. It is not necessary to enter a company name.

Name:

Company:

### รูปที่7 การใส่ข้อมูลของผู้ติดตั้ง

ให้ใส่ข้อมูลชื่อผู้ติดตั้งและชื่อบริษัท แล้วคลิกปุ่ม Next

### Setup

Locate your 10 digit 'CD Key' and enter it in the space below. You will find this number on the yellow sticker of your CD liner notes or CD sleeve.

CD Key:

-

OK

### รูปที่8 ช่องใส่ CD-Key

ใส่ CD-Key ที่ช่องใส่จำนวนสิบหลัก แล้วคลิก OK

**Setup Type**

Click the type of Setup you prefer, then click Next.

Typical Installed with the most common options. Recommended for most users.

Minimum Installed with minimum required options.

Custom You may choose the options you want to install. Recommended for advanced users.

---

Destination Folder

Program Files  C:\MSSQL7

Data Files  C:\MSSQL7

---

	Required	Available
Space on program files drive:	82842 K	2147483 K
Space on system drive:	33392 K	2147483 K
Space on data files drive:	30597 K	2147483 K

Help < Back Next > Cancel

### รูปที่ 9 ประเภทของการติดตั้ง

ควรเลือกแบบ Custom เป็นการติดตั้งเพราะจะสามารถกำหนดตัวเลือกได้ในทุกส่วนประกอบ จากนั้นคลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป

**Select Components**

Select the components you want to install, clear the components you do not want to install.

Components:

- Server Components
- Management Tools
- Client Connectivity
- Books Online
- Development Tools

Sub-Components:

- SQL Server
- Upgrade Tools
- Replication Support
- Full-Text Search

Description

Installs the core functional components of the SQL Server relational database.

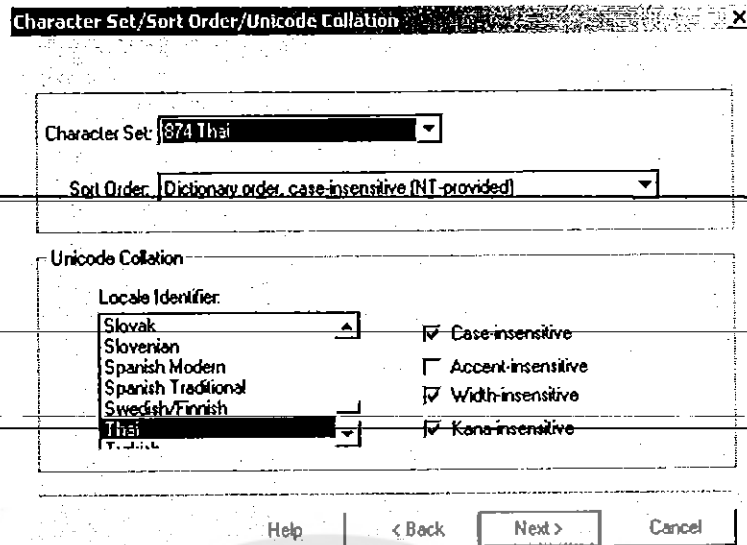
---

	Required	Available
Space on program files drive:	87200 K	2147483 K
Space on system drive:	57599 K	2147483 K
Space on data files drive:	30597 K	2147483 K

Help < Back Next > Cancel

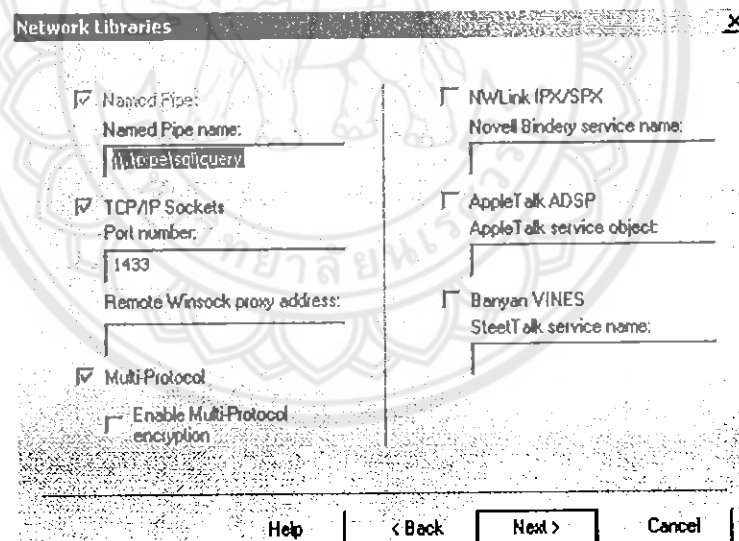
### รูปที่ 10 ส่วนประกอบของ Server Components

เลือก Components ที่ต้องการติดตั้ง จากนั้นคลิกปุ่ม Next



รูปที่ 11 การกำหนดคุณลักษณะของ Window (Thai Enable)

คลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป




รูปที่ 12 ประเภทของ Network Libraries

ให้คลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป

Services Accounts															
<input checked="" type="radio"/> Use the same account for each service. Auto start SQL Server Service. <input type="radio"/> Customize the settings for each service.															
<table border="1"> <thead> <tr> <th>Services</th> <th>Service Settings</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/> SQL Server</td> <td><input type="radio"/> Use the Local System account</td> </tr> <tr> <td><input checked="" type="radio"/> SQL Server Agent</td> <td><input checked="" type="radio"/> Use a Domain User account</td> </tr> <tr> <td></td> <td>Username: Administrator</td> </tr> <tr> <td></td> <td>Password: [REDACTED]</td> </tr> <tr> <td></td> <td>Domain: FIRST</td> </tr> <tr> <td></td> <td><input type="checkbox"/> Auto Start Service</td> </tr> </tbody> </table>	Services	Service Settings	<input type="radio"/> SQL Server	<input type="radio"/> Use the Local System account	<input checked="" type="radio"/> SQL Server Agent	<input checked="" type="radio"/> Use a Domain User account		Username: Administrator		Password: [REDACTED]		Domain: FIRST		<input type="checkbox"/> Auto Start Service	
Services	Service Settings														
<input type="radio"/> SQL Server	<input type="radio"/> Use the Local System account														
<input checked="" type="radio"/> SQL Server Agent	<input checked="" type="radio"/> Use a Domain User account														
	Username: Administrator														
	Password: [REDACTED]														
	Domain: FIRST														
	<input type="checkbox"/> Auto Start Service														
<div style="text-align: right;"> <input type="button" value="Help"/> <input type="button" value=" &lt; Back"/> <input type="button" value=" Next &gt;"/> <input type="button" value=" Cancel"/> </div>															

รูปที่13 กำหนด User Account สำหรับ run SQL Server

ควรเลือก Use a Domain User account เพราะเป็น account ที่อยู่บน domain ของ window โดยจะเป็นชื่อที่ log in เข้า window มา แต่ถ้าไม่มี log in เข้า window ให้เลือก Use the Local System account จากนั้นคลิก Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป


Start Copying Files	
	Setup has enough information to prompt you for the licensing mode information and start copying the program files. Click Back to review or change the settings, otherwise click Next to proceed.
<div style="text-align: right;"> <input type="button" value=" &lt; Back"/> <input type="button" value=" Next &gt;"/> <input type="button" value=" Cancel"/> </div>	


รูปที่14 การยืนยันก่อนการคัดลอก file

คลิกปุ่ม Next เพื่อเข้าสู่ขั้นตอนการติดตั้งต่อไป

### Choose Licensing Mode

Microsoft SQL Server 7.0 supports two client licensing modes:

 **Per Server** - Each concurrent connection to this server requires a separate CLIENT ACCESS LICENSE.

 **Per Seat** - Each computer that accesses Microsoft SQL Server 7.0 requires a separate CLIENT ACCESS LICENSE.

Please click Help to fully understand the implications of the licensing mode you choose.

Licensing Mode

Per Server for:  concurrent connections

Per Seat

Use Licensing (located in the Administrative Tools program group) to record the number of CLIENT ACCESS LICENSES purchased and avoid violation of the license agreement.

รูปที่ 15 กำหนดประเภทของ License modes

เลือก Per server แล้วคลิกที่ Add Licenses แล้วใส่จำนวนเครื่องที่ต้องการให้ติดต่อกับ SQL Server จากนั้นคลิกปุ่ม Continue

### Setup Complete



Setup has finished installing Microsoft SQL Server 7.0 on your computer.

Click Finish to complete Setup.

< Back

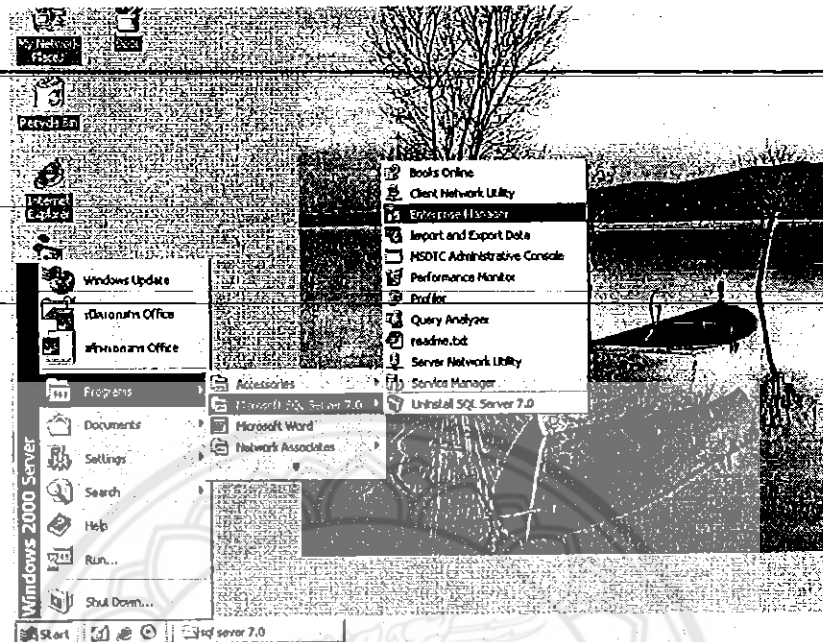
Finish

รูปที่ 16 รายงานการติดตั้งเสร็จสมบูรณ์

คลิกปุ่ม Finish เพื่อเป็นการเสร็จสิ้นการติดตั้ง

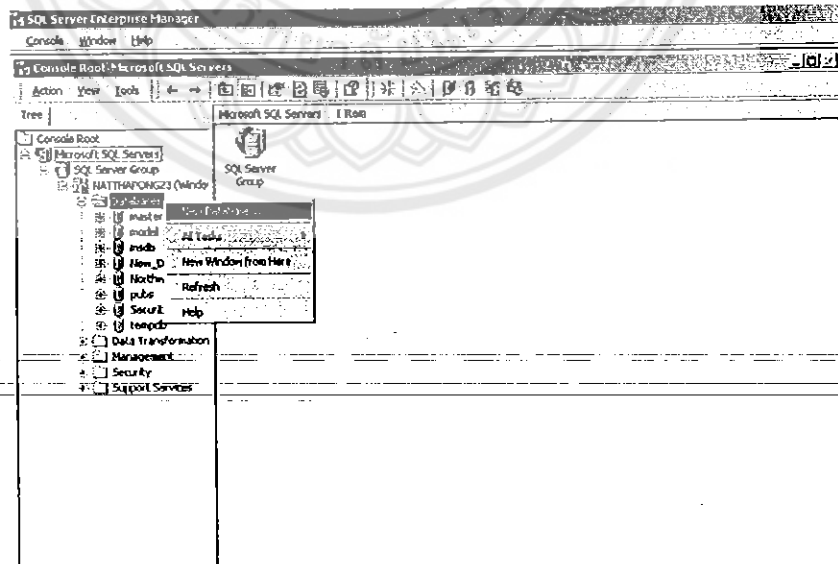


### 3. การสร้าง Database และ Table



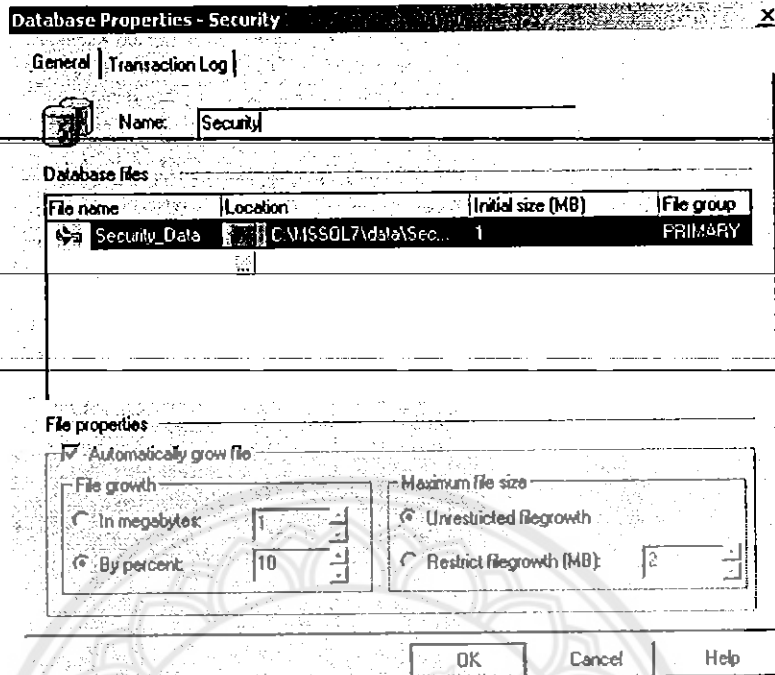
รูปที่17 การสร้าง Database โดยใช้ SQL Enterprise Manager

เลือก Start Menu, Program, Microsoft SQL Server 7.0 แล้วคลิก Enterprise Manager



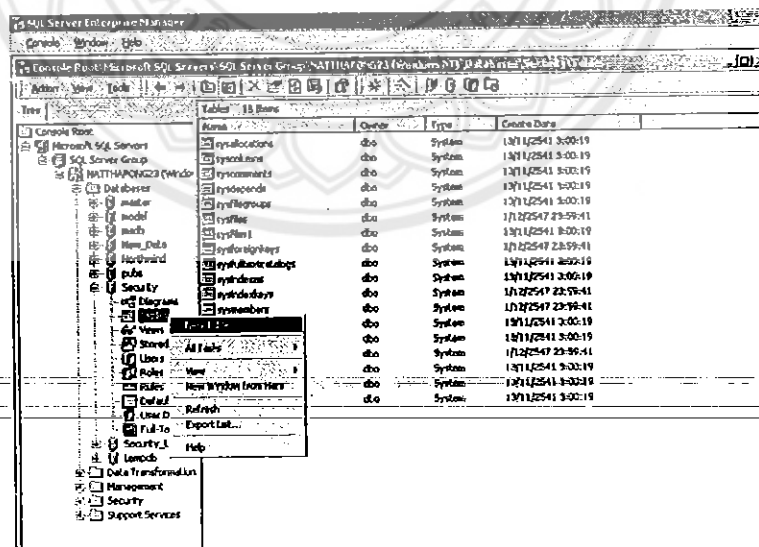
รูปที่18 การเลือกเมนูเพื่อสร้าง Database

ขยาย Folder ไปยัง database แล้วคลิกขวา เลือก New Database



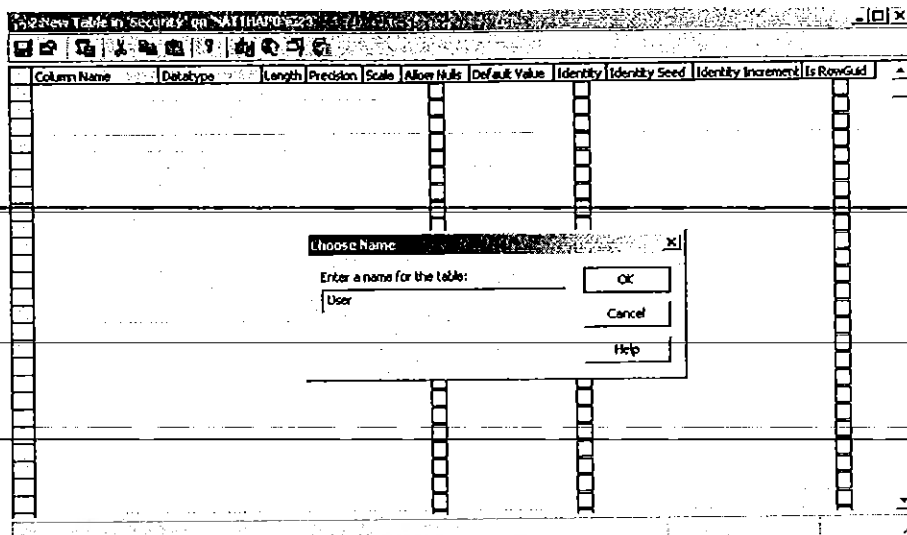
รูปที่ 19 การสร้างฐานข้อมูล

โดยใช้ชื่อ Database ที่ต้องการสร้าง



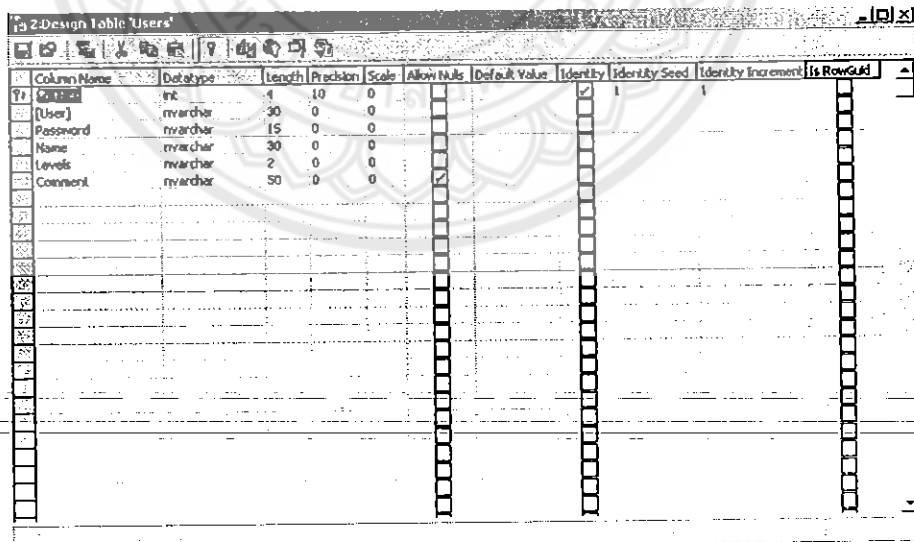
รูปที่ 20 การเลือกเมนูเพื่อสร้าง Table

ขยาย Folder ไปยังชื่อ database ที่เราต้องการ แล้วคลิกขวา เลือก New Table



รูปที่ 21 การตั้งชื่อ Table

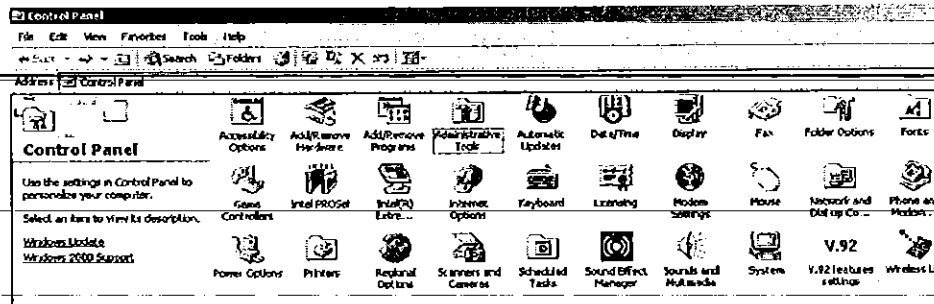
ใส่ชื่อที่ต้องการสร้าง Table ลงในช่อง Enter a name for the table



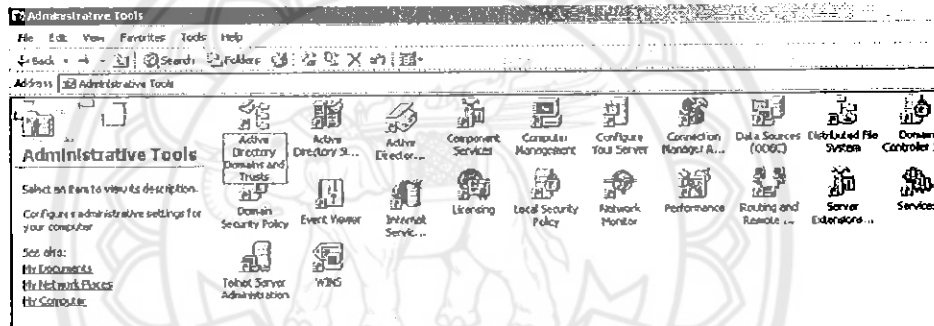
รูปที่ 22 การกำหนดโครงสร้างของ Table

กำหนดชื่อ Column และชนิดของข้อมูลใน Table

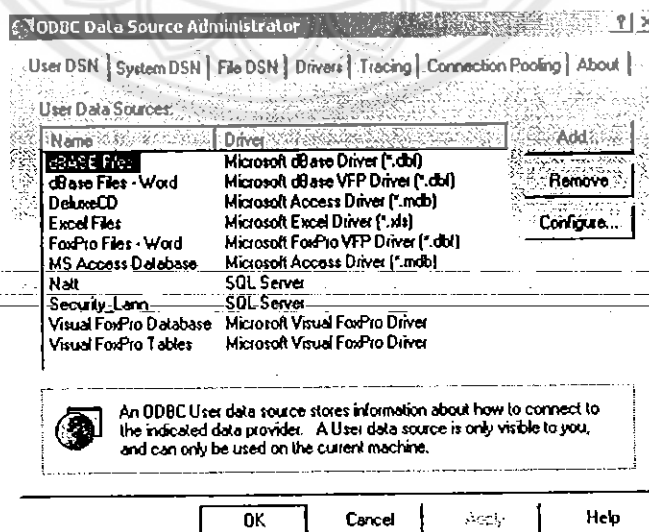
## 4 การ Setการเชื่อมต่อของฐานข้อมูลด้วยOpen Database Connectivity (ODBC)



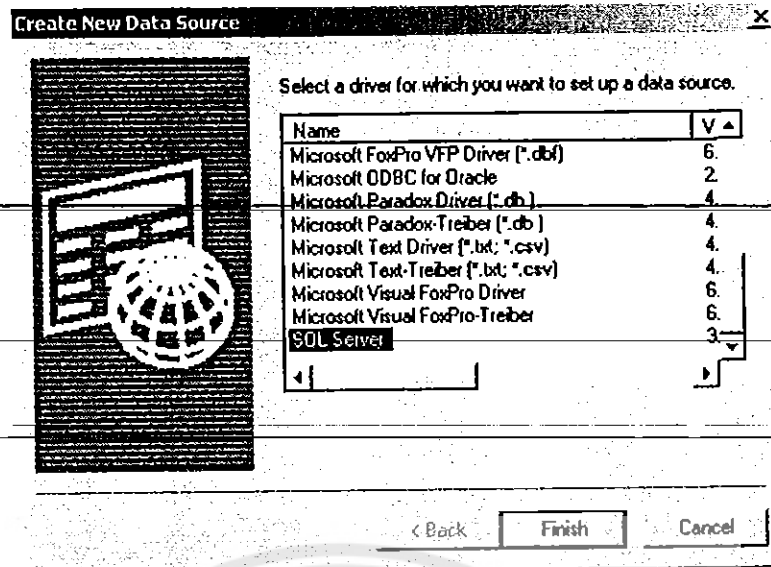
รูปที่23 เมื่อเข้าไปใน Control Panel



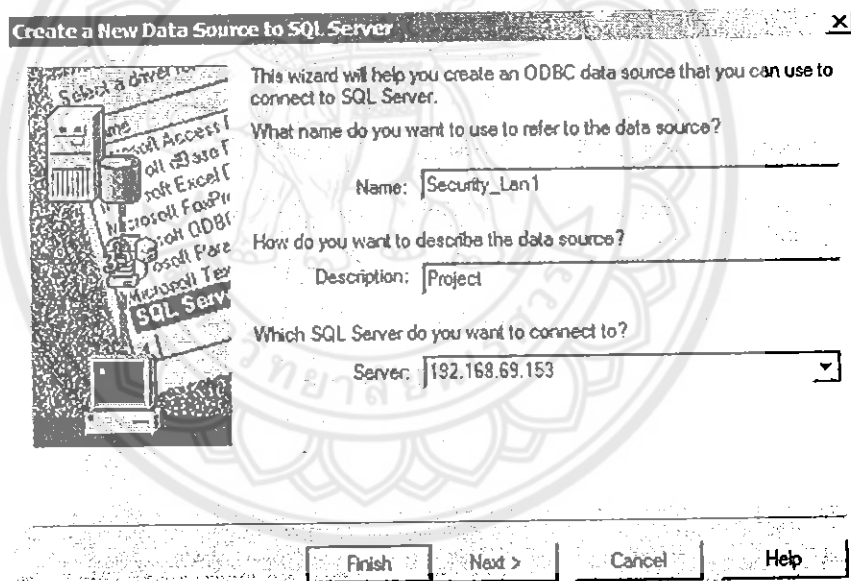
รูปที่24 เมื่อเข้าไปใน Administrative Tools



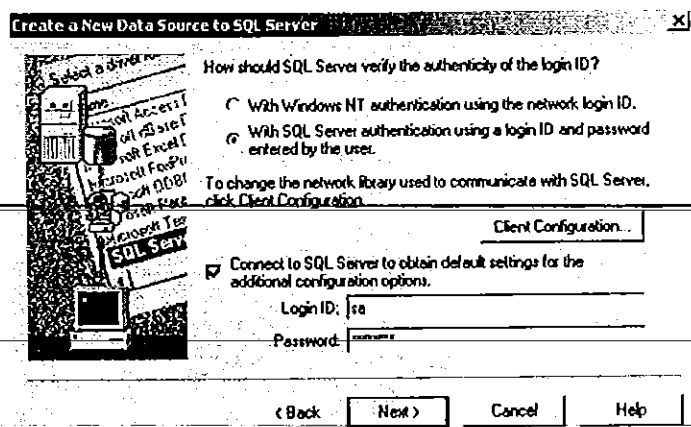
รูปที่25 การสร้างODBC



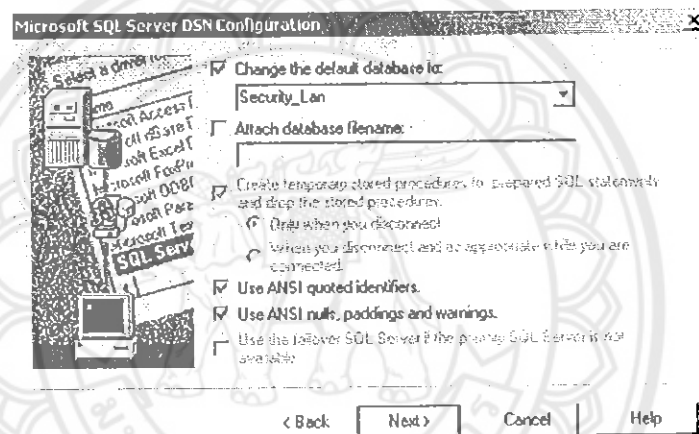
รูปที่26 การเลือกชนิดตัวเชื่อมฐานข้อมูล



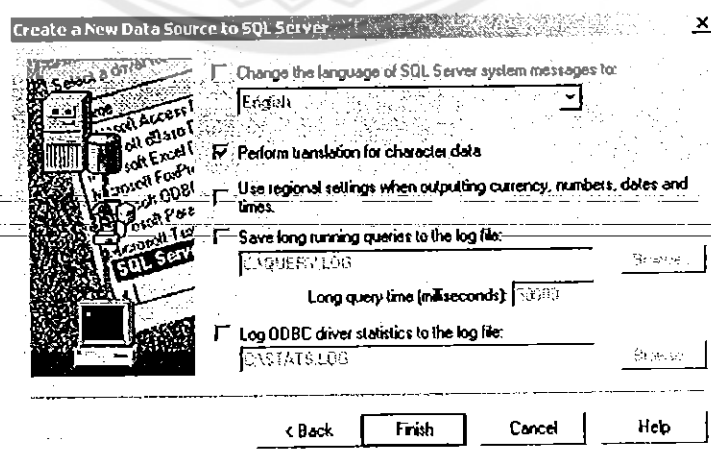
รูปที่27 การตั้งชื่อตัวเชื่อมฐานข้อมูลและเลือกชื่อเครื่อง Server ที่ต้องการติดต่อ



รูปที่28 การตั้ง Login ID และ Password



รูปที่29 การเลือกชื่อฐานข้อมูลในการติดต่อ



รูปที่30 การเลือกภาษาในการติดต่อฐานข้อมูล

**ODBC Microsoft SQL Server Setup**

A new ODBC data source will be created with the following configuration:

Microsoft SQL Server ODBC Driver Version 03.70.1063

Data Source Name: Security\_Lan1  
 Data Source Description:  
 Server: natthapong23  
 Database: (Default)  
 Language: (Default)  
 Translate Character Data: Yes  
 Log Long Running Queries: No  
 Log Driver Statistics: No  
 Use Integrated Security: No  
 Use Regional Settings: No  
 Prepared Statements Option: Drop temporary procedures on disconnect  
 Use Failover Server: No  
 Use ANSI Quoted Identifiers: Yes  
 Use ANSI Null, Paddings and Warnings: Yes

Test Data Source... OK Cancel

รูปที่ 31 สร้างต้นการเชื่อมต่อฐานข้อมูล (ODBC)

**SQL Server ODBC Data Source Test**

Test Results:

Microsoft SQL Server ODBC Driver Version 03.70.1063

Running connectivity tests...

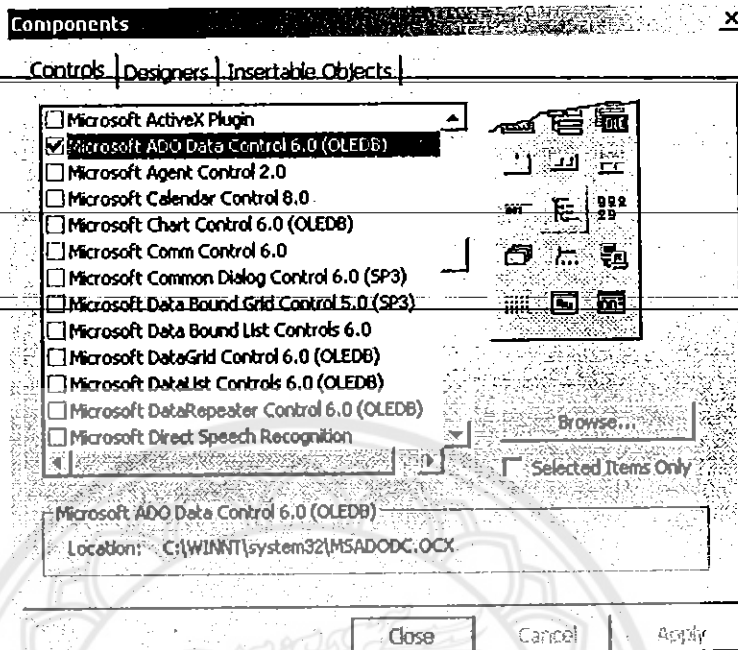
Attempting connection  
 Connection established  
 Verifying option settings  
 Disconnecting from server

TESTS COMPLETED SUCCESSFULLY!

OK

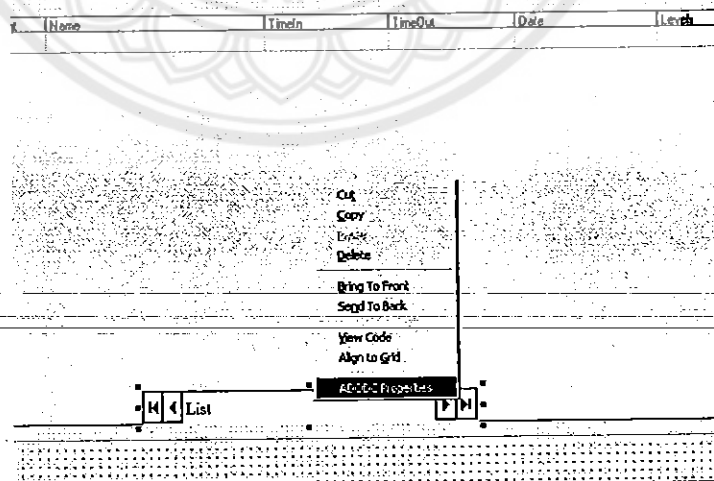
รูปที่ 32 การเชื่อมต่อสำเร็จ

## 5 การเชื่อม Visual Basic เข้ากับ Database (SQL Server)



รูปที่ 33 การติดตั้ง ADO บน Toolbox

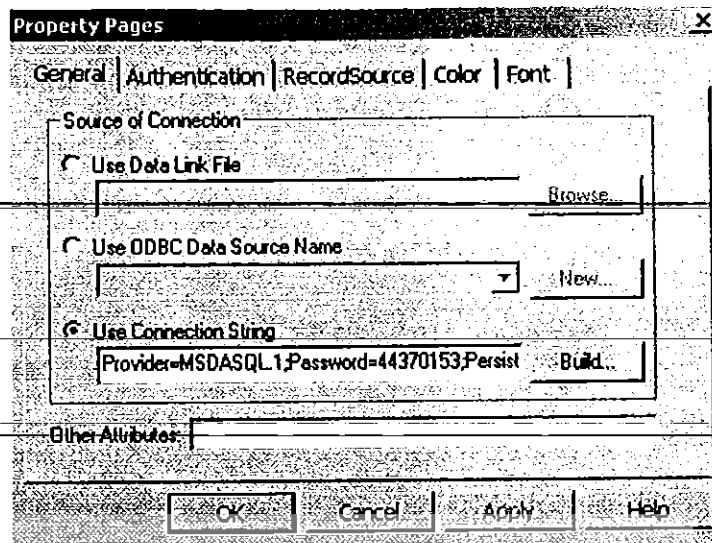
คลิกขวา Toolbox จากนั้นเลือกที่ Components จะปรากฏหน้าจอ ดังรูปที่ 33



รูปที่ 34 การทำ Object ADODC

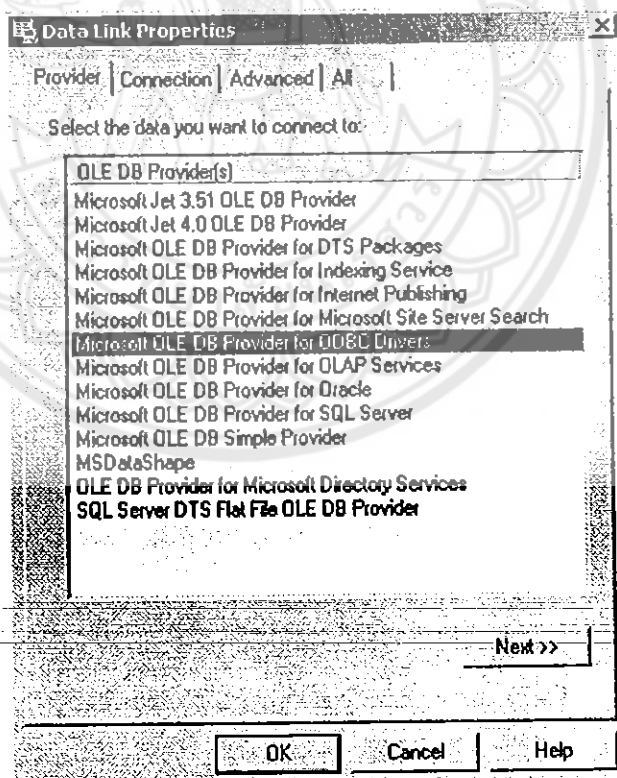
เมื่อนำ ADODC มาวางไว้บน Form แล้ว ให้คลิกขวาเลือก ADODC Properties





รูปที่35 การกำหนด Properties ของ ADODC

เลือกปุ่ม Build เพื่อทำขั้นตอนต่อไป



รูปที่36 Datalink Properties

คลิกปุ่ม Next เพื่อทำขั้นตอนต่อไป

Data Link Properties

Provider Connection | Advanced | All

Specify the following to connect to ODBC data:

1. Specify the source of data:

Use data source name  
Security\_Len1 Refresh

Use connection string  
Connection string: Build

2. Enter information to log on to the server:

User name: sa  
Password: \*\*\*\*\*

Blank password  Allow saving password

3. Enter the initial catalog to use:

Test Connection

OK Cancel Help

รูปที่ 37 หน้าจอ Connection

เลือก SQL Server ที่ต้องการติดต่อแล้ว Login ด้วย Username และ Password จากนั้นคลิกปุ่ม Test Connection

Data Link Properties

Provider Connection | Advanced | All

Specify the following to connect to ODBC data:

1. Specify the source of data:

Use data source name  
Security\_Len1 Refresh

Use connection string  
Connection string: Build

2. Enter information to log on to the server:

User name: sa  
Password: \*\*\*\*\*

Blank password  Allow saving password

3. Enter the initial catalog to use:

Test Connection

OK Cancel Help

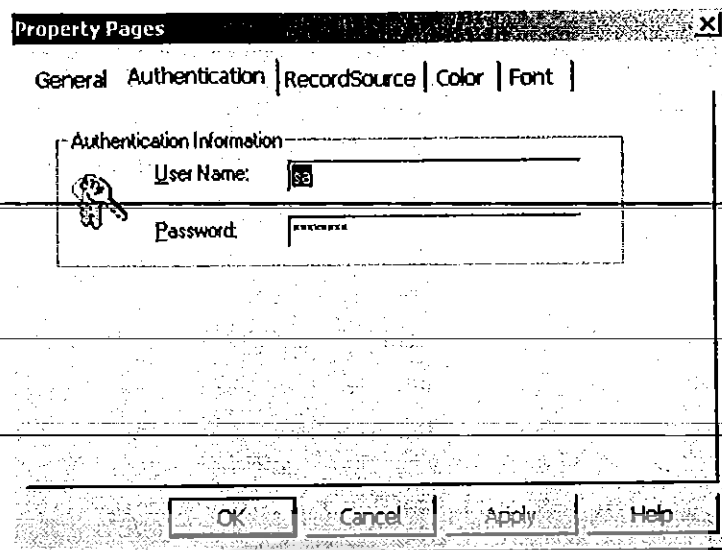
Microsoft Data Link

Test connection succeeded.

OK

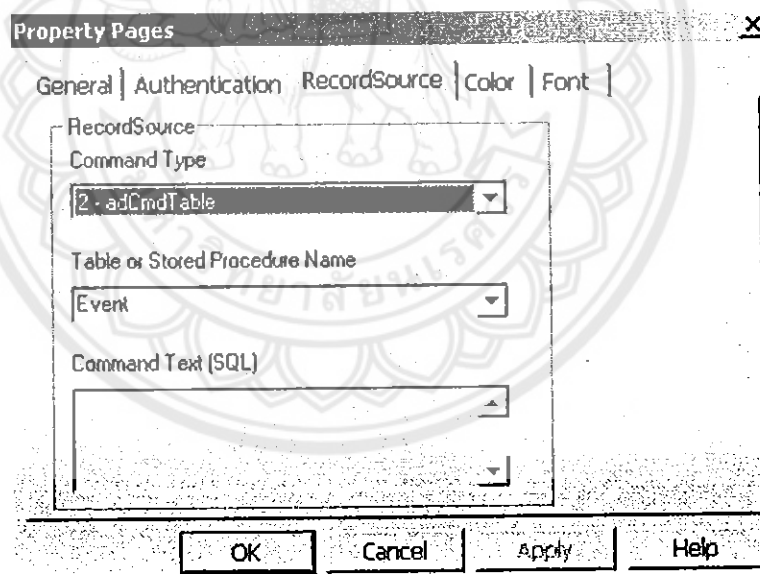
OK Cancel Help

รูปที่ 38 การเชื่อมต่อ Database สำเร็จ



รูปที่39 หน้าจอ Authentication

เมื่อใส่ Username และ Password แล้วคลิก OK



รูปที่40 หน้าจอ Recordsource

เลือก adCmdTable แล้วเลือกตารางที่ต้องการติดต่อคลิก OK เป็นการเสร็จสิ้นการ Connect

## 6 การใช้งานโปรแกรม

### 6.1 การเข้าอาคาร

- รูดบัตรบาร์โค้ด แล้วกดรหัสผ่าน ตามด้วย # เป็นการ Enter ถ้าบาร์โค้ดและรหัสผ่าน

ถูกต้อง ประตูก็จะเปิดพร้อมบันทึกชื่อ, เวลาเข้า, วันที่ และชื่อประตู

### 6.2 การออกอาคาร

- รูดบัตรบาร์โค้ด แล้วเดินผ่าน Sensor ประตูก็จะเปิด พร้อมบันทึกเวลาออก





## Source Code

---

### Source Code Visual Basic 6.0

---

'Form1: Event

Option Explicit

Dim DataIn(34), ID, DataOUT As Integer

Dim indexR As Integer

Dim DelayOFF(20) As Integer

Dim Barcode, PassID As String

Dim ChkSum As Boolean

Private Sub Form\_Load()

Timer2.Enabled = True

indexR = 0

ChkSum = False

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

End If

' เปิด Port1 ของโปรแกรม

MSComm1.Settings = "9600,N,8,1"

MSComm1.CommPort = 1

MSComm1.InputLen = 1

MSComm1.PortOpen = True

MSComm1.RThreshold = 1

End Sub

Private Sub mnuadmin\_Click()

Form2.Show

' เปิด Form2

End Sub

Private Sub mnuexit\_Click()

```
Form3.Show
```

```
End Sub
```

```
Private Sub MSComm1_OnComm()
```

---

```
Dim str As Variant
```

```
Dim i, sum As Integer
```

```
Dim Namestr, Datestr As String
```

```
On Error GoTo Error_Handler
```

```
'ถ้าโปรแกรม Error ไปที่ Error_Handler
```

---

```
str = MSComm1.Input
```

```
DataIn(indexR) = Asc(str)
```

```
indexR = indexR + 1
```

```
'หาค่า ID ว่าส่งมาจากเครื่องไหน
```

```
If indexR = 2 Then
```

```
    ID = DataIn(1)
```

```
End If
```

---

```
'ตรวจสอบการออกจากอาคาร
```

```
If indexR = 24 Then
```

```
    If (DataIn(0) = 15) And (DataIn(23) = 15) Then
```

```
        indexR = 0
```

```
        Barcode = ""
```

```
        For i = 2 To 22
```

```
            If (DataIn(i) <> 13) Then
```

```
                sum = DataIn(i)
```

```
                Select Case sum
```

---

```
                    Case 49
```

```
                        Barcode = Barcode + "1"
```

```
                    Case 50
```

```
                        Barcode = Barcode + "2"
```

```
                    Case 51
```

```
                        Barcode = Barcode + "3"
```

```
                    Case 52
```

```
Barcode = Barcode + "4"
```

```
Case 53
```

```
Barcode = Barcode + "5"
```

```
Case 54
```

```
Barcode = Barcode + "6"
```

```
Case 55
```

```
Barcode = Barcode + "7"
```

```
Case 56
```

```
Barcode = Barcode + "8"
```

```
Case 57
```

```
Barcode = Barcode + "9"
```

```
Case 48
```

```
Barcode = Barcode + "0"
```

```
Case Else
```

```
Barcode = Barcode + ""
```

```
End Select
```

```
Else
```

```
i = 22
```

```
End If
```

```
Next i
```

```
' ตรวจสอบ Data Base ของ User(Barcode) ที่ตาราง Users
```

```
Adodc2.Refresh
```

```
Adodc2.Recordset.MoveFirst
```

```
Adodc2.Recordset.Find "User = " & Barcode & "" , , adSearchForward
```

```
If Adodc2.Recordset.EOF Then
```

```
DataOUT = 252 ' ข้อมูลไม่ถูกต้องส่ง 252 ไปให้บอร์ดควบคุม
```

```
Else
```

```
' เก็บชื่อมาไว้ก่อน
```

```
Namestr = Adodc2.Recordset.Fields![Name]
```

```
Namestr = Trim(Namestr)
```

```
Datestr = Format(Date, "DD/MM/YYYY")
```



```

Adodc1.RecordSource = "SELECT * FROM Event WHERE Event.Name =" &
Namestr & " AND Event.Date =" & Datestr & " "

```

```

Adodc1.Recordset.Find "Name =" & Namestr & "", , adSearchForward

```

```

If Adodc1.Recordset.EOF Then

```

```

    DataOUT = 252 ' ข้อมูลไม่ถูกต้องส่ง252ไปให้บอร์ดควบคุม

```

```

Else

```

```

    DataOUT = 250 ' ข้อมูลถูกต้องส่ง250ไปให้บอร์ดควบคุม ประตูปิด

```

```

' หาชื่อที่ตาราง Event แล้วแสดงเวลาออก

```

```

Adodc1.Recordset.MoveLast

```

```

Adodc1.Recordset.Find "Name =" & Namestr & "", , adSearchBackward

```

```

Adodc1.Recordset.Fields![Timeout] = CStr(Time)

```

```

Adodc1.Recordset.Update

```

```

Adodc1.Recordset.Requery

```

```

Adodc1.Recordset.MoveLast

```

```

End If

```

```

Adodc1.RecordSource = "SELECT * FROM Event WHERE Event.Date =" & Datestr & " "

```

```

End If

```

```

' การส่งค่าไปให้บอร์ดควบคุม

```

```

MSComm1.Output = Chr$(128)

```

```

MSComm1.Output = Chr$(ID)

```

```

MSComm1.Output = Chr$(DataOUT)

```

```

Timer2.Enabled = True

```

```

' clear ค่า Barcode และ Password

```

```

PassID = ""

```

```

Barcode = ""

```

```

indexR = 0

```

```

For i = 0 To 34

```

```

    DataIn(i) = 0

```

```

Next i

```

```

End If

```

End If

---

'ตรวจสอบการเข้าอาคาร

---

If indexR = 35 Then

If (DataIn(0) = 0) And (DataIn(34) = 13) Then

---

'คำนวณค่า Check Sum

sum = 0

For i = 1 To 32

sum = sum + DataIn(i)

If sum > 255 Then

sum = sum - 256

End If

Next i

If (sum = DataIn(33)) Then

Else

'ถ้าข้อมูลไม่ถูกต้องให้ส่งข้อมูลไปบอกชุด MCS ให้ส่งข้อมูลชุดนี้มาใหม่

' Error

DataOUT = 254

End If

'หาค่า Barcode ที่ส่งมา

Barcode = ""

For i = 2 To 22

If (DataIn(i) <> 13) Then

---

sum = DataIn(i)

---

Select Case sum

Case 49

Barcode = Barcode + "1"

Case 50

Barcode = Barcode + "2"

Case 51

Barcode = Barcode + "3"

Case 52

Barcode = Barcode + "4"

Case 53

~~Barcode = Barcode + "5"~~

Case 54

Barcode = Barcode + "6"

Case 55

~~Barcode = Barcode + "7"~~

Case 56

Barcode = Barcode + "8"

Case 57

Barcode = Barcode + "9"

Case 48

Barcode = Barcode + "0"

Case Else

Barcode = Barcode + ""

End Select

Else

i = 22

End If

Next i

'รหัส Password ที่ส่งมา

PassID = ""

For i = 23 To 32

sum = DataIn(i)

If (sum <> 12) And (sum <> 0) Then

If sum = 11 Then

sum = 0

End If

PassID = PassID + CStr(sum)

End If

Next i

End If

' ค้นหาข้อมูลใน Database

Adodc2.Refresh

Adodc2.Recordset.MoveFirst

Adodc2.Recordset.Find "User = " & Barcode & "" , , adSearchForward

Adodc2.Recordset.Find "Password = " & PassID & "" , , adSearchForward

If Adodc2.Recordset.EOF Then

DataOUT = 252 ' ข้อมูลไม่ถูกต้องส่ง252ไปให้บอร์ดควบคุม

Else

If Adodc2.Recordset![Password] = PassID Then

Adodc3.Refresh

Adodc3.Recordset.MoveFirst

Adodc3.Recordset.Find "IDDoor = " & ID & "" , , adSearchForward

DataOUT = 250 ' ข้อมูลถูกต้องส่ง250ไปให้บอร์ดควบคุม ประตูเปิด

' Update ข้อมูล ชื่อ, เวลาเข้า, วัน, ประตู ลงDatabase

Adodc1.Refresh

Adodc1.Recordset.AddNew

Adodc1.Recordset.Fields![Name] = Adodc2.Recordset.Fields![Name]

Adodc1.Recordset.Fields![TimeIn] = CStr(Time)

Adodc1.Recordset.Fields![Levels] = Adodc3.Recordset.Fields![Name]

Datestr = Format(Date, "DD/MM/YYYY")

Adodc1.Recordset.Fields![Date] = Datestr

Adodc1.Recordset.Update

Adodc1.Recordset.Requery

Adodc1.Recordset.MoveLast

End If

End If

' การส่งค่าไปให้บอร์ดควบคุม

MSComm1.Output = Chr\$(128)

MSComm1.Output = Chr\$(ID)

```
MSComm1.Output = Chr$(DataOUT)
```

```
Timer2.Enabled = True
```

```
'clear ค่า Barcode และ Password
```

```
PassID = ""
```

```
Barcode = ""
```

```
indexR = 0
```

```
For i = 0 To 34
```

```
DataIn(i) = 0
```

```
Next i
```

```
End If
```

```
GoTo end_sub
```

```
'ไปที่ end_sub
```

```
Error_Handler:
```

```
Form9.Show
```

```
Form_Load
```

```
end_sub:
```

```
End Sub
```

```
Private Sub muabout_Click()
```

```
Form10.Show
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
Dim txt As String
```

```
'แสดงเวลาที่ Form1
```

```
txt = Format$(Time, "hh:mm:ss") & Format$(Timer - Fix(Timer), ".00")
```

```
Label2.Caption = txt
```

```
Label3.Caption = Date
```

```
End Sub
```

```
Private Sub Timer2_Timer()
```

```
Dim i As Integer
```

```

For i = 1 To 20
    MSComm1.Output = Chr$(128)
    MSComm1.Output = Chr$(i)
MSComm1.Output = Chr$(251)
Next i
Timer2.Enabled = False
End Sub

```

```

Private Sub Timer3_Timer()
    Dim sql As String
        ' Refresh หน้าจอทุก5วินาที
    sql = "SELECT ID_Event, Name, TimeIn, TimeOut, Date, Levels "
    sql = sql & " FROM Event"

    With Adodc1
        .CommandType = adCmdText
        .RecordSource = sql
        .Refresh
    End With
End Sub

```

```

Private Sub Timer4_Timer()
    Unload Form9
End Sub

```

---

### **'Form2: Check Username และ Password ของผู้ดูแลระบบ**

```

Option Explicit
Dim Conn As New ADODB.Connection
Dim rsTmp As New ADODB.Recordset
Dim Invalid As Integer
Dim rsToAffect As Long

```

```
Private Const strConn = "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=Security_Lan;Data Source=NATTHAPONG23"
```

---

```
Private Sub Command1_Click()
```

```
Dim sql As String
```

```
' Check Username และ Password ที่ตาราง Admins
```

```
sql = "SELECT Username, Password"
```

```
sql = sql & " FROM Admins"
```

```
sql = sql & " WHERE (Username='" & Text1.Text & "'"
```

```
sql = sql & "AND(Password='" & Text2.Text & "'")"
```

```
If Text1.Text = "" Then
```

```
MsgBox "Have not Data!!!!", vbYes + vbCritical, "Error"
```

```
Else
```

```
With Adodc1
```

```
.CommandType = adCmdText
```

```
.RecordSource = sql
```

```
.Refresh
```

```
If Adodc1.Recordset.RecordCount < 0 Then
```

```
Load Form4
```

```
Form4.Show
```

```
Unload Form2
```

```
Else
```

```
MsgBox "Username and Password Wrong!!!!", vbYes + vbCritical, "Error"
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
End If
```

```
End With
```

```
End If
```

```
End Sub
```

---

```

Private Sub Command2_Click()
If MsgBox("Are you exit ?", vbYesNo + vbQuestion, "Exit") = vbYes Then
Unload Form2
End If
End Sub

```

### 'Form3: Check Password และ ออกจากการทำงาน

Option Explicit

```

Private Sub Command1_Click()
Dim sql As String
' Check Password ที่ตาราง Admins เพื่อออกจาก โปรแกรม
sql = "SELECT Password"
sql = sql & " FROM Admins"
sql = sql & " WHERE (Password='" & Text1.Text & "'"
If Text1.Text = "" Then
MsgBox " Have not Data!!!!", vbYes + vbCritical, "Error"
Else
With Adodc1
.CommandType = adCmdText
.RecordSource = sql
.Refresh
If Adodc1.Recordset.RecordCount <> 0 Then
If MsgBox("Are you exit program ?", vbYesNo + vbQuestion, "Exit program") = vbYes
Then
End
End If
Else
MsgBox "Password Wrong!!!!", vbYes + vbCritical, "Error"

```



Text1.Text = ""

End If

End With

End If

End Sub

---

Private Sub Command2\_Click()

Unload Form3

End Sub

---

**'Form4: เลือกรายละเอียดการ Set ระบบ**

Option Explicit

---

Private Sub Command1\_Click()

Form5.Show

Unload Form4

End Sub

---

Private Sub Command2\_Click()

Form6.Show

Unload Form4

End Sub

---

Private Sub Command3\_Click()

Form7.Show

Unload Form4

End Sub

---

Private Sub Command4\_Click()

Form8.Show

Unload Form4

End Sub

---

Private Sub Command6\_Click()

---

Unload Form4

---

End Sub

---

**'Form5: เพิ่ม, ลค, แก้ไข ผู้ดูแลระบบ**

---

Option Explicit

---

Private Sub Command1\_Click()

If (Text1.Text = "") Or (Text2.Text = "") Or (Text3.Text = "") Or (Text4.Text = "") Then

MsgBox "Not verify data", vbOKOnly + vbCritical, "Error"

Else

Text1.Enabled = True

Text2.Enabled = True

Text3.Enabled = True

Text4.Enabled = True

Text5.Enabled = True

Adodc1.Recordset.AddNew 'เพิ่มผู้ดูแลระบบ ที่ตาราง Admins

Adodc1.Enabled = False

Command1.Enabled = False

Command2.Enabled = False

Command3.Enabled = False

---

Command4.Enabled = False

Command5.Visible = True

Command6.Visible = True

End If

End Sub

---

```
Private Sub Command2_Click()
If (Text1.Text = "") Or (Text2.Text = "") Or (Text3.Text = "") Or (Text4.Text = "") Then
MsgBox "Not verify data", vbOKOnly + vbCritical, "Error"
```

---

```
Else
```

```
Text1.Enabled = True
```

```
Text2.Enabled = True
```

```
Text3.Enabled = True
```

---

```
Text4.Enabled = True
```

```
Text5.Enabled = True
```

```
Adodc1.Recordset.Update 'แก้ไขข้อมูลระบบ ที่ตาราง Admins
```

```
Command1.Enabled = False
```

```
Command2.Enabled = False
```

```
Command3.Enabled = False
```

```
Command4.Enabled = False
```

```
Command5.Visible = True
```

```
Command6.Visible = True
```

```
End If
```

```
End Sub
```

---

```
Private Sub Command3_Click()
```

```
If (Text1.Text = "") Or (Text2.Text = "") Or (Text3.Text = "") Or (Text4.Text = "") Then
```

```
MsgBox "Not verify data", vbOKOnly + vbCritical, "Error"
```

---

```
Else
```

```
If MsgBox("Are you delete data record?", vbYesNo + vbQuestion, "Delete") = vbYes Then
```

```
With Adodc1.Recordset 'ลบข้อมูลระบบ ที่ตาราง Admins
```

```
.Delete
```

```
.MoveNext
```

```
If Adodc1.Recordset.EOF Then
```

```
Adodc1.Recordset.MoveLast
```

```
End If
```

End With

End If

End If

End Sub

---

Private Sub Command4\_Click()

Form4.Show

Unload Form5

End Sub

---

Private Sub Command5\_Click()

If (Text1.Text = "") Or (Text2.Text = "") Or (Text3.Text = "") Or (Text4.Text = "") Then

MsgBox "Not verify data", vbOKOnly + vbCritical, "Error"

Else

Adodc1.Enabled = True

Adodc1.Recordset.MoveNext

Command1.Enabled = True

Command2.Enabled = True

Command3.Enabled = True

Command4.Enabled = True

Command5.Visible = False

Command6.Visible = False

Text1.Enabled = False

Text2.Enabled = False

Text3.Enabled = False

Text4.Enabled = False

Text5.Enabled = False

End If

End Sub

---

```
Private Sub Command6_Click()
```

```
Adodc1.Enabled = True
```

```
Command1.Enabled = True
```

```
Command2.Enabled = True
```

```
Command3.Enabled = True
```

```
Command4.Enabled = True
```

```
Command5.Visible = False
```

```
Command6.Visible = False
```

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Text3.Enabled = False
```

```
Text4.Enabled = False
```

```
Text5.Enabled = False
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Command5.Visible = False
```

```
Command6.Visible = False
```

```
Text1.Enabled = False
```

```
Text2.Enabled = False
```

```
Text3.Enabled = False
```

```
Text4.Enabled = False
```

```
Text5.Enabled = False
```

```
End Sub
```

**Form6: เพิ่ม, ลด, แก้ไข ผู้ใช้**

Option Explicit

```
Private Sub Command1_Click()
```

```
Form4.Show
```

```
Unload Form6
```

End Sub

---

Private Sub DataGrid1\_BeforeDelete(cancel As Integer)

~~If MsgBox("Are you delete ?", vbYesNo + vbQuestion, "Delete") = vbYes Then~~

cancel = False

Else

cancel = True

End If

End Sub

---

**'Form7: แสดงรายงานการเข้า-ออก**

Option Explicit

---

Private Sub Command1\_Click()

Form4.Show

Unload Form7

End Sub

---

Private Sub Command2\_Click()

With DataReport1

'แสดงรายงานการเข้า-ออก

Set .DataSource = Adodc1

.Caption = "Print"

.Show

End With

End Sub

---

Private Sub Command3\_Click()

If MsgBox("Are you delete record?", vbYesNo + vbQuestion, "Delete Record") = vbYes Then

With Adodc1.Recordset

.Delete

'ลบรายชื่อของผู้ใช้

.MoveNext

If Adodc1.Recordset.EOF Then

```

        Adodc1.Recordset.MoveLast
    End If
End With
End If
End Sub

```

### Form8: การค้นหาข้อมูล

Option Explicit

```
Private Sub Command1_Click()
```

```
Form4.Show
```

```
Unload Form8
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Dim sql As String
```

```
' ค้นหาข้อมูลของผู้เข้าใช้อาคารตามรายชื่อ ที่ตาราง Event
```

```
sql = "SELECT ID__Event, Name, TimeIn, TimeOut, Date, Levels "
```

```
sql = sql & " FROM Event"
```

```
sql = sql & " WHERE (Name='" & Text1.Text & "')"
```

```
If Text1.Text = "" Then
```

```
MsgBox "Not Have Data In Text Box!!!!", vbYes + vbCritical, "Error"
```

```
Else
```

```
Adodc1.Refresh
```

```
With Adodc1
```

```
.CommandType = adCmdText
```

```
.RecordSource = sql
```

```
.Refresh
```

```
End With
```

```
End If
```

```
End Sub
```

---

Private Sub Command3\_Click()

With DataReport2

Set .DataSource = Adodc1

.Caption = "Print"

.Show

End With

End Sub

---

### Source Code Assembly

```

Data EQU P3.4 ; ขา T0
CLK EQU P3.2 ; ขา Interrupt 0
RELAYEQU P1.7 ; ควบคุมการปิดเปิด Relay เพื่อควบคุมประตู
IDChk EQU 00h ; Check ID Flag
Error EQU 01h ; Error Resend data to Computer
OUT EQU 02h
TX EQU P3.5 ; TX Flag

```

```

IndexBC EQU 21h ; Index to Record BC
IndexPW EQU 22h ; Index to Record PW
BuffSend EQU 23h ; Buffer to Send data to Keyboard
ROK EQU 24h ; Receive OK Flag
Kpdata EQU 25h ; Key PAD Read Buffer

```

---

```

ID EQU 2Fh ; ID ของบอร์ด

```

---

```

ChkSum EQU 4Fh ; Check Sum

```

```

indexR EQU 5Fh ; Buffer to Recieve from computer

```

```

org 0000h

```

```

ljmp Initial

```

```

org 0023h

```



```
ljmp    GetData
```

```
;
```

```
org    0030h
```

```
Initial: setb    EA
```

```
setb    ES
```

```
mov     TMOD,#20h
```

```
mov     TH1,#0FDh
```

```
mov     PCON,#00h
```

```
mov     SCON,#50h
```

```
setb    TRI
```

```
mov     ID,#02h           ; ระบุ ID ของ บอร์ด
```

```
mov     IndexBC,#30h      ; เก็บข้อมูล BarCode ไว้ที่ตำแหน่ง 30H - 44H ( 20 ตัว)
```

```
mov     IndexPW,#45h     ; เก็บข้อมูล PassWord จาก Key PAD ไว้ที่ตำแหน่ง 45H - 4FH (10 ตัว)
```

```
mov     ROK,#00h         ; ROK = 2 จะส่งข้อมูลทั้ง BC และ PW ไปให้ COM
```

```
mov     indexR,#60h      ; ตั้งค่าตัวรับข้อมูล ไว้ที่ตำแหน่ง 60h
```

```
clr     Error            ; Flag Error if Computer Recieve Data Error to Resend
```

```
setb    P3.7            ; LED Status ON ->> OFF
```

```
clr     RELAY           ; Relay OFF ประตูไม่เปิด
```

```
clr     IDChk
```

```
mov     R0,#30h
```

```
clrAll: mov     @R0,#00h
```

```
inc     R0
```

```
cjne   R0,#50h,clrAll
```

```
clr     OUT
```

```
clr     TX
```

```
;
```

```
===== Main Program =====
```

```
Main:
```

```
jn     P3.4,ChkOut
```

```
acall  GetKeyPAD         ; อ่านค่าจาก KeyBoard
```

```
acall  ChkLogIn         ; ตรวจสอบว่ามีการใส่ Barcode และ Password หรือยัง
```

```

}
    jb    Error,ResendCom    ; ถ้าคอมรับข้อมูลผิดก็ส่งข้อมูลไปให้ใหม่
    sjmp  Main

```

---

ResendCom:

```

    mov  ROK,#03h
    acall ChkLogIn

```

---

```

    sjmp  Main

```

---

ChkOut:

```

    acall Delay_100ms
    jnb  P3.4,$
    acall Delay_100ms
    jb   ROK.1,SndOut
    sjmp Main

```

SndOut:

```

    push acc
    push 00
    mov  A,#15 ; Start Bits = 15
    acall TX_BYTE
    mov  A,ID ; ID
    acall TX_BYTE
    mov  R0,#30h

```

sndo:

```

    mov  A,@R0 ; BarCode
    acall TX_BYTE
    inc  R0
    cjne R0,#45h,sndo
    mov  A,#15
    mov  indexR,#60h
    acall TX_BYTE ; Stop
    clr  ROK.1
    mov  IndexBC,#30h

```

```

pop    00
pop    acc
sjmp   Main

```

---

```

;=====END Main=====

```

---

```

; รับข้อมูลที่ส่งมาจากคอมพิวเตอร์ 50h - 52h

```

---

GetData:

```

clr    RI
push   Acc
mov    A,SBUF
jnb    IDChk,checkID
sjmp   GetCom

```

CheckID:

```

cjne   A,ID,LastWd
setb   IDChk

```

GetCom:

```

push   00
mov    R0,indexR
mov    @R0,A
inc    R0
mov    indexR,R0
cjne   R0,#62h,ExtIntCom ; เช็คว่ารับข้อมูลครบ 2 ตัวหรือยัง (ID + Data)

```

```

mov    A,60h
cjne   A,ID,clrDCom      ; ตรวจสอบว่า ID ที่รับมานั้นตรงกับตัวเครื่องหรือเปล่า
mov    A,61h
cjne   A,#254,chkopen    ; คอมรับข้อมูลผิดพลาดให้ส่งข้อมูลไปใหม่
setb   Error
sjmp   clrDCom

```

chkopen:

```

cjne A,#250,chkclose ;คอมสั่งให้ เปิดประตู
clr P3.7
setb RELAY
sjmp clrDCom

```

chkclose:

;คอมสั่งให้ปิดประตู

```

cjne A,#251,chkError
setb P3.7
clr RELAY
sjmp clrDCom

```

chkError:

;ถ้าคีย์รหัสไม่ถูกต้อง LED จะกระพริบแล้วดับ

```

cjne A,#252,clrDCom
clr P3.7
acall Delay_100ms
acall Delay_100ms
acall Delay_100ms
setb P3.7
acall Delay_100ms
acall Delay_100ms
acall Delay_100ms
clr P3.7
acall Delay_100ms
acall Delay_100ms
acall Delay_100ms
setb P3.7
acall Delay_100ms
acall Delay_100ms
acall Delay_100ms
clr P3.7
acall Delay_100ms
acall Delay_100ms
acall Delay_100ms

```

```

    setb    P3.7
    sjmp    clrDCom

```

ExtIntCom:

```

    pop     00
    pop     Acc
    reti

```

clrDCom:

```

    mov     IndexR,#60h
    clr     IDChk
    sjmp    ExtIntCom

```

; รับข้อมูลจาก Barcode (RS-232)

LastWd:

```

    cjne    A,#0Ah,RecData
    setb    ROK.1
    mov     IndexBC,#30h
    sjmp    extR232

```

RecData:

```

    push    00
    mov     R0,IndexBC
    mov     @R0,A
    inc     R0
    mov     IndexBC,R0
    pop     00

```

extR232:

```

    pop     Acc
    reti

```

;

ChkLogIn:

```

    push    Acc

```

```

mov    A,ROK
cjne  A,#03h,extLG ; เช็คว่ามีการใส่ข้อมูลครบทั้ง BarCode และ Password แล้ว
mov    R1,#2Fh      ; ส่งจาก ID


---


acall  CheckSum     ; check sum


---


mov    A,#00h      ; ส่ง Start Bits


---


acall  TX_BYTE


---



```

#### SndData:

```

mov    A,@R1
inc    R1
acall  TX_BYTE
cjne  R1,#50h,SndData
mov    a,#13
mov    indexR,#60h
acall  TX_BYTE     ; Stop Bits
mov    ROK,#00h
clr   Error

```

#### extLG:

```

pop    Acc
ret

```

#### Checksum:

```

push  00


---


push  Acc


---


push  B


---


mov   A,#00h


---


mov   B,#00h


---


mov   R0,#2Fh

```

#### lpSum:

```

mov   B,@R0

```

```

inc    R0
Add    A,B
cjne   R0,#4Eh,lpSum


---


mov    4Fh,A           ; inu Check sum 4FH


---


pop    B


---


pop    Acc


---


pop    00


---


ret

```

GetKeyPAD:

```

push   Acc
mov    A,P1
and    A,#10000000B
orl    A,#01111000B
mov    P1,A
nop
mov    A,P1
and    A,#01111111B
cjne   A,#01111000B,KPScan
sjmp   extKP

```

KPScan:

```

cjne   A,#70h,ROW2
mov    KPDATA,#1
acall  KPScanCOL


---


sjmp   KPWaitRelease

```

ROW2:

```

cjne   A,#68h,ROW3
mov    KPDATA,#4
acall  KPScanCOL
sjmp   KPWaitRelease

```

ROW3:

```

cjne   A,#58h,ROW4

```

```

mov   KPDATA,#7
acall KPScanCOL
sjmp  KPWaitRelease

```

ROW4:

```

cjne  A,#38h,extKP
mov   KPDATA,#10
acall KPScanCOL
sjmp  KPWaitRelease

```

KPWaitRelease:

```

; รอกนกว่าจะปล่อยปุ่มกด
mov   A,P1
anl  A,#10000000B
orl  A,#01111000B
mov  P1,A
nop
mov  A,P1
anl  A,#01111111B
cjne A,#01111000B,KPWaitRelease
push 00
mov  R0,IndexPW
cjne R0,#45h,GetKeyData ; กดตัวแรกให้กรีขของเก่าออกก่อน

```

clrKPD:

```

mov  @R0,#00h
inc  R0
cjne R0,#4Fh,clrKPD
mov  R0,#45h

```

GetKeyData:

```

mov  @R0,KPDATA
inc  R0

```



```

mov   IndexPW,R0
pop   00
mov   A,KPDATA

```

---

```

cjne  A,#12,chkCLR

```

Check ENTER

```

setb  ROK.0

```

---

```

mov   IndexPW,#45h

```

---

```

sjmp  extKP

```

chkCLR:

```

cjne  A,#10,extKP      ; check CLEAR
clr   ROK.0
push  00
mov   R0,#45h
clrKP: mov  @R0,#00h
inc   R0
cjne  R0,#4Fh,clrKP
mov   IndexPW,#45h
pop   00

```

---

```

extKP:
pop   Acc
ret

```

KPScanCOL:

```

mov   A,P1

```

---

```

anl   A,#10000000B

```

---

```

orl   A,#00000111B
mov   P1,A
nop
mov   A,P1
anl   A,#0000111B
cjne  A,#03h,COL2

```

```

mov  A,KPDATA
add  A,#0
mov  KPDATA,A
sjmp extcol

```

COL2:

```

cjne A,#05h,COL3
mov  A,KPDATA
add  A,#1
mov  KPDATA,A
sjmp extcol

```

COL3:

```

cjne A,#06h,extcol
mov  A,KPDATA
add  A,#2
mov  KPDATA,A

```

extcol:

```
ret
```

```

;
;ออกจากห้อง

```

Delay\_100ms:

```
push 07h
```

```
push 06h
```

```
mov  R7,#100
```

Delay\_100ms2:

```
mov  R6,#0E6h
```

Delay\_100ms1:

```
nop
```

```

nop
djnz R6,Delay_100ms1
djnz R7,Delay_100ms2

```

---

```

pop 06h

```

```

pop 07h

```

```

ret

```

---

### Delay\_1Sec:

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

lcall Delay_100ms

```

```

ret

```

---

### TX\_BYTE:

```

setb TX

```

```

CLR ES

```

```

MOV SBUF,A

```

---

```

JNB TI,$

```

```

CLR TI

```

```

setb ES

```

```

clr TX

```

```

RET

```

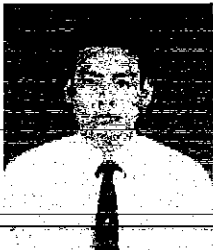
---

```

END

```

## ประวัติผู้เขียนโครงการ



ชื่อ นามสกุล ชินบุญ เอี่ยมฉิม  
 เกิดวันที่ 16 มิถุนายน 2525  
 ภูมิลำเนา 268/15 ถนนบรมไตรโลกนารถ  
 ตำบลในเมือง อำเภอเมือง  
 จังหวัดพิษณุโลก 65000

### ประวัติการศึกษา

- จบการศึกษาชั้นประถมศึกษาจากโรงเรียนอนุบาล  
 โรจนวิทย์มาลาเปียง อำเภอเมือง จังหวัดพิษณุโลก
- จบการศึกษาชั้นมัธยมศึกษาจากโรงเรียนพิษณุโลก  
 พิตยาคม อำเภอเมือง จังหวัดพิษณุโลก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4  
 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-Mail : sri\_prasert5@hotmail.com



ชื่อ นามสกุล ณัฐพงษ์ ตำนียงงาม  
 เกิดวันที่ 23 มิถุนายน 2523  
 ภูมิลำเนา 806 หมู่1 ตำบลบ่อพลอย อำเภอบ่อพลอย  
 จังหวัดกาญจนบุรี 71160

### ประวัติการศึกษา

- จบการศึกษาชั้นประถมศึกษาจากโรงเรียนกาญจนจินดา  
 อำเภอเมือง จังหวัดกาญจนบุรี
- จบการศึกษาชั้นมัธยมศึกษาจากโรงเรียนวิสุทธิรังษี  
 อำเภอเมือง จังหวัดกาญจนบุรี
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4  
 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
 มหาวิทยาลัยนเรศวร

E-Mail : natthapong\_23@yahoo.com