ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ที่จอดรถอัจฉริยะ

INTELLIGENT PARKING

โดย

ศฤงคาร อดุลศิริ

ชินวัฒน์ ฉัตรวิริยะเจริญ

ศิลป์ธวัฒน์ ศตกมลวุฒิ

อาจารย์ที่ปรึกษา

อาจารย์อนันตพัฒน์ อนันตชัย

วศ.บ.
ศ๒๙๓ ก
2550

เลขหมู่.....
เลขทะเบียน 04782
วัน,เดือน,ปี - 8 ต.ค. 2551

b.11971984
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTELLIGENT PARKING

SARINGKARN ADUNSIRI

CHINNAWAT CHATVIRIYAJALEARN

SILPTHAWAT SATAKAMOLWUT

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2008

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2550

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ที่จอดรถอัจฉริยะ

INTELLIGENT PARKING

ผู้จัดทำ

1. นาย ศฤงคาร	**อดุลศิริ**	**รหัสประจำตัว 47070042**
2. นาย ชินวัฒน์	**ฉัตรวิรยะเจริญ**	**รหัสประจำตัว 47070070**
3. นาย ศิลป์ธวัฒน์	**ศตกมลวุฒิ**	**รหัสประจำตัว 47070092**

.....อาจารย์ที่ปรึกษา

(อาจารย์ อนันตพัฒน์ อนันตชัย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อ	ที่จอดรถอัจฉริยะ
นักศึกษา	นาย ศฤงคาร อดุลศิริ 47070042
	นาย ชินวัฒน์ ฉัตรวิรยะเจริญ 47070070
	นาย ศิลป์ธวัฒน์ ศตกมลวุฒิ 47070092
ปริญญา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2550
อาจารย์ที่ปรึกษา	อาจารย์ อนันตพัฒน์ อนันตชัย

บทคัดย่อ

เนื่องจากการสถานที่จอดรถนั้น ปัจจุบันต้องอาศัยแรงงานคน ซึ่งในบางครั้งไม่สามารถทำงานได้อย่างเต็มที่ซึ่งเป็นเรื่องที่ต้องประสบปัญหามาก ทั้งด้านความสะดวกและด้านความถูกต้อง ดังนั้น จึงจัดทำระบบที่จอดรถอัจฉริยะขึ้นมา ซึ่งสามารถจำลองสถานที่จอดรถ โดยสามารถตรวจสอบ ได้ตลอด โดยข้อมูลที่ได้ตรงกับความเป็นจริง ทั้งนี้เพื่อ อำนวยความสะดวกและลดข้อผิดพลาดในการตรวจสอบที่จอดรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Intelligent Parking
Student	Mr.Saringkarn Adunsiri 47070042
	Mr.Chinnawat Chatviriyajalearn 47070070
	Mr.Silpthawat Satakamolwut 47070092
Advisor	Bachelor of Science
Programme	Information Technology
Faculty	Information Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2007
Advisor	Mr. Anuntaput Anuntachai

ABSTRACT

Nowadays, the status of parking places is generally monitored by human, which can cause error and delay and may not meet the requirement of users.This project develops a parking monitoring system that electronically checks the status of each parking place and display the result through various channels. Users can view the result through websites or a mobile device. This improves the service as well as reduces the error cursed in the original system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์สำเร็จลุล่วงได้อย่างดี เพราะได้รับความเมตตากรุณา คำแนะนำและคำปรึกษาจาก อาจารย์ อนันตพัฒน์ อนันตชัย ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ คณะผู้จัดทำรู้สึกทราบซึ้งในความอนุเคราะห์จากท่าน และขอกราบขอบพระคุณเป็นอย่างสูง

กราบขอบพระคุณบิดา มารดา และขอขอบคุณครอบครัวของกลุ่มผู้จัดทำ ที่ได้ให้การสนับสนุนในการศึกษาและจัดทำทั้งทางด้านกำลังใจและการเงิน

ขอขอบพระคุณอาจารย์ประจำคณะเทคโนโลยีสารสนเทศ เป็นอย่างสูงที่สละเวลาอันมีค่าให้ข้อคิดแนวทางในบางจุดที่ผู้จัดทำติดปัญหาบางอย่าง ซึ่งมีส่วนช่วยให้ปริญญานิพนธ์สำเร็จไปด้วยดี

ขอขอบคุณเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศ ที่อนุเคราะห์ในการอำนวยความสะดวกต่างๆ ในการทำงานตั้งแต่ต้น รวมถึงพี่ๆ เพื่อนๆนักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่างๆและให้กำลังใจต่อคณะผู้จัดทำโดยตลอด

ศฤงคาร อดุลศิริ
ชินวัฒน์ฉัตรวิริยะเจริญ
ศิลป์ธวัฒน์ ศตกมลวุฒิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ..... I

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญรูป.....VI

สารบัญตาราง.....VIII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและความสำคัญของปัญหา.....1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....1

1.3 สมมติฐานของการศึกษา.....1

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย.....2

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน.....2

1.6 ขอบเขตการวิจัย.....2

1.7 ขั้นตอนการศึกษา.....3

บทที่ 2 ทฤษฎีพื้นฐาน และเทคโนโลยีที่เกี่ยวข้อง.....4

2.1 ทฤษฎี ตัวรับ และ ตัวส่ง sensor infrared (แบบตัดแสง).....4

2.2 IC เบอร์ 555.....5

2.3 การสื่อสารข้อมูลแบบอนุกรม.....11

2.4 รีจิสเตอร์ควบคุมพอร์ตอนุกรม.....14

2.5 การส่งข้อมูลทางพอร์ตอนุกรม.....17

2.6 การคำนวณอัตราบอดของพอร์ตอนุกรม.....22

2.7 การสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์หลายตัว.....26

2.8 โครงสร้างไมโครคอนโทรลเลอร์.....27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 3 การวิเคราะห์และการออกแบบระบบ.....37

3.1 การวิเคราะห์.....37

3.2 การออกแบบ.....38

3.3 แผนภาพขั้นตอนการทำงานของระบบ.....41

3.4 สรุปการทำงาน.....42

บทที่ 4 การทดลองและผลการทดลอง.....43

4.1 การทดลอง.....43

4.2 ผลการทดลอง.....43

บทที่ 5 สรุปโครงงานและข้อเสนอแนะ.....50

5.1 สรุปโครงการ.....50

5.2 ข้อเสนอแนะ.....50

บรรณานุกรม.....51

ภาคผนวก.....52

ภาคผนวก ก 53

ภาคผนวก ข 65

ประวัติผู้เขียน.....67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	Infrared Emitting Diode	4
2.2	วงจรภายในของหน่วยรับอินฟาเรดสำเร็จรูป	4
2.3	ส่วนประกอบของไอซีเบอร์ 555	5
2.4	ไอซีเบอร์ 555	5
2.5	วงจร Monostable Multivibator	6
2.6	วงจร Astable Multivibator	8
2.7	การส่งข้อมูลแบบอนุกรม จำนวน 8 บิต จะส่งทีละบิตจนครบ 1 ไบต์	11
2.8	การส่งข้อมูลแบบขนาน จำนวน 8 บิต จะส่งทีละ 8 บิต หรือ 1 ไบต์	11
2.9	ผังเวลาของการส่งข้อมูลแบบอนุกรมจำนวน 8 บิต พร้อมกับบิตเริ่มต้น บิตพาริตี้ และบิตหยุด	13
2.10	บล็อกไดอะแกรมของพอร์ตอนุกรม	14
2.11	ผังเวลาของการส่งและรับข้อมูลในโหมด 0	18
2.12	การใช้งานพอร์ตอนุกรมในโหมด 0 เพื่อขยายเอาต์พุตของ 8051	18
2.13	รูปแบบการรับและส่งข้อมูลในโหมด 1	19
2.14	รูปแบบการรับส่งข้อมูลในโหมด 2	20
2.15	แหล่งกำเนิดอัตราบอดของพอร์ตอนุกรมสำหรับ MCS-51	22
2.16	แนวทางการเชื่อมต่อระดับสัญญาณแบบ TTL จาก 8051 กับระดับสัญญาณของ RS-232C	24
2.17	ไอซีเบอร์ MAX232 ซึ่งเป็นวงจรเชื่อมต่อแบบ RS-232C โดยการใช้ไฟเลี้ยง +5V เพียงชุดเดียว	25
2.18	การสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์หลายตัว	26
2.19	โครงสร้างภายในของไมโครคอนโทรลเลอร์ AT89S52	27
2.20	การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ AT89S52	28
2.21	แสดงแผนภาพการจัดสรรหน่วยความจำ หน่วยความจำ ภายใน	31
2.22	แสดงรีจิสเตอร์หน้าที่พิเศษใน ไมโครคอนโทรลเลอร์ AT89S52	33
2.23	รีจิสเตอร์ควบคุมพอร์ตอนุกรม	34
2.24	วงจรภายในของ MAX232	36
2.25	การจัดขาของ MAX232	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.26 วงจรเชื่อมต่อ MAX232 กับพอร์ตอนุกรม....36

3.1 หน้าจอหลักของโปรแกรม....38

3.2 Entity Relationship Model....39

3.3 หน้าแรกของเว็ปแอปพิเคชั่น....40

3.4 แผนภาพการขั้นตอนการทำงานของระบบ....41

3.5 แผนภาพแสดงสถาปัตยกรรมของระบบ....42

4.1 7 เซคเม้น ก่อนเปิดพอร์ต....43

4.2 แอปพลิเคชั่นก่อนเปิดพอร์ต....44

4.3 ฐานข้อมูลตารางสถานะ ก่อนเปิดพอร์ต....44

4.4 ฐานข้อมูลตารางชั้น ก่อนเปิดพอร์ต....45

4.5 7 เซคเม้นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด....45

4.6 แอปพลิเคชั่นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด....45

4.7 ฐานข้อมูลตารางสถานะขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด....46

4.8 ฐานข้อมูลตารางชั้นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด....46

4.9 7 เซคเม้นขณะเปิดพอร์ต โดยมีรถมาจอด....46

4.10 แอปพลิเคชั่นขณะเปิดพอร์ต โดยมีรถมาจอด....47

4.11 ฐานข้อมูลตารางสถานะขณะเปิดพอร์ต โดยมีรถมาจอด....47

4.12 ฐานข้อมูลตารางชั้นขณะเปิดพอร์ต โดยมีรถมาจอด....48

4.13 7 เซคเม้นเมื่อปิดพอร์ต....48

4.14 แอปพลิเคชั่นเมื่อปิดพอร์ต....48

4.15 ฐานข้อมูลตารางสถานะเมื่อปิดพอร์ต....49

4.16 ฐานข้อมูลตารางชั้นเมื่อปิดพอร์ต....49

A.1 หน้าแรกของเว็ปแอปพลิเคชั่น....65

A.2 เลือกรูปแบบการตรวจสอบ....65

A.3 รูปแบบการตรวจสอบแบบเว็ป....66

A.4 รูปแบบการตรวจสอบแบบมือถือ....66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 การกำหนดโหมดของพอร์ตอนุกรม 15
2.2 การเลือกใช้อัตราบอดและค่ากำหนดเริ่มต้น 23
2.3 แสดงการทำงานรีจิสเตอร์ควบคุมพอร์ตอนุกรม.....34
3.1 ข้อมูลของฐานข้อมูล.....39
3.2 พจนานุกรมข้อมูลของตาราง Status.....39
3.3 พจนานุกรมข้อมูลของตาราง Floor Detail.....39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันข้อมูลข่าวสารไม่ว่าจะใหญ่โตหรือเล็กน้อยเพียงใดก็สามารถเลือกสรรนำมาใช้ในชีวิตประจำวันได้อย่างไม่มีวันหมดสิ้น ซึ่งในแต่ละวันคนเราเดินทางไปมาหลากหลายสถานที่ ทั้งข้อมูลสถานที่ หรือเส้นทางการเดินทางก็ย่อมเป็นข้อมูลที่ต้องการ แต่หากพึ่งพาเพียงแต่หนังสือก็คงจะไม่ทันการ เพราะทุกอย่างเป็นการแข่งกับเวลาทั้งสิ้น สิ่งที่จะต้องเข้ามาเติมเต็มความต้องการจึงหนีไม่พ้นเทคโนโลยี

เทคโนโลยีถือเป็นทางออกที่ดีในการแสวงหาข้อมูลให้ได้มาอย่างรวดเร็ว ไม่ว่าจะเป็นโทรทัศน์หรือคอมพิวเตอร์ แต่ท้ายที่สุดแล้วก็ยังไม่สามารถรองรับความต้องการบางอย่างได้อยู่ดี ความต้องการข้อมูลที่เร่งด่วนจนถึงขั้นสามารถที่จะวางแผนอนาคตได้ถือเป็นสิ่งที่จำเป็น แม้แต่ข้อมูลเล็กๆน้อยๆก็สามารถจะนำพาไปยังสิ่งที่ยิ่งใหญ่กว่าได้

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

โครงงานนี้มุ่งหวังเพื่อศึกษาเทคโนโลยีที่มีอยู่ในปัจจุบันนำมาประยุกต์ใช้งานเพื่อให้ได้ข้อมูลสภาพภายในของอาคารที่จอดรถได้ โดยการส่งผ่านข้อมูลโดยอาศัยอุปกรณ์ไมโครคอนโทรลเลอร์และ 7-Segment พร้อมทั้งการใช้งานร่วมกับอุปกรณ์รับ-ส่งสัญญาณเซ็นเซอร์ จึงเป็นการใช้งานที่ทำให้การตรวจสอบข้อมูลบางสิ่งภายในสถานที่หนึ่งๆเป็นไปได้อย่างรวดเร็วและครอบคลุม

1.3 สมมติฐานของการศึกษา

การรับ-ส่งสัญญาณของตัวเซ็นเซอร์ถือเป็นปัจจัยสำคัญของการศึกษา ซึ่งตัวรับแบบตัดแสงจะถูกติดตั้งเอาไว้บริเวณที่จอดรถแต่ล่ะจุด และจะทำงานต่อเนื่องซึ่งตัวรับแบบตัดแสงจะเป็นตัวคอยส่งข้อมูลสัญญาณคอยตรวจเช็คสถานะของจุดที่กำหนด โดยที่สัญญาณนั้นจะถูกส่งไปประมวลผลโดยอุปกรณ์ไมโครคอนโทรลเลอร์ ผลที่ได้จะถูกแสดงผ่านออกทางหน้าจอคอมพิวเตอร์เพื่อแสดงให้เห็นถึงสถานะของที่จอดรถในแต่ล่ะจุด(ว่าง-ไม่ว่าง) ซึ่งสถานะของที่จอดรถนั้นจะแปรผันตามความต่อเนื่องของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

1.หลักการทำงานของMCS51

2.หลักการทำงานของ 7 segments

3.หลักการส่งข้อมูลแบบ Serial

4.หลักการเกี่ยวกับฐานข้อมูล

1.5 การเปรียบเทียบระหว่างวิธีการที่นำเสนอกับวิธีการแบบพื้นฐาน

วิธีการประยุกต์นำอุปกรณ์รับ-ส่งสัญญาณมาใช้ร่วมกับอุปกรณ์ไมโครคอนโทรลเลอร์เมื่อเปรียบเทียบกับวิธีการพื้นฐานที่ใช้เพียงแรงของมนุษย์เท่านั้น ย่อมเห็นความแตกต่างได้อย่างชัดเจน เนื่องด้วยการใช้แรงมนุษย์นั้นมีข้อจำกัดอยู่มาก ทั้งเรื่องของความรวดเร็วและความถูกต้อง วิธีการที่นำเสนอนั้นจะช่วยให้เราสามารถจำลองสภาพของสถานที่จอดรถมาไว้ในคอมพิวเตอร์ ทำให้สามารถตรวจสอบสถานะของแต่ล่ะจุดได้อย่างสะดวก รวดเร็ว และมีความถูกต้อง ทั้งนี้ยังสามารถทำงานได้ตลอดเวลาตามที่ต้องการอีกด้วย

1.6 ขอบเขตการวิจัย

การวิจัยต่างๆย่อมต้องเป็นไปเพื่อให้ได้ผลตามสมมติฐานที่ตั้งเอาไว้ ซึ่งจะต้องวิจัยในเรื่องของข้อมูลสัญญาณของอุปกรณ์รับ-ส่งสัญญาณเซ็นเซอร์ว่าจะใช้ไมโครคอนโทรลเลอร์รับสัญญาณมาประมวลผลเพื่อส่งเข้าสู่คอมพิวเตอร์ได้อย่างไร และมีการศึกษาในเรื่องของฐานข้อมูล เพื่อที่จะเก็บข้อมูลไว้และใช้แอพพลิเคชั่นดึงค่าออกมาแสดงให้ได้ทางหน้าจอคอมพิวเตอร์ หลังจากนั้นจะทำการดึงข้อมูลในฐานข้อมูลที่มีไปแสดงผลบนเว็บแอปพลิเคชั่นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.7 ขั้นตอนของการศึกษา

วิทยานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกันคือ

บทที่ 1 กล่าวถึงความเป็นมาของงานวิจัย ความมุ่งหมายและวัตถุประสงค์ สมมติฐาน ทฤษฎีที่ใช้ ขอบเขตของการวิจัย และขั้นตอนการศึกษา

บทที่ 2 กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และรูปแบบส่วนประกอบต่างๆของแผงวงจร ไมโครคอนโทรลเลอร์ รวมถึงไอซีด้วย

บทที่ 3 กล่าวถึงการวิเคราะห์ระบบและการออกแบบระบบ ซึ่งแบ่งเป็น 3 ส่วน คือ ตัวรับสัญญาณแบบตัดแสง (Sensor Infrared) ไมโครคอนโทรลเลอร์ (Microcontroller) และ ฐานข้อมูล

บทที่ 4 กล่าวถึงวิธีการทดลองและผลการทดลอง โดยแบ่งผลการทดลองเป็น 4 ส่วนคือ แอปพลิเคชั่น (Application) เว็บแอปพลิเคชั่น (Web Application) 7 เซ็คเม้น (7 segments) และ ฐานข้อมูล

บทที่ 5 เป็นบทสรุปผลการวิจัยและข้อเสนอแนะ

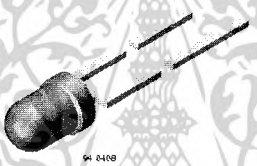
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

2.1 ทฤษฎี ตัวรับ และ ตัวส่ง sensor infrared (แบบตัดแสง)

2.1.1 ตัวส่ง (Led Emitting)

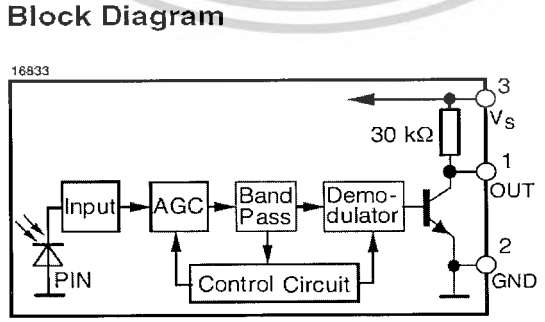
LED เป็นอุปกรณ์สารกึ่งตัวนำที่เปล่งคลื่นแม่เหล็กไฟฟ้าออกมา ในช่วงความยาวคลื่นแคบๆ ออกมาเพื่อให้กระแสไฟฟ้าเข้าไปในทิศทางที่ถูกต้อง ช่วงที่คลื่นปล่อยออกมานั้นขึ้นอยู่กับชนิดของสารกึ่งตัวนำที่นำมาใช้งาน ซึ่ง LED สามารถเปล่งคลื่นออกมาได้หลายแบบทั้งแบบมองเห็นได้ และ ตัวรับแบบตัดแสง (ตาเปล่าไม่สามารถมองเห็นได้)



รูปที่ 2.1 Infrared Emitting Diode

2.1.2 ตัวรับ (Infrared Receiving Module)

หน่วยรับ มีด้วยกันหลายแบบ ทั้งแบบไดโอดและเป็นแบบสำเร็จรูป โดยถ้าเป็นแบบไดโอดต้องทำการเพิ่มส่วนของวงจรมอดดูเลตเพิ่มเข้ามาเพื่อมอดดูเลตสัญญาณ ที่ส่งมาจากภาคส่ง ถ้าเป็นแบบสำเร็จรูปจะมีส่วนของมอดดูเลตมาให้ภายใน

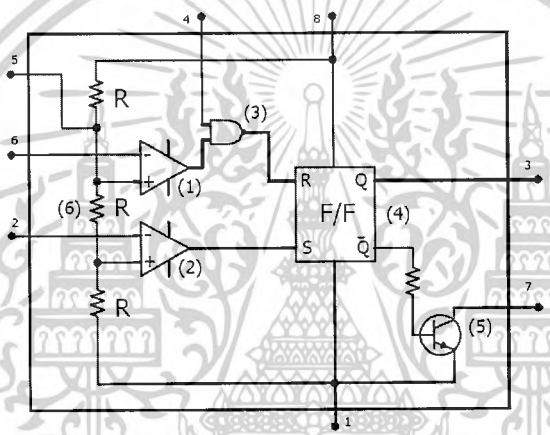


รูปที่ 2.2 วงจรภายในของหน่วยรับอินฟราเรดสำเร็จรูป

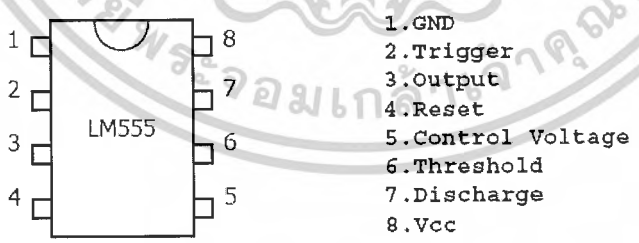
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 IC เบอร์ 555

IC เบอร์ 555 เป็นไอซี ที่นิยมใช้กันมากในการนำไปสร้างสัญญาณรูปคลื่นแบบต่างๆ เช่น สัญญาณ SquareWave , สัญญาณพัลส์ สัญญาณ ramp และวงจรตั้งเวลา ไอซีเบอร์ 555 เป็นอุปกรณ์วงจรรวมที่มีอุปกรณ์อิเล็คทรอนิคส์อื่นๆ อยู่ภายใน และมีส่วนที่ต้องต่อภายนอกเพื่อควบคุมการทำงาน และใช้งานเป็นลักษณะต่างๆ ซึ่งง่ายต่อการออกแบบ และง่ายในการสร้างสัญญาณพัลซ์ความถี่ต่างๆ อีกทั้งสามารถเข้าใจการทำงานได้ง่ายนอกจากไอซีเบอร์ 555 แล้วยังมีไอซีเบอร์ 556 ที่เป็นแบบ Dual Timer ประกอบด้วย ไอซีเบอร์ 555 จำนวน 2 ตัว อยู่ภายในตัวเดียวกัน เพื่อใช้เป็นวงจรตั้งเวลา และสะดวกในการออกแบบวงจรที่ต้องใช้ไอซีเบอร์ 555 หลายๆตัว



รูปที่ 2.3 ส่วนประกอบของไอซีเบอร์ 555



รูปที่ 2.4 ไอซีเบอร์ 555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานเบื้องต้น ของไอซีเบอร์ 555

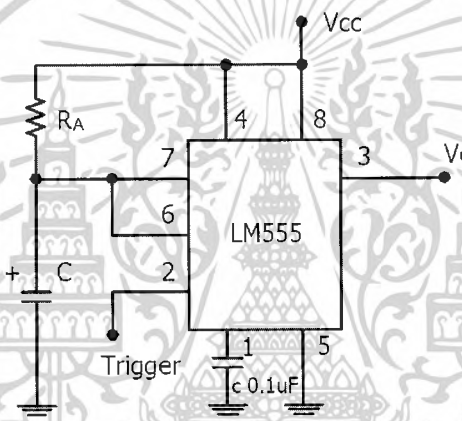
โดยอาศัยการทำงานเป็น 2 แบบ คือ

- เป็นตัวผลิตสัญญาณค่าความถี่ต่างๆ และ
- เป็นวงจรนับสัญญาณ

จึงแบ่งการทำงานเป็นวงจรที่สำคัญ ได้ 2 ชนิดคือ

2.2.1 วงจรโมโนสเตเบิล Monostable Multivibator

คือวงจรที่สร้างสัญญาณพัลซ์ขึ้นมา 1 ลูก หลังจากมีการทริก(กระตุ้น)ให้วงจรเกิดสัญญาณ มีความกว้างที่สามารถกำหนดได้ด้วยค่า RC



รูปที่ 2.5 วงจร Monostable Multivibator

หลักการทำงาน

ขณะที่ไม่มีสัญญาณ Trig ที่ขา 2 Vtrig = VCC แรงดันคร่อมคาปาซิสเตอร์จะเป็นศูนย์ เพราะว่า VCC จะมีกระแสไหลผ่าน RA ผ่าน Tr ลงกราวด์ ได้ Vo = 0เมื่อมีสัญญาณ trig ที่ขา 2 คือ Vtrig = 0 Tr ภายในจะ “Off” VCC จะทำการเก็บประจุ (Charge C) จนกระทั่งเกิดแรงดันตกคร่อมคาปาซิสเตอร์ ประมาณ 2VCC/3 คาปาซิสเตอร์จะคายประจุผ่าน Tr ขณะที่คาปาซิสเตอร์คายประจุ output จะเป็น logic “1” ได้ Vo = VCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งช่วงการทำงานเป็น 4 ช่วง

ช่วงที่ 1 ขณะที่ Vtrig = VCC Comparator (2) ตัวล่างจะทำให้ output เป็น “0” (ขาลบมีศักดามากกว่าขาบวก)Comparator (1) ตัวบนจะทำให้ output เป็น “1” เพราะว่าขา 6 ต่อกับขา 7 ผ่าน Tr ลง GND ทำให้ Tr “ON” ได้output ของ NAND gate เป็น ”0” เพราะ input ทั้งคู่เป็น “1” ดังนั้น R = 0 , S = 0 จะทำให้ Q และQ ไม่เปลี่ยนแปลงคือ Q = ”0” , Q = ”1” ถ้า Q = “1” Tr จะ “ON” คาปาซิสเตอร์จะไม่มีการเก็บประจุ เนื่องจากกระแสจาก VCC จะไหลผ่าน R และ Tr ลง GND

ช่วงที่ 2 ขณะที่ trig ด้วย Vtrig = 0 V output ของ Comparato(2) ตัวล่างจะเปลี่ยนจาก “0” เป็น “1” ส่วนOutput ตัว Comparator(1) ตัวบน ยังคงเหมือนเดิม ดังนั้นที่วงจร F/F ขา R = “0” , S = “1” จะเป็นการ Set F/F ได้ Q = “1” และQ =”0” เมื่อ Q =”0” ทำให้ Tr เกิดการ “Off” ส่งผลให้คาปาซิสเตอร์จะทำการเก็บประจุ

ช่วงที่ 3 เมื่อสัญญาณ trig กลับเป็น VCC อีกครั้ง Output ของ comparator (2) ตัวล่างจะเปลี่ยนจาก “1” เป็น “0”ทำให้ output ของ comparator ที่ส่งไปให้วงจร F/F ไม่เปลี่ยนแปลง

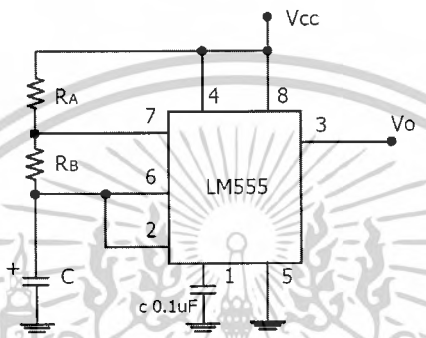
ช่วงที่ 4 เมื่อคาปาซิสเตอร์ทำการเก็บประจุจนกระทั่ง VC มีค่าเท่ากับ 2/3VCC หรือมากกว่านั้น ส่งผลให้ Output ของ comparator(1) ตัวบน เปลี่ยนจาก “1” เป็น “0” ทำให้ Output ของ NAND gate = “1” ที่วงจร F/F ขา R = “1” และ S = “0” ดังนั้น Q = “0” ,Q = “1” ทำให้ Tr “ON” อีกครั้ง คาปาซิสเตอร์ จะทำการคายประจุออกผ่าน Tr ลง GND กลับสู่สภาพเดิม วงจร Monostable Multivibator นี้สามารถสร้างสัญญาณพัลซ์ขึ้นมาจำนวน1 ลูก หลังจากมีการ trig ที่ขา 2 โดยความกว้างพัลซ์ขึ้นอยู่กับค่า RA และ C สมการในการ charge ประจุของ Capacitor จาก 0 ถึง 2/3VCC คือ

$$
\begin{aligned}
vC(t) &= VCC(1-e^{-t/RC}) \\
2/3VCC &= VCC(1-e^{-t/RC}) \\
e^{-t/RC} &= +1/3 \\
-t/RAC &= \ln 1/3 \\
&= -1.1 \\
t &= 1.1\ RAC \\
T &= 1.1\ RAC
\end{aligned}
\qquad (2.1)
$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2. วงจรอะสเตเบิล Astable Multivibrator

คือวงจรสร้างสัญญาณ Square wave มีความถี่ที่สามารถกำหนดได้ตามความต้องการ ด้วยค่าคาปาซิสเตอร์ Cและค่าความต้านทาน RA และ RBโดยในวงจรนี้นอกจากจะสามารถกำหนดความถี่ของสัญญาณ Square wave ด้วยค่า C และค่า RT = RA + RBแล้วยังสามารถที่จะกำหนดช่วงเวลา “ON” และ “OFF” ของสัญญาณด้วยค่าที่แตกต่างกันของ RA และ RB ได้อีกด้วย



รูปที่ 2.6 วงจร Astable Multivibator

หลักการทำงาน

ใช้หลักการเก็บประจุ และคายประจุของคาปาซิสเตอร์โดยจะทำการเก็บประจุผ่าน RA, RB และจะคายประจุผ่านRB และ Tr ภายในลง GND

- ขณะที่คาปาซิสเตอร์เก็บประจุ จนถึงระดับแรงดัน 2/3VCC ขณะนั้นแรงดัน ของ V0 = VCC
- ขณะที่คาปาซิสเตอร์คายประจุออก เกิดแรงดันตกคร่อมมีช่วงจาก 2/3VCC ถึง VCC/3 ขณะนั้นแรงดันของ V0 = 0 (ศูนย์โวลท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งขั้นตอนการทำงานเป็น 5 ช่วง

ช่วงที่ 1 เป็นช่วงที่ทำการเปิดวงจร คาปาซิสเตอร์จะทำการเก็บประจุจาก VCC ผ่าน RA ,RB Output ของ comparatorตัวบนและล่างจะเป็น “1” ทำให้ Output ของ Nand gate จะเป็น “0” ส่งผลให้ F/F ขา R = “0” และ S = “1” เป็นการ set F/F Output Q = “1” , Q = “0” คาปาซิสเตอร์จะทำการเก็บประจุไปเรื่อยจนมีแรงดันตกคร่อม เท่ากับ VCC/3

ช่วงที่ 2 ช่วงที่คาปาซิสเตอร์ทำการเก็บประจุเกิน VCC/3 จะทำให้ comparator(2) ตัวล่างเป็น “0” F/F ขา R = “0” , S= “0” ได้ Output V0 เหมือนเดิม

ช่วงที่ 3 เมื่อคาปาซิสเตอร์ถูกเก็บประจุจนกระทั่งมีค่ามากกว่า 2VCC/3 output ของ comparator(1)ตัวบน จะเปลี่ยนเป็น “0” Output ของ NAND gate จะเป็น logic “1” ดังนั้น F/F R = “1” , S = “0” เป็นการทำให้ F/F Q = “0” และ Q = “1” Transistor จะ “ON”

ช่วงที่ 4 คาปาซิสเตอร์จะทำการคายประจุผ่าน RB และ Tr ลง GND ทำให้Output ของ comparator(1)ตัวบน เป็น“1” comparator(2) ตัวล่างเป็น “0” ทำให้ F/F ขา R = “0”, S = ”0” Output ของ F/F จะคงเดิม Q = “0” ,Q = “1”

ช่วงที่ 5 เมื่อคาปาซิสเตอร์ทำการคายประจุ จนแรงดันคร่อม C = VCC/3 ทำให้ comparator(1)ตัวบน เป็น “1”comparator(2)ตัวล่าง เป็น “1” ทำให้ F/F ขา R = “0” , S = “1” เป็นการทำให้ F/F ได้ Q = “1” , Q = “0” Tr จะ “Off” คาปาซิสเตอร์จะทำการเก็บประจุใหม่อีกครั้ง ในขั้นตอนที่ 2 และเป็นเช่นนี้ไปเรื่อยๆ การวิเคราะห์เวลาในการ charge และ discharge Capacitor

เวลาในการ charge- C จาก 0 ถึง VCC/3 ใช้เวลา t1

$$vC(t) = VCC(1-e^{-t/RC}) \text{ ----------1}$$

$$VCC/3 = VCC(1-e^{-t/RC})$$

$$-t1/RC = \ln 2/3 = -0.405$$

$$t1 = 0.405\ RC \text{ ----------2}$$

เวลาในการ charge C จาก 0 ถึง 2/3VCC ใช้เวลา t2 (2.2)

$$2/3VCC = VCC(1-e^{-t2/RC})$$

$$-t2/RC = \ln 1/3 = -1.098$$

$$t2 = 1.098\ RC \text{ ----------3}$$

$$T1 = t2 - t1 = (1.098 - 0.405)RC$$

$$T1 = 0.693\ RC$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ R = RA + RB

T1 = 0.69(RA + RB)C

เวลาในการ discharge C จาก 2/3VCC ถึง VCC/3

สูตร vC(t) = Ve-t/RC

VCC/3 = 2/3VCC e-t/RC (2.2) ต่อ

T = -RC ln 1/3 = 0.69 RC

T2 = 0.69(RB)C

T = T1 + T2

T = 0.69(RA + 2RB)C

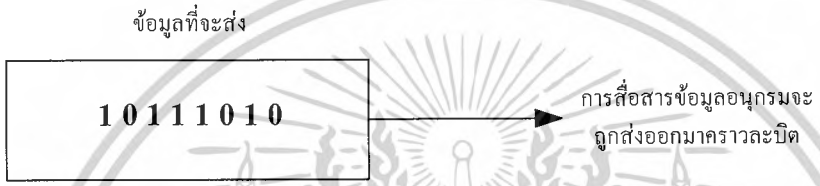
ที่ความถี่ของ square wave

F = 1 / 0.69(RA + 2RB)C

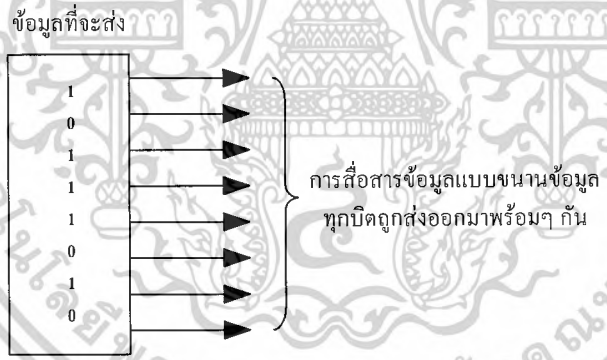
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอนุกรม เป็นการรับและส่งข้อมูลคราวละ 1 บิต เป็นลำดับจนสิ้นสุดกลุ่มข้อมูล การสื่อสารแบบนี้แตกต่างจากการสื่อสารแบบขนาน เนื่องจากการส่งข้อมูลแบบขนานจะโอนย้ายข้อมูลพร้อมกัน จึงต้องใช้จำนวนเส้นของสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรม ต้องการเส้นสัญญาณเพียง 2 หรือ 3 เส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะในการสื่อสารกับอุปกรณ์ภายนอกที่เป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก ดังรูปที่ 1



รูปที่ 2.7 การส่งข้อมูลแบบอนุกรม จำนวน 8 บิต จะส่งทีละบิตจนครบ 1 ไบต์



รูปที่ 2.8 การส่งข้อมูลแบบขนาน จำนวน 8 บิต จะส่งทีละ 8 บิต หรือ 1 ไบต์

2.3.1 จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารแบบอนุกรม เป็นการรับส่งข้อมูลเป็นกลุ่มของบิตข้อมูล (Bit Steam) ทีละบิต ดังนั้นจึงต้องพิจารณาถึงความเร็วในการรับส่งข้อมูลเป็นอันดับแรก โดยทั่วไปความเร็วจะระบุกันในหน่วยที่เรียกว่า ***อัตราบอด (Baud Rate)*** คือ เป็นหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที ตามค่ามาตรฐานเหล่านี้ เช่น 110, 150, 300, 1200, 2400, 4800 และ 9600 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ถ้าต้องการส่งข้อมูล 8 บิต ด้วยอัตราบอด 2400 การส่งข้อมูล 1 บิต จะใช้เวลาเท่ากับ 1 / 2400 เท่ากับ 416 ไมโครวินาที ดังนั้นเวลาที่ใช้ในการส่งข้อมูล 8 บิต มีค่าเท่ากับ 8 X 416 เท่ากับ 3,328 ไมโครวินาที

2.3.2 รูปแบบของการส่งข้อมูลอนุกรม

วิธีการที่จะทำให้ข้อมูลในการสื่อสารอนุกรมถูกต้องมากยิ่งขึ้นนั้น จะใช้วิธีการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง คือ

2.3.2.1 บิตเริ่มต้น (Start Bit)

มีหน้าที่สำหรับบอกให้ฮาร์ดแวร์ทราบว่า ถึงตำแหน่งเริ่มต้นของบิตข้อมูลกลุ่มใหม่แล้ว เพื่อที่จะปรับจังหวะของสัญญาณรับข้อมูลให้ตรงกัน ดังนั้นบิตเริ่มต้นนี้ จึงถูกเพิ่มเข้าไปก่อนมีการส่งข้อมูลจริง โดยทั่วไปแล้วค่าของบิตเริ่มต้นมักจะมีระดับลอจิกตรงข้ามกับระดับลอจิกของสายส่งข้อมูลเมื่อไม่มีการส่งข้อมูล (Idle State) ตัวอย่างเช่น หากสภาวะของสายเมื่อไม่มีการส่งข้อมูลเป็นลอจิกสูง บิตเริ่มต้นก็จะเป็นลอจิกต่ำ เป็นต้น

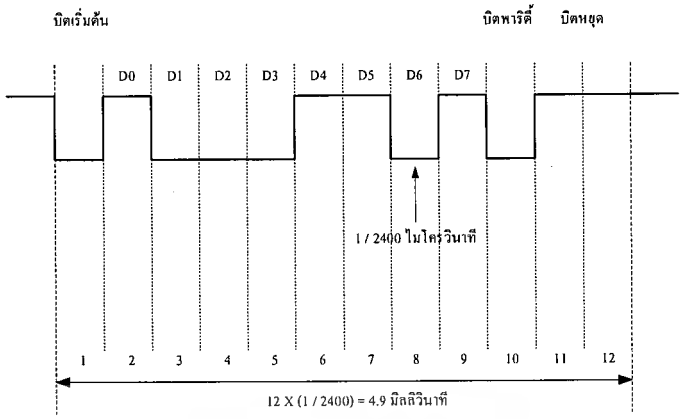
2.3.2.2 บิตพาริตี้ (Parity Bit)

มีหน้าที่เพื่อตรวจสอบความผิดพลาดของข้อมูล โดยจะนำไปต่อท้ายบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนของบิตข้อมูลที่เป็น 1 ซึ่งพาริตี้นี้จะเป็นได้ 2 ลักษณะ คือ ***พาริตี้คู่ (Even Parity) และ พาริตี้คี่ (Odd Parity)*** ตัวอย่าง เช่น ระบบที่เป็นพาริตี้คู่ จะนำบิตข้อมูลที่เป็น 1 มาพิจารณาว่าเป็นคู่หรือไม่ ถ้าเป็นคู่บิตพาริตี้จะเป็น 0 และ ถ้าเป็นคี่พาริตี้จะเป็น 1 ส่วนทางด้านรับ จะตรวจสอบจำนวนบิตข้อมูลที่มีค่าเป็น 1 ทั้งหมด รวมทั้งพาริตี้ด้วย ถ้ามีจำนวนเป็นคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง ถ้าหากเป็นคี่แสดงว่าข้อมูลที่รับเข้ามาผิดพลาด

2.3.2.3 บิตหยุด (Stop Bit)

บิตหยุด เป็นบิตที่เพิ่มเข้าไปเพื่อบอกขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตหยุดนี้จะมีมากกว่า 1 บิตก็ได้ คือ 1 บิต, 1 บิต ครึ่ง และ 2 บิต ดังนั้นในกรณีของการส่งข้อมูล 8 บิต ที่เพิ่มเติมเข้าไปโดยสมบูรณ์ คือ บิตเริ่มต้น บิตพาริตี้ และ บิตหยุด รวมทั้งสิ้นจะเป็น 12 บิต ดังผังเวลาในรูปที่ 2.9 ตัวอย่างเช่น หากข้อมูลส่งออกไปด้วยอัตราบอด 2400 เวลารวมในการส่งข้อมูล 1 ไบต์ จะมีค่าเป็น 12 X 416 ไมโครวินาที เท่ากับ 4.99 มิลลิวินาที เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



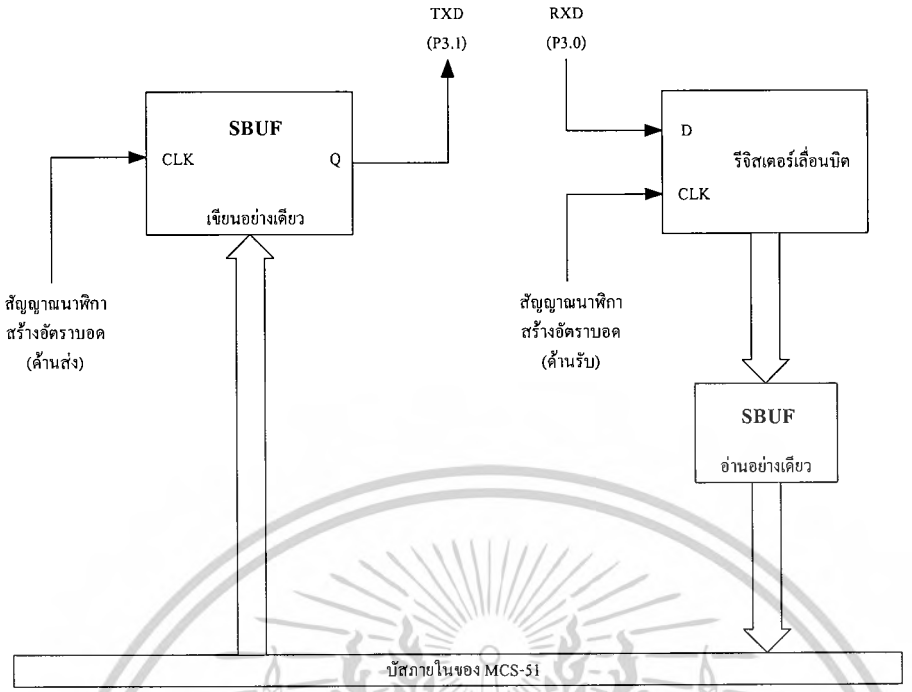
รูปที่ 2.9 ผังเวลาของการส่งข้อมูลแบบอนุกรมจำนวน 8 บิต พร้อมกับบิตเริ่มต้น,บิตพาริตี้ และบิตหยุด

2.3.3 การรับ-ส่งข้อมูลแบบอนุกรมของ MCS-51

MCS-51 มีพอร์ตอนุกรม (Serial Port) ที่สามารถสั่งให้ทำงานได้หลายโหมด (Mode) อยู่ภายในชิป (Chip) ซึ่งการทำงานจะเป็น ***แบบฟูลดูเพลกซ์ (Full Duplex หมายถึงสามารถรับและส่งข้อมูลได้ในช่วงเวลาเดียวกัน)*** ในการทำงานจะมีรีจิสเตอร์ (Register) ที่ทำหน้าที่ในการรับข้อมูล ซึ่งจะนำตัวอักษรที่รับได้มาเก็บไว้ในบัฟเฟอร์ (Buffer) ในขณะที่กำลังรับตัวอักษรตัวที่สอง ด้วยเหตุนี้ถ้าให้ ซีพียู (**C**entral **P**rocessing **U**nit) อ่านตัวอักษรตัวแรกออกไป ก่อนที่จะรับตัวอักษรตัวที่สองเสร็จ ข้อมูลก็จะไม่สูญหาย ส่วนบัฟเฟอร์อีกตัวหนึ่งจะใช้สำหรับเก็บข้อมูลก่อนที่จะส่งออกไป ***บัฟเฟอร์ทั้งทางด้านรับและด้านส่งนี้จะใช้ชื่อเดียวกัน คือ รีจิสเตอร์ SBUF (SBUF:Serial Port Buffer)*** ซึ่งจะอยู่ที่แอดเดรส (Address) 099H ในหน่วยความจำภายใน

รูปที่ 2.10 แสดงโครงสร้างทางกายภาพของรีจิสเตอร์ SBUF ซึ่งจะเห็นว่ารีจิสเตอร์ทั้ง 2 ตัวจะอยู่ที่แอดเดรสเดียวกัน ซึ่งรีจิสเตอร์ตัวหนึ่งจะสามารถเขียนได้อย่างเดียว ทำหน้าที่เก็บข้อมูลที่จะส่งออกไปภายนอก ส่วนรีจิสเตอร์อีกตัวหนึ่งจะสามารถอ่านได้อย่างเดียว ทำหน้าที่เก็บข้อมูลที่รับได้ทางพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 บล็อกไดอะแกรมของพอร์ตอนุกรม

2.4 รีจิสเตอร์ควบคุมพอร์ตอนุกรม

รีจิสเตอร์ที่ใช้ในการควบคุมพอร์ตอนุกรมมี 2 ตัวคือ รีจิสเตอร์ SCON และ PCON รีจิสเตอร์ SCON จะใช้สำหรับควบคุมรูปแบบของการรับส่งข้อมูล (Data formatted) ส่วนรีจิสเตอร์ PCON จะใช้สำหรับควบคุมอัตราความเร็วของการรับส่งข้อมูล (Data rate) ส่วนขา RXD (P3.0) และขา TXD(P3.1) จะเป็นขาที่ใช้ต่อกับวงจรภายนอกสำหรับรับหรือส่งข้อมูล ส่วนรายละเอียดของรีจิสเตอร์ SCON และ PCON มีรายละเอียดดังนี้ คือ

รีจิสเตอร์ SCON (Serial Port Control Register) อยู่ในหน่วยความจำที่แอดเดรส 098H

	7	6	5	4	3	2	1	0	Bit
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SCON มีรายละเอียดดังนี้ คือ

บิต	สัญลักษณ์	รายละเอียด
7	SM0	กำหนดโหมดการทำงานของพอร์ตอนุกรม
6	SM1	กำหนดโหมดการทำงานของพอร์ตอนุกรม

ตารางที่ 2.1 การกำหนดโหมดของพอร์ตอนุกรม

โหมด	SM0	SM1	รายละเอียด
0	0	0	รีจิสเตอร์เลื่อนบิต (Shift Register); มีอัตรารับส่งข้อมูล = $f_{OSE}/12$
1	0	1	UART ชนิด 8 บิต; มีอัตรารับส่งข้อมูลกำหนดเองได้
2	1	0	UART ชนิด 9 บิต; มีอัตรารับส่งข้อมูล = $f_{OSC}/32$ หรือ $f_{OSC}/64$
3	1	1	UART ชนิด 9 บิต; มีอัตรารับส่งข้อมูลกำหนดเองได้

บิต	สัญลักษณ์	รายละเอียด
5	SM2	บิตเลือกการใช้งานพอร์ตสื่อสารข้อมูลแบบอนุกรมในโหมด 2 และ 3 เพื่อใช้ติดต่อระหว่างซีพียูด้วยกันเอง ***โหมด 2 และ 3*** บิต RI จะเป็น 1 เมื่อบิตที่ 9 ที่รับเข้ามามีค่าเป็น 1 แต่ถ้าบิตที่ 9 ที่รับเข้ามาเป็น 0 บิต RI จะถูกเคลียร์ ***โหมด 1*** บิต RI จะเป็น 1 เมื่อ Stop bit ที่เข้ามายังพอร์ตอนุกรมถูกต้อง แต่ถ้า Stop bit ไม่เข้ามายังพอร์ตอนุกรม ซึ่งเกิดจากปัญหาในการส่งข้อมูล จะทำให้บิต RI มีค่าเป็น 0 ***โหมด 0*** บิตนี้จะถูกเคลียร์เป็น 0 เสมอ
4	REN	บิตควบคุมการอนุญาตให้มีการรับข้อมูลเข้ามาทางขา RxDถ้าบิตนี้เป็น 1 ก็จะรับข้อมูลจากภายนอกได้ แต่ถ้าบิตนี้เป็น 0 ก็จะไม่รับข้อมูลจากภายนอก ทำให้บิตนี้เป็น 1 หรือ 0 ทำได้โดยใช้คำสั่ง SETB bitหรือ CLR bit (บิตนี้อยู่ที่แอดเดรส 9CH)
3	TB8	การส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิตนี้เก็บข้อมูลบิตที่ 9 ส่วนโหมดอื่นจะไม่ใช้งานในบิตนี้การกำหนดค่าให้กับบิตนี้ทำได้โดยใช้คำสั่งSETB bit หรือ CLR bit (บิตนี้อยู่ที่ตำแหน่ง 9BH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2	RB8	การรับข้อมูลแบบอนุกรมในโหมด 2 และ 3 จะใช้บิตนี้เก็บข้อมูลบิตที่ 9 ส่วนในโหมด 1 นั้น บิตนี้จะเก็บ Stop Bit และจะไม่ใช้งานบิตนี้ในโหมด 0 (บิตนี้อยู่ที่ตำแหน่ง 9AH)
1	TI	แฟลกของการอินเตอร์รัพท์เมื่อส่งข้อมูล จะถูกกำหนดโดยฮาร์ดแวร์ในโหมด 0 จะถูกเซตให้เป็น 1 เมื่อส่งข้อมูลในรีจิสเตอร์ SBUF ออกไปครบทั้ง 8 บิตเรียบร้อยแล้ว ส่วนในโหมดอื่นจะถูกเซตให้เป็น 1 เมื่อเริ่มส่งStop bit การเคลียร์บิตนี้สามารถทำได้ โดยใช้คำสั่ง CLR bit (บิตนี้อยู่ที่แอดเดรส 99H)
0	RI	แฟลกของการอินเตอร์รัพท์เมื่อรับข้อมูล จะถูกกำหนดโดยฮาร์ดแวร์ ในโหมด 0 จะถูกเซตให้เป็น 1 หลังจากรับข้อมูลครบทั้ง 8 บิต เรียบร้อยแล้วส่วนในโหมดอื่นจะถูกเซตให้เป็น 1 เมื่อรับ Stop bit เข้ามาครึ่งหนึ่ง การบิตนี้สามารถทำได้ โดยใช้คำสั่ง CLR bit (บิตนี้อยู่ที่แอดเดรส 98H)

หมายเหตุ *การอ้างตำแหน่งบิตของรีจิสเตอร์นี้สามารถอ้างได้โดยใช้ SCON.0 ถึง SCON.7*

รีจิสเตอร์ PCON (Power Control Register) อยู่ในหน่วยความจำที่แอดเดรส 087H

	7	6	5	4	3	2	1	0	Bit
PCON	SMOD	-	-	-	GF1	GF0	PD	IDL	

รายละเอียดของรีจิสเตอร์ PCON มีดังนี้ คือ

บิต	**สัญลักษณ์**	**รายละเอียด**
7	SMOD	เป็นบิตที่ใช้ในการกำหนดอัตราการรับ-ส่งข้อมูล (Baud Rate) ของพอร์ตอนุกรม ในโหมด 1 และ 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้ตามอัตราการเกิด Overflow ในไทเมอร์ 1 ซึ่งถ้าบิตนี้เป็น 1 จะทำให้อัตราการส่งข้อมูลเพิ่มขึ้น 2 เท่า
6-4	-	ไม่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

3-2	GF0, GF1	แฟลกใช้งานทั่วไป (General Purpose Flag Bit) ซึ่งจะไม่เกี่ยวข้องกับการใดๆ ทั้งสิ้น สามารถกำหนดได้โดยผู้ใช้โปรแกรม
1	PD	ถ้าบิตนี้ถูกเซ็ทให้เป็น 1 จะเข้าสู่การทำงานใน Power Down Mode ทันที
0	IDL	ถ้าบิตนี้ถูกเซ็ทให้เป็น 1 แล้ว 8051 จะเข้าสู่การทำงานใน Idle Mode ทันที

หมายเหตุ *รีจิสเตอร์ PCON ไม่สามารถอ้างอิงแอดเดรสแบบบิตได้*

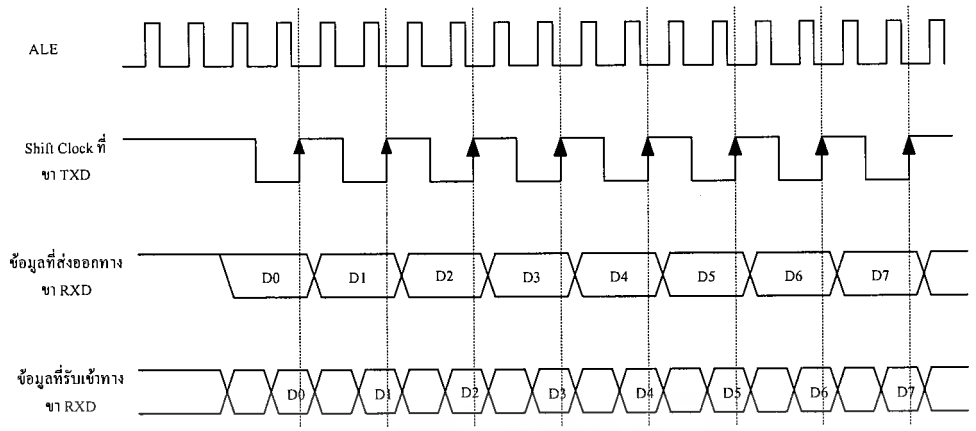
8051 เป็นไมโครคอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยีแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS จะมีข้อดีที่ใช้กำลังไฟฟ้าต่ำกว่าแบบ HMOS นอกจากนี้ 8051 ยังมีข้อดีอีกที่สามารถลดการใช้กำลังไฟฟ้าได้ โดยการทำงานใน Idle Mode และ Power Down Mode ใน Idle Mode นั้นสัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน อินเตอร์รัพท์,พอร์ตอนุกรมและไทเมอร์ สำหรับส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยง แต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร ดังนั้นกำลังไฟฟ้าจึงลดลงมาก ส่วนใน Power Down Mode ออสซิลเลเตอร์จะหยุดทำงาน ทำให้ไม่มีสัญญาณนาฬิกาไปเลี้ยงส่วนใดๆ ของวงจรเลย แต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูญหายไป

2.5 การส่งข้อมูลทางพอร์ตอนุกรม

การส่งข้อมูลทางพอร์ตอนุกรมของ 8051 แบ่งออกเป็น 4 โหมด คือ

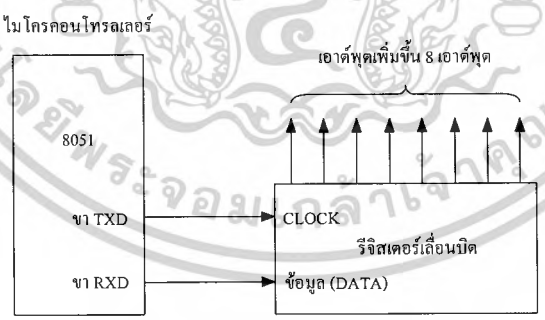
การส่งข้อมูลทางพอร์ตอนุกรมในโหมด 0 ในโหมดนี้จะต้องกำหนดให้บิต SM0และ SM1 ที่อยู่ในรีจิสเตอร์ SCON มีค่าเป็น 0 ทั้งคู่ ซึ่งจะทำให้รีจิสเตอร์ SBUF รับหรือส่งข้อมูลขนาด 8 บิต โดยจะใช้ขา RXD เพียงขาเดียวเท่านั้น ต่อกับสายสัญญาณที่จะใช้ส่งข้อมูลภายนอก ส่วนขา TXD จะถูกต่อกับส่วนกำเนิดสัญญาณนาฬิกา (Clock) ที่อยู่ภายใน 8051 โดยจะใช้ส่งสัญญาณนาฬิกาเพื่อกำหนดความถี่อ้างอิงในการส่งข้อมูลออกมาภายนอกดังแสดงด้วยผังเวลาในรูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
04782



รูปที่ 2.11 ผังเวลาของการส่งและรับข้อมูลในโหมด 0

จากรูปที่ 2.11 จะเห็นว่าความถี่ของพัลซ์ที่ส่งออกมาที่ขา TXD จะมีค่าเท่ากับ 1/12 ของความถี่ออสซิสเลเตอร์ เมื่อทำการส่งข้อมูลออก ข้อมูลจะถูกเลื่อนออกมาทีละบิตผ่านทางขา RXD และในเวลาเดียวกัน สัญญาณพัลซ์อ้างอิงจะถูกส่งออกมาทางขา TXD ด้วย เมื่อรับข้อมูลเข้า จะรับผ่านทางขา RXD เช่นเดียวกัน การรับข้อมูลจะเกิดขึ้นเมื่อเซตบิตให้ REN = 1 และให้บิต RI = 0 เมื่อบิต RI ถูกเคลียร์ สัญญาณพัลซ์จะถูกส่งออกที่ ขา TXD เพื่อทำการซิงโครไนซ์กับข้อมูลที่รับเข้ามาที่ขอบด้านบวกของพัลซ์ ดังรูปที่ 4 โดยทั่วไปแล้ว ในโหมดนี้มักจะใช้ในการขยายความสามารถทางด้านเอาต์พุตของ 8051 โดยใช้ชิปช่วยในการเปลี่ยนข้อมูลอนุกรมเป็นขนาน (Serial-to-Parallel shift register) จะช่วยให้ทำการขยายเอาต์พุตเพิ่มได้อีก 8 เส้น ดังรูปที่ 5

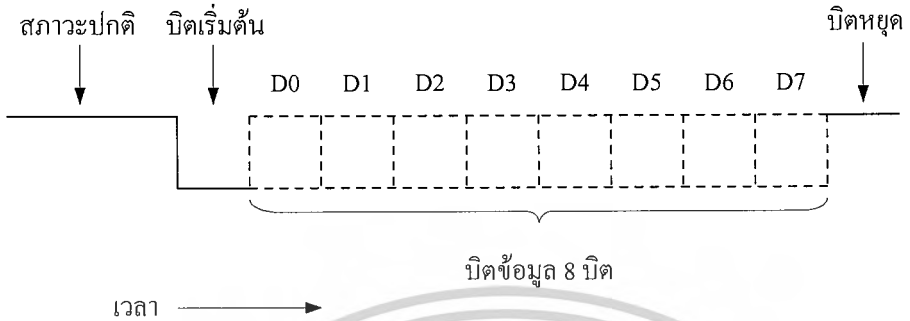


รูปที่2.12 การใช้งานพอร์ตอนุกรมในโหมด 0 เพื่อขยายเอาต์พุตของ 8051

การส่งข้อมูลทางพอร์ตอนุกรมในโหมด 1 (Standard UART) ในโหมดนี้จะต้องกำหนดให้ SM0 และ SM1 ในรีจิสเตอร์ SCON มีค่าเป็น 0 และ 1 ซึ่งจะเป็นการกำหนดให้รีจิสเตอร์ SBUF กลายเป็นตัวรับส่งข้อมูลขนาด 10 บิต แบบฟูลดูเพล็กซ์ ซึ่งจะสามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน (ต่างกับโหมด 0 คือจะสามารถรับหรือส่งในเวลาหนึ่งๆ ได้เพียงอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เดียวเท่านั้น) โดยจะใช้ขา RxD รับสัญญาณอนุกรมที่เข้ามาและ TxD ส่งข้อมูลแบบอนุกรมออกไปภายนอก ดังรูปที่ 2.13



รูปที่ 2.13 รูปแบบการรับและส่งข้อมูลในโหมด 1

การส่งข้อมูลจะเริ่มต้นด้วยการส่ง Start bit (เป็น 0 เสมอ) ออกไปก่อน แล้วตามด้วยบิตข้อมูลอีก 8 บิต (รับและส่งบิตต่ำสุดก่อน) จากนั้นจึงส่ง Stop bit (เป็น 1 เสมอ) ส่วนแฟลก TI จะเซตเมื่อส่งข้อมูลครบทั้ง 10 บิต

การรับข้อมูลจะเริ่มต้นเมื่อลอจิกที่ขา RxD เปลี่ยนจากลอจิก 1 เป็น 0 ข้อมูลจะถูกเลื่อนเข้ามาในบัฟเฟอร์สำหรับรับข้อมูล เมื่อรับครบทั้ง 8 บิตแล้วจึงจะโหลดไปไว้ที่รีจิสเตอร์ SBUF พร้อมกับบิตหยุดจะถูกเก็บไว้ที่บิต RB8 ของ SCON และบิต RI จะเซตเป็น 1 เพื่อให้ทราบว่าได้รับข้อมูลครบ 1 ไบต์แล้ว และจะเริ่มรีเซตตัวเองเพื่อกลับไปรอรับการเปลี่ยนสภาวะจาก 1 เป็น 0 ที่ขา RxD อีก ทั้งนี้กระบวนการรับข้อมูลจะเป็นจริงเมื่อ

- บิต RI = 0 และ
- บิต SM2 =0 หรือ ถ้า SM2 = 1 และบิตหยุดที่รับเข้ามาเป็น 1 ด้วย

การคำนวณอัตราบอด (Baud Rate) ในโหมด 1 เราจะใช้ไทเมอร์ 1 เป็นตัวสร้างอัตราบอด เมื่อเรากำหนดการรับส่งข้อมูลแบบอนุกรมในโหมด 1 แล้ว ไทเมอร์ 1 จะถูกใช้งานในโหมด 2 ซึ่งจะเป็นโหมดการโหลดค่าอัตโนมัติ ซึ่งสามารถทำการคำนวณได้ดังนี้

$$f_{Buad1} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator frequency}}{12 \text{ x } [\, 256 - (TH1) \,]} \qquad (2.3)$$

เมื่อ SMOD จะเป็นบิตควบคุมอยู่ในรีจิสเตอร์ PCON ถ้าเป็น 0 จะเป็นความถี่ปกติ และถ้าเป็น 1 จะเพิ่มความถี่เป็น 2 เท่า ถ้าไม่ใช้ไทเมอร์ 1 ในโหมด 2 จะได้อัตราบอดเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f_{Buad1} = \frac{2^{SMOD}}{32} \times (\text{Timer 1 Overflow Rate}) \quad (2.4)$$

ไทเมอร์ 1 จะทำงานโดยการใช้สัญญาณนาฬิกาจากภายใน หรือทำงานเป็นเคาน์เตอร์ก็ได้ โดยการรับสัญญาณนาฬิกาจากภายนอกผ่านทางขา T1 ถ้าเลือกใช้อัตราบอดมาตรฐาน ควรเลือกใช้ความถี่ของคริสตอลให้ถูกต้องคือ 11.059 MHz

ตัวอย่าง 1 สมมติว่าถ้าต้องการอัตราบอดเป็น 9600 Hz แล้ว ค่าที่ต้องกำหนดให้กับ TH1 คือ

วิธีทำ

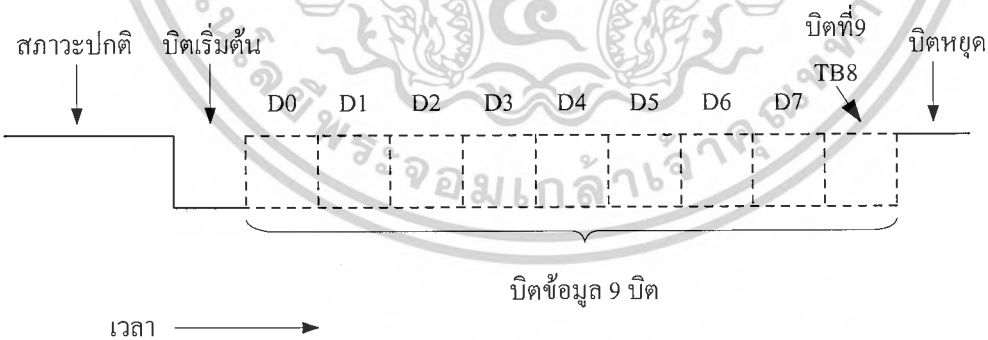
$$f_{Buad1} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator frequency}}{12 \text{ x } [\, 256 - (TH1) \,]}$$

$$TH1 = 256 - \frac{2^0}{32} \times \frac{11.0592 \text{ X } 10^6}{12 \text{ X } 9600} \quad (2.5)$$

$$TH1 = 256 - 3 = 253 = 0FDH$$

เมื่อกำหนดให้ SMOD = 0

การส่งข้อมูลแบบอนุกรมในโหมด 2 (Multiprocessor mode) ในโหมด 2 จะคล้ายกับโหมด 1 ต่างกันที่โหมด 2 จะเป็นการรับส่งข้อมูลแบบ 11 บิต คือ Start bit 1 บิต (เป็น 0 เสมอ) บิตข้อมูล 8 บิต (รับและส่งบิตต่ำสุดก่อน) บิตที่กำหนดเอง 1 บิต (โปรแกรมให้เป็น 0 หรือ 1 ได้เอง) และ Stop bit อีก 1 บิต (เป็น 1 เสมอ) ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 รูปแบบการรับส่งข้อมูลในโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการส่งข้อมูล บิตที่ 9 ข้อมูลจะได้จากบิต TB8 ในรีจิสเตอร์ SCON ส่วนเวลารับข้อมูลบิตที่ 9 จะถูกเก็บเข้าไว้ในบิต RB8

เราสามารถคำนวณอัตราบอดได้ดังนี้

$$f_{Buad2} = \frac{2^{SMOD}}{64} \times \text{Oscilator frequency} \qquad (2.6)$$

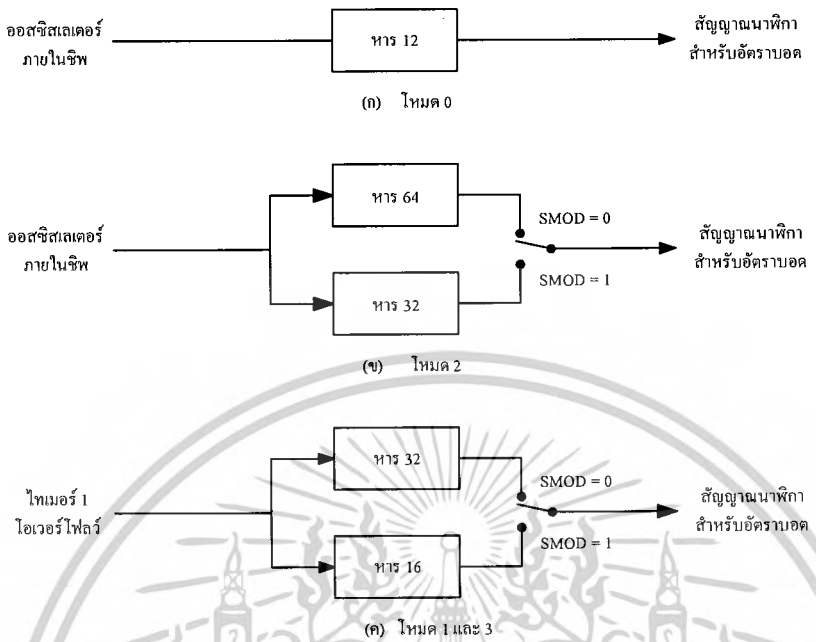
เมื่อ Oscilator frequency = ความถี่ออสซิลเลเตอร์

ข้อกำหนดในการเซตบิต RI ในโหมด 2 จะเหมือนกับในโหมด 1 คือ RI จะต้องเป็น 0 ก่อนบิตที่สุดท้ายจะรับเข้ามาใน 8051 และ SM2 จะต้องเป็น 0 (หรือบิตที่ 9 ของข้อมูลจะต้องเป็น 1) การรับส่งข้อมูลในโหมดนี้มักจะใช้ในการรับส่งข้อมูลระหว่างไมโครโปรเซสเซอร์กับไมโครโปรเซสเซอร์ด้วยกัน (ซึ่งจะต่อสายสัญญาณเข้าหากันโดยตรง) เนื่องจากอัตรารับส่งข้อมูลจะสูงมากกว่าโหมด 1

การส่งข้อมูลแบบอนุกรมในโหมด 3 ในโหมดนี้จะเหมือนกับโหมด 2 ทุกประการ ยกเว้นอัตราบอดเท่านั้นจะหาได้แน่นอนกว่าในโหมด 2 โดยใช้การคำนวณเหมือนโหมด 1 และใช้ไทเมอร์ 1 มาทำการสร้างความถี่สำหรับการส่งสัญญาณแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การคำนวณอัตราบอดของพอร์ตอนุกรม



รูปที่ 2.15 แหล่งกำเนิดอัตราบอดของพอร์ตอนุกรมสำหรับ MCS-51

โหมด 0 อัตราบอดจะมีค่าคงที่ อัตราบอดจะได้จากความถี่ที่เกิดจากคริสตอลหารด้วย 12 ดังรูปที่ 2.15

โหมด 2 อัตราบอดจะได้จากความถี่ของคริสตอลหารด้วย 64 จากรูปที่ 2.15 จะเห็นว่า อัตราบอดเปลี่ยนแปลงได้ โดยการควบคุมบิต SMOD (ในรีจิสเตอร์ PCON) ในโหมด 1,2 และ 3 นั้นถ้า SMOD = 1 จะทำให้อัตราบอด เพิ่มขึ้นเป็น 2 เท่า

โหมด 1 และ 3 อัตราบอดได้จากการเกิดโอเวอร์โฟลว์ (Overflow) ของไทเมอร์ 1 แล้วหารด้วย 32 (หรือ 16 เมื่อ SMOD = 1) (เนื่องจากความถี่ที่ได้จากโอเวอร์โฟลว์ของไทเมอร์ 1 สูงเกินไปในการที่จะนำมาสร้างเป็นสัญญาณนาฬิกาของอัตราบอดให้พอร์ตอนุกรม) ดังรูปที่ 2.15

การใช้ไทเมอร์ 1 เป็นตัวกำเนิดอัตราบอด ทำได้โดยการกำหนดให้ไทเมอร์ 1 ทำงานเป็นแบบ 8 บิต โหลดค่าเองอัตโนมัติ (โหมด 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง 2 ถ้าต้องการอัตราบอดเท่ากับ 1200

วิธีทำ

$$f_{Buad1} = \frac{2^{SMOD}}{32} \times \frac{\text{Oscillator frequency}}{12 \text{ x } [\, 256 - (TH1) \,]}$$

$$TH1 = 256 - \frac{2^0}{32} \times \frac{11.0592 \text{ x } 10^6}{12 \text{ x } 1200} \qquad (2.7)$$

$$TH1 = 256 - 26$$

$$TH1 = 230 = 0E6H$$

สำหรับค่าอื่นๆ จะสรุปไว้เป็นตารางการเลือกอัตราบอดและค่ากำหนดเริ่มต้น ดังตารางที่ 2.2

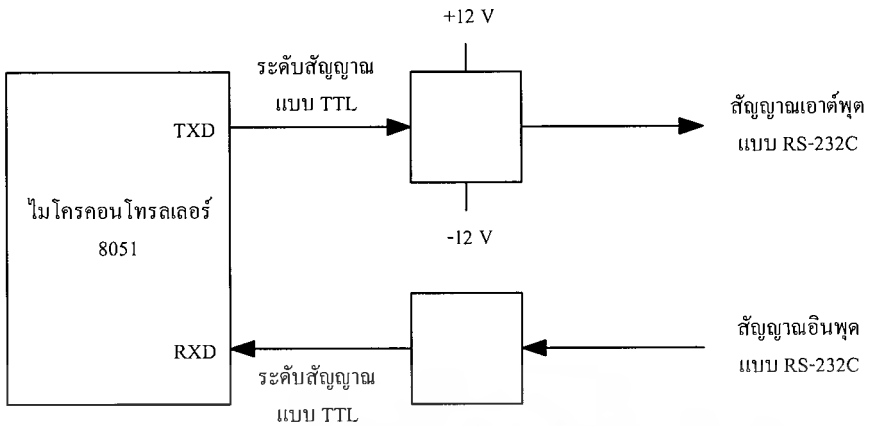
ตารางที่ 2.2 การเลือกใช้อัตราบอดและค่ากำหนดเริ่มต้น

BAUD RATE	CRYSTAL FREQUENCY (MHz)	SMOD	TH1 RELOAD VALUE	ACTUAL BAUD RATE	ERROR (%)
9600	12.000	1	-7 (0F9H)	8923	7
2400	12.000	0	-13 (0F3H)	1202	0.16
1200	12.000	0	-26 (0E6H)	1202	0.16
19200	11.059	1	-3 (0FDH)	19200	0
9600	11.059	0	-3 (0FDH)	9600	0
2400	11.059	0	-12 (0F4H)	2400	0
1200	11.059	0	-24 (0E8H)	1200	0

การเชื่อมต่อกับมาตรฐาน RS-232C

ในการเชื่อมต่อแบบอนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่างๆ มักจะใช้การเชื่อมต่อตามมาตรฐาน RS-232C เพื่อให้การใช้งานมีรูปแบบการเชื่อมต่อที่สอดคล้องกัน โดยทั่วไประดับลอจิกตามมาตรฐาน RS-232C จะแตกต่างจากมาตรฐานของดิจิตอลทั่วๆ ไปคือ ระดับลอจิก 1 จะมีค่าแรงดันอยู่ในช่วง -3V ถึง -20V และระดับลอจิก 0 จะมีค่าแรงดันอยู่ในช่วง +3V ถึง +20V ดังนั้นในการใช้งาน จะต้องเพิ่มอุปกรณ์หรือวงจรพิเศษเข้าไป เพื่อเปลี่ยนระดับแรงดัน 0V ถึง +5V จากขาของ 8051 เป็นระดับแรงดันที่สูงกว่า +3V หรือ ต่ำกว่า -3V ก่อนที่จะต่อกับระดับสัญญาณของ RS-232C ดังรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แนวทางการเชื่อมต่อระดับสัญญาณแบบ TTL จาก 8051 กับระดับสัญญาณของ RS-232C

ปัจจุบันได้มีการผลิตไอซี วงจรรวมที่ประกอบด้วยวงจรรับและส่งแบบ RS-232C อยู่ภายในตัว และต้องการไฟ-เลี้ยงขนาด +5V เท่านั้น เนื่องจากมีวงจรเปลี่ยนระดับแรงดัน (DC to DC Converter) อยู่ภายในไอซีเรียบร้อยแล้ว ดังในรูปที่ 10 ซึ่งทำให้การสร้างวงจรรับและส่งข้อมูลตามมาตรฐาน RS-232C ทำได้ง่ายและสะดวกมากยิ่งขึ้น

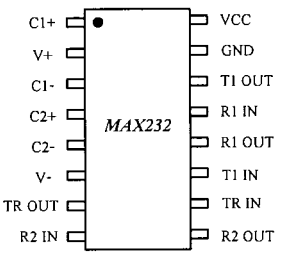
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

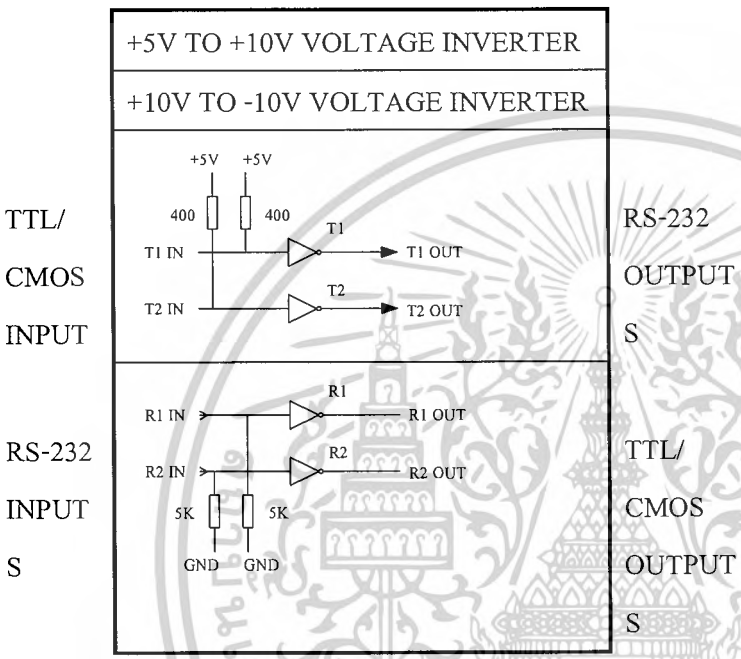
MAX232

RS-232C DRIVERS / RECEICERS

Block Diagram

Pin Configuration

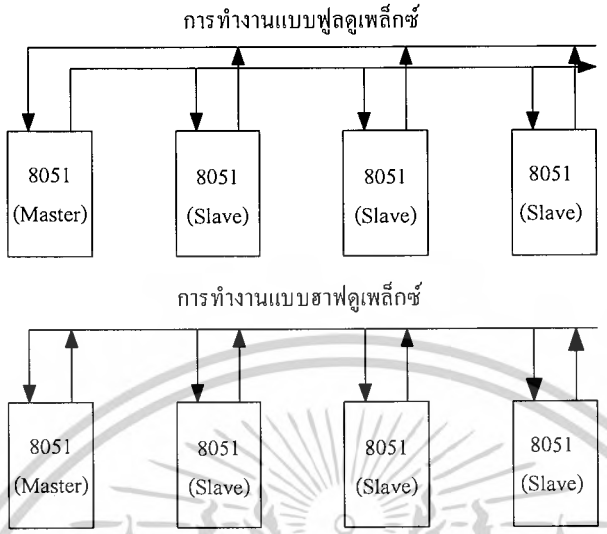




รูปที่ 2.17 ไอซีเบอร์ MAX232 ซึ่งเป็นวงจรเชื่อมต่อแบบ RS-232C โดยการใช้ไฟเลี้ยง +5V เพียงชุดเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์หลายตัว

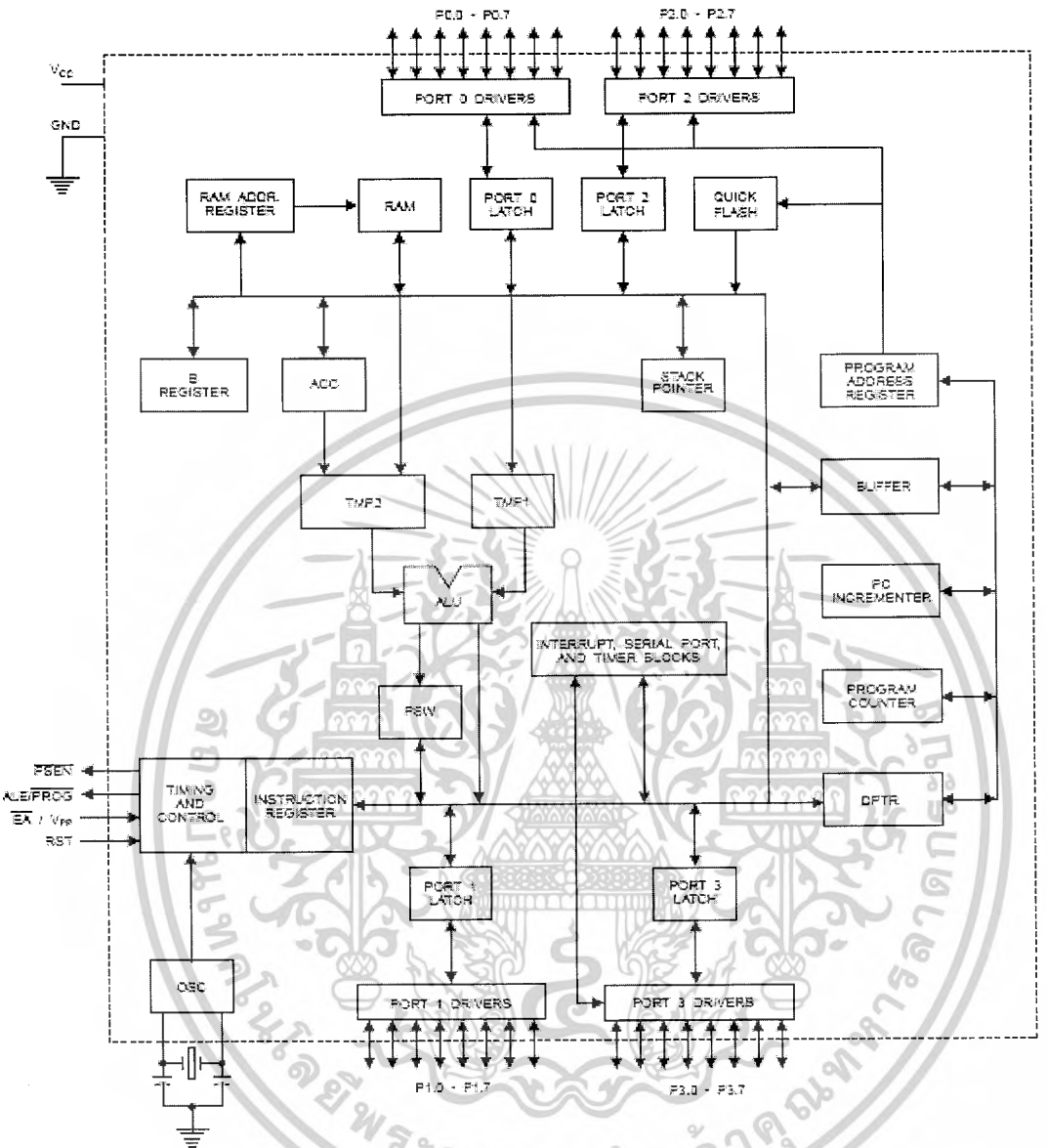


รูปที่ 2.18 การสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์หลายตัว

เราสามารถใช้โหมด 2 และ 3 ในการสื่อสารข้อมูล ระหว่างไมโครคอนโทรลเลอร์หลายตัวได้ (Multiprocessor Communication) ดังรูปที่ 7 เมื่อไมโครคอนโทรลเลอร์หลักต้องการจะส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์รองตัวใด เริ่มต้นจะส่งแอดเดรสไบต์ไปก่อน เพื่อบอกว่าไมโครคอนโทรลเลอร์รองตัวนั้นต้องการจะติดต่อด้วย (แอดเดรสไบต์กับข้อมูลแตกต่างกันคือแอดเดรสไบต์ จะให้บิตที่ 9 เป็น 1 ส่วนข้อมูลจะให้บิตที่ 9 เป็น 0) เมื่อส่งแอดเดรสไบต์ออกไปไมโครคอนโทรลเลอร์รองทุกตัวจะเกิดการอินเตอร์รัพท์ และตรวจสอบแอดเดรสไบต์ที่ดึงเข้ามาว่าตรงกับแอดเดรสของตัวเองหรือไม่ ถ้าตรง ไมโครคอนโทรลเลอร์ตัวนั้นจะเคลียร์บิต SM2 และเตรียมรับข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์หลักต่อไป ถ้าแอดเดรสไบต์ไม่ตรงกันไมโครคอนโทรลเลอร์รองจะให้บิต SM2 = 1 เหมือนเดิม และจะไม่สนใจข้อมูลที่ตามมา จนกว่าไมโครคอนโทรลเลอร์หลัก จะส่งแอดเดรสไบต์มาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 โครงสร้างไมโครคอนโทรลเลอร์ AT89S52



รูปที่ 2.19 โครงสร้างภายในของไมโครคอนโทรลเลอร์ AT89S52

2.8.1 คุณสมบัติของไมโครคอนโทรลเลอร์ AT89S52

- เป็นไมโครคอนโทรลเลอร์ที่มีซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช 8 กิโลไบท์ และหน่วยความจำเสมือน 256 ไบท์ 32 ส่วนรับและส่งออกข้อมูล
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม 256 x 8 บิตใน บางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาท์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- 3 ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต
- สามารถรองรับแหล่งกำเนิด 2 ระดับอินเทอร์รัปได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

2.8.2 การจัดขาของไมโครคอนโทรลเลอร์ AT89S52

รูปที่ 2.20 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ AT89S52

ไมโครคอนโทรลเลอร์ AT89S52 จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานซึ่งเหมือนกับ MCS-51ทั่วไป ขา Vcc ใช้สำหรับเชื่อมต่อแหล่งกำเนิดไฟวงจร +5V ขา GND เป็นขากราวด์สำหรับต่อกับกราวด์ของระบบ

2.8.3 พอร์ตของไมโครคอนโทรลเลอร์ AT89S52

2.8.3.1 พอร์ต 0

ขาพอร์ต 0 รูปที่ 2.14 P0.0-P0.7 มี 8 ขา สามารถกำหนดให้เป็นทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาตำแหน่งไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วยเพื่อสลับการทำงานเป็นไปได้ทั้งขาติดต่อตำแหน่งและขาข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3.2 พอร์ต 1

ขาพอร์ต 1 รูปที่ 2.14 P1.0-P1.7 มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

2.8.3.3 พอร์ต 2

ขาพอร์ต 2 รูปที่ 2.14 P2.0-P2.7 มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาตำแหน่งไบต์สูงของหน่วยความจำภายนอก (A8-A15)

2.8.3.4 พอร์ต 3

ขาพอร์ต 3 รูปที่ 2.14 P3.0-P3.7 มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ขา RxD ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม

P3.1 ขา TxD ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม

P3.2 ขา INT0 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0

P3.3 ขา INT1 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1

P3.4 ขา T0 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0

P3.5 ขา T1ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1

P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา ALE/PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆ ออกมา

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น “ 1 ” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม+12V

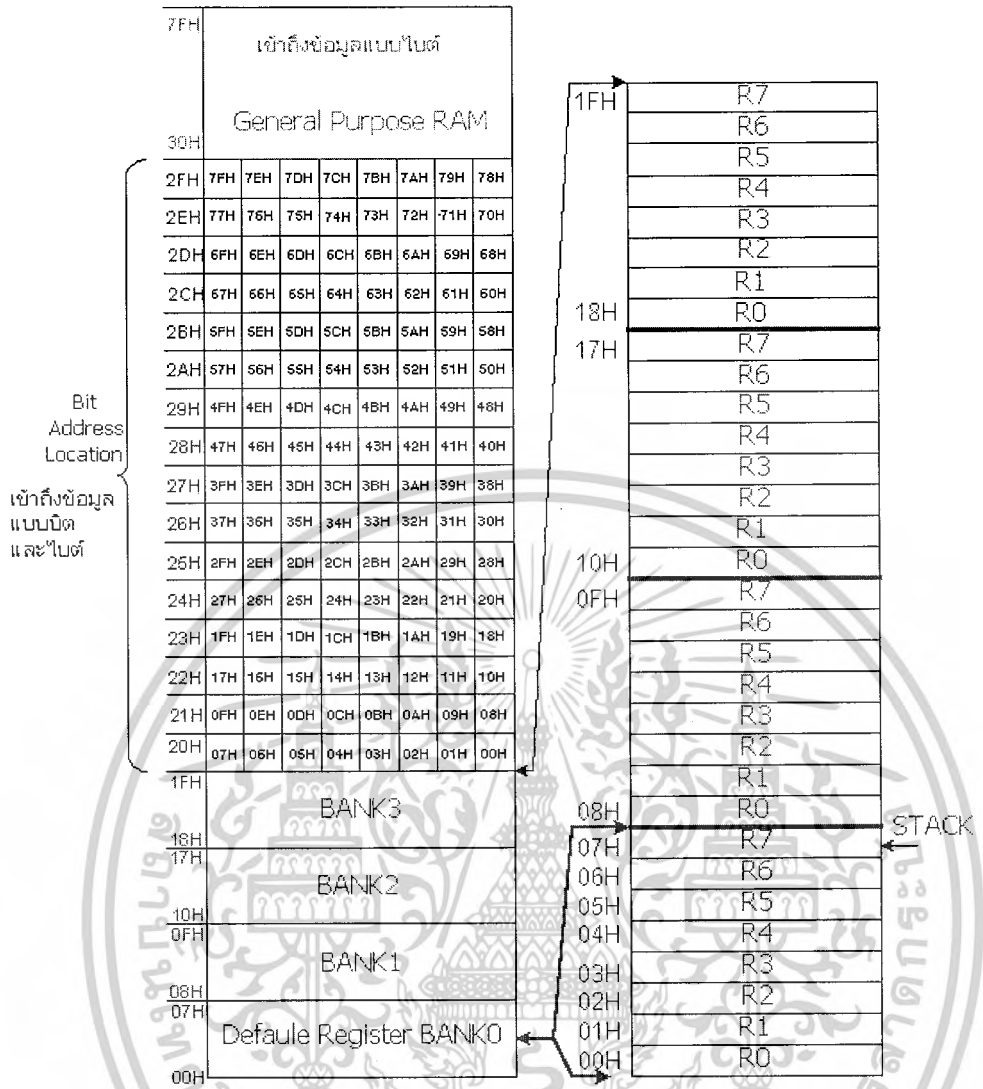
ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานไมโครคอนโทรลเลอร์

2.8.3.5 หน่วยความจำภายในของ ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์สนับสนุนชุดคำสั่งที่ช่วยให้โปรแกรมเมอร์สามารถทำการอ้างอิงตำแหน่งของ RAM ภายในได้หลายวิธี โดยเราสามารถอ้างอิงหน่วยความจำ 32 ตำแหน่งแรกเป็นรีจิสเตอร์ หรืออ้างอิงเป็นตำแหน่งหนึ่งในหน่วยความจำก็ได้ การเข้าถึงข้อมูลในตำแหน่งที่ถูกอ้างอิงเป็น register นั้นเราสามารถทำได้โดยใช้คำสั่งที่มีขนาดเพียงแค่ 1 byte ซึ่งการใช้คำสั่งชนิดนี้จะช่วยให้โปรแกรมเมอร์สามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพ

หน่วยความจำของ ไมโครคอนโทรลเลอร์ นี้จะมีพื้นที่ส่วนหนึ่งขนาด 16 byte ซึ่งเราสามารถทำการอ้างอิงข้อมูลหน่วยความจำส่วนนี้ทีละบิตหรือทีละไบต์ก็ได้ โดยคำสั่งทั่วไปจะสามารถทำการอ้างอิงข้อมูลทีละไบต์จากหน่วยความจำส่วนนี้ แต่ถ้าเราต้องการอ้างอิงข้อมูลทีละบิตในหน่วยความจำส่วนนี้เราจะต้องใช้คำสั่งพิเศษ คำสั่งพิเศษนี้มีประโยชน์มากเมื่อเราต้องการทำการประมวลข้อมูลที่รับมาจากอุปกรณ์ภายนอก เช่น ในงานที่มีการใช้ระบบควบคุม ในหัวข้อต่อไปเราจะพบว่าคำสั่งที่ติดต่อกับ I/O เพื่อทำการเคลื่อนย้ายข้อมูลของ ไมโครคอนโทรลเลอร์ สามารถเคลื่อนย้ายข้อมูลที่มีขนาดเป็นไบต์หรือบิตก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงแผนภาพการจัดสรรหน่วยความจำ หน่วยความจำ ภายใน

รูปที่ 2.21แสดงแผนภาพการจัดสรรหน่วยความจำภายในช่วง 128 ตำแหน่งแรกของไมโครคอนโทรลเลอร์ จากรูปเราจะพบว่า 32 ตำแหน่งแรกในหน่วยความจำจะถูกกำหนดให้เป็นรีจิสเตอร์แบงค์จำนวน 4 แบงค์ ในแต่ละแบงค์จะมีรีจิสเตอร์ 8 ตัว รีจิสเตอร์ในที่นี้เป็นตำแหน่งในหน่วยความจำสำหรับเขียนและอ่านข้อมูลขนาด 8 บิต ซึ่งเราสามารถอ้างอิงได้โดยใช้คำสั่งที่มีขนาด 1 byte และเนื่องจากว่าคำสั่งขนาด 1 byte นี้จะใช้บิตเพียงแค่ 3 บิต ในการระบุตำแหน่งของรีจิสเตอร์ที่เราต้องการดังนั้นเราจึงใช้คำสั่งนี้อ้างรีจิสเตอร์ ที่แตกต่างกันได้เพียง 8 ตัว

แบงก์รีจิสเตอร์ที่กำลังใช้งานอยู่ในจะถูกเลือกโดยการตั้งค่าที่แฟลกสำหรับเลือกแบงก์ เมื่อแบงก์รีจิสเตอร์หนึ่งๆ ถูกเลือกแล้ว การเคลื่อนย้ายข้อมูลจะเกิดขึ้นกับรีจิสเตอร์ R0 และ R7 ของแบงก์นั้น ถ้าเราต้องการเปลี่ยนไปใช้รีจิสเตอร์ในแบงก์อื่น เราต้องกำหนดค่าในแฟลกสำหรับเลือกแบงก์ก่อนจึงจะสามารถเลือกแบงก์ที่จะใช้งานตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.21 แสดงพื้นที่ขนาด 16 byte ในหน่วยความจำตั้งแต่ตำแหน่งที่ 20H ถึง 30H ซึ่งเป็นส่วนของ RAM ที่เราสามารถอ้างอิงข้อมูลเป็นบิตได้ โดยเราสามารถใช้คำสั่งพิเศษเพื่ออ้างอิงข้อมูลในหน่วยความจำส่วนนี้ทีละบิต หรือใช้คำสั่งทั่วไปเพื่ออ้างอิงข้อมูลทีละไบต์ก็ได้ การใช้คำสั่งพิเศษใน ไมโครคอนโทรลเลอร์ ช่วยให้โปรแกรมเมอร์สามารถทำการคอมพลีเมนท์ (complement) บิตใดก็ได้หรือทำการแอนด์ (AND) บิตหนึ่งกับอีกบิตหนึ่งที่อยู่ในแอ็คคอมูเลเตอร์(accumulator) หรืออยู่ในรีจิสเตอร์สถานะเป็นต้น

พื้นที่ส่วนสุดท้ายในหน่วยความจำที่มีขนาด 80 ไบท์ นั้นมีชื่อว่า scratchpad ซึ่งโปรแกรมเมอร์สามารถนำข้อมูลที่เปลี่ยนแปลงได้ไปเก็บ

2.8.3.6 การหาค่าเวลาใน 1 แมชชีนไซเคิล

การทำงานของตัวจับเวลาในโหมดต่างๆ จะใช้สัญญาณนาฬิกาภายในตัวไมโครคอนโทรลเลอร์ ค่าเวลาเริ่มต้นในการนับหรือตั้งเวลาอยู่ที่รีจิสเตอร์ THx ,TLx และจะเพิ่มขึ้นทุกครั้งของการนับ โดย1 ครั้งของการนับ คือการเพิ่มขึ้นทุก 1 แมชชีนไซเคิล (ความถี่สัญญาณนาฬิกาหารด้วย 12) เพราะฉะนั้นการหาค่าการนับจะต้องคำนึงถึง ความถี่ของคริสตอลที่นำมาใช้ด้วย เราสามารถหาเวลาใน 1 แมชชีนไซเคิลได้จากสมการ **T= MC x 12/f X-tal**

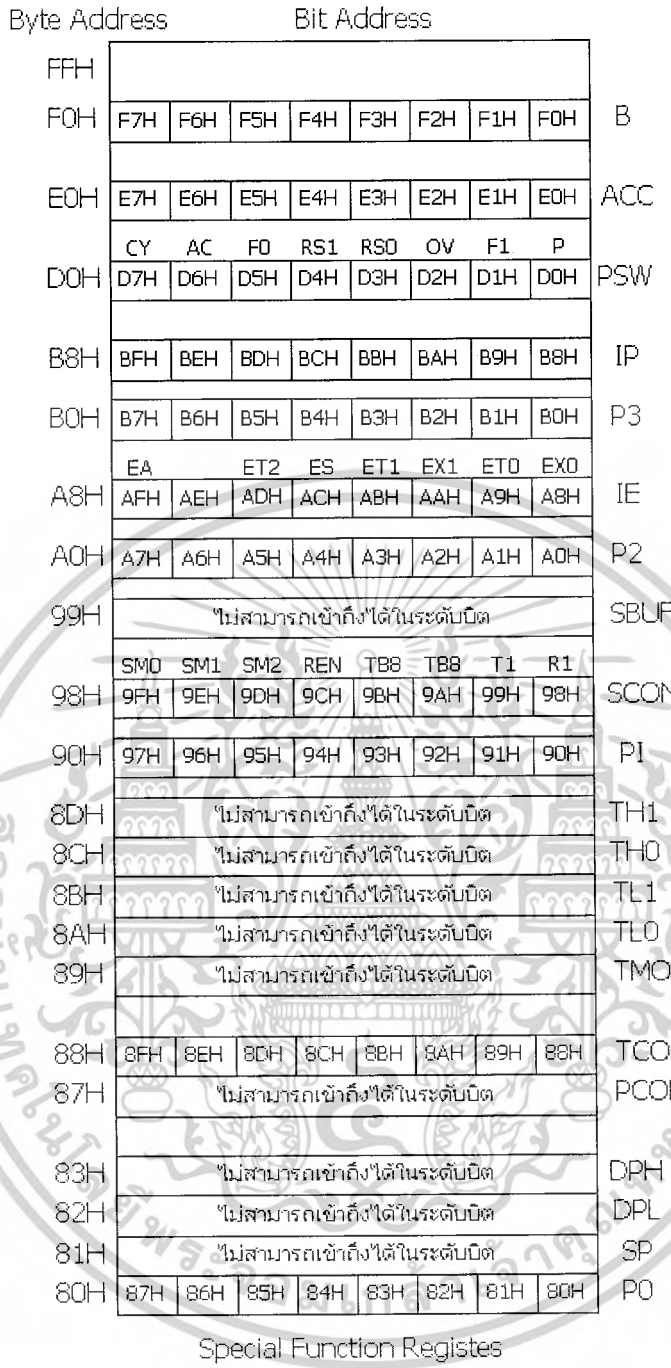
ถ้ากำหนดให้ X-tal 11.0592 MHz

$T= 1x12/11.0592x10^{6}$ (2.8)

$T= 1.085\ x10^{-6}$ sec หรือ 1.085 microsecond

ดังนั้นจึงสรุปได้ว่าในการทำงานของไมโครคอนโทรลเลอร์ 1 แมชชีนไซเคิล ถ้าหากใช้คริสตอล 11.0592 MHz จะใช้เวลา 1.085 microsecond

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 แสดงรีจิสเตอร์หน้าที่พิเศษใน ไมโครคอนโทรลเลอร์ AT89S52

2.8.4 การติดต่อสื่อสารระหว่างคอมพิวเตอร์ และ ไมโครคอนโทรลเลอร์ผ่านพอร์ตอนุกรม

2.8.4.1 คุณสมบัติของพอร์ตอนุกรม

คุณสมบัติที่สำคัญข้อหนึ่งของ คือ พอร์ตแบบอนุกรมที่มีอยู่ในตัวชิป AT89S52 ใน ไมโครโปรเซสเซอร์ทั่วไป การเพิ่มพอร์ตอนุกรมจะต้องเพิ่ม UART และวงจรควบคุมไอซีที่ สนับสนุนการเชื่อมต่อแบบอนุกรม ซึ่งอุปกรณ์เหล่านี้มีความซับซ้อนและมีราคาค่อนข้างสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อแบบอนุกรมของ AT89S52 นั้นเป็นแบบส่งข้อมูลไปและส่งข้อมูลกลับ (full duplex) ได้ในเวลาเดียวกัน การเชื่อมต่อแบบอนุกรมนี้จะมีการพักข้อมูลที่รับเข้ามา (Receive-buffered) แสดงว่าก่อนที่ข้อมูลชุดแรกจะถูกส่งไปยังไมโครโปรเซสเซอร์ พอร์ตอนุกรมจะสามารถรองรับข้อมูลอนุกรมชุดที่สองที่รับเข้ามาจะถูกนำไปเก็บในแลตช์ข้อมูล มิฉะนั้นข้อมูลชิ้นแรกที่อยู่ในนั้นจะถูกเขียนทับ

การเชื่อมต่อแบบอนุกรมนี้มีการใช้รีจิสเตอร์ 2 ตัว ที่ทำหน้าที่รับและส่งข้อมูล เราติดต่อกับรีจิสเตอร์ทั้งสองตัวนี้ได้โดยการอ้างอิงรีจิสเตอร์พิเศษตัวหนึ่งที่มีชื่อว่า SBUF ถ้าเราทำการเขียนข้อมูลลง SBUF แสดงว่าเราได้เขียนข้อมูลลงรีจิสเตอร์ส่งข้อมูล แต่ถ้าเราทำการอ่านข้อมูลจาก SBUF แสดงว่าเราอ่านข้อมูลที่อยู่ในรีจิสเตอร์รับข้อมูล จะเห็นได้ว่ารีจิสเตอร์ที่ทำหน้าที่รับและส่งข้อมูลทั้งสองนี้มีค่าตำแหน่งค่าเดียวกัน

รีจิสเตอร์ควบคุมพอร์ตอนุกรม (Serial port control register :SCON) จะเก็บข้อมูลที่ควบคุมการทำงานของพอร์ตอนุกรม

SM1	SM2	SM3	REN	TB8	RB8	TI	RI

รูปที่ 2.23 รีจิสเตอร์ควบคุมพอร์ตอนุกรม

ตารางที่ 2.3 แสดงการทำงานรีจิสเตอร์ควบคุมพอร์ตอนุกรม

SM1 SCON.7	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM2 SCON.6	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM3 SCON.5	บิตนี้จะถูกนำมาใช้ในการสื่อสารระหว่างไมโครโปรเซสเซอร์หลายตัวในโหมดการทำงานที่ 2 และ 3 สำหรับในโหมด 2 หรือโหมด 3 นี้ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เกิดการเปลี่ยนแปลงหากค่าในบิตที่ 9 ที่รับเข้ามามีค่าเป็น 0 สำหรับในโหมด 1 ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เปลี่ยนแปลงถ้าเกิดการรับบิตสิ้นสุดข้อมูลผิด สำหรับในโหมด 0 บิต SM2 จะมีค่าเป็น 0
REN SCON.4	บิตนี้จะถูกเซตหรือเคลียร์โดยโปรแกรมเพื่อทำการรับ/ไม่รับข้อมูล
TB8 SCON.3	บิตที่ 9 ในข้อมูลอนุกรมที่จะถูกส่งไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RB8 SCON.2	บิตนี้จะมีค่าเท่ากับบิตที่ 9 ในข้อมูลอนุกรมที่รับเข้ามาในโหมดที่ 2 และโหมดที่ 3 สำหรับในโหมดที่ 1 ถ้าบิต SM2 มีค่าเป็น 0 บิต RB8 จะมีค่าเท่ากับค่าในบิตสิ้นสุดข้อมูล สำหรับในโหมด 0 บิต RB8 จะไม่ถูกนำมาใช้งาน
TI SCON.1	แฟลก transmit interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตออกในโหมด 0 หรือเมื่อได้ทำการส่งข้อมูลจนพบบิตสิ้นสุดข้อมูลในโหมดอื่น หลังการส่งข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์
RI SCON.0	แฟลก receive interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตเข้ามาในโหมด 0 หรือที่จุดครึ่งทางของช่องบิตสิ้นสุดข้อมูลในโหมดอื่น (ยกเว้นกรณี SM2 มีค่าเป็น 1) หลังการรับข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์

ตารางที่ 2.3 (ต่อ)

บิต SM0 และ SM1 จะใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม ซึ่งมีทั้งหมด 4 โหมดคือ โหมด 0,1,2 และ 3 ซึ่งมีการทำงานแตกต่างกันออกไปโดย

ในโหมด 0 พอร์ตอนุกรมจะเป็นเหมือนรีจิสเตอร์ที่เลื่อนค่าได้ (shift register) ซึ่งจะเลื่อนค่าตามสัญญาณนาฬิกาที่มีอัตราความถี่เท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ใน 8051 สัญญาณนาฬิกานี้เป็นตัวกำหนดอัตราความเร็วในการเคลื่อนย้ายข้อมูลแบบอนุกรมที่เราเรียกว่า อัตราบอด (baud rate)

ในโหมดที่ 1 พอร์ตอนุกรมจะรับและส่งข้อมูลขนาด 10 บิต ซึ่งข้อมูลขนาด 10 บิตนี้ประกอบด้วยบิตเริ่มต้นข้อมูลขนาด 8 บิต และบิตสิ้นสุดข้อมูล 1 บิต ในโหมดการทำงานนี้ timer ตัวหนึ่งใน 8051 จะนำมาใช้ในการกำหนดอัตราเร็วในการเคลื่อนย้ายข้อมูลของพอร์ตอนุกรม

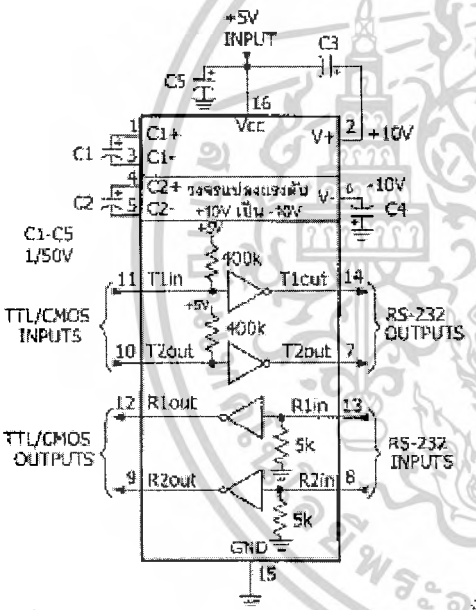
ในโหมดที่ 2 พอร์ตอนุกรมจะทำการรับและส่งข้อมูลทีละ 11 บิต ซึ่งข้อมูลขนาด 11 บิตนี้ประกอบด้วยข้อมูลขนาด 8 บิต บิตที่ 9 (เป็นบิตที่เราสามารถกำหนดค่าได้) บิตเริ่มต้นข้อมูล และบิตสิ้นสุดข้อมูลโดยบิตที่ 9 นี้จะมีค่าตรงกับบิต TB8 ของรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) ในโหมดการทำงานนี้เราสามารถเลือก baud rate ให้มีค่าเท่าเป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ใน AT89S52 ได้การทำงานของพอร์ตอนุกรมในโหมดที่ 3 จะทำการรับและส่งข้อมูลทีละ 11 บิตและมีบิตที่ 9 ที่กำหนดค่าได้ โดยบิตนี้จะมีค่าตรงกับบิต RB8 ในรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) โหมดที่ 3 นี้จะทำหน้าที่เช่นเดียวกับโหมดที่ 2 ยกเว้นแต่ว่าอัตราเร็วในการเคลื่อนย้ายข้อมูลสามารถเปลี่ยนแปลงได้ ไมโครโปรเซสเซอร์ AT89S52 หลายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

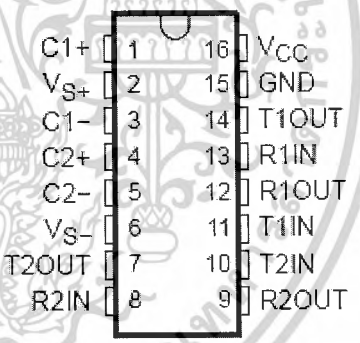
สามารถติดต่อกันได้โดยการเชื่อมต่อกันแบบอนุกรม เมื่อมีข้อมูลส่งมายังไมโครคอนโทรเลอร์ บิตที่ 9 ในข้อมูลที่รับเข้ามาจะทำให้เกิดการอินเตอร์รัปต์ซึ่งจะบอกให้ไมโครคอนโทรลเลอร์รู้ว่าพอร์ตอนุกรมของมันได้ทำการรับข้อมูลเข้ามาเก็บในที่พักข้อมูล และมันควรทำการโอนย้ายข้อมูลนี้เพื่อนำไปใช้ต่อไป ใน AT89S52 อัตราบอดในโหมด 1 และโหมด 3 จะถูกกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทเมอร์เบอร์ 1 และเราสามารถเพิ่มอัตราบอด ให้เป็นสองเท่าได้โดยการกำหนดค่าของบิต SMOD ในรีจิสเตอร์ควบคุมให้มีค่าเป็น 1

2.8.4.2 การติดต่อสื่อสารผ่านทางพอร์ตอนุกรม

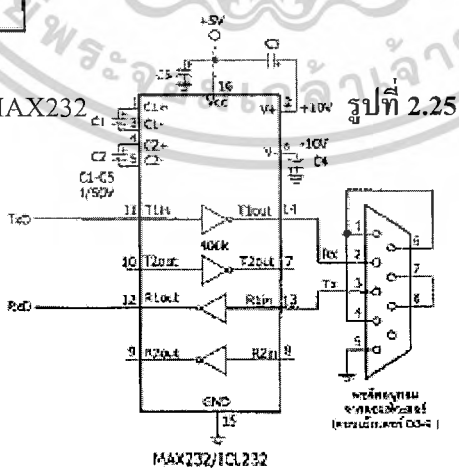
การเชื่อมต่อคอมพิวเตอร์กับไมโครคอนโทรลเลอร์โดยตรงไม่สามารถทำได้เนื่องจากระดับสัญญาณของไมโครคอนโทรลเลอร์ไม่สูงพอจึงต้องอาศัยการเชื่อมต่อผ่านไอซีตัวหนึ่งที่ทำหน้าที่แปลงระดับสัญญาณให้สูงจนสามารถทำให้เกิดการติดต่อกันได้ โดยในที่นี้จะใช้ไอซีที่ MAX232



รูปที่ 2.24 วงจรภายในของ MAX232



รูปที่ 2.25 การจัดขาของ MAX232



รูปที่ 2.26 วงจรเชื่อมต่อ MAX232 กับพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบระบบ

3.1 การวิเคราะห์

ที่จอดรถโดยทั่วไปนั้นมักจะใช้การดูแลโดยแรงงานคน ซึ่งทำให้เกิดข้อจำกัดในด้านการทำงาน ดังนั้นจึงได้ทำการวิเคราะห์ระบบ โดยจำแนกการทำงานต่างๆออกเป็นส่วนๆ โดยสามารถแบ่งออกได้เป็น 5 ส่วน ดังนี้

3.1.1 ตัวรับสัญญาณแบบตัดแสง (Sensor infrared)

เป็นอุปกรณ์ที่ใช้ในการตรวจจับสถานะของที่จอดรถ เพื่อดูว่ามีการเปลี่ยนแปลงของที่จอดรถหรือไม่ โดยจะทำการตรวจสอบอยู่ตลอดเวลา โดยจะมีการส่งค่าไปยังไมโครคอนโทรเลอร์ (Micro controller) ให้ทราบถึงสถานะของที่จอดรถ โดยในแต่ละที่จอดรถจะมี ตัวตรวจจับสัญญาณอินฟราเรด ประจำอยู่ทุกที่จอดรถ

3.1.2 ไมโครคอนโทรลเลอร์ (Micro controller)

เป็นแผงวงจรที่ใช้ในการรับข้อมูลจากตัวรับสัญญาณอินฟราเรด (Sensor infrared) โดยหลังจากรับค่าแล้ว จะทำการตรวจสอบว่าตัวรับสัญญาณอินฟราเรด (Sensor infrared) ตัวใดส่งมาและส่งค่าใดมา หลังจากนั้นไมโครคอนโทรลเลอร์ (Micro controller) จะทำการประมวลผล เพื่อแยกแยะค่าที่ได้ แล้วจะทำการส่งค่าเป็นรหัสแอสกี้ (ASCII) ไปยังคอมพิวเตอร์ โดยส่งผ่านทางซีเรียลพอร์ต (Serial Port) RS-232

3.1.3 แอปพลิเคชั่น (Application)

หลังจากที่ได้รับข้อมูลเป็นรหัสแอสกี้ (ASCII) ทาง ซีเรียลพอร์ต (Serial Port) RS-232 แล้ว แอปพลิเคชั่น (Application) จะทำการตรวจสอบรหัสแอสกี้ที่ได้มา แล้วแปลงรหัสแอสกี้ (ASCII) เป็นตัวอักษร หลังจากนั้นจึงนำไปตรวจสอบค่าว่าเป็นตัวอักษรใด เพื่อนำไปแยกแยะกรณีที่จะกระทำกับฐานข้อมูล โดยแต่ละตัวอักษรจะกระทำกับฐานข้อมูลแตกต่างกันไป

3.1.4 ฐานข้อมูล (Database)

จะเก็บข้อมูลสถานะของที่จอดรถไว้โดยจะมีการเปลี่ยนแปลงอยู่ตลอดเวลา โดยการรับคำสั่งเปลี่ยนแปลงจากแอปพลิเคชั่น (Application) มาเปลี่ยนแปลงฐานข้อมูล เพื่อให้ข้อมูลในฐานข้อมูล ตรงตามความเป็นจริงมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 เว็บแอปพลิเคชั่น (Web Application)

จะแสดงข้อมูลจากฐานข้อมูลโดยจะแสดงออกได้ 2 รูปแบบ คือในรูปแบบที่เหมาะสำหรับบนเว็บ และรูปแบบที่เหมาะสำหรับมือถือ

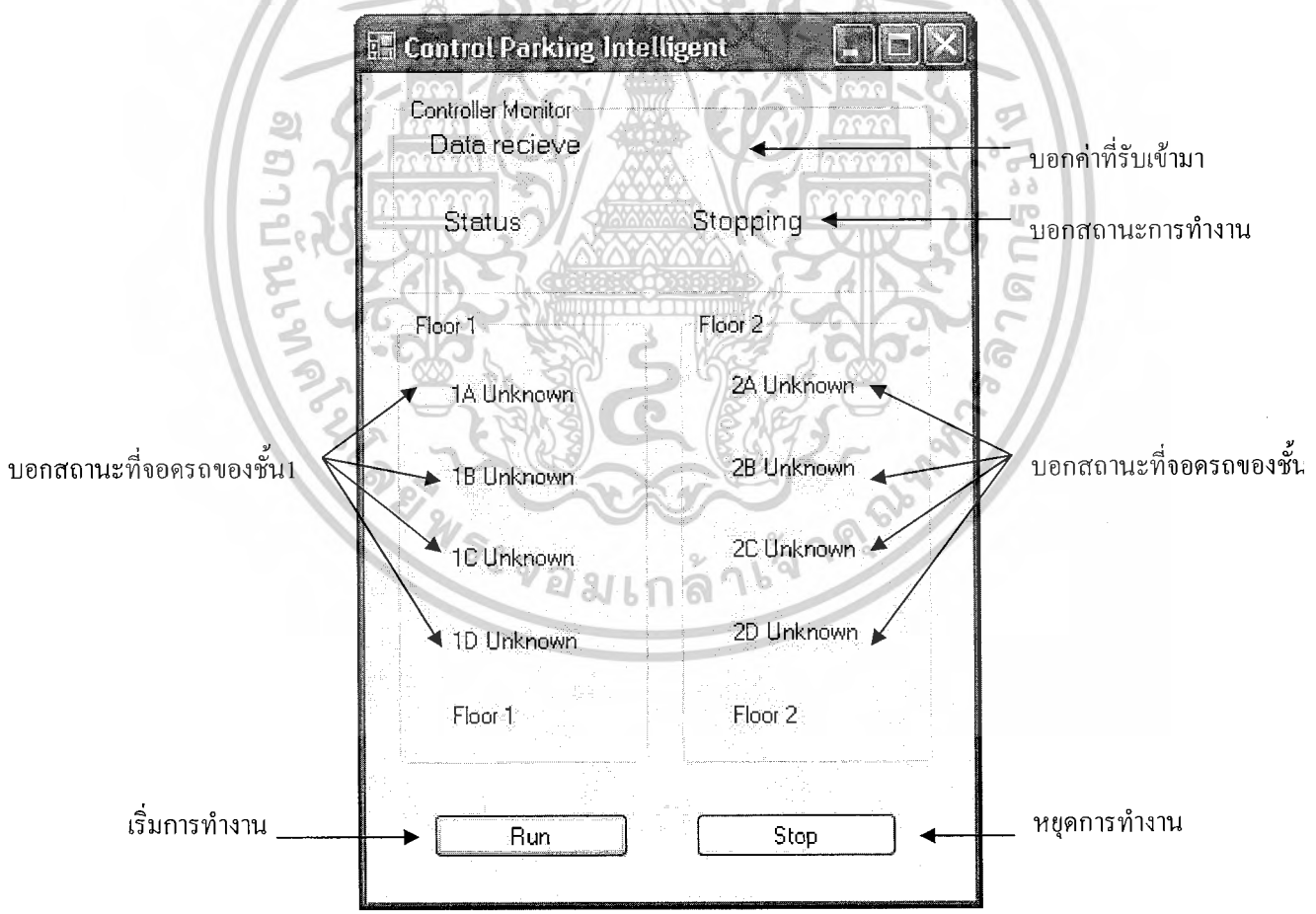
3.2 การออกแบบ

จากการวิเคราะห์ระบบแล้ว ทำให้ทราบได้ว่า มีส่วนที่จำเป็นต้องออกแบบ อยู่ทั้งหมด 3 ส่วนดังต่อไปนี้

3.2.1 ตัวรับส่งสัญญาณแบบตัดแสง (Sensor infrared)

ระยะทดสอบประมาณ 60 เซนติเมตร ความถี่ 38 KHz

3.2.2 แอปพลิเคชั่น (Application)



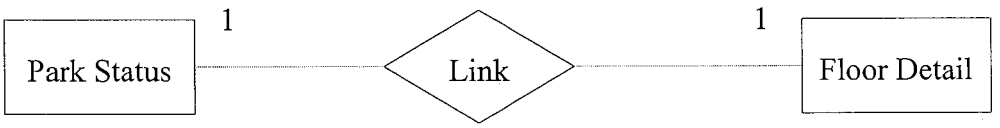
รูปที่ 3.1 หน้าจอหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ฐานข้อมูล (Database)

3.2.3.1 แบบจำลองความสัมพันธ์ของเอนติตี้ (Entity Relationship Model)

สามารถเขียนออกมาได้ในรูปแบบของ ER-Model ดังรูป



รูปที่ 3.2 Entity Relationship Model

3.2.3.2 พจนานุกรมข้อมูล (Data Dictionary)

ตารางที่ 3.1 ข้อมูลของฐานข้อมูล

ชื่อตาราง	รายละเอียด
Park Status	จัดเก็บข้อมูลสถานะของที่จอดรถ
Floor Detail	จัดเก็บข้อมูลรายละเอียดของชั้นจอดรถทั้งหมด

3.2.3.3 รายละเอียดข้อมูลในตาราง

ตารางที่ 3.2 พจนานุกรมข้อมูลของตาราง Status

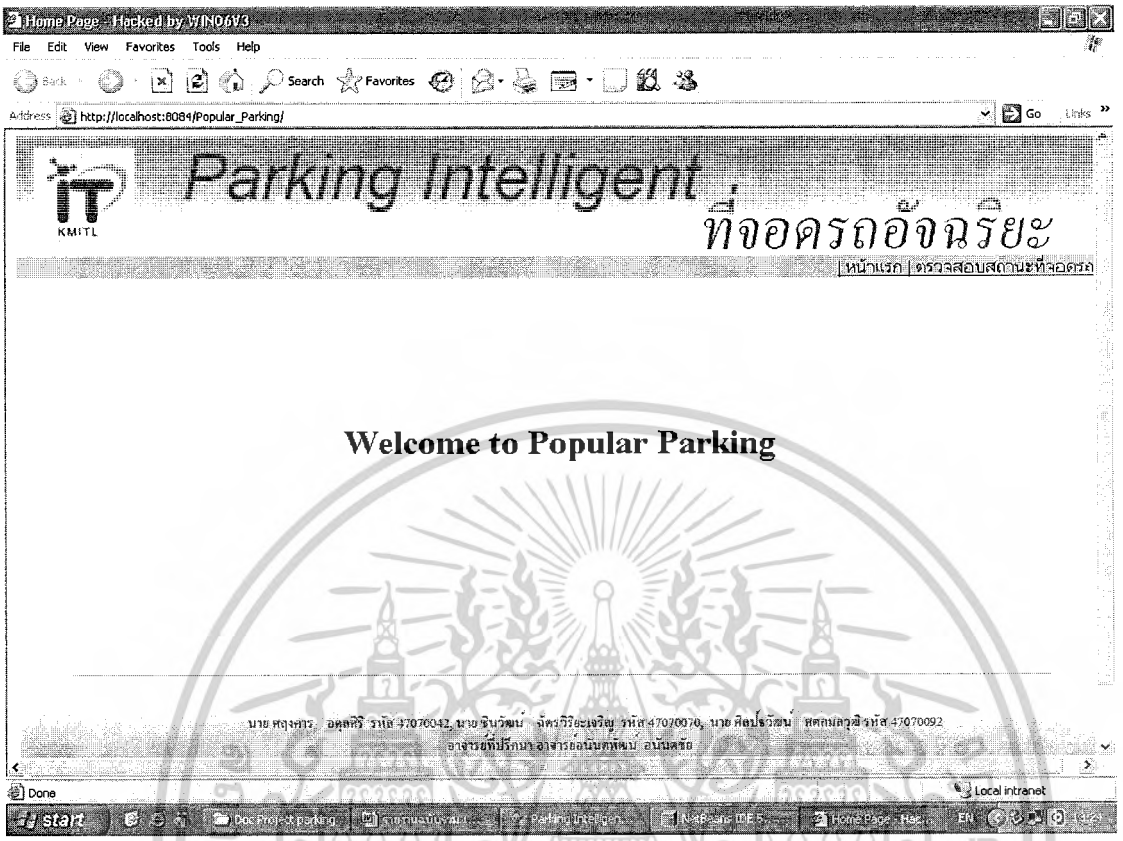
Attribute Name	Description	Types	Length	Key	FK Referenced Table
ID	รหัสที่จอดรถ	Varchar	4	PK	-
floor	ชั้นของที่จอดรถ	Varchar	2	FK	Floor Detail
status	สถานะของที่จอดรถ	Varchar	10	-	-

ตารางที่ 3.3 พจนานุกรมข้อมูลของตาราง Floor Detail

Attribute Name	Description	Types	Length	Key	FK Referenced Table
floor	ชั้นของที่จอดรถ	Varchar	2	PK	-
total	จำนวนของที่จอดรถที่มี	Number	2	-	-
have	จำนวนที่จอดรถที่ว่าง	Number	2	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

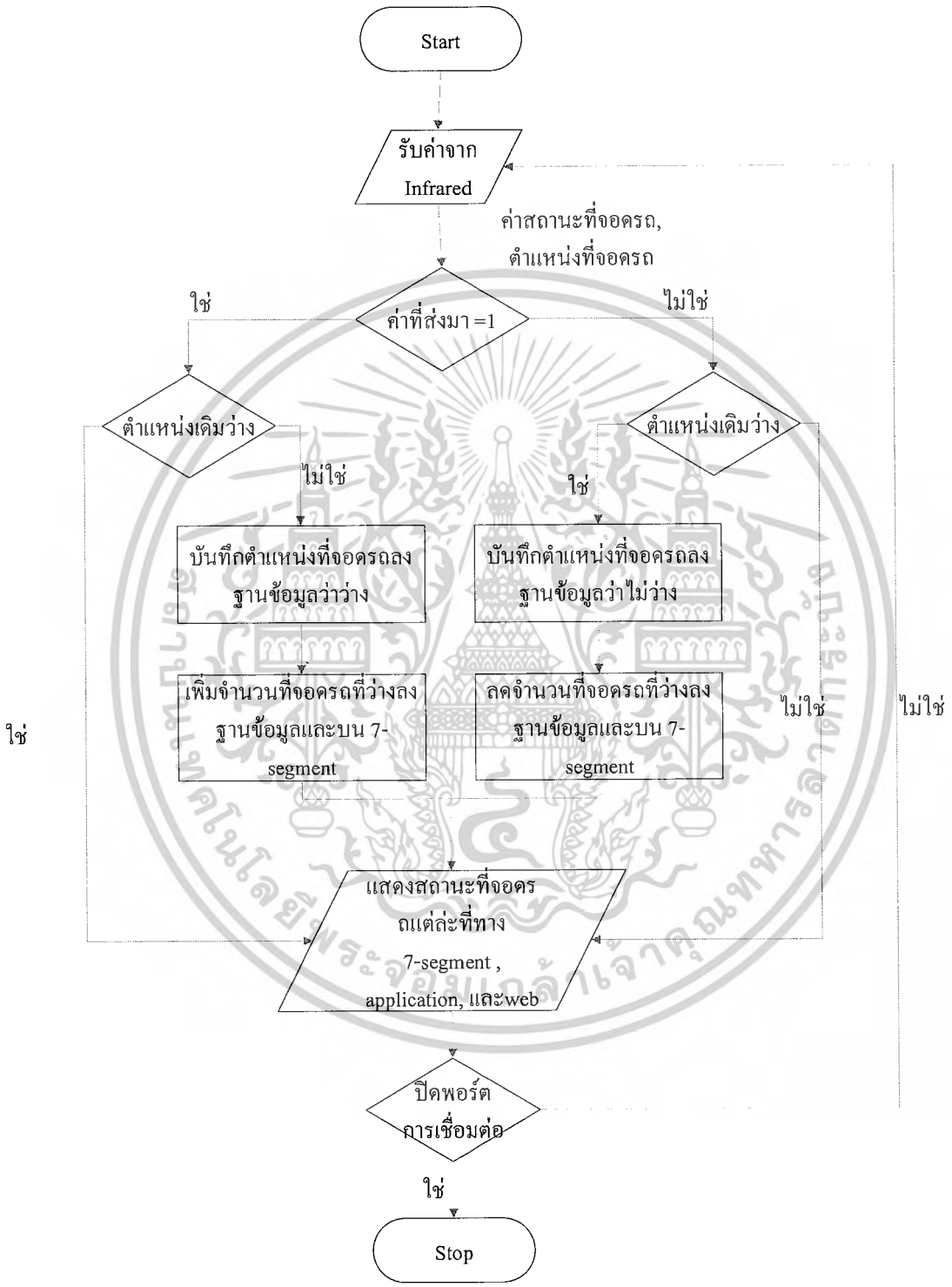
3.2.4 เว็บแอปพลิเคชั่น



รูปที่ 3.3 หน้าแรกของเว็บแอปพลิเคชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 แผนภาพขั้นตอนการทำงานของระบบ (Flow Chart Diagram)



รูปที่ 3.4 แผนภาพการขั้นตอนการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 สรุปการทำงาน

ระบบที่จอดรถนั้นจะทำการตรวจสอบสถานะที่จอดรรถด้วยตัวรับสัญญาณอินฟราเรด (Sensor Infrared) แล้วจึงส่งค่าต่อไปให้ ไมโครคอนโทรเลอร์ (Microcontroller) ซึ่งจะนำค่าที่ได้ไปใช้เปรียบเทียบหากรณีต่างๆ หลังจากนั้นจะทำการส่ง ข้อมูลออกไปเป็นรหัสแอสกี้ (ASCII) ซึ่งจะส่งไปยังคอมพิวเตอร์ ทาง พอร์ต RS-232 หลังจากนั้นจะนำค่าที่ได้ไปเปรียบกรณีต่างๆโดยแอปพลิเคชั่น (Application) หลังจากนั้นเว็บแอปพลิเคชั่นจะทำการดึงค่าไปแสดงผลจากฐานข้อมูล ดังรูป

รูปที่ 3.5 แผนภาพแสดงสถาปัตยกรรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

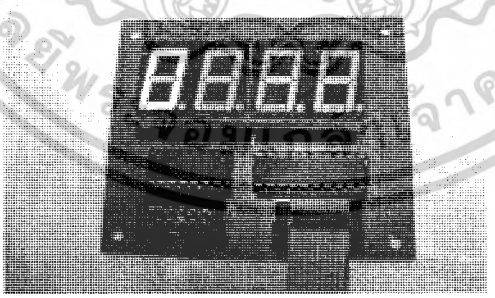
วิธีการทดลองจะทำการทดลองโดย จำลองที่จอดรถลงเหลือขนาดเล็ก ทั้งนี้เพื่อความสะดวกในการทดลอง โดยที่จะทำการนำวัตถุมาขวางตัวรับสัญญาณ หลังจากนั้นจะตรวจสอบว่ามีการส่งค่าไปยังไมโครคอนโทรลเลอร์ (Microcontroller) หรือไม่ โดยสามารถตรวจสอบได้โดยแอปพลิเคชั่น (Application) หลังจากนั้นจะดูในฐานข้อมูลว่ามีการเปลี่ยนแปลงของฐานข้อมูลหรือไม่อย่างไร

4.2 ผลการทดลอง

จะแบ่งส่วนการสังเกตเป็น 3 ส่วน คือ 7 เซคเม้น (7 segments), แอปพลิเคชั่น (Application) และ ฐานข้อมูล โดยจะแบ่งสถานะการทดลองเป็น 4 สถานะดังนี้

4.2.1 ก่อนเปิดพอร์ตการเชื่อมต่อ

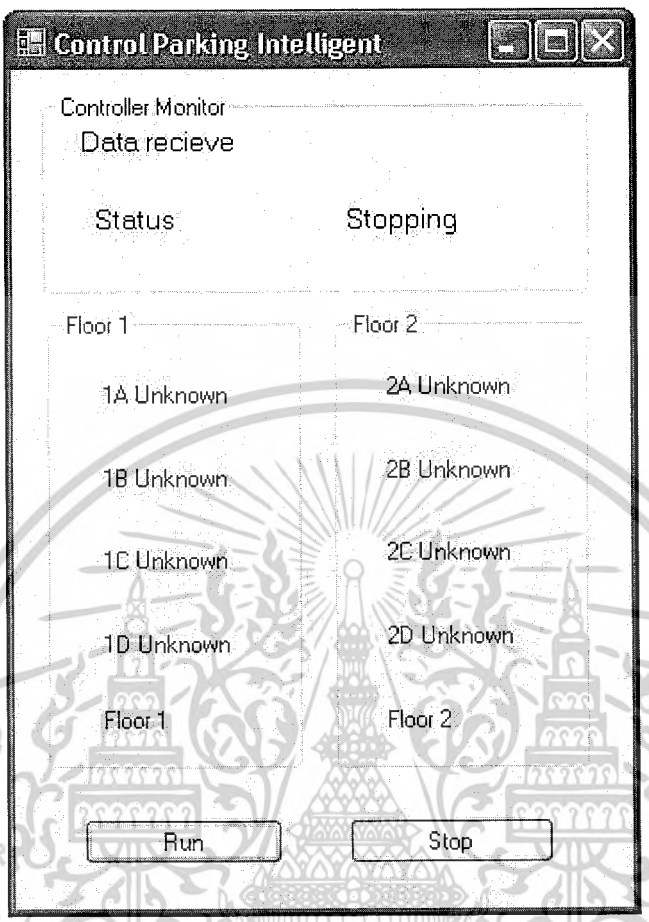
4.2.1.1 7 เซคเม้น (7 segments)



รูปที่ 4.1 7 เซคเม้นก่อนเปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2 แอปพลิเคชั่น (Application)



รูปที่ 4.2 แอปพลิเคชั่นก่อนเปิดพอร์ต

4.2.1.3 ฐานข้อมูล

4.2.1.3.1 ตารางสถานะ

id	floor	status
1A	1	Vacancy
1B	1	Vacancy
1C	1	Vacancy
1D	1	Vacancy
2A	2	Vacancy
2B	2	Vacancy
2C	2	Vacancy
2D	2	Vacancy

รูปที่ 4.3 ฐานข้อมูลตารางสถานะ ก่อนเปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

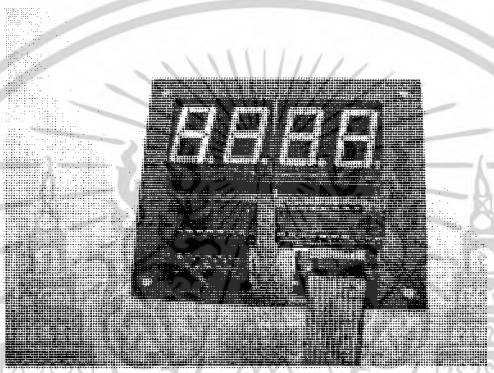
4.2.1.3.2 ตารางชั้น

	f...	total	have
▶	1	4	4
	2	4	4

รูปที่ 4.4 ฐานข้อมูลตารางชั้น ก่อนเปิดพอร์ต

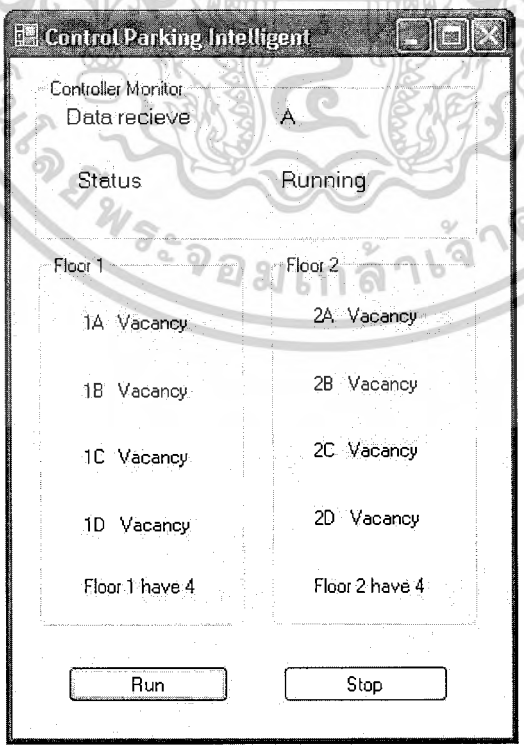
4.2.2 เปิดพอร์ตการเชื่อมต่อโดยยังไม่มีรถมาจอด

4.2.2.1 7 เซคเม้น (7 segments)



รูปที่ 4.5 7 เซคเม้นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด

4.2.2.2 แอปพลิเคชั่น (Application)



รูปที่ 4.6 แอปพลิเคชั่นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 ฐานข้อมูล

4.2.2.3.1 ตารางสถานะ

id	floor	status
1A	1	Vacancy
1B	1	Vacancy
1C	1	Vacancy
1D	1	Vacancy
2A	2	Vacancy
2B	2	Vacancy
2C	2	Vacancy
2D	2	Vacancy

รูปที่ 4.7 ฐานข้อมูลตารางสถานะขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด

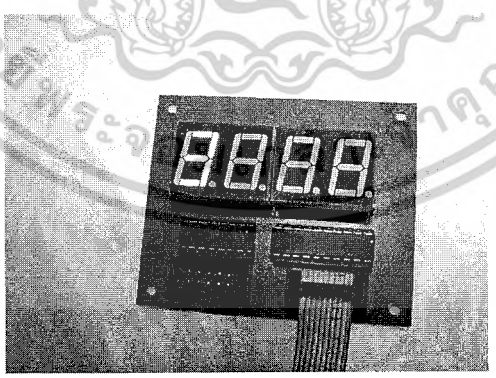
4.2.2.3.2 ตารางชั้น

f...	total	have
1	4	4
2	4	4

รูปที่ 4.8 ฐานข้อมูลตารางชั้นขณะเปิดพอร์ต โดยยังไม่มีรถมาจอด

4.2.3 เมื่อมีรถมาจอด

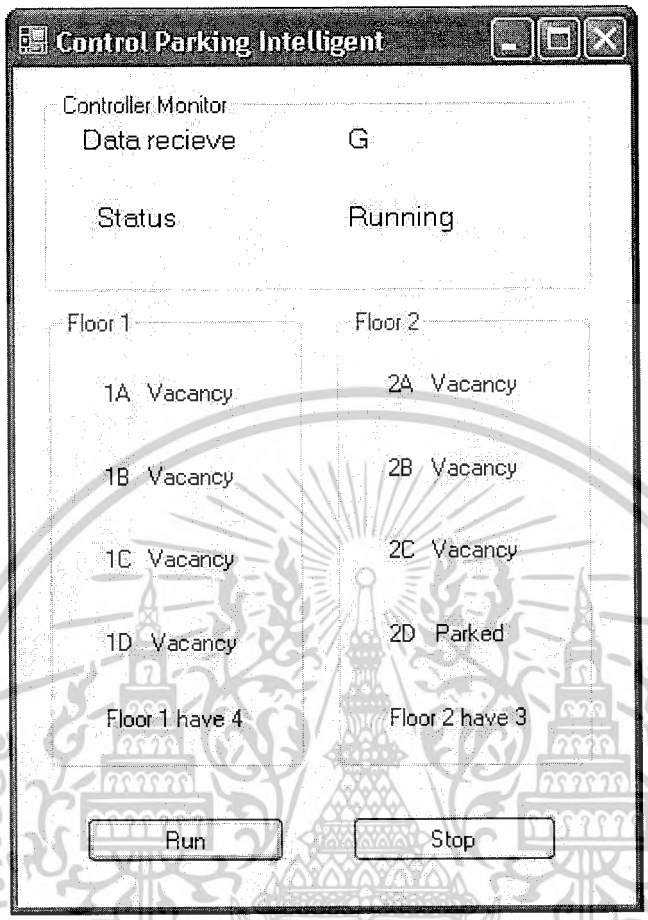
4.2.3.1 7 เซคเม้น (7 segments)



รูปที่ 4.9 7 เซคเม้นขณะเปิดพอร์ต โดยมีรถมาจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3.2 แอปพลิเคชั่น (Application)



รูปที่ 4.10 แอปพลิเคชั่นขณะเปิดพอร์ต โดยมีรถมาจอด

4.2.3.3 ฐานข้อมูล

4.2.3.3.1 ตารางสถานะ

id	floor	status
1A	1	Vacancy
1B	1	Vacancy
1C	1	Vacancy
1D	1	Vacancy
2A	2	Vacancy
2B	2	Vacancy
2C	2	Vacancy
2D	2	Parked

รูปที่ 4.11 ฐานข้อมูลตารางสถานะขณะเปิดพอร์ต โดยมีรถมาจอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

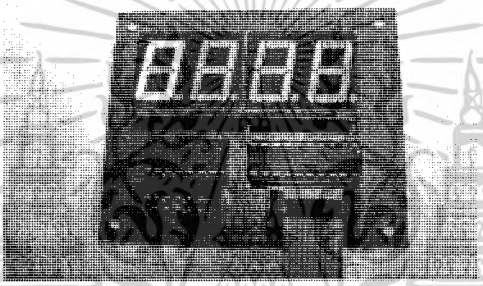
4.2.3.3.2 ตารางชั้น

f...	total	have
1	4	4
2	4	3

รูปที่ 4.12 ฐานข้อมูลตารางชั้นขณะเปิดพอร์ต โดยมีรถมาจอด

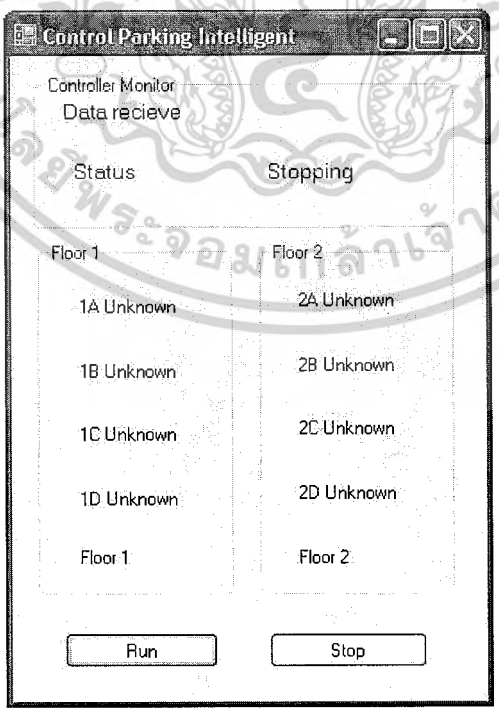
4.2.4 หลังปิดพอร์ตการเชื่อมต่อ

4.2.4.1 7 เซคเม้น (7 segments)



รูปที่ 4.13 7 เซคเม้นเมื่อปิดพอร์ต

4.2.4.2 แอปพลิเคชั่น (Application)



รูปที่ 4.14 แอปพลิเคชั่นเมื่อปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.3 ฐานข้อมูล

4.2.4.3.1 ตารางสถานะ

id	floor	status
1A	1	Vacancy
1B	1	Vacancy
1C	1	Vacancy
1D	1	Vacancy
2A	2	Vacancy
2B	2	Vacancy
2C	2	Vacancy
2D	2	Parked

รูปที่ 4.15 ฐานข้อมูลตารางสถานะเมื่อปิดพอร์ต

4.2.4.3.2 ตารางชั้น

f...	total	have
1	4	4
2	4	3

รูปที่ 4.16 ฐานข้อมูลตารางชั้นเมื่อปิดพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปโครงการและข้อเสนอแนะ

5.1 สรุปผลโครงการ

ตัวตรวจรับสัญญาณสามารถที่จะตรวจสถานะของที่จอดรถได้ โดยในที่นี้ตัวรับสัญญาณเป็นเพียงอุปกรณ์ต้นแบบ ที่พยายามออกแบบเพื่อประหยัดต้นทุน จึงใช้ ตัวรับส่งสัญญาณ ที่มีกำลังการส่งต่ำ โดยจากผลการทดลองแสดงให้เห็นได้ว่า ระบบสามารถตรวจสอบได้จริงถึงการมีและไม่มีอยู่ของวัตถุ โดยที่มีข้อจำกัดว่าทั้งตัวรับและตัวส่งมีความจำเป็นที่จะต้องตั้งตรงกันเพื่อที่จะได้รับส่งสัญญาณได้

เมื่อรับข้อมูลมาแล้วจะทำการตรวจสอบแล้วส่งค่าไปให้แอปพลิเคชั่น โดยที่การส่งค่านั้นสามารถที่จะส่งได้หลากหลายตั้งแต่ 1 หลักข้อมูลถึง 8 หลัก ทำให้สามารถส่งข้อมูลได้มากมายทั้งนี้การส่งไปยังแอปพลิเคชั่นเพื่อทำการตรวจสอบกรณีก่อนทำการเก็บค่าลงในฐานข้อมูล ซึ่งจะทำการเก็บค่าไว้ตลอดเวลา ทำให้มั่นใจได้ว่าข้อมูลในฐานข้อมูลเป็นที่ถูกต้อง นอกจากนี้ยังนำข้อมูลจากฐานข้อมูลไปแสดงผลบนเว็บแอปพลิเคชั่นได้อีกด้วย ด้วยโครงการดังกล่าวนี้ทำให้สามารถประหยัดทรัพยากรมนุษย์ได้ อีกทั้งยังประหยัดเวลาและสะดวกอีกด้วย

5.2 ข้อเสนอแนะ

จากโครงการนี้จะเห็นได้ว่ามีการเก็บข้อมูลไว้ในฐานข้อมูล ซึ่งฐานข้อมูลนี้สามารถนำไปใช้ประโยชน์ต่อได้ นอกเหนือจากการสร้างเว็บเพื่อให้บริการแก่ผู้ใช้อื่นๆ คือ อาจมีการเก็บข้อมูลที่มีไว้เพื่อวิเคราะห์และวางแผนในการบริหารและจัดการที่จอดรถนั้นๆให้เหมาะสมมากที่สุด โดยฐานข้อมูลนี้จะเก็บแบบตามความเป็นจริงทำให้มั่นใจได้ว่า จะได้ข้อมูลที่ตรงกับความเป็นจริงมากที่สุด

ทั้งนี้ การใช้ฐานข้อมูล MY SQL เป็นที่นิยมมาก จึงมีความสะดวกในการพัฒนาต่อโดยไม่ว่าจะใช้ภาษาใดในการพัฒนาเว็บเพื่อให้บริการก็จะรองรับกับ MY SQL ทั้งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ดอนสัน ปงผาบ. 2549. **ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน 2.** พิมพ์ครั้งที่ 1. กรุงเทพฯ :บริษัท ซีเอ็ดยูเคชั่น จำกัด

ปริญญานิพนธ์หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. **การจำลองกอล์ฟเสมือนจริง.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

Source code Application

```
Imports System
Imports System.IO
Imports System.IO.Ports
Imports System.Threading
Imports MySql.Data.MySqlClient


Public Class Form1

    '============= Open DB Connection ====================
    Dim conn As MySqlConnection
    Dim data As DataTable
    Dim da As MySqlDataAdapter
    Dim cb As MySqlCommandBuilder
    Dim count As Integer = 0



    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        If Not conn Is Nothing Then conn.Close()

        Dim connStr As String
        connStr = String.Format("server={0};user id={1};
password={2}; database=mysql; pooling=false", _
        "127.0.0.1", "root", "admin")

        Label5.Text = "1A Unknown"
        Label6.Text = "1B Unknown"
        Label7.Text = "1C Unknown"
        Label8.Text = "1D Unknown"
        Label9.Text = "2D Unknown"
        Label10.Text = "2C Unknown"
        Label11.Text = "2B Unknown"
        Label12.Text = "2A Unknown"
        Try
            conn = New MySqlConnection(connStr)
            conn.Open()
            conn.ChangeDatabase("parking")

        Catch ex As MySqlException
            MessageBox.Show("Error connecting to the server: " +
ex.Message)
        End Try
        '============== Close DB Connection ====================
    End Sub



    '================ Click Run ============================
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If count = 1 Then
            SerialPort1.Open()
            Label2.Text = "Running"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            count = count - 1
        Else
            Exit Sub
        End If

    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        If count = 0 Then
            SerialPort1.Close()
            Label2.Text = "Stopping"
            Label16.Text = ""
            count = count + 1
            Label5.Text = "1A Unknown"
            Label6.Text = "1B Unknown"
            Label7.Text = "1C Unknown"
            Label8.Text = "1D Unknown"
            Label9.Text = "2D Unknown"
            Label10.Text = "2C Unknown"
            Label11.Text = "2B Unknown"
            Label12.Text = "2A Unknown"
            Label17.Text = "Floor 1"
            Label18.Text = "Floor 2"
        Else
            Exit Sub
        End If

    End Sub

    Private Sub SerialPort1_DataReceived(ByVal sender As
System.Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
        CheckForIllegalCrossThreadCalls = False
        Dim Buffer As String
        Buffer = Chr(SerialPort1.ReadChar)
        Label16.Text = Buffer

        '================ Check Status
================================

        If Buffer = "A" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1A'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())


            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand


            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1A'", conn)    'update สถานะที่จอดรถ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label5.Text = "1A   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label5.Text = "1A   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                Dim com5 As MySqlCommand = New MySqlCommand("UPDATE
floor SET have = " & Have & " where floor = '1'", conn)   'update จำนวนที่
จอดรถในตาราง
                com5.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "B" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1A'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand


            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่เรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label5.Text = "1A   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                Dim com5 As MySqlCommand = New MySqlCommand("UPDATE
floor SET have = " & Have & " where floor = '1'", conn)   'update จำนวนที่
จอดรถในตาราง
                com5.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่เรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label5.Text = "1A   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If

        ElseIf Buffer = "C" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1B'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label6.Text = "1B   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label6.Text = "1B   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""

                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "D" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1B'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Label6.Text = "1B   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1B'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label6.Text = "1B   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "E" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1C'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1C'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label7.Text = "1C   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1C'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label7.Text = "1C   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "F" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1C'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label7.Text = "1C   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)   'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label7.Text = "1C   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "G" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1D'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1D'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label8.Text = "1D   Vacancy"
                Label17.Text = "Floor 1 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '1D'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label8.Text = "1D   Vacancy"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "H" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '1D'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '1'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1D'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label8.Text = "1D   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '1'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '1D'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label8.Text = "1D   Parked"
                Label17.Text = "Floor 1 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "I" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2A'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2A'", conn)   'update สถานะที่จอดรถ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label12.Text = "2A   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label12.Text = "2A   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "J" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2A'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label12.Text = "2A   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2A'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label12.Text = "2A   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "K" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2B'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label11.Text = "2B   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label11.Text = "2B   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)   'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "L" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2B'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label11.Text = "2B   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)   'update จำนวนที่จอดรถในตาราง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2B'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label11.Text = "2B   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "M" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2C'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label10.Text = "2C   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label10.Text = "2C   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)   'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "N" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2C'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label10.Text = "2C   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)   'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2C'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label10.Text = "2C   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
            End If

        ElseIf Buffer = "O" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2D'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2D'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label9.Text = "2D   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""

            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Vacancy' where id = '2D'", conn)    'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have + 1
'เพิ่มที่ว่างของที่จอดรถ
                Label9.Text = "2D   Vacancy"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)   'update จำนวนที่จอดรถในตาราง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            End If

        ElseIf Buffer = "P" Then 'ตรวจสอบค่าที่รับมาจากบอร์ดไมโครว่าใช่ค่านี้หรือไม่

            Dim com1 As MySqlCommand = New MySqlCommand("SELECT
status FROM status WHERE id = '2D'", conn)
            Dim Strstatus As String = CStr(com1.ExecuteScalar())

            Dim com2 As MySqlCommand = New MySqlCommand("SELECT have
FROM floor WHERE floor = '2'", conn)
            Dim Have As Integer = CInt(com2.ExecuteScalar())
            Dim com As MySqlCommand

            If Strstatus = "Vacancy" Then
'ตรวจสอบว่าค่าเป็น Vacancy ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2D'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Have = Have - 1
'เพิ่มที่ว่างของที่จอดรถ
                Label9.Text = "2D   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
                com = New MySqlCommand("UPDATE floor SET have = " &
Have & " where floor = '2'", conn)  'update จำนวนที่จอดรถในตาราง
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
            ElseIf Strstatus = "Parked" Then
'ตรวจสอบว่าค่าเป็น Parked ใช่หรือไม่
                com = New MySqlCommand("UPDATE status SET status =
'Parked' where id = '2D'", conn)   'update สถานะที่จอดรถ
                com.ExecuteNonQuery()
'ทำการเรียกใช้คำสั่ง
                Label9.Text = "2D   Parked"
                Label18.Text = "Floor 2 have " & Have.ToString & ""
            End If
        End If


    End Sub

End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

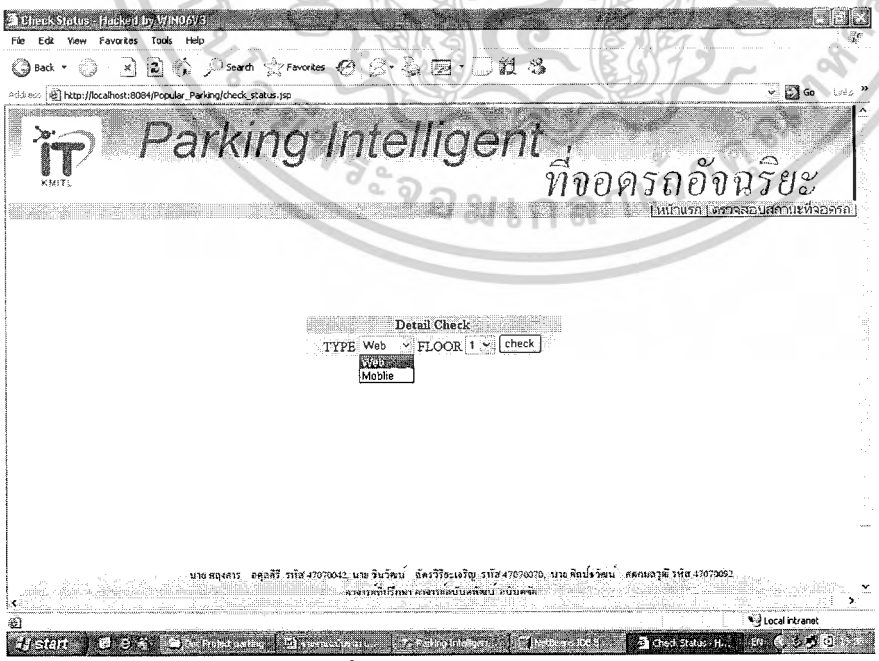
วิธีการใช้งานเว็บแอปพลิเคชั่น

หน้าแรกของเว็บแอปพลิเคชั่น

ตรวจสอบสถานะที่จอด

รูปที่ A.1 หน้าแรกของเว็บแอปพลิเคชั่น

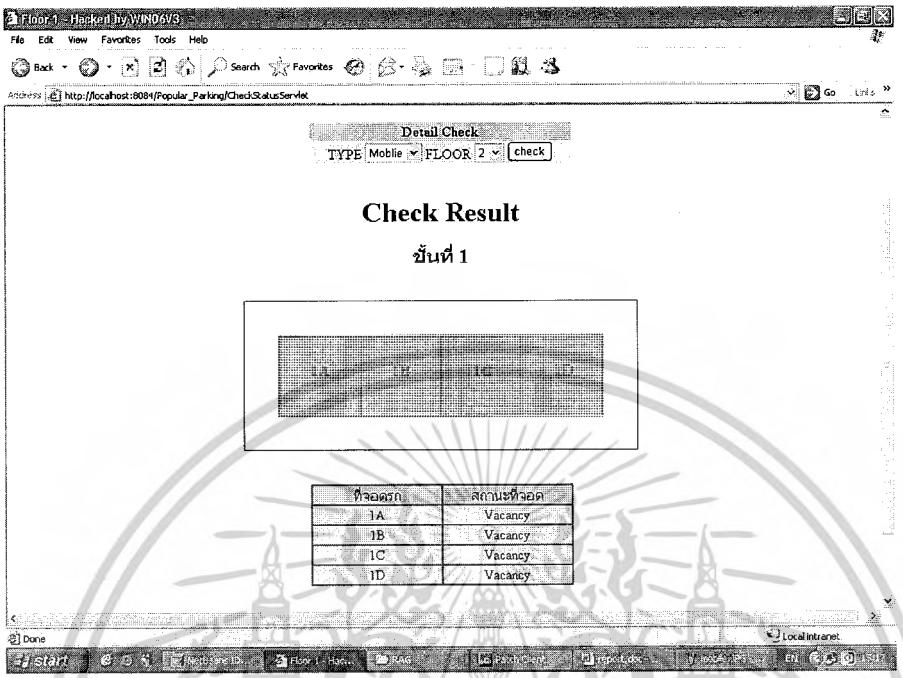
เลือกตรวจสอบสถานะที่จอดรถ



รูปที่ A.2 เลือกรูปแบบการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตรวจสอบแบบเว็ป



รูปที่ A.3 รูปแบบการตรวจสอบแบบเว็ป

การตรวจสอบแบบมือถือ

รูปที่ A.4 รูปแบบการตรวจสอบแบบมือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน นาย ศิลป์ธวัฒน์ ศตกมลวุฒิ

วันเดือนปีเกิด 23 กรกฎาคม 2529

สถานที่เกิด กรุงเทพมหานคร

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนเศรษฐบุตรบำเพ็ญ ปีการศึกษา 2544

ศึกษาระดับปริญญาตรี วิทยาศาสตรบัณฑิตสาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

ชื่อผู้เขียน นาย ชินวัฒน์ ฉัตรวิริยะเจริญ

วันเดือนปีเกิด 19 ตุลาคม 2529

สถานที่เกิด กรุงเทพมหานคร

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนนวมินทราชินูทิศ บดินทรเดชา ปีการศึกษา 2544

ศึกษาระดับปริญญาตรี วิทยาศาสตรบัณฑิตสาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

ชื่อผู้เขียน นาย ศฤงคาร อดุลศิริ

วันเดือนปีเกิด 3 กรกฎาคม 2527

สถานที่เกิด ราชบุรี

ประวัติการศึกษา

ศึกษาระดับมัธยมศึกษาตอนปลาย โรงเรียนพระปฐมวิทยาลัย ปีการศึกษา 2544

ศึกษาระดับปริญญาตรี วิทยาศาสตรบัณฑิตสาขาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้