สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

เครื่องจัดคิวอัตโนมัติ

Auto Queuing Machine

โดย

นาย ธวัชชัย และหมัน
นาย ธัญญา ลิขิตวาศ
นาย นครินทร์ เรืองศรี

ว.พ.
ส395ค
2547

เลขหมู่.....
เลขทะเบียน.....61350
วัน,เดือน,ปี...17 ก.ค. 2549

b.....11515413.....
i.....

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจัดคิวอัตโนมัติ

Auto Queuing Machine

โดย

นาย ธวัชชัย	และหมัน	เลขประจำตัว 44010209
นาย ธัญญา	ลิขิตวาศ	เลขประจำตัว 44010210
นาย นครินทร์	เรืองศรี	เลขประจำตัว 44010233

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงานเรื่อง	เครื่องจัดคิวอัตโนมัติ			
	Auto Queuing Machine			
จัดทำโดย	นายธวัชชัย	และหมัน	รหัส	44010209
	นายธัญญา	ลิขิตวาศ	รหัส	44010210
	นายนครินทร์	เรืองศรี	รหัส	44010233
อาจารย์ที่ปรึกษา	ผศ. พลผดุง	ผดุงกุล		

รายงานฉบับนี้ได้ผ่านการตรวจสอบจากอาจารย์ที่ปรึกษาแล้ว

ลงชื่อ.....อาจารย์ที่ปรึกษา

(ผศ. พลผดุง ผดุงกุล)

วันที่...../...../.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจัดลำดับคิวอัตโนมัติ

นายธวัชชัย และหมัน
นายธัญญา ลิขิตวาศ
นายนครินทร์ เรืองศรี
ผศ. พลผดุง ผดุงกุล (อาจารย์ ที่ปรึกษา)
ภาคการศึกษาที่ 2 ปีการศึกษาที่ 2547

บทคัดย่อ

โครงงานนี้เป็นโครงงานที่เกี่ยวกับการให้บริการแก่ลูกค้าหรือบุคคลที่มาใช้บริการขององค์กร โดยสร้างขึ้นเพื่อมีวัตถุประสงค์ให้มีความถูกต้องและรวดเร็วทำให้ผู้ที่มาใช้บริการรู้สึกประทับใจ โดยโครงงานนี้จะใช้การแบ่งหน้าที่การให้บริการออกเป็นหน่วยย่อยๆหลายหน่วยประกอบด้วย หน่วยประมวลผลหลัก ช่องรับบริการ ส่วนแสดงผลหลักเป็นตัวเลข ส่วนกระจายเสียง และส่วนรอรับการเลือกอินพุทจากผู้ที่มาใช้บริการหรืออาจเรียกว่าส่วนแจกหมายเลข โดยที่เริ่มแรกผู้ที่มาใช้บริการจะมาเลือกรูปแบบการให้บริการจากส่วนรอรับการเลือกอินพุทซึ่งจะได้รับหมายเลขคิวของตนเอง จากนั้นหน่วยประมวลผลหลักก็จะรอการขอข้อมูลของช่องรับบริการเพื่อที่จะให้บริการ หน่วยประมวลผลหลักก็จะส่งข้อมูลของผู้ที่มาใช้บริการไปให้โดยเรียงตามลำดับ หลังจากนั้นก็จะส่งข้อมูลไปให้ส่วนแสดงผลหลักและส่วนกระจายเสียงตามลำดับ เมื่อข้อมูลถูกนำไปประมวลผลเรียบร้อยก็จะมีหมายเลขผู้ใช้บริการที่พร้อมให้บริการและหมายเลขของช่องรับบริการแสดงที่ส่วนแสดงผลหลักพร้อมทั้งมีการขานเรียกจากส่วนกระจายเสียงด้วย และจะมีการให้บริการในแบบเดียวกันนี้กับผู้ที่มาใช้บริการทุกราย นอกจากนี้จะมีการเชื่อมต่อกับคอมพิวเตอร์เพื่อเรียกดูหรือตั้งค่าต่างๆของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto Queuing Machine

Thawatchai Lehman

Thunya Likhitwas

Nakarin Ruangsri

Asst . Prof . Polphadung Phadungkul (Advisor)

2nd Semester Academic Year 2004

Abstract

This project is to develop a system that provides a service to customers of an organization. It aims to impress the customers by convenience and quickness. This project is divided to many small units such as master processor unit, counter service unit, main display unit, speaker unit and queue distribution unit. First, customer will choose a service type at queue distribution unit and it will give him/her a number. If there is a vacant counter, master will send this data to the vacant counter and also send it to main display unit and speaker unit as well. After all communications have finished and are ready to serve the customer, the main display will show the number of customers and the number of counters which are ready to serve and the speaker unit will call the customer to deal at the counter. This is all of the service cycle for serving customers. Furthermore, this system provides interfacing with computer to read data from the system and to set initial data to system.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่โครงงานชิ้นนี้สามารถสำเร็จลุล่วงได้ด้วยดีนั้น ทางผู้จัดทำขอขอบคุณอาจารย์ที่ปรึกษา อาจารย์พลผดุง ผดุงกุลที่ได้ให้แนวคิด คำปรึกษาและความช่วยเหลือต่างๆในการดำเนินงานตลอดจน สนับสนุนอุปกรณ์ในการสร้างโครงงาน ขอขอบคุณรุ่นพี่ เพื่อนๆทุกคนที่ได้ให้ความช่วยเหลืออย่างดี ในด้านต่างๆ และขอขอบคุณ บิดา มารดา รวมทั้งผู้มีพระคุณทุกท่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
บทที่ 1บทนำ	1
1.1 ความเป็นมาและโครงสร้างของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ	1
1.5 ข้อกำหนดของโครงงาน	1
1.6 รายละเอียดและหน้าที่ของส่วนประกอบของโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการพื้นฐาน	5
2.1 การเชื่อมต่อไมโครคอนโทรเลอร์แบบ Multiprocessor	5
2.2 ไอซีสร้างฐานเวลา (Real time Clock : RTC) DS 1307	7
2.3 IC บันทึกเสียง (ISD 2590)	11
2.4 โมดูล LCD	14
2.5 การเชื่อมต่อโดยใช้มาตรฐาน RS-485	18
2.6 ระบบบัสแบบ I^2C	19
2.7 การเขียนโปรแกรมสื่อสารแบบอนุกรม	21
บทที่ 3 การออกแบบ	29
3.1 การติดต่อแบบ Multiprocessor	33
3.2 หน่วยประมวลผลหลัก (Master Controller)	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
3.3 หน่วยประมวลผลส่วนให้บริการ (Counter Service Unit)	34
3.4 ส่วนแจกคิว (Queue Distribution Unit)	37
3.5 ส่วนแสดงผลหลัก (Main Display)	39
3.6 ส่วนควบคุมการการกระจายเสียง (Speaker Unit)	41
3.7 การเชื่อมต่อส่วนประมวลผลกลางไปยังคอมพิวเตอร์ผ่าน USB Port	44
บทที่ 4 ผลการทดสอบ	46
บทที่ 5 สรุป ประเมิน และวิจารณ์ผลการทำโครงงาน	54
หนังสืออ้างอิง	56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 1.1 แสดงการเชื่อมต่อของอุปกรณ์ทั้งหมด	4
รูปที่ 2.1 แสดงลักษณะของRegister SCON	5
รูปที่ 2.2 การสื่อสารระบบ Multiprocessor	7
รูปที่ 2.3 ไอซีสร้างฐานเวลาจริง(RTC) DS1307	8
รูปที่ 2.4 โครงสร้างภายในของ IC RTC เบอร์ DS1307	9
รูปที่ 2.5 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล	10
รูปที่ 2.6 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล	11
รูปที่ 2.7 LCD เบอร์ MTC-S16206	14
รูปที่ 2.8 แสดงคำสั่งเลือกโหมดการป้อนข้อมูล	15
รูปที่ 2.9 แสดงคำสั่งควบคุมการแสดงผล	16
รูปที่ 2.10 แสดงคำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร	16
รูปที่ 2.11 แสดงคำสั่งกำหนดฟังก์ชันการทำงาน	16
รูปที่ 2.12 แสดงคำสั่งอ่านFlag Busy และAddress	17
รูปที่ 3.1 เครือข่ายของไมโครคอนโทรเลอร์ที่เชื่อต่อกันหลายตัว	29
รูปที่ 3.2 ไบต์กำหนดแอดเดรส (Address byte)	30
รูปที่ 3.3 ไบต์กำหนดข้อมูล (Data byte)	30
รูปที่ 3.4 การเชื่อมต่อไมโครคอนโทรเลอร์เข้ากับ SN 75176	31
รูปที่ 3.5 แสดง Protocol ที่ใช้ในการสื่อสาร	32-33
รูปที่ 3.6 วงหน่วยประมวลผลหลัก	34
รูปที่ 3.7 วงจรที่ช่องรับบริการ	36
รูปที่ 3.8 วงจรหน่วยประมวลผลปุ่มเรียกบัตรคิว	38
รูปที่ 3.9 วงจรส่วนแสดงผลหลัก	40
รูปที่ 3.10 การต่อวงจรสำหรับการบันทึกเสียง	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 3.11 วงจรการเชื่อมต่อกับ ISD 2590 สำหรับการใช้งานจริง	43
รูปที่ 3.12 รูปแบบวงจรการเชื่อมต่อ USB	44
รูปที่ 4.1 การเชื่อมต่อของระบบ	47
รูปที่ 4.2 การทำงานของส่วนแจกคิว	48
รูปที่ 4.3 การทำงานของส่วนรับบริการ	49
รูปที่ 4.4 แสดงการรับข้อมูลจากระบบของคอมพิวเตอร์	50
รูปที่ 4.5 แสดงผลการเก็บค่าข้อมูลที่ได้ไปไว้ที่ฐานข้อมูล	51
รูปที่ 4.6 แสดงการเรียกดูค่าเริ่มต้นของระบบผ่านคอมพิวเตอร์	52
รูปที่ 4.7 แสดงค่าเริ่มต้นของระบบที่ถูกตั้งค่าใหม่แล้ว	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงงาน

ปัจจุบันมีองค์กรทั้งภาครัฐและภาคเอกชนที่ต้องมีการให้บริการแก่ประชาชนหรือลูกค้าที่มาใช้บริการดังนั้นการให้บริการที่สะดวกสบายและรวดเร็วเพื่อให้ผู้ที่มาใช้บริการเกิดความพึงพอใจจึงเป็นสิ่งหนึ่งที่มีความสำคัญอย่างมากในธุรกิจ อีกทั้งยังทำให้เกิดความเป็นระเบียบเรียบร้อยและถูกต้องชัดเจน จึงได้มีการพัฒนาโครงงานชิ้นหนึ่งที่จะช่วยเพิ่มประสิทธิภาพขององค์กร และยังช่วยเพิ่มประสิทธิภาพของพนักงานนั่นก็คือโครงงานเครื่องจัดคิวอัตโนมัติ โดยเป็นอุปกรณ์ที่จะทำหน้าที่แจกคิวให้กับผู้ที่มาใช้บริการ เรียกคิวตามที่แจกไปและยังสามารถบันทึกผลการปฏิบัติงานของพนักงานเพื่อนำไปใช้ในการปรับปรุงการให้บริการให้ดียิ่งขึ้นไปได้อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการใช้งานไมโครคอนโทรเลอร์ที่ทำงานแบบMultiprocessor
2. เพื่อศึกษาการออกแบบวงจรดิจิตอลที่ใช้งานร่วมกับไมโครคอนโทรเลอร์
3. เพื่อศึกษาการเขียนโปรแกรมโดยใช้ภาษาซี
4. เพื่อศึกษาการใช้งานอุปกรณ์ร่วมต่างๆเช่น LCD , RS-485 และ ISD 2590
5. เพื่อพัฒนาระบบที่มีอยู่เดิมให้มีคุณภาพดียิ่งขึ้น

1.3 ขอบเขตของโครงงาน

1. ระบบสามารถทำงานร่วมกันเป็นแบบ Multiprocessor ได้
2. สามารถให้บริการผู้ที่มาใช้บริการได้ไม่จำกัดจำนวนเนื่องจากใช้การวนกลับมาใช้หมายเลขเดิมที่ใช้บริการเสร็จไปแล้ว
3. สามารถเพิ่มจำนวนช่องรับบริการได้ตามความต้องการ
4. สามารถเพิ่มปุ่มกดเพื่อเพิ่มจำนวนหน้าที่ของการให้บริการ
5. สามารถเรียกดูข้อมูลของระบบผ่านทางคอมพิวเตอร์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถใช้ไมโครคอนโทรเลอร์ร่วมกับอุปกรณ์อื่นๆได้
2. สามารถออกแบบวงจรดิจิตอลที่ใช้งานจริงได้
3. สามารถนำไปใช้งานจริงได้ตามองค์กรทั่วไปที่ต้องการการจัดลำดับก่อนหลังของผู้ที่มาใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ข้อกำหนดของโครงงาน

1. มีระบบที่ยืดหยุ่นสามารถเพิ่มหรือลดจำนวนหน่วยปฏิบัติการบางหน่วยได้เช่นช่องรับบริการ
2. มีการแสดงผลที่ชัดเจนและเข้าใจง่าย
3. มีความถูกต้องและแน่นอน

1.6 รายละเอียดและหน้าที่ของส่วนประกอบของโครงงาน

1. หน่วยประมวลผลหลัก (Master Controller) เป็นส่วนที่ควบคุมการทำงานทั้งหมด โดยมีหน้าที่คือ

- รับข้อมูลการแจกคิวจากอุปกรณ์ที่ทำหน้าที่แจกลำดับคิวให้กับผู้ที่มาใช้บริการ
- แจกคิวไปยังช่องให้ส่วนให้บริการ (Counter Service) และควบคุมการติดต่อสื่อสารกับส่วนให้บริการทุกๆตัว
- ส่งข้อมูลตัวเลขที่จะให้แสดงออกเป็นเสียง ไปให้กับตัวประมวลผลของส่วนกระจายเสียงและควบคุมการทำงานของส่วนกระจายเสียง ให้ทำงานตามที่กำหนด
- ส่งคิวที่จะเรียกไปให้กับตัวประมวลผลของส่วนแสดงผลหลักเพื่อแสดงลำดับคิวที่ต้องการเรียก

2. หน่วยแจกคิวให้กับผู้ที่มาใช้บริการ (Queue Controller) ทำหน้าที่ในการรับอินพุทจากการกดของผู้ที่มาใช้บริการและแจกคิวตามปุ่มที่กด

- รอรับการกดปุ่มจากผู้ที่มาใช้บริการโดยที่มีปุ่มกด 2 ปุ่มคือหน้าที่ที่ 1 มีลำดับคิวที่ 1-100 และหน้าที่ที่ 2 มีลำดับคิวที่ 101-200
- แจกหมายเลขลำดับคิวให้กับผู้ที่มาใช้บริการ

3. ส่วนให้บริการ (Counter Service) ทำหน้าที่ในการควบคุมการเรียกคิว การข้ามคิวและการเรียกคิวซ้ำโดยจะส่งคำสั่งต่างๆเช่นการขอคิวที่จะต้องให้บริการต่อไปไปที่หน่วยประมวลผลหลัก

- สามารถเลือกรับ Officer ID จากการใช้ Bar Code หรือจากการกด Key Pad ก็ได้
- มีจอแสดงผลแบบ LCD
- แสดงหมายเลขที่เรียกเพื่อให้บริการ
- สามารถเรียกหมายเลขได้ตั้งแต่ 1-999 แล้วกลับมาเริ่มที่ 1 ใหม่
- สามารถข้ามหมายเลขที่เรียกแล้วยังไม่มาใช้บริการได้
- สามารถเรียกซ้ำหมายเลขเดิมที่เรียกไปแล้วแต่ยังไม่มาใช้บริการ
- สามารถยกเลิกหมายเลขที่ไม่มาใช้บริการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน่วยควบคุมการแสดงผลหลัก (Main Display Unit) ทำหน้าที่ในการควบคุมการแสดงผลผ่านบอร์ดแสดงผลหลัก โดยจะรับหมายเลขคิวที่ต้องการแสดงมาจากหน่วยประมวลผลหลักแล้วนำมาประมวลผลก่อนที่จะสั่งให้บอร์ดแสดงผลหลักแสดงหมายเลขที่ต้องการ

- มีบอร์ดแสดงหมายเลขคิวหลัก (Main Display) ทำหน้าที่ในการแสดงผลหมายเลขคิวของผู้ที่มาใช้บริการและหมายเลขของช่องให้บริการ (Counter) ที่พร้อมจะให้การบริการ
- มี 7-Segment จำนวน 5 หลักจำนวน 2 แถวซึ่งประกอบไปด้วยหมายเลขของผู้ที่มาใช้บริการ 3 หลัก และหมายเลขของช่องให้บริการ (Counter) 2 หลักโดยที่แถวแรกจะแสดงลำดับคิวล่าสุดที่กำลังเรียกและแถวที่สองจะแสดงลำดับคิวก่อนหน้าที่เรียก
- มี 7-Segment จำนวน 3 หลักจำนวน 1 แถวทำหน้าที่ในการแสดงจำนวนผู้ที่มาใช้บริการที่กำลังรอคิวอยู่

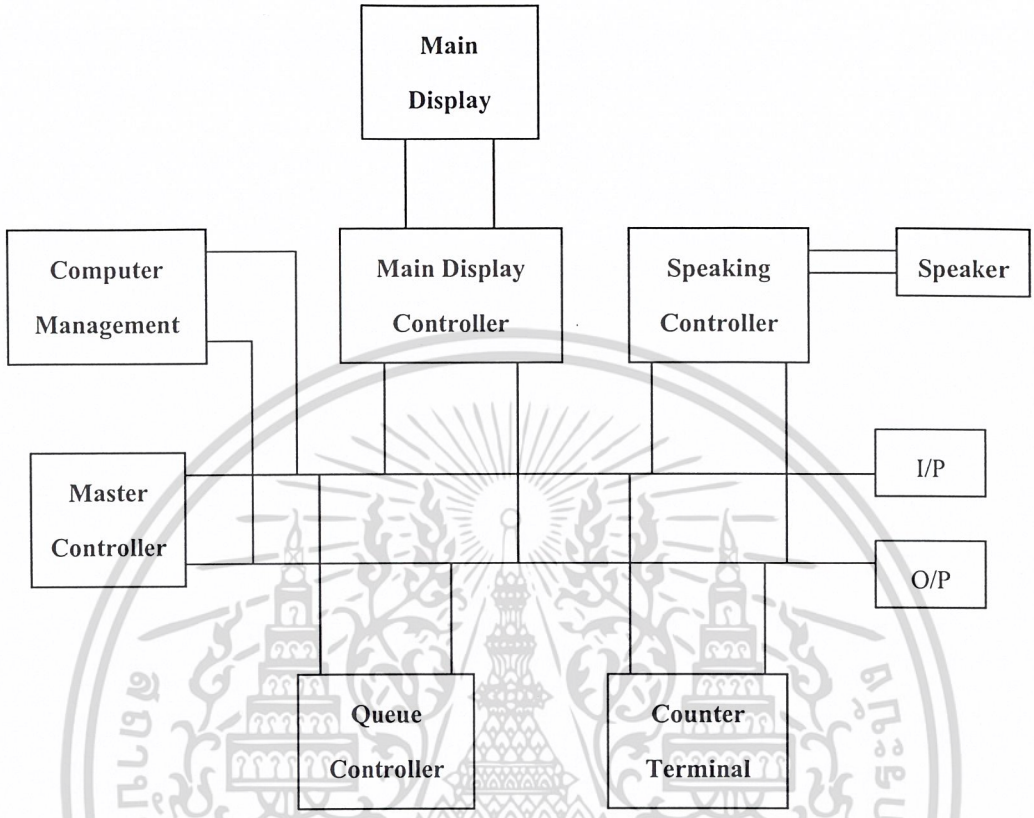
5. หน่วยประมวลผลเสียงพูด (Speaking Controller) ทำหน้าที่ควบคุม IC ที่ทำหน้าที่ในการส่งเสียงเรียกหมายเลขที่พร้อมจะให้บริการ โดยจะรับหมายเลขที่ต้องการเรียกมาจากหน่วยประมวลผลหลัก

- ชุดกระจายเสียงเรียก (Speaker) เป็นชุดอุปกรณ์ที่ทำหน้าที่เรียกหมายเลขที่พร้อมให้บริการตามที่หน่วยประมวลผลเสียงพูดสั่งมาโดยที่มีอุปกรณ์หลักคือ ISD 2590
- การเรียกจะบอกหมายเลขคิวที่พร้อมให้บริการและบอกหมายเลขของช่องให้บริการที่พร้อมให้บริการ

6. คอมพิวเตอร์ (Computer) ติดต่อสื่อสารกับหน่วยประมวลผลหลักผ่านพอร์ตอนุกรม

- นำข้อมูลการทำงานของระบบมาเก็บใน Database ของคอมพิวเตอร์ได้ โดยเก็บเป็น Database ของโปรแกรม Microsoft Access
- สามารถตั้งค่าช่วงของหน้าที่ของการให้บริการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงการเชื่อมต่อของอุปกรณ์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการพื้นฐาน

2.1 การเชื่อมต่อไมโครคอนโทรเลอร์แบบ Multiprocessor

การเชื่อมต่อไมโครคอนโทรเลอร์เพื่อการรับส่งข้อมูลแบบอนุกรมมีอยู่ 2 แบบด้วยกันคือ

1. Single Processor Mode ใช้ในการเชื่อมต่อกันระหว่างไมโครคอนโทรเลอร์ 2 ตัวเข้าหากัน
2. Multiprocessor Mode ในการใช้งานโหมดนี้จะใช้ไมโครคอนโทรเลอร์ 1 ตัวเป็นตัวแม่ (Master) ส่วนตัวที่มาต่อที่เหลือจะเป็นตัวลูก (Slave)

ส่วน Register ที่ใช้ในการตั้งค่าการเชื่อมต่อการรับส่งข้อมูลแบบอนุกรมคือ Serial Control Port Register (SCON) เป็น Register ที่อยู่ใน Special Function Register ตำแหน่งที่ 98H

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

รูปที่ 2.1 แสดงลักษณะของRegister SCON

SM1	SM0	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วเท่ากับ 1/12 เท่าของ Oscillator ที่จ่ายให้กับไมโครคอนโทรเลอร์
0	1	1	8 บิต UART ความเร็วในการรับส่งข้อมูลสามารถกำหนดได้จาก Timer 0 และ Timer 1
1	0	2	9 บิต UART ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 เท่า หรือ 1/64 เท่าของ Oscillator 1ที่จ่ายให้กับไมโครคอนโทรเลอร์
1	1	3	9 บิต UART ความเร็วในการรับส่งข้อมูลสามารถกำหนดได้จาก Timer 0 และ Timer1

ตารางที่ 2.1 แสดงรายละเอียดการทำงานในโหมดต่างๆของการติดต่อแบบอนุกรม

SM2 คือบิตที่ใช้ในการเลือกการทำงานแบบ Single Processor Environment หรือแบบ Multiprocessor โดยที่ถ้า SM2 เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 : เลือกใช้ Single Processor Environment ใช้ได้กับทุกโหมดการใช้งาน

1 : เลือกใช้ Multiprocessor ใช้ได้กับโหมด 2 และ 3 เท่านั้น

เมื่อเลือกทำงานแบบ Multiprocessor แล้วถ้าข้อมูลบิตที่ 9 ที่ได้รับมีค่าเป็น 1 จะทำให้บิต RI Set แต่ถ้าบิตที่ 9 ที่ได้รับเป็น 0 RI จะไม่ Set

REN (Receive Enable) เป็นบิตควบคุมให้รับหรือไม่รับข้อมูล

0 : ไม่รับข้อมูลที่ส่งเข้ามา

1 : รับข้อมูลที่ส่งเข้ามาได้

TB8 (Transmit bit D8) ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2,3 ให้ใส่บิตนี้

RB8 (Receive Bit D8) ข้อมูลที่บิต 9 ที่รับเข้ามาจะเก็บในบิตนี้

TI แฟลกซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI แฟลกซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์

สมการคำนวณ Boud Rate Mode 1,3 = $\frac{2^{SMOD} \times CPUoscillator}{32 \times 12 \times [256 - (TH1)]}$

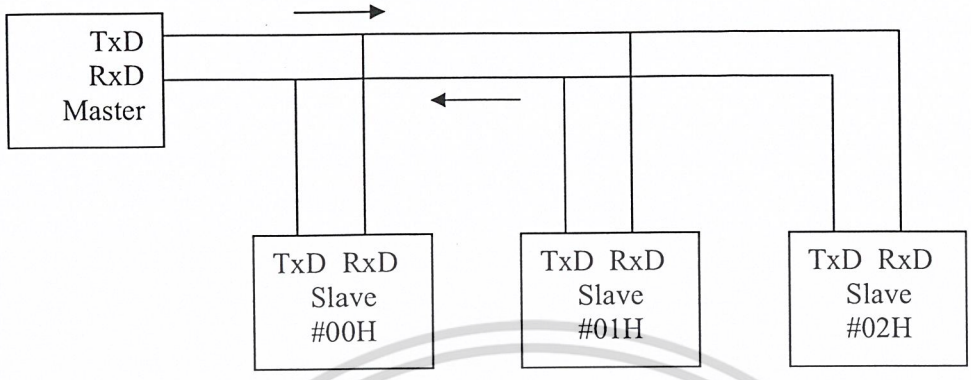
ในการใช้งานพอร์ทสื่อสารอนุกรมโดยใช้หลักการของโปรเซสเซอร์ มี 2 รูปแบบ คือ

- ตัวแม่ 1 ตัว ส่งข้อมูลไปให้ตัวลูกได้หลายตัว
- ตัวลูกหลายตัวส่งข้อมูลไปให้ตัวแม่ 1 ตัว

ตัวแม่ 1 ตัวส่งข้อมูลไปให้ตัวลูกได้หลายตัว

จะใช้ตัวแม่ 1 ตัวส่งข้อมูลไปให้ตัวลูก โดยตัวลูกจะต้องมีรหัสประจำตัว เช่น 00H, 01H , 02H,...,FFH) รูปแบบการส่งจะต้องส่ง Address byte ออกไปก่อนตามด้วย data byte จำนวนกี่ไบต์ก็ได้ ตามรูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การสื่อสารระบบ Multiprocessor

ข้อแตกต่างระหว่าง Address byte กับ data byte อยู่ที่บิต TB8

ถ้า TB8 = 1 หมายถึง Address byte

TB8 = 0 หมายถึง Data byte

ข้อแตกต่างระหว่าง Single Processor Mode กับ Multiprocessor Mode คือ

- Single Processor Mode

RI จะเท่ากับ 1 เมื่อรับข้อมูลเสร็จ 1 byte

- Multiprocessor Mode

RI จะเท่ากับ 1 เมื่อรับข้อมูลเสร็จ 1 byte และค่าในบิต TB8 ทางด้านส่งเมื่อเข้ามายังภาครับ (RB8) จะต้องเท่ากับ 1 เท่านั้น

2.2 ไอซีสร้างฐานเวลา (Real time Clock : RTC) DS 1307

2.2.1 คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซี RTC ให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรธินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100

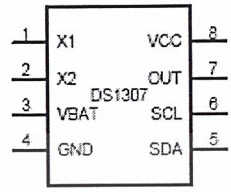
- มีหน่วยความจำ Non Volatile RAM 56 ไบต์อยู่ภายใน สามารถเก็บข้อมูลทั่วไปได้

- ใช้เชื่อมต่อแบบบัส I^2C

- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 รายละเอียดขาต่อใช้งานของ DS1307



รูปที่ 2.3 ไอซีสร้างฐานเวลาจริง(RTC) DS1307

ในรูปที่ 2.3 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้

V_{CC},GND ต่อกับไฟเลี้ยง +5V

V_{BAT} ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไปแม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสม คือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr. หรือมากกว่าจะสามารถรักษาข้อมูลได้นาน 10 ปีที่ 25 องศาเซลเซียส

SDA , SCL เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

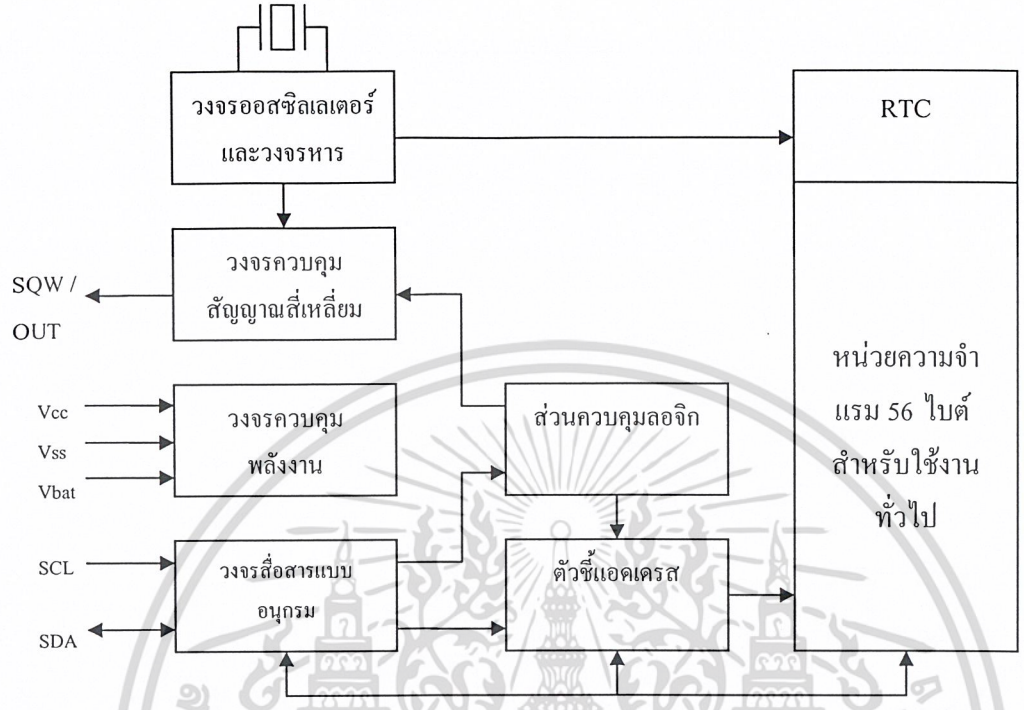
SQW OUT ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1 Hz , 4.096 kHz , 8.192 kHz และ 32 kHz

X1 , X2 ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อใช้เป็นฐานเวลาในการสร้างคาบเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุด้วย

2.2.3 การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์ Slave เสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบที่กำหนดไว้ในการติดต่อแบบ I^2C ในรูปที่ 2.4 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุมค่าความถี่ของสัญญาณนี้สามารถเลือกค่าได้ 4 ค่าคือ 1 Hz , 4.096 kHz , 8.192 kHz และ 32 kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำNon Volatile RAM ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 โครงสร้างภายในของ IC RTC เบอร์ DS1307

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะไฟเลี้ยงของไอซี หากไฟเลี้ยงต่ำกว่า $1.25V_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดทำงาน รีเซตค่าตัวนับแอดเดรสภายใน ทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งาน DS1307 ต้องระมัดระวังอย่าให้ไฟเลี้ยงตกต่ำกว่า $1.25V_{BAT}$ ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรวงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัสI^2C

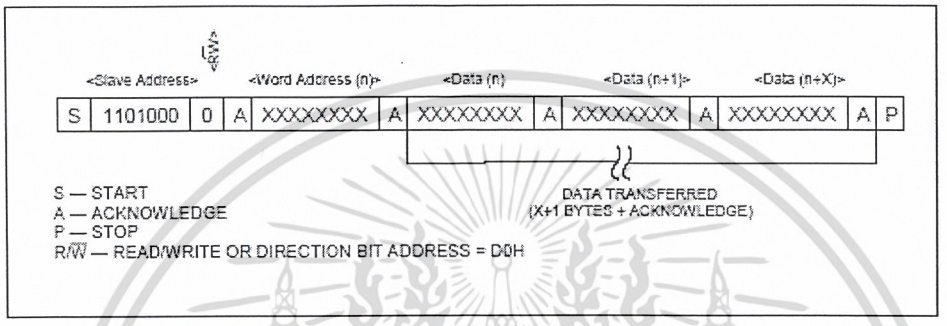
2.2.4 โหมดการทำงานของ DS1307

มีด้วยกัน 2 โหมด โหมดเขียนข้อมูลและโหมดอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูล

2.2.5 โหมดการเขียนข้อมูล

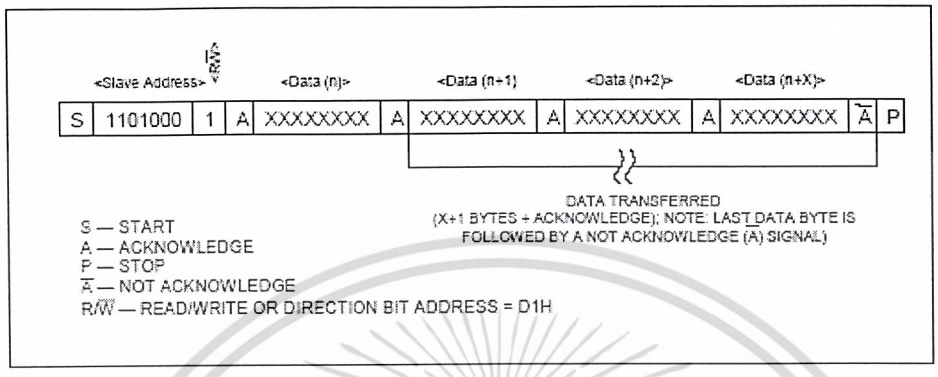


รูปที่ 2.5 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล

มีรูปแบบดังในรูปที่ 2.5 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (START: S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นก็จะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับกลับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลแต่ละแอดเดรสจะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้ง จึงสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนข้อมูลเรียบร้อยแล้วให้ส่งสภาวะหยุด(STOP:S) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 โหมดการอ่านข้อมูล



รูปที่ 2.6 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

มีรูปแบบแสดงดังในรูปที่ 2.6 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่าย ๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะทีจะต้องเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนล่วงหน้า

2.3 IC บันทึกเสียง (ISD 2590)

ISD 2590 เป็น IC สำหรับการบันทึกและเล่นเสียงที่ได้บันทึกไว้ โดยที่สามารถบันทึกและเล่นได้เป็นเวลา 90 วินาที IC ชนิดนี้สร้างโดยใช้ CMOS เทคโนโลยีซึ่งประกอบไปด้วยแหล่งกำเนิดความถี่ (Oscillator)ภายใน ตัวกรองความถี่ วงจรขยายเสียง วงจรกรองที่ช่วยให้เสียงที่ได้มีความเรียบ ระบบการตัดเสียงรบกวน วงจรควบคุมอัตราขยายอัตโนมัติ และหน่วยความจำแบบ Flash ที่สามารถบันทึกได้เป็นหลักพันครั้ง เสียงที่เก็บจะเป็นการเก็บโดยตรงไปที่หน่วยความจำและไม่ต้องใช้พลังงานในการเก็บรักษาข้อมูล อีกทั้งยังสามารถเก็บได้เป็นเวลานานมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณลักษณะของ IC บันทึกเสียง

- สามารถบันทึกและเล่นเสียงได้จาก IC ตัวเดียวกัน
- คุณภาพเสียงสูงเพราะเก็บเสียงตามที่ได้รับเลย ไม่มีการบีบอัด
- สามารถบันทึกเสียงได้ 90 วินาที
- สามารถบันทึกเสียงได้ต่อกันเรื่อยๆจนสิ้นสุดหน่วยความจำ
- รูปแบบการใช้งานที่ง่ายและตายตัว
- สามารถติดต่อกับไมโครคอนโทรลเลอร์หรือใช้สวิตช์แบบแมนนวลก็ได้
- มีระบบประหยัดไฟ(สำหรับการต่อแบบแมนนวล)
- สามารถเข้าถึงหน่วยความจำได้ทุกๆตำแหน่ง
- คำพูดที่เก็บไว้ไม่มีการสูญหาย
- สามารถเก็บคำพูดไว้ได้ 100 ปี
- สามารถบันทึกซ้ำได้ 1000 ครั้ง
- ใช้ไฟเพียง 5 V.
- มีแหล่งกำเนิดสัญญาณนาฬิกาภายใน
- อุณหภูมิของขา(Die) = 0-50˚C และอุณหภูมิของ Package = 0-70˚C

รายละเอียดขาต่างๆของ ISD 2590

Address/Mode Input : การที่ ISD 2590 จะรับคำสั่งว่าเป็นตัวบอก Address หรือ Mode ต้องดูที่สองบิตสูงสุดของขา Address ถ้าทั้งสองบิตเป็น 0 ค่าที่รับจะเป็น Address แต่ถ้าบิตทั้ง 2 เป็น 1 ค่าที่รับมาจะเป็นการเลือก Mode การทำงานซึ่งมีอยู่ 7 Mode โดยที่ขา Address จะรับค่าเข้าไปก็ต่อเมื่อขา $\overline{CE}$ มีสัญญาณขอบขาลงเท่านั้น

AUX IN : เมื่อขา $\overline{CE}$ เป็น 1 และเป็นการทำงานในช่วงการเล่นเสียง ขานี้ใช้ในการเชื่อมต่อรับสัญญาณเสียงเข้ามาจากภายนอกแล้วส่งต่อไปที่วงจรขยาย แต่ถ้ามีการใช้เสียงที่เก็บไว้ใน Memory ขานี้ก็จะไม่ใช้

V_{SSA} , V_{SSD} : ทั้งสองขานี้คือ Digital Ground และ Analog Ground โดยที่กราวนด์ทั้งสองจะแยกกันเพื่อกันสัญญาณรบกวน

SP+/SP- : เป็นขาเอาท์พุทที่จะให้เสียงที่เล่นออกมามีความสามารถในการขับโหลด 16 Ω โดยให้กำลังได้สูงสุด 50 mW

V_{CCA}, V_{CCD} : ขาไฟเลี้ยงทั้งสองขาจะแยกกันเช่นเดียวกันกับขากราวนด์เพื่อลดสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MIC : เป็นขาที่รับสัญญาณเสียงจากภายนอกเข้าไปผ่านวงจรขยาย Pre Amplifier ที่ใช้ AGC ในการควบคุมอัตราขยาย

MIC REF : เป็นขาที่ช่วยในการกำจัดสัญญาณรบกวนของเสียงที่เข้าไปภายใน IC

AGC : Automatic Gain Control ใช้ในการปรับค่าอัตราขยายของ Pre Amplifier เพื่อชดเชยช่วงของเสียงที่เข้ามา AGC เป็นตัวที่ทำให้สามารถบันทึกเสียงตั้งแต่เสียงกระซิบไปจนถึงเสียงดังมากโดยที่ไม่มีการลดทอนสัญญาณ

ANA IN : Analog Input เป็นขาที่ใช้ส่งสัญญาณอนาลอกผ่านไปที่ Chip

ANA OUT : Analog Output เป็นขาที่ให้ Pre Amplifier Output โดยที่อัตราขยายจะถูกควบคุมโดย AGC

$\overline{OVF}$: ขานี้จะเป็น Logic 0 เมื่อสิ้นสุดตำแหน่งของ Memory นอกจากนั้นขา Overflow นี้ยังช่วยในการนำ ISD 2590 หลายๆตัวมาต่อกันโดยที่นำไปต่อเข้ากับขา $\overline{CE}$

$\overline{CE}$: Chip Enable จะถูกทำให้เป็น 0 เมื่อมีการต้องการที่จะใช้การบันทึกเสียงหรือเล่นเสียง

PD : Power Down โดยที่เมื่อไม่มีการใช้งานตัว Chip ขา PD ควรจะให้เป็น 1 เพื่อให้ ISD 2590 ทำงานในระบบ Stan By หรือเมื่อ $\overline{OVF}$ มีค่าเป็น 0 ขา PD ควรมีค่า 1 เพื่อ Reset ตำแหน่งของ Address Pointer ไปที่จุดเริ่มต้นใหม่

$\overline{EOM}$: End of Message ขานี้จะให้ Logic 0 ทุกๆครั้งที่ข้อความสิ้นสุด 1 ข้อความ

XCLK : ถ้าเราต้องการใช้สัญญาณนาฬิกาที่เที่ยงตรงกว่าที่ทีอยู่ภายในเราก็สามารถใส่สัญญาณเข้าไปที่ขานี้ได้ แต่ถ้าขานี้ไม่ถูกใช้ควรที่จะต่อลงกราวนด์

$P/\overline{R}$: ขานี้จะเป็นขาที่ใช้เลือกว่าจะทำงานเป็นการเล่นเสียงหรือทำงานเป็นการบันทึกเสียง โดยถ้าต้องการเล่นเสียงขานี้ต้องรับค่า 1 แต่ถ้าต้องการบันทึกเสียงขานี้ต้องรับค่า 0 โดยที่ Chip จะรับค่าเข้าไปก็ต่อเมื่อขา $\overline{CE}$ เป็นขอบขาลง

รายละเอียด

ISD 2590 เป็นอุปกรณ์ที่มี Sampling Frequency 5.3 kHz และ Bandwidth 2.3 kHz เพื่อให้ผู้ใช้สามารถอัดเสียงได้อย่างมีประสิทธิภาพ ใน Series ของ ISD ถ้าเราต้องการเพิ่มเวลาในการบันทึกเสียงจะทำให้ความถี่ Sampling และ Bandwidth ลดลง ซึ่งอาจจะทำให้คุณภาพเสียงลดลง

เสียงที่ได้รับการ Sampling แล้วจะถูกบันทึกโดยตรงไปที่ Memory โดยที่เสียงที่บันทึกนี้จะไม่มีการบีบอัดและจะอยู่อย่างนั้นจนกว่าจะมีการบันทึกทับไป โดยที่การบันทึกจะเป็นการบันทึกโดยตรงและการบันทึกของเสียงดนตรีจะไม่สามารถแบบนี้ทำได้ถ้าใช้การบันทึกเสียงแบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

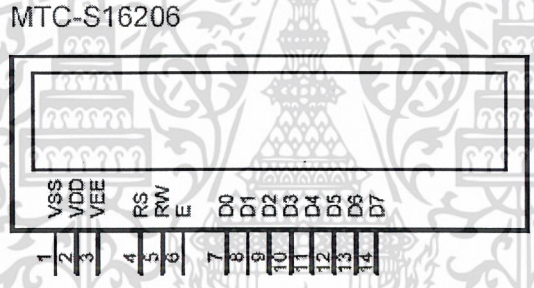
การเก็บเสียงใน Memory แบบ EEPROM จะเป็นการเก็บไว้โดยที่ไม่ต้องใช้พลังงาน ข้อมูลเสียงที่เก็บจะสามารถเก็บไว้ได้ 100 ปีโดยที่ไม่มีการใช้พลังงานและสามารถบันทึกซ้ำได้ 100,000 ครั้ง

การเชื่อมต่อ(Interface) สามารถทำได้ทั้งกับไมโครคอนโทรเลอร์และกับสวิทช์ทั่วไปโดยการส่งสัญญาณควบคุมต่างๆไปที่ ISD 2590 เช่นตำแหน่งของหน่วยความจำ , การเลือกว่าจะเล่นเสียงหรือบันทึกเสียง

2.4 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16X2)

รายละเอียดของ LCD (MTC-S16206)

สำหรับโมดุล LCD เป็นขนาด 16 ตัวอักษร 2 บรรทัด เป็นโมดูล LCD ที่มีโครงสร้างเป็นมาตรฐานเบอร์ MTC-S16206โมดุล LCDขนาด 16X2 มีขาใช้งานทั้งสิ้น 14 ขา สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้



รูปที่ 2.7 LCD เบอร์ MTC-S16206

V_{SS} (ขา1) : ต่อกราวนด์

V_{DD} (ขา2) : ต่อไฟเลี้ยง 5+ โวลต์

V_{OUT} (ขา3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR (Instruction Register)หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR (Data Register) โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W (ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับโมดุล LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา6) : เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดุล LCD ให้ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D0-D7 (ขา7 -14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต อนึ่งขาRS, R/W และ E จะใช้งานร่วมกันโดยมีความสัมพันธ์แสดงในตาราง

2.4.2 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดุล LCD ของชิปควบคุม โมดุล LCD ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear display) Space DDRAM

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างหรือ Space! ข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์กลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น “1”

2. คำสั่ง Return home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งเคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายมือสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือข้อมูลคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

0	0	0	0	0	1	I/D	S

รูปที่ 2.8 แสดงคำสั่งเลือกโหมดการป้อนข้อมูล

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูลถ้าหากบิต S เป็น “1”เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางด้านซ้ายแต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว แอดเดรสของ DDRAMเพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้นแต่ถ้าหากบิตนี้เป็น “0”แอดเดรสจะลดลงดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ แอดเดรสของ DDRAM เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คำสั่งควบคุมการแสดงผล

0	0	0	0	1	D	C	B

รูปที่ 2.9 แสดงคำสั่งควบคุมการแสดงผล

บิต D ใช้ควบคุมการปิดเปิดจอแสดงผล ถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0”จะเป็นการปิดเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์กระพริบดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH (8รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0CH เป็นการสั่งให้ปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

0	0	0	1	S/C	R/L	*	*

รูปที่ 2.10 แสดงคำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

การควบคุมการเลื่อนเคอร์เซอร์และและตัวอักษรบนจอแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ RL ซึ่งสามารถสรุปได้ดังนี้

S/C	RL	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนเคอร์เซอร์ใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนเคอร์เซอร์ใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

0	0	1	DL	N	F	*	*

รูปที่ 2.11 แสดงคำสั่งกำหนดฟังก์ชันการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าบิตนี้เป็น “0” จะแสดงผล 1 บรรทัดถ้าเป็น “1” จะแสดง 2 บรรทัดในกรณีที่จอแสดงผลแสดงได้มากกว่า 2 บรรทัด และต้องการแสดงผลมากกว่า2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “1” จุดที่น่าสังเกตคือโมดุล LCD แบบ 16 ตัวอักษร 1 บรรทัดแม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น “1” เนื่องจากแอดเดรสของ DDRAM แบ่งเป็นสองช่องคือ 00H และ 40H

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น “0”จะเป็นการแสดงผลแบบ5 x 7จุด และถ้าเป็น “1” จะแสดงผลเป็นเป็นแบบ5 x 10จุดข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดุล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัดและเลือกความละเอียดเป็น 5x7

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0”บิต 6 เป็น“1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูลโดยบิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วยหากบิต N เป็น “0” แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น “1”แอดเดรสของ DDRAM จะมีสองช่วงคือ 8CH-87H และ 0C0H-0C7H

9. คำสั่งอ่านFlag Busy และAddress

BF	A	A	A	A	A	A	A

รูปที่ 2.12 แสดงคำสั่งอ่านFlag Busy และAddress

เป็นคำสั่งที่ใช้อ่าน Flag Busy (BF) โดย Flag นี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0”แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผล

61350
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอยู่ ยังไม่พร้อมรังข้อมูลหรือคำสั่ง เมื่อต้องการอ่าน Flag ต้องกำหนดให้ขา R/W เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง นอกจากนี้ ยังใช้เป็นคำสั่งที่ใช้อ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต-0 บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.5 การเชื่อมต่อโดยใช้มาตรฐาน RS-485

ลักษณะการเชื่อมต่อตามมาตรฐาน RS-485

มาตรฐาน RS-485 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-422 ให้มีประสิทธิภาพในการสื่อสารเพิ่มขึ้นเป็นอย่างมากโดยมีการพัฒนาให้วงจรของตัวขับสัญญาณเป็นแบบ 3 สถานะ (Tri state) ทำให้ส่งข้อมูลได้สองทิศทางบนสายคู่เดียว และสามารถต่อเครือข่ายแบบ Multi drop ซึ่งอุปกรณ์หลาย ๆ ตัวสามารถรับส่งข้อมูลแบบ Half-Duplex บนสายสัญญาณคู่เดียวได้

เหตุผลในการเลือกใช้มาตรฐาน RS-485

1. คุณสมบัติในการสื่อสารสองทิศทางในคนละเวลา (Half-Duplex)

นอกจากนี้ตามมาตรฐาน RS-485 ยังสามารถเชื่อมต่อเป็นเครือข่ายได้มากถึง 32 จุด (Unit Loads: ULs) บนสายสัญญาณคู่เดียว

2. คุณสมบัติทางการเชื่อมต่อเป็นเครือข่าย

ตามมาตรฐาน RS-485 เครือข่ายสามารถเชื่อมต่อได้หลายจุดบนสายสัญญาณเพียงคู่เดียว (Multiple Transceivers) ซึ่งมีพื้นฐานอยู่บนการเชื่อมต่อเป็นเครือข่ายแบบบัส (Bus Type Network) จึงสามารถเชื่อมต่อได้หลายแบบโดยอาศัยการแปลงให้เป็นเครือข่ายเสมือนเพื่อให้เกิดประสิทธิภาพสูงสุดในการทำงาน

3. คุณสมบัติของสายสัญญาณที่ใช้

สายสัญญาณที่ใช้ตามมาตรฐาน RS-485 สามารถใช้สายเกลียวคู่ (Twist pair) ซึ่งเป็นสายสัญญาณโทรศัพท์ใช้งานโดยทั่วไปโดยไม่มีผลกระทบต่อการสื่อสารเลย แต่ถ้าต้องการคุณภาพการสื่อสารที่สูงขึ้น อาจใช้สายสัญญาณที่มีคุณภาพดีกว่าสายเกลียวคู่ เช่น สายโคแอกเชียล (Coaxial)

4. คุณสมบัติทางด้านอัตราเร็วและระยะทางในการส่งข้อมูล

ในการส่งข้อมูลตามมาตรฐาน RS-485 สามารถส่งข้อมูลได้สูงสุดถึง 10Mbps และส่งข้อมูลได้ไกลที่สุดถึง 400 ฟุต (1200 เมตร) โดยความสัมพันธ์ของอัตราเร็วในการส่งข้อมูลและระยะทางในการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คุณสมบัติทางด้านสัญญาณรบกวน

สัญญาณรบกวนมีผลต่อการสื่อสารตามมาตรฐาน RS-485 น้อยมากถ้าเลือกอัตราเร็วและระยะทางในการส่งข้อมูลให้เหมาะสม เนื่องจากมาตรฐาน RS-485 การสื่อสารเป็นแบบ Current Loop และใช้ความต่างศักย์ของคู่สายสัญญาณในการส่งข้อมูล ทำให้สามารถทนต่อสัญญาณรบกวนได้ดี โดยเฉพาะสัญญาณรบกวนแบบ Common mode

2.6 ระบบบัสแบบ I^2C

ระบบบัสแบบ I²C ย่อมาจาก Inter - IC Communication ซึ่งพัฒนาโดยห้องวิจัยของ Phillips ในประเทศ Netherlands เมื่อปี ค.ศ 1980 .สำหรับเชื่อมต่อชิปต่าง ๆ กับระบบไมโครคอนโทรเลอร์แบบอนุกรมโดยใช้สัญญาณเพียงสองเส้น เส้นหนึ่งเป็นสายข้อมูล อีกเส้นหนึ่งเป็นสายสัญญาณนาฬิกาสำหรับกำหนดจังหวะการสื่อสารข้อมูล ปัจจุบันได้มีชิปสนับสนุนการทำงานขอไมโครคอนโทรลเลอร์หลายตัวที่ใช้การเชื่อมต่อระบบบัสแบบนี้

สายข้อมูลที่ใช้รับส่งข้อมูลแบบอนุกรมมีชื่อเรียกว่า Serial Data Line (SDA) ส่วนสายสัญญาณนาฬิกาควบคุมมีชื่อว่า Serial Clock Line (SCL) สายทั้งสองเส้นนี้จะรับส่งข้อมูลได้สองทิศทาง วงจรทางเอาต์พุตของอุปกรณ์ที่ใช้บัสแบบนี้จะเป็นแบบวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector) ดังนั้นการต่อบัสแบบนี้จะต้องมีตัวต้านทานต่อพูลอัพกับแรงดันไฟ 5+ โวลต์ด้วย

ระบบบัสแบบนี้รับส่งข้อมูลได้สองทิศทางด้วยความเร็วสูงถึง 100 กิโลบิตต่อวินาที และสามารถใช้กับไอซีที่ใช้แรงดันไฟฟ้าต่างกันได้ ระบบบัสแบบนี้สามารถเชื่อมต่ออุปกรณ์ได้หลายตัว โดยที่อุปกรณ์แต่ละตัวที่ต่ออยู่กับระบบบัสแบบนี้สามารถส่งข้อมูลถึงกันได้โดยใช้รูปแบบการส่งข้อมูลหรือ Protocol ที่อุปกรณ์ทุกตัวรู้จัก อุปกรณ์แต่ละชนิดจะมีค่าแอดเดรสประจำตัวมัน อุปกรณ์ต้องการให้อุปกรณ์ตัวใดรับข้อมูลตัวส่งจะส่งแอดเดรสของอุปกรณ์ตัวนั้นออกไปก่อน ถ้าหากอุปกรณ์ตัวใดมีแอดเดรสตรงกันก็จะรับข้อมูลนั้นไป ถ้าหากอุปกรณ์ที่ต้องการส่งข้อมูลอุปกรณ์ตัวนั้นจะเรียกว่า Master โดยจะเป็นตัวสร้างจังหวะสัญญาณต่าง ๆ บนระบบบัส ส่วนอุปกรณ์ที่ถูกควบคุมหรือเป็นตัวรับข้อมูลจะเรียกว่า Slave

ถ้าหากทำการเชื่อมต่อไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ใช้บัสแบบ I²C หลายตัว ถ้าไมโครคอนโทรลเลอร์ต้องการส่งข้อมูล ตัวมันจะทำหน้าที่เป็นมาสเตอร์ และจะส่งสัญญาณไปบนบัสโดยส่งค่าแอดเดรสออกไปก่อน ถ้าค่าแอดเดรสนี้ตรงกับแอดเดรสของชิปตัวใด ชิปตัวนั้นจะส่งสัญญาณตอบรับ (ACK) ออกมา และชิปตัวนั้นก็จะถูกไมโครคอนโทรลเลอร์ควบคุม แต่ถ้าไม่ตรงกับตัวใดก็ไม่มีอะไรเกิดขึ้น โดยที่การส่งข้อมูลต่าง ๆ จะต้องเกิดขึ้นเมื่อระบบบัสว่างเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 สภาวะของ Bus I^2C

1. บัสว่าง (Bus not busy) สภาวะนี้ค่าลอจิกบนสาย SDA และ SCL จะเป็นลอจิกสูงทั้งคู่
2. เริ่มส่งข้อมูล(Start data transfer) สภาวะนี้สาย SCL จะเป็นลอจิกสูง แต่สาย SDA จะเปลี่ยนจากลอจิกสูงไปเป็นลอจิกต่ำ เรียกว่าวภาวะเริ่มต้น (start)
3. สภาวะหยุด (Stop) สภาวะนี้สาย SCL จะเป็นลอจิกสูง แต่สาย SDA จะเปลี่ยนจากลอจิกต่ำไปเป็นลอจิกสูง
4. สภาวะมีข้อมูล (Data valid) สภาวะนี้จะอยู่ระหว่างสภาวะเริ่มต้นและสภาวะหยุด โดยการรับส่งข้อมูลต่าง ๆ จะเกิดในสภาวะนี้ การรับส่งข้อมูลแต่ละบิตจะใช้สัญญาณนาฬิกาหนึ่งลูก โดยข้อมูลบน SDA จะต้องคงที่ ขณะที่ SCL จะเป็นลอจิกสูง และบิตข้อมูลใน SDA จะเปลี่ยนแปลงได้ขณะที่ SCL เป็นลอจิกต่ำ ถ้าหากบิตบน SDA เปลี่ยนแปลงขณะที่ SCL เป็นลอจิกสูง ระบบจะตีความว่าเป็นสภาวะเริ่มต้นส่งข้อมูล หรือสภาวะหยุดแทน
5. สภาวะตอบรับ (Acknowledge) เมื่ออุปกรณ์มาสเตอร์ส่งข้อมูลออกมาครบหนึ่งไบต์แล้วในช่วงสัญญาณ SCL มาสเตอร์จะส่งข้อมูลลอจิกสูงออกมาและถ้าตัวรับได้รับข้อมูลครบแล้วมันจะส่งสัญญาณตอบรับ (ACK) โดยทำให้ระดับลอจิกสูงบนสัญญาณ SDA ให้กลับเป็นระดับลอจิกต่ำ แต่ถ้าตัวรับได้รับข้อมูลไม่ถูกต้องตัวรับจะบังคับให้ตัวส่งหยุดส่งในสภาวะรอ
ในการโอนถ่ายข้อมูลระหว่างสภาวะเริ่มและสภาวะหยุดสามารถโอนถ่ายข้อมูลได้ไม่จำกัด แต่เมื่อ Slave ได้รับข้อมูลแต่ละไบต์แล้ว มันจะส่งสัญญาณกลับมาทุกครั้ง

2.6.2 ขั้นตอนการเขียนข้อมูลที่ใช้กับระบบบัสแบบ I^2C มีสามขั้นตอนดังนี้

1. เขียนข้อมูลอ้างแอดเดรส (Addressing) ข้อมูลไบต์แรกที่มาสเตอร์จะส่งออกไปคือข้อมูลที่ใช้อ้างแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ โดยทั่วไปแล้วข้อมูลที่กำหนดแอดเดรสจะมีจำนวน 8 บิต บิตต่ำสุดจะเป็นตัวระบุว่าอุปกรณ์ที่ต้องการติดต่อนั้นเราจะใช้อ่านหรือเขียนข้อมูลลงไป ส่วน 7 บิตบนจะเป็นบิตแอดเดรสซึ่งแบ่งออกเป็นบิตแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิตบน ซึ่งจะถูกโปรแกรมมาจากโรงงานที่ผลิตชิปแต่ละประเภท ส่วนอีก 3 บิตต่อมาจะเป็นบิตแอดเดรสที่ผู้ใช้สามารถโปรแกรมได้ (programmable address bit)
2. การเขียนไบต์ควบคุม (Control byte) ข้อมูลไบต์นี้จะขึ้นกับอุปกรณ์แต่ละประเภท เพื่อกำหนดการทำงานต่าง ๆ ของตัวมัน
3. การเขียนไบต์ข้อมูล (Data byte) เป็นข้อมูลที่จะโอนถ่ายในระบบ ระหว่างสภาวะเริ่มต้นและสภาวะหยุดจะมีการอ่านเขียนข้อมูลจำนวนกี่ไบต์ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การเขียนโปรแกรมสื่อสารแบบอนุกรม

ตามปกติการเขียนโปรแกรมสื่อสารแบบอนุกรมไม่ว่าจะเป็น Microsoft Visual C++, Boland C++ Builder และ Delphi มักมีการใช้ ActiveX หรือ คอมโพเนนท์ มาช่วยให้การเขียนโปรแกรมทำได้ง่ายขึ้น ซึ่ง ActiveX และ คอมโพเนนท์ที่นิยมใช้กันในปัจจุบันได้แก่ MSComm และ ComPort Library

2.7.1 การใช้งานคอลโทรล MSComm

คอลโทรล MSComm เป็นคอมโพเนนท์ชนิด Active ของบริษัทไมโครซอฟต์ ซึ่งใช้สำหรับควบคุมการสื่อสารข้อมูลแบบอนุกรม ด้วยความสามารถทางด้าน Active จึงทำให้คอมโพเนนท์ตัวนี้สามารถนามใช้ร่วมกับการเขียนโปรแกรมหลายภาษา เช่น Visual Basic, Visual C++, Borland C++ Builder และ Delphi คอลโทรล MSComm จะอยู่ในรูปของไฟล์ MsComm32.OCX ซึ่งถูกติดตั้งมากับ Visual Basic ส่วนในกรณีของ Visual C++ จะต้องทำการเลือกให้มีการติดตั้ง ActiveX Control ไว้ในขั้นตอนของการติดตั้งโปรแกรม

คอลโทรล MSComm จัดเตรียมทางเลือกสำหรับจัดการการสื่อสารไว้สองทางดังนี้

1. การสื่อสารโดยใช้การขับจากเหตุการณ์ (Event-Driven Communication) การสื่อสารแบบนี้เป็นวิธีการที่มีประสิทธิภาพในการจัดการการโต้ตอบของพอร์ตอนุกรม ในหลายๆสถานการณ์เราอาจต้องการให้มีการกระตุ้นในทันทีทันใดที่มีเหตุการณ์เกิดขึ้น เช่น เมื่อมีการรับไบต์ข้อมูลเข้ามา หรือมีการเปลี่ยนแปลงเกิดขึ้นในสายสัญญาณ CD (Carrier Detect) หรือ RTS (Request Tio Send) ในกรณีดังกล่าวจะใช้อีเวนต์ OnComm ของคอนโทรล MSComm ในการดักจับและจัดการกับการสื่อสารของเหตุการณ์เหล่านี้
2. ใช้การตรวจสอบอีเวนต์จากค่าพร็อพเพอร์ตี้ ComEvent หลังจากฟังก์ชันแต่ละตัวที่ทำให้เกิดการเปลี่ยนแปลงขึ้นภายในโปรแกรมทำงานเสร็จสิ้นลง วิธีการนี้เหมาะกับโปรแกรมที่มีขนาดเล็ก

ในการเขียนโปรแกรมนั้นคอนโทรล MSComm แต่ละตัวจะใช้กับพอร์ตอนุกรมได้เพียงพอร์ตเดียว ดังนั้นถ้าหากต้องการติดต่อกับพอร์ตอนุกรมมากกว่าหนึ่งพอร์ตในเวลาเดียวกัน เราจำเป็นต้องใช้คอนโทรล MSComm มากกว่าหนึ่งตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 คุณสมบัติที่สำคัญของคอนโทรล MSComm

1. Comm Port

ใช้ตั้งและรับค่าของหมายเลขพอร์ตที่ต้องการติดต่อสื่อสารด้วย โดยค่านี้จะอยู่ระหว่าง 1 ถึง 16

2. Port Open

ใช้สำหรับตั้งและรับค่าสถานะของพอร์ตสื่อสารในกรณีที่เปิดและปิดพอร์ต โดยถ้าต้องการเปิดพอร์ตจะเซ็ตเป็น TRUE และถ้าต้องการปิดพอร์ตจะเซ็ตเป็น FALSE คอนโทรล MSComm จะแจ้งข้อผิดพลาดในกรณีที่ไม่มีพอร์ตที่เราพยายามจะเปิดด้วยโค้ด 68 (Device unavailable)

3. Input

ใช้สำหรับรับและลบไบต์ข้อมูลออกจากบัฟเฟอร์รับข้อมูล ซึ่งข้อมูลที่รับเข้ามานั้นจะอยู่ในโหมดตัวอักษร หรือ โหมดไบนารี จะ ขึ้นอยู่กับการเซ็ตค่าพร็อพเพอร์ตี้ InputMode

4. Output

ใช้สำหรับเขียนสตรีมข้อมูลไปยังบัฟเฟอร์ส่ง การกำหนดค่าให้แก่พร็อพเพอร์ตี้นี้จะต้องใช้ตังแปรแบบ Variant เท่านั้น ซึ่งข้อมูลที่ใช้ในการส่งนั้นสามรถเป็นโหมดตัวอักษรหรือไบนารีก็ได้ โดยในการส่งข้อมูลที่เป็นตัวอักษรนั้น Variant ที่ใช้จะต้องบรรจุด้วยสตริงที่ต้องการส่งออกไป ส่วนการส่งข้อมูลไบนารีเราต้องกำหนดให้ Variant บรรจุด้วยอาเรย์ของไบต์ข้อมูล

5. Settings

พร็อพเพอร์ตี้ตัวนี้ใช้สำหรับการตั้งค่าอัตราบอด, พาริตี้, จำนวนบิตของข้อมูล และบิตสิ้นสุด โดยที่อัตราบอดเป็นอัตราเร็วของการส่งข้อมูล สำหรับพาริตี้เป็นส่วนที่ใช้สำหรับการตรวจสอบความถูกต้องขอข้อมูลซึ่งตามปกติมักไม่ใช้ ในส่วนของจำนวนบิตข้อมูลจะใช้ในการกำหนดจำนวนซึ่งส่งไปในแต่ละครั้ง และบิตสิ้นสุดทำหน้าที่เป็นตัวบอกจุดจบของบิตข้อมูล สำหรับรูปแบบของสตริงที่ใช้กับการตั้งค่าคุณสมบัติ มีลักษณะดังนี้

BBBB, P, D, S

โดยที่ BBBB คือค่าอัตราบอด, P เป็นค่าพาริตี้, D คือจำนวนบิตข้อมูล และ S คือจำนวนของบิตสิ้นสุด ซึ่งค่าเริ่มต้นของสตริงนี้คือ “9600, N, 8, 1” นั่นคืออัตราการส่งข้อมูลเท่ากับ 9600 บอด, ไม่มีการใช้พาริตี้บิต, ความยาวของข้อมูลเท่ากับ 8 บิต และมีบิตสิ้นสุดความยาว 1 บิต สำหรับค่าที่เป็นไปได้ในการเซ็ตพร็อพเพอร์ตี้ Settings มีดังนี้

- อัตราบอด (บิตต่อวินาที): 110, 300, 600, 1200, 2400, 9600, (ค่าเริ่มต้น), 14400, 19200, 28800, 38400, 56000, 128000, 256000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พาริตี้: E (พาริตี้คู่), M (Mark คือการเซ็ตพาริตี้บิตให้เป็น 1), N (ไม่ใช้: ค่าเริ่มต้น), O (พาริตี้คี่), S (Space คือการเซ็ตพาริตี้บิตให้เป็น 0)
- จำนวนบิตข้อมูล: 4, 5, 6, 7, 8 (ค่าเริ่มต้น)
- จำนวนของบิตสิ้นสุด: 1(ค่าเริ่มต้น), 1.5, 2

6. InBuffer Size และ OutBuffer Size

พร็อพเพอร์ตี้ InBufferSize และ OutBufferSize เป็นการกำหนดปริมาณหน่วยความจำที่ถูกจองไว้สำหรับบัฟเฟอร์รับและบัฟเฟอร์ส่ง การตั้งให้ค่านี้มากเกินไปจะทำให้สิ้นเปลืองหน่วยความจำของระบบและถ้ามีค่าน้อยเกินไปจะทำให้เกิดการล้นของข้อมูลและเกิดความเสียหายแก่ข้อมูลได้แต่ เนื่องจากประสิทธิภาพของเครื่องคอมในปัจจุบันมีความสามารถเพิ่มขึ้นเป็นอย่างมาก ดังนั้นการปรับค่าการจองหน่วยความจำนี้ให้มีค่าสูงขึ้นจึงไม่ส่งผลกระทบต่อประสิทธิภาพของระบบมากนัก ในการใช้งานจริงนั้นเรามักใช้ค่าเริ่มต้นของมันไปก่อนนั่นคือ InBuffer Size มีค่าเท่ากับ 1024 ไบต์ และ OutBuffer Size มีค่าเท่ากับ 512 ไบต์ และหากค่าดังกล่าวไม่เพียงพอในการใช้งานจึงปรับเพิ่มให้เหมาะสมต่อไป

7. R Threshold และ S Threshold

พร็อพเพอร์ตี้ทั้งสองตัวนี้ใช้สำหรับการเซ็ต และอ่านค่าจำนวนของไบต์ข้อมูลซึ่งถูกรับเข้ามาในบัฟเฟอร์และบัฟเฟอร์ส่งก่อนที่อีเวนต์ OnComm จะถูกกระตุ้นขึ้นมา อีเวนต์ OnComm จะถูกใช้สำหรับตรวจจับและตอบสนองต่อการเปลี่ยนแปลงในขั้นตอนของการสื่อสาร การเซ็ตค่าพร็อพเพอร์ตี้ทั้งสองตัวนี้ให้เป็น 0 จะเป็นการหลีกเลี่ยงการกระตุ้นจากอีเวนต์ OnComm และการปรับค่าทั้งสองให้เป็นค่าอื่นที่ไม่เป็น 0 จะทำให้อีเวนต์ OnComm ถูกกระตุ้นขึ้นทุกครั้งที่มีการรับไบต์ข้อมูลเข้ามาในบัฟเฟอร์

8. Input Len และ EOF Enable

การปรับค่า InputLen ให้เท่ากับ 0 จะเป็นการกำหนดให้คอนโทรลทำการอ่านข้อมูลทั้งหมดซึ่งบรรจุอยู่ในบัฟเฟอร์รับ ส่วนพร็อพเพอร์ตี้ EOFEnable จะใช้สำหรับหยุดการรับข้อมูล อละกระตุ้นอีเวนต์ OnComm เมื่อมีการรับไบต์ข้อมูลซึ่งเป็น EOF (End Of File) หรือจุดสิ้นสุดของไฟล์เข้ามา

9. Input Mode

พร็อพเพอร์ตี้ตัวนี้เป็นตัวกำหนดลักษณะของข้อมูลที่รับเข้ามา ซึ่งข้อมูลนี้สามารถเป็นได้ทั้งสตริง(ตัวอักษร) หรือไบต์อะเรย์ (ข้อมูลไบนารี) การเซ็ตค่านี้ให้เป็น 0 หมายถึงการรับตัวข้อมูลแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สตริง และการเซ็ตให้เป็น 1 จะเป็นการรับข้อมูลแบบไบนารีซึ่งรวมถึงตัวอักขระและสัญลักษณ์ต่างๆ ซึ่งไม่สามารถแสดงผลได้ เช่น NULL เป็นต้น

10. Handshaking

พร็อพเพอร์ตี้ Handshaking ใช้สำหรับเซ็ตและรับค่าการทำแฮนด์เช็กโพรโตคอลของฮาร์ดแวร์ การทำแฮนด์เช็กจะช่วยทำให้มั่นใจได้ว่าไม่มีการสูญหายของข้อมูลเกิดขึ้นถ้าเกิดการล้นของบัฟเฟอร์ ซึ่งมีสาเหตุมาจากการที่ข้อมูลวิ่งมาถึงพอร์ตเร็วเกินกว่าที่อุปกรณ์สื่อสารจะทำการเคลื่อนย้ายข้อมูลเข้าไปเก็บยังบัฟเฟอร์สำหรับรับข้อมูล สำหรับการเซ็ตค่าของ Handshaking มีดังนี้

ค่าคงที่	ค่า	คำอธิบาย
comNone	0	ไม่มีการทำแฮนด์เช็ก(ค่าเริ่มต้น)
comXonXoff	1	มีการทำแฮนด์เช็ก XonXoff
comRTS	2	มีการทำแฮนด์เช็ก RTS/CTS(Request To Send)
comERSXOnXOff	3	มีการทำแฮนด์เช็ก RTS และ XonXoff

ตารางที่ 2.2 การเซ็ตค่าแฮนด์เช็ก

11.การเปิดพอร์ตอนุกรม

ในการเปิดพอร์ตอนุกรมมีพร็อพเพอร์ตี้ที่เกี่ยวข้องได้แก่ CommPort, PortOpen และ Settings ซึ่งใน Visual C++ จะใช้ฟังก์ชันสมชิกของคลาส CMSComm ในการเซ็ตค่าพร็อพเพอร์ตี้ดังกล่าวได้แก่ SetCommPort, SetSettings และ SetPortOpen สำหรับตัวอย่างของการเปิดพอร์ตอนุกรมสามารถเขียนได้ดังนี้

```
1. m_Comm.SetCommPort(PortNumber);
2. m_Comm.SetSettings("9600,N,8,1");
3. m_Comm.SetPortOpen(TRUE);
```

สมมติให้ m_Comm เป็นออปเจ็คของคอนโทรล MSComm ซึ่งอยู่ในคลาส CMSComm บรรทัดที่ 1 เป็นการเลือกพอร์ตตามตัวแปร PortNumber บรรทัดที่ 2 เป็นการเซ็ตค่าอัตราบอด, พาริตี้, ความยาวของบิตข้อมูล และความยาวของบิตสิ้นสุด ส่วนบรรทัดที่ 3 เป็นการเปิดพอร์ตที่ได้เลือกเอาไว้ในบรรทัดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. การปิดพอร์ตอนุกรม

ในการปิดพอร์ตอนุกรมมีพร็อพเพอร์ตี้ที่เกี่ยวข้องได้แก่ PortOpen ซึ่งตัวอย่างการปิดพอร์ตสำหรับ VisualC++ สามารถเขียนได้ดังนี้

```
1. m_Comm.SetPortOpen(FALSE);
```

2.7.3 การส่งข้อมูลผ่านคอนโทรล MSComm

1. การส่งข้อมูลในโหมดตัวอักษร

การส่งข้อมูลเป็นตวอักษรผ่านคอนโทรล MSComm ใน Visual C++ สามารถทำได้โดยใช้ฟังก์ชัน SetOutput เพื่อเซ็ตค่าให้แก่พร็อพเพอร์ตี้ Output ซึ่งพร็อพเพอร์ตี้นี้จะต้องกำหนดให้เป็น Variant ที่บรรจุสตริงเอาไว้ สำหรับโค้ดการส่งข้อมูลในโหมดอักษรสามารถเขียนได้ดังนี้

```
1. CString strOutput;
2. strOutput = _T("Hello");
3. m_Comm.SetOutput(COLeVariant(strOutput));
```

ในบรรทัดที่ 1 เป็นการประกาศตัวแปร strOutput เป็นชนิด CString บรรทัดที่ 2 เป็นการกำหนดค่าให้แก่ตัวแปร strOutput และในบรรทัดที่ 3 เป็นการแปลง strOutput ให้เป็นตัวแปรแบบ Variant แล้วส่งเป็นพารามิเตอร์ให้แก่ฟังก์ชัน SetOutput ทั้งนี้เนื่องจากฟังก์ชัน SetOutput รับพารามิเตอร์เป็นแบบ Variant ดังนั้นจึงต้องมีการแปลงชนิดของตัวแปรแบบสตริงให้เป็น Varient

2. การส่งข้อมูลในโหมดไบนารี

เนื่องจากการส่งข้อมูลในบางครั้งอาจมีความต้องการส่งค่าซึ่งไม่สามารถแสดงเป็นตัวอักษรได้ เช่น NULL หรือค่าพิเศษอื่นๆ ดังนั้นการส่งข้อมูลในลักษณะนี้จึงจำเป็นต้องใช้การส่งข้อมูลในโหมดไบนารี การส่งข้อมูลในโหมดนี้จะใช้ฟังก์ชันเดียวกับโหมดตัวอักษร นั่นคือ SetOutput แต่ค่าพร็อพเพอร์ตี้จะต้องเป็น Variant ซึ่งบรรจุอะเรย์ของไบต์ข้อมูลไบนารีหรือไบต์อะเรย์ (Byte Array) เอาไว้สำหรับโค้ดในการส่งข้อมูลสามารถเขียนได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1. CByteArray arrBytes;
2. unsignde char sendByte;
3. arrBytes.Add(sendByte);
4. sendBytes.SetOutput(COleVariant(arrBytes));
```

การส่งข้อมูลในโหมดไบนารีมักกระทำผ่านตัวแปรชนิด BYTE หรือ unsigned char แต่เนื่องจากฟังก์ชัน SetOutput รับพารามิเตอร์เป็นแบบ Variant ซึ่งบรรจุไบต์อะเรย์เอาไว้ดังนั้นการส่งข้อมูลที่เป็นชนิด BYTE หรือ unsigned char ด้วยคอนโทรล MSComm ขั้นตอนแรกต้องทำการจัดรูปแบบข้อมูลก่อน ในบรรทัดที่ 1 และ 2 เป็นการประกาศตัวแปรชนิด CByteArray หรือ unsigned char ตามลำดับส่วนบรรทัดที่ 3 เป็นการใส่ค่าตัวแปร ด sendByte เข้าไปใน arrBytes และในบรรทัดสุดท้ายเป็นการแปลง arrBytes ให้เป็นแบบ Variant แล้วส่งเป็นพารามิเตอร์ให้แก่ฟังก์ชัน SetOutput

2.7.4 การรับข้อมูลผ่านคอนโทรล MSComm

1. การรับข้อมูลในโหมดตัวอักษร

การรับข้อมูลในโหมดตัวอักษรผ่านคอนโทรล MSComm หรือ Visual C++ สามารถทำได้โดยใช้ฟังก์ชัน GetInput ซึ่งค่าส่งกลับของฟังก์ชันนี้จะเป็นตัวแปรชนิด Variant ที่อาจบรรจุด้วยสตริงหรืออาจเป็นอะเรย์ของไบต์ข้อมูลก็ได้ ทั้งนี้ขึ้นอยู่กับการเซ็ตโหมดของการรับอินพุท สำหรับการเซ็ตโหมดของการรับอินพุทนั้นสามารถทำได้ 2 วิธี คือการกำหนดค่าให้แก่พร็อพเพอร์ตี้ InputMode ที่ตัวคอนโทรลโดยตรงในขั้นตอนของการออกแบบโปรแกรม และอีกวิธีหนึ่งเป็นการใช้ฟังก์ชัน SetInputMode เพื่อทำการเซ็ตค่าให้แก่พร็อพเพอร์ตี้ InputMode สำหรับโค้ดของการรับข้อมูลในโหมดตัวอักษรสามารถเขียนได้ดังนี้

```
1. COleVariant myVar;
2. CString str;
3. m_Comm.SetInputMode(0);
4. myVar.Attach(m_Comm.GetInput());
5. str. = CSring(myVar.bstrVal);
```

สำหรับโค้ดโปรแกรมข้างต้นสามารถอธิบายได้ดังนี้

บรรทัดที่ 1 เป็นการประกาศตัวแปร myVar เป็นชนิด COleVariant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรทัดที่ 2 เป็นการประกาศตัวแปร str เป็นชนิด CSring

บรรทัดที่ 3 เป็นการเซ็ตโหมดการรับข้อมูลเป็นแบบไบนารี

บรรทัดที่ 4 เป็นการรับค่าจากคอนโทรลแล้วส่งค่าให้แก่ตัวแปร myVar

บรรทัดที่ 5 เป็นการสำสตริงจาก Variant มากำหนดค่าให้แก่ตัวแปร str ซึ่งสตริงที่ได้มานี้อาจนำไปประมวลผลเพื่อใช้กับงานอื่นๆ ต่อไป

2. การรับข้อมูลในโหมดไบนารี

การรับข้อมูลไบนารีผ่านคอนโทรล MSComm สามารถทำได้โดยใช้ฟังก์ชัน GetInput เช่นกัน แต่อาจมีความยุ่งยากมากกว่าเล็กน้อย สำหรับโค้ดของการรับข้อมูลในโหมดตัวอักษรสามารถเขียนได้ดังนี้

```
1.  long lLen;
2.  BYTE *pAccess;
3.  COleVariant myVar;
4.  CString str;
5.  HRESULT hr;
6.  myVar.Attach(m_Comm.GetInput());
7.  hr = SafeArrayGetUBound(myVar.parray, 1, &lLen);
8.  if  (hr == S_OK)
9.  {
10.     hr = SafeArrayAccessData(myVar.parray,(void**)&pAccess);
11.     if (hr == S_OK)
12.     {
13.         for (int i = 0; i <= lLen; i++)
14.         {
15.             //ProcessData(pAccess[i]);
16.         }
17.         SafeArrayUnaccessData(myVar.parry);
18.     }
19. }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดข้างต้นสามารถอธิบายได้ดังนี้

บรรทัดที่ 6 เป็นการกำหนดค่าให้แก่ตัวแปร myVar โดยรับข้อมูลจากคอนโทรล MSComm ซึ่งค่าที่รับเข้ามานี้จะเก็บอยู่ในฟิลด์ parray ของตัวแปร myVar

บรรทัดที่ 7 เป็นการหาค่าขนาดของฟิลด์อะเรย์ parray ซึ่งค่าที่ได้จะเป็นขนาดที่เริ่มนับตั้งแต่ 0 และถูกส่งไปในตัวแปร lLen ฟังก์ชัน SafeArrayGetUBound จะส่งค่ากลับเป็นผลของการทำงานนั่นคือ HERSULT ถ้าการทำงานสำเร็จค่าที่ส่งกลับมาจะมีค่าเท่ากับ S_OK

บรรทัดที่ 10 เป็นการเข้าถึงข้อมูลที่มีอยู่ในฟิลด์อะเรย์ parray โดยใช้พอยน์เตอร์ pAccess เป็นตัวชี้ข้อมูล ขั้นตอนนี้จะมีการตรวจสอบผลการทำงานของฟังก์ชันเช่นกัน โดยถ้าผลการทำงานสำเร็จฟังก์ชันจะส่งค่า S_OK กลับมา

บรรทัดที่ 15 คือตำแหน่งซึ่งเราสามารถใส่โค้ดสำหรับทำงานกับข้อมูลแต่ละไบต์ซึ่งรับเข้ามาจากคอนโทรล MSComm

บรรทัดที่ 17 เป็นการปลดข้อมูลออกจากตัวแปร myVar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

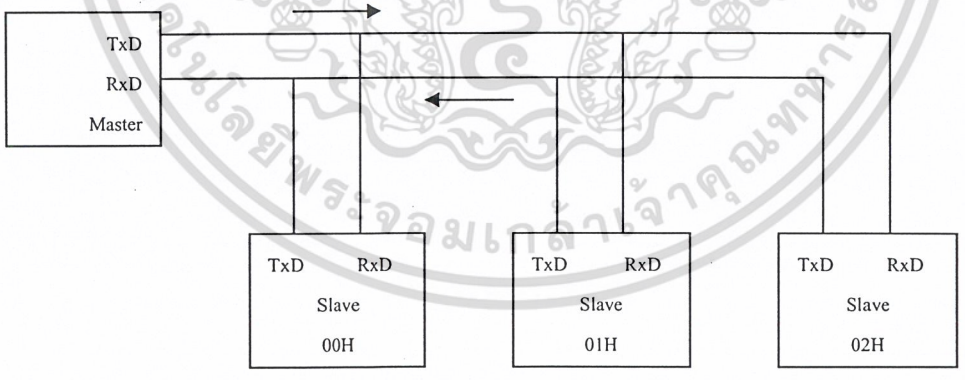
บทที่ 3

การออกแบบ

ในโครงงานนี้หลักการควบคุมเป็นแบบมีไมโครคอนโทรเลอร์ตัวหลักเป็นตัวควบคุมซึ่งเรียกว่าตัว Master ที่จะคอยควบคุมและสั่งการไปที่ไมโครคอนโทรเลอร์ตัวอื่นๆซึ่งเรียกว่า Slave โดยที่ตัว Slave แต่ละตัวก็จะมี Address ของตัวมันเองเพื่อที่จะให้ไมโครคอนโทรเลอร์รู้ว่าจะต้องติดต่อกับ Slave ตัวไหน และมาตรฐานที่จะใช้ติดต่อกันระหว่างไมโครคอนโทรเลอร์ทุกตัวจะใช้มาตรฐานแบบ RS-485 ในการเชื่อมต่อ เพราะเป็นการส่งข้อมูลแบบ Differential ทำให้ตัดสัญญาณรบกวนได้ดีและ IC ที่ใช้ในการรับส่งข้อมูลในมาตรฐาน RS-485 ซึ่งก็คือ SN 75176 นั้นสามารถสามารถติดต่อกันได้มากที่สุด 32 ตัว นั่นหมายความว่าเราสามารถมี Slave ได้ถึง 31 ตัว และรูปแบบการต่อไมโครคอนโทรเลอร์เข้ากับ SN 75176 สามารถทำได้ดังรูป

3.1 การติดต่อแบบ Multiprocessor

การติดต่อสื่อสารระหว่างไมโครคอนโทรเลอร์ในโครงงานนี้จะใช้การติดต่อสื่อสารผ่านพอร์ตสื่อสารแบบอนุกรมที่มีอยู่ในตัวของไมโครคอนโทรเลอร์(MCS-51) โดยใช้คุณสมบัติการเชื่อมต่อ แบบ Multiprocessorในการเชื่อมต่อระบบต่างๆเข้าด้วยกัน โดยที่มีตัวประมวลผลหลัก (Master Controller) ทำหน้าที่ในการรับส่งและควบคุมระบบต่างๆให้เป็นไปตามที่กำหนด



รูปที่ 3.1 เครือข่ายของไมโครคอนโทรเลอร์ที่เชื่อต่อกันหลายตัว

การทำงานของพอร์ตอนุกรมของ MCS-51 ในโหมด 2 และ 3 จะมีการทำงานพิเศษซึ่งสามารถทำงานแบบ Multiprocessor ดังที่กล่าวไว้ในเบื้องต้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ข้อมูลที่เป็นตัวกำหนด Address (Address Byte)

การเริ่มต้นกระบวนการรับส่งทุกอย่างจะเริ่มด้วย Master จะส่ง Address Byte ของตัว Slave ที่ต้องการติดต่อด้วยไปที่ Bus ข้อมูล โดยตัว Slave ทุกตัวจะได้รับ Address Byte ที่เหมือนกันเมื่อได้รับแล้วก็จะพิจารณาว่าเป็น Address byte ของตัวเองหรือเปล่า ถ้าใช่ก็จะตอบรับมายัง Master และเตรียมส่งข้อมูลในลำดับต่อไป

Start Bit = 0	D0	D1	D2	D3	D4	D5	D6	D7	D8 = 1	Stop Bit = 1

รูปที่ 3.2 ไบต์กำหนดแอดเดรส (Address byte)

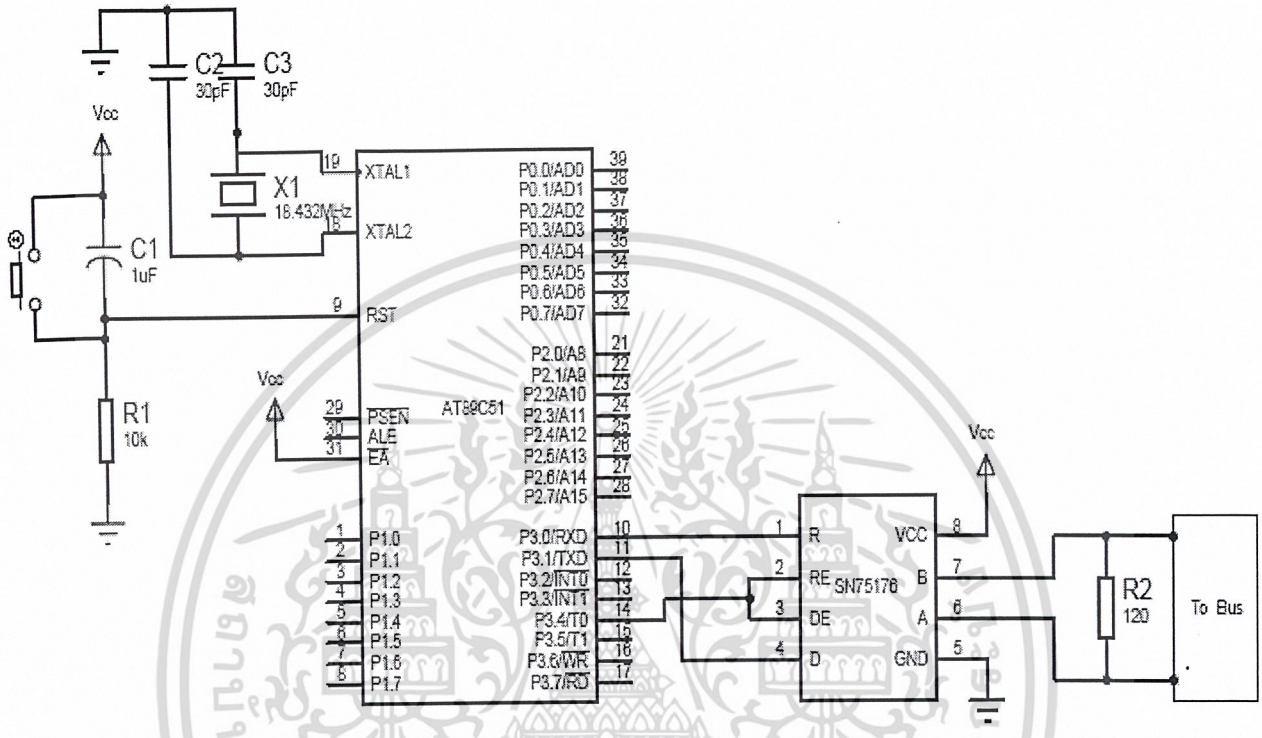
3.1.2 ข้อมูลที่เป็นข้อมูลข่าวสาร (Data byte)

เป็นไบต์ที่ใช้แทนข้อมูลที่นำไปใช้งาน ลักษณะข้อมูลเป็นดังรูปที่ 3.3 โดยเมื่อ Slave ตอบรับกับ Master ตัว Slave และ Master จะเปลี่ยนโหมดการทำงานของตัวเองจาก Multiprocessor mode มาเป็น Single Processor Mode เพื่อให้สามารถรับส่งโดยตรงกันได้เลยและข้อมูลบิตที่ 8 จะต้องมีค่าเท่ากับ 0 เสมอในช่วงการส่งข้อมูล และเมื่อสิ้นสุดการส่งข้อมูลข่าวสาร(Data Byte) Master และ Slave ก็จะกลับไปทำงานในโหมด Multiprocessor อีกครั้ง

Start Bit = 0	D0	D1	D2	D3	D4	D5	D6	D7	D8 = 0	Stop Bit = 1

รูปที่ 3.3 ไบต์กำหนดข้อมูล (Data byte)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การเชื่อมต่อไมโครคอนโทรเลอร์เข้ากับ SN 75176

Protocol ที่ใช้ในการสื่อสาร

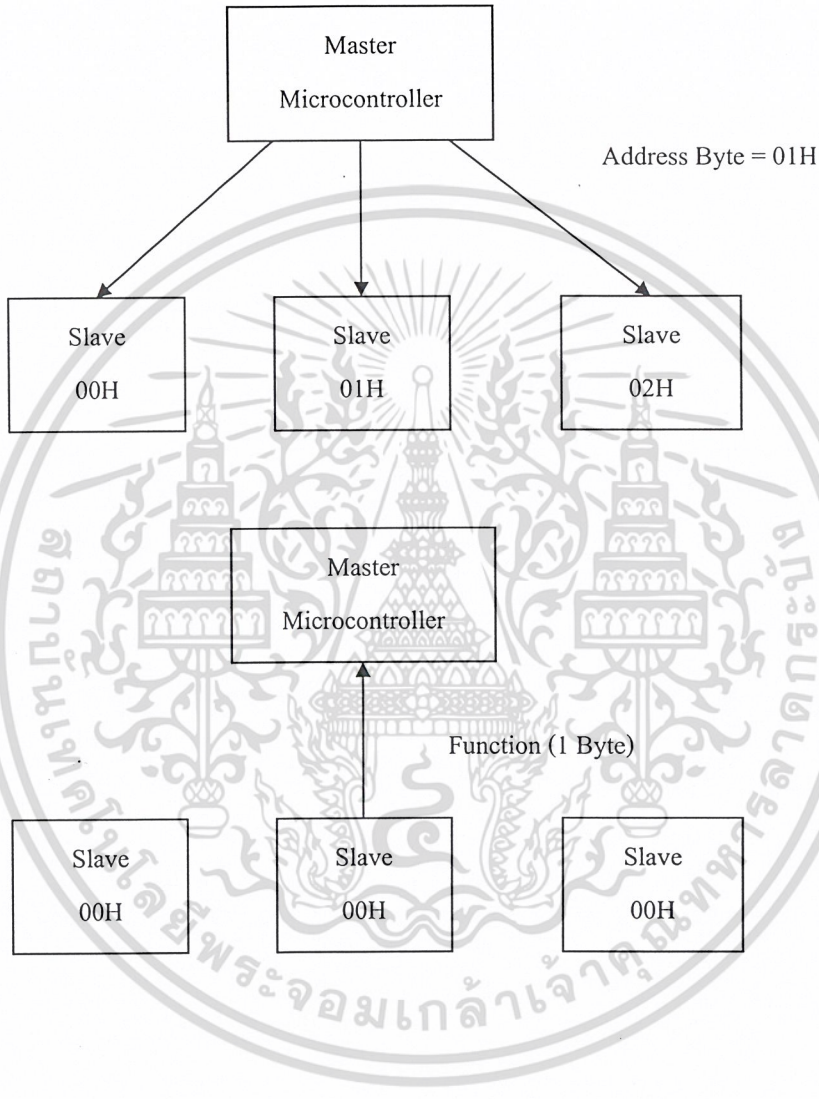
สำหรับ Protocol ที่ใช้ในการสื่อสารนั้นได้ออกแบบมาให้มีลักษณะคือข้อมูลที่ใช้ในการส่ง 1 Byte จะมี 9 บิตดังนี้

Address Byte : xxxxxxxx1

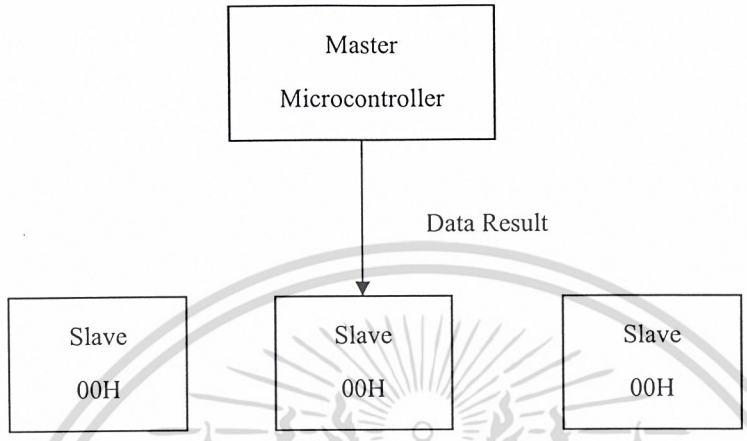
Data Byte : xxxxxxxx0

การติดต่อจะเริ่มที่ตัว Master จะส่ง Address มาให้กับ Slave ทุกตัวผ่านทาง Data Bus โดยที่ Slave ทุกตัวจะรับ Address นั้นเข้าไปเช็คดูว่าใช่ Address ของตนหรือไม่ ถ้าไม่ใช่ก็จะอยู่เฉยๆแล้วรอจนกว่าจะมีการส่ง Address ครั้งต่อไปมา แต่ถ้าใช่ก็จะส่งคำสั่งที่ต้องการให้ Master ทำไปให้กับ Master แล้วก็จะรอจน Master ส่งข้อมูลตามที่ต้องการกลับมาให้ หลังจาก Master ส่งข้อมูลไปให้กับ Slave ข้างต้นแล้วก็จะวนส่ง Address ของ Slave ตัวอื่นๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



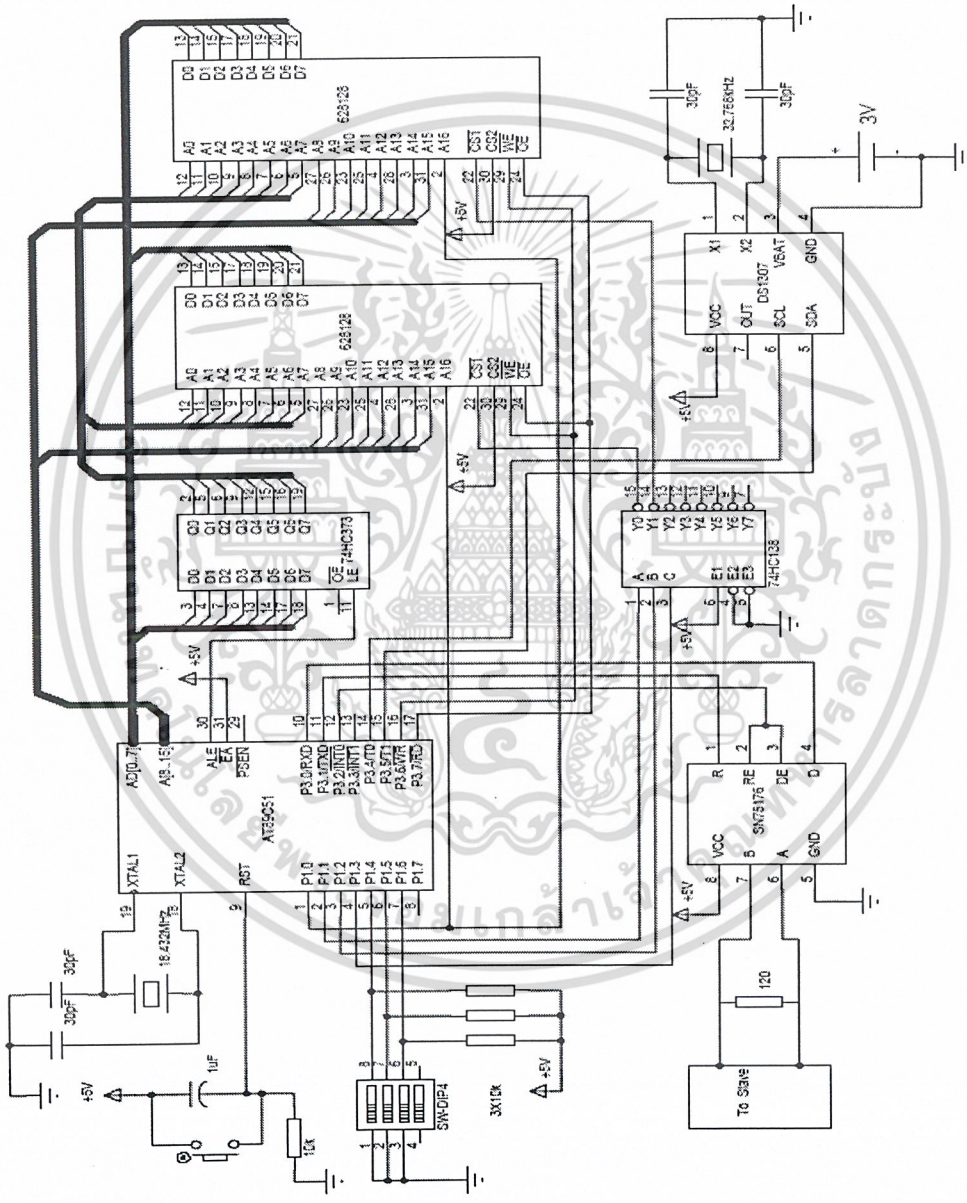
รูปที่ 3.5 แสดง Protocol ที่ใช้ในการสื่อสาร

3.2 หน่วยประมวลผลหลัก (Master Controller)

ตัว Master นี้จะเป็นตัวควบคุมการทำงานหลักจะมีหน้าที่วนส่ง Address Byte ให้กับ Slave ทีละตัวทุกตัว ถ้า Slave ตัวใดมี Address ตรงกับที่ Master ส่งมาก็จะส่งคำสั่งไปขอข้อมูลที่ต้องการมาจาก Master ซึ่งส่วนประกอบของหน่วยประมวลผลหลักมีดังนี้

- ไมโครคอนโทรเลอร์ (MCS-51) เบอร์ AT89C5120PC ใช้ในการควบคุมการทำงานของระบบ โดยใช้แหล่งกำเนิดความถี่ (XTAL Oscillator) 18.432MHz.
- หน่วยความจำ(Memory) ภายนอก เป็นส่วนที่ใช้ในการเก็บข้อมูลการปฏิบัติงานของพนักงาน ซึ่งประกอบด้วย 74HC373 เป็น IC Latch ใช้ในการเก็บค่า Address Low Byte ที่ออกมาจากพอร์ต 0 เพราะใช้พอร์ต 0 เป็นทางออกของ Address Low Byte และ เป็นทางเข้าออกของข้อมูลที่ติดต่อกับหน่วยความจำ
- 74HC138 เป็น IC Decoder ใช้ในการเลือกหน่วยความจำ(Memory)ที่ต้องอ่าน โดยสามารถรองรับหน่วยความจำได้มากที่สุด 8 ตัว โดยที่ใช้ P1.1-P1.3 ของ Master ในการควบคุมและใช้ Dip Switch ในการเลือกว่ามีหน่วยความจำต่ออยู่ในระบบกี่ตัว โดยที่ Dip Switch ต่อกับ P1.4-1.6 ของ Master และยังมี LED ที่ P1.7 เพื่อใช้บอกสถานะของหน่วยความจำว่าถึงค่าวิกฤตที่หน่วยความจำใกล้เต็มแล้วหรือยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงหน่วยประมวลผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 หน่วยประมวลผลส่วนให้บริการ (Counter Service Unit)

ในส่วนนี้จะมีการทำงานหลักๆอยู่ 3 ส่วนคือ

3.3.1 ส่วนแสดงผลหมายเลขช่องให้บริการ(Counter)

จะใช้ 7-Segment ในการแสดงผล โดยที่ 7-Segment นี้จะต่อผ่าน IC 8255 ที่ใช้ในการของพอร์ตของไมโครคอนโทรเลอร์

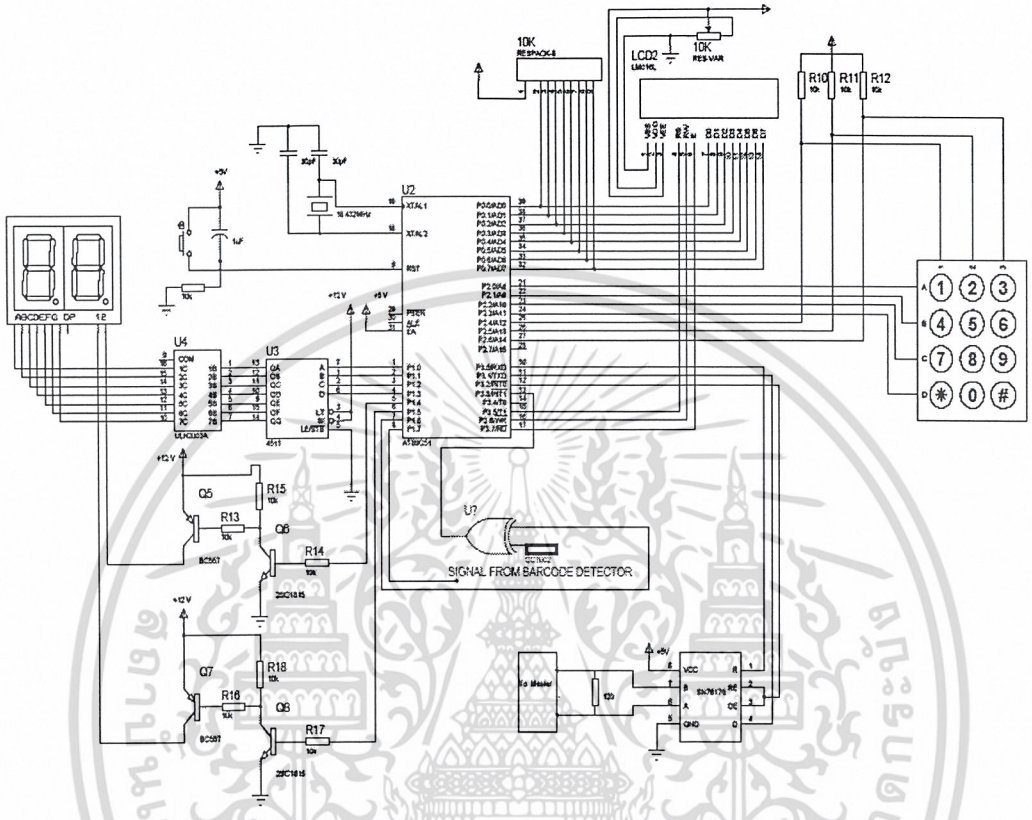
3.3.2 ส่วนรับอินพุท

ใช้ Matrix Switch ทำหน้าที่เป็นตัวรับอินพุทจากพนักงาน โดยที่อินพุทจะนำไปใช้ในการประมวลผลและร้องขอข้อมูลจาก Master เช่น ขอคิวต่อไป , ขอข้ามคิว , ขอยกเลิกคิว และยังใช้ในการใส่รหัสของพนักงานที่มานั่งประจำที่ช่องให้บริการ(Counter) โดยที่ Matrix Switch นี้จะต่อผ่าน IC 8255

3.3.3 ส่วนแสดงผล LCD

ใช้เพื่อตอบสนองพนักงานกับระบบเพื่อให้พนักงานได้ทราบว่าคิวที่กำลังจะให้บริการหรือกำลังให้บริการอยู่เป็นหมายเลขใด และยังสามารถแสดงผลอื่นๆได้อีกด้วย โดยที่ LCD จะติดต่อโดยตรงกับพอร์ตของไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรที่ส่วนให้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนแจกคิว (Queue Distribution Unit)

ลักษณะหน่วยประมวลผลนี้ประกอบด้วยส่วนต่าง ๆ ตามรูปที่ 3.8 ดังนี้

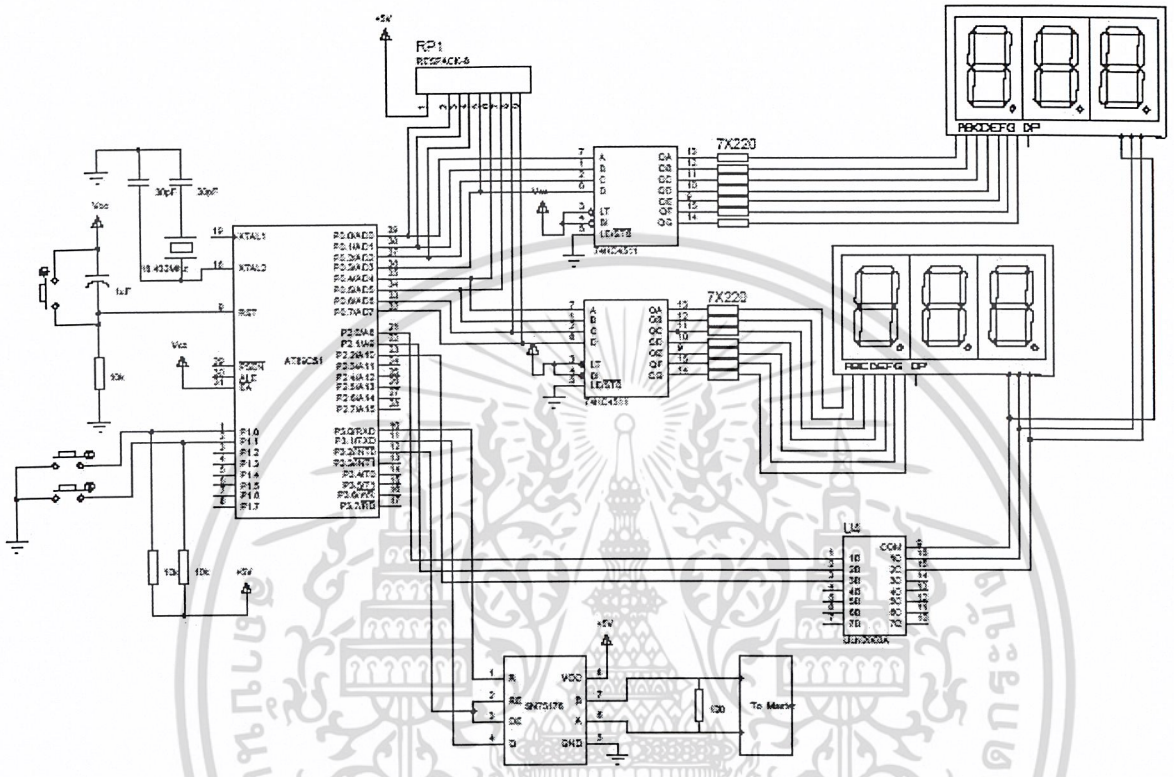
3.4.1 จอแสดงผล 7 –Segment

มีไว้สำหรับแสดงค่าของหมายเลขคิวที่ผู้ใช้บริการได้รับ และมีไว้เพื่อแสดงจำนวนของผู้ใช้บริการที่กำลังรอคิวอยู่ทั้งหมด การแสดงผลของ 7 –Segment ควบคุมโดย MCS-51 โดยจะส่งค่าแบบ BCD มาเก็บที่ 4511 ทั้งสองตัว โดยตัวแรกทำหน้าที่ในการเก็บแต่ละหลักของหมายเลขคิว และตัวที่สองทำหน้าที่ในการเก็บแต่ละหลักของจำนวนผู้รอคิวโดยจะใช้การแสดงแต่ละหลักของทั้ง 2 ชุด พร้อมกันโดยการสั่งทรานซิสเตอร์ที่ควบคุมแต่ละหลัก ON ทำให้กระแสไหลผ่าน 7 –Segment หลังจากนั้นก็หน่วงเวลาช่วงหนึ่ง แล้วก็สั่งทรานซิสเตอร์ OFF แล้วส่งข้อมูลหลักใหม่ไปเก็บใน 4511 ใหม่แล้วแสดงหลักถัดไป ทำอย่างนี้วนไปเรื่อย ๆ

3.4.2 ปุ่มกดบัตรคิว

เป็นอินพุทในการที่ส่วนควบคุมปุ่มเรียกบัตรคิวจะทราบว่ามีผู้ใช้บริการมาใช้บริการ พอมีอินพุทเข้ามาหน่วยประมวลผลจะตรวจสอบว่าผู้ใช้บริการกดปุ่มใดเพื่อที่ทราบได้ว่าผู้ใช้บริการมาใช้บริการด้านใด โดยในการออกแบบออกแบบให้มีปุ่มกดได้มากที่สุด 8 ปุ่มกด ซึ่งทำให้แยกลักษณะงานได้มากที่สุด คือ 8 ลักษณะงาน การตรวจสอบปุ่มกดจะใช้การ Polling เพื่อตรวจสอบ ถ้าเกิดกรณีที่มีการกด 2 ปุ่มพร้อมกันจะไม่พิจารณาการกดนั้น เมื่อตรวจสอบเจอการกดปุ่มระบบก็จะรอการส่ง address ของ master มาถึง เพื่อที่แจ้งให้ทราบว่ามีการกดปุ่มและให้ master ส่งคิวกลับมาเพื่อแสดงผลให้ผู้ใช้บริการทราบต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรหน่วยประมวลผลปุ่มเรียกบัตรคิว

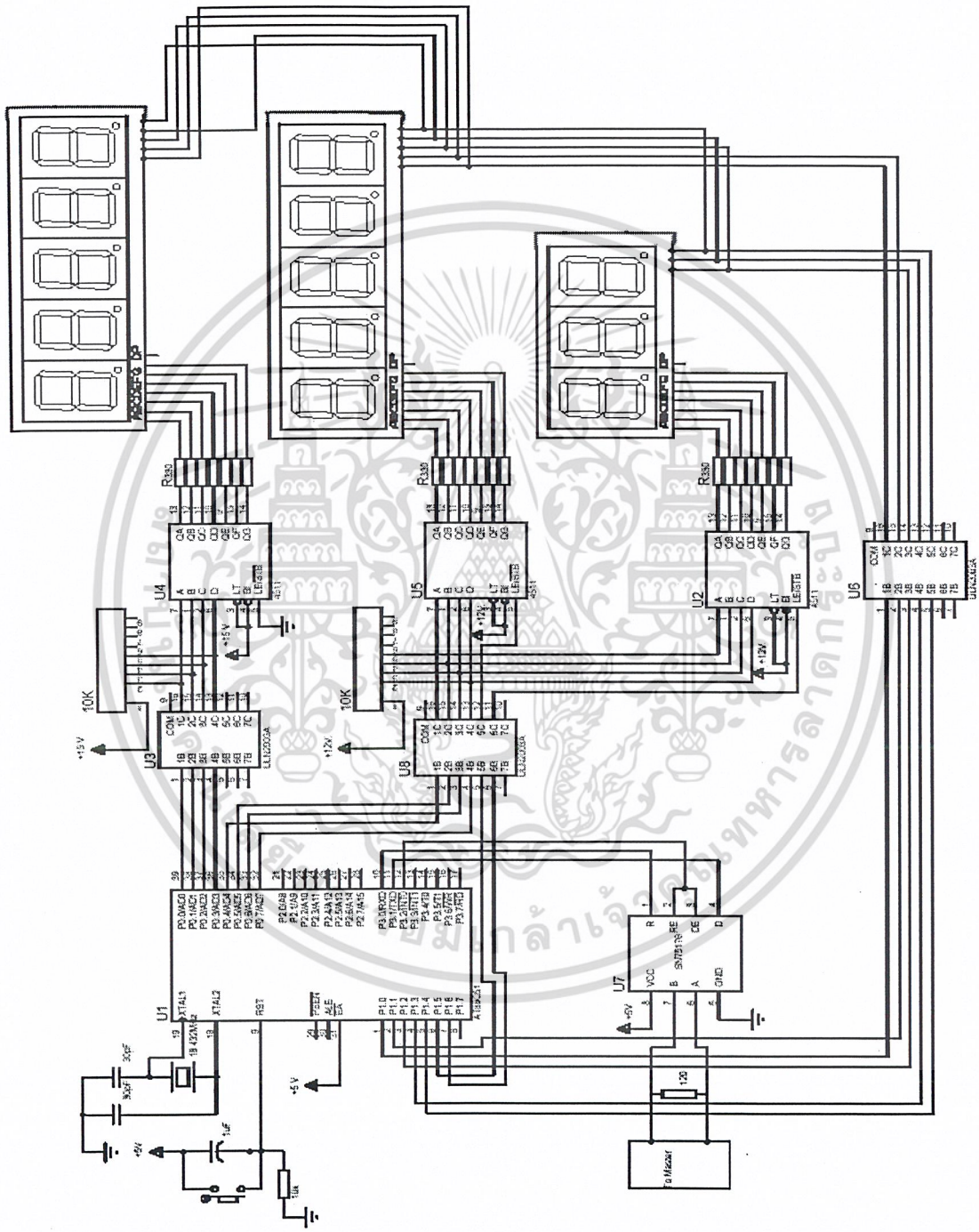
3.5 ส่วนแสดงผลหลัก (Main Display)

มีหน้าที่ในการแสดงหมายเลขคิวที่พร้อมจะให้บริการ โดยส่วนแสดงผลหลักจะประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 3.9 จะมีการแสดงหมายเลขช่องให้บริการ และหมายเลขคิวของผู้ใช้บริการ โดยจะแสดงเป็น 7-Seqment ชุดใหญ่กับชุดเล็ก โดย 7-Seqment ชุดใหญ่จะเป็นการแสดงคิวที่เพิ่งประกาศและจะแสดงเป็นตัวเลขกระพริบชั่วขณะและก็จะนิ่ง ส่วนชุดเล็กจะแสดงจะเป็นการแสดงคิวที่ประกาศไปแล้ว โดยถ้ามีการแสดงคิวใหม่เข้ามาคิวที่แสดงในชุดใหญ่ก็จะถูกเลื่อนไปแสดงใน 7-Seqment ชุดเล็ก ในการควบคุมการแสดงเนื่องจาก 7-Seqment ชุดใหญ่และชุดเล็กใช้แรงดันในการทำให้มีแสงออกมาต่างกันโดยชุดใหญ่ใช้แรงดันสูงประมาณ 12 V ซึ่งสูงกว่าแรงดันที่จ่ายให้กับ MCS-51 ซึ่งในการทำให้ 7-Seqment ติดได้จะใช้ IC CD4511 ในการขับ(driver) ซึ่งสามารถใช้ V_{DD} Supply สูงได้ถึง 15 V ทำให้มีปัญหาในการติดต่อกับ MSC-51 เพราะ V_{IL} และ V_{OH} ของ IC CD4511 สูงขึ้น ดังนั้นจึงเพิ่มส่วนของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Comparator เข้าไปเพื่อปรับระดับ Level TTL ส่วน 7-Seqment ชุดเล็กใช้ระดับแรงดันเดียวกับ MCS-51 ทำให้ไม่จำเป็นต้องปรับระดับ Level TTL ส่วนการควบคุมการแสดงผลจะมีลักษณะเดียวกับการแสดงผลของวงจรปุ่มเรียกบัตรคิว แต่จะใช้การแสดงผลหลักที่มากกว่าโดย 2 หลักแรกใช้การแสดงผลหมายเลขช่องให้บริการ และ 3 หลักหลังใช้แสดงหมายเลขคิวของผู้ใช้บริการ และการติดต่อกับ master ก็มีลักษณะเดียวกับส่วนอื่นคือ จะรอ master ส่ง address มาให้และถ้า address ตรงกันก็จะทำการส่งคำสั่งกลับไปเพื่อให้ master ส่งหมายเลขที่จะแสดงมาให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรส่วนแสดงผลหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ส่วนควบคุมการการกระจายเสียง (Speaker Unit)

การเชื่อมต่อไมโครคอนโทรเลอร์กับ ISD 2590 นั้นเป็นการเชื่อมต่อแบบขนานโดยที่จะมีการเชื่อมกันแบบบิตต่อบิต เมื่อเราต้องการให้ขาของ ISD 2590 มีค่าเป็นเท่าไรก็ส่งค่านั้นไปที่พอร์ตของไมโครคอนโทรเลอร์ที่ต่ออยู่กับขานั้นๆโดยที่การเชื่อมต่อจะเป็นดังนี้

$\overline{CE}$ ต่อกับ P0.0 เพื่อใช้เป็นตัวเลือกว่าจะให้ ISD 2590 ทำงานเมื่อใด

$P/\overline{R}$ ต่อกับ P0.1 เพื่อใช้เลือกว่าจะทำงานในส่วนของการบันทึกเสียงหรือการเล่นเสียง

PD ต่อกับ P0.2 เพื่อให้สัญญาณว่าต้องการเปิดการใช้งาน ISD 2590 หลังจากที่ Stand By อยู่

A0 ต่อกับ P0.6 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A1 ต่อกับ P0.7 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A2 ต่อกับ P2.7 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A3 ต่อกับ P2.6 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A4 ต่อกับ P2.5 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A5 ต่อกับ P2.4 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A6 ต่อกับ P2.3 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A7 ต่อกับ P2.2 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

A8 ต่อกับ P2.1 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

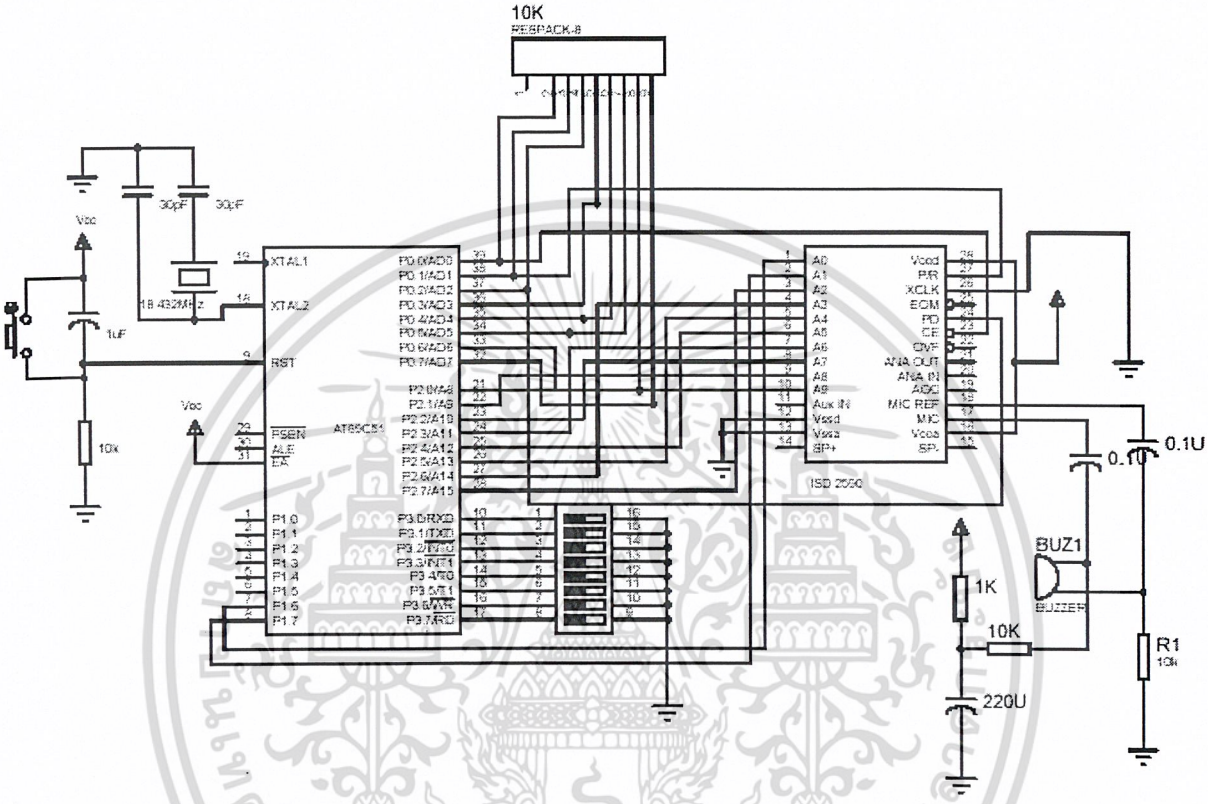
A9 ต่อกับ P2.0 ใช้บอกตำแหน่งของหน่วยความจำที่ต้องการเข้าถึงให้กับ ISD 2590

การออกแบบโปรแกรมสำหรับการใช้งาน ISD 2590

การออกแบบโปรแกรมสำหรับใช้งาน ISD 2590 นั้นจะต้องแบ่งออกเป็น 2 ส่วนคือเริ่มแรกนั้นต้องมีการบันทึกเสียงให้กับ ISD 2590 ก่อนเพื่อใช้เป็นข้อมูลให้กับการเรียกใช้ในการเล่นเสียงเมื่อนำเอาไปต่อในระบบจริง ส่วนที่สองก็คือการเล่นเสียงเมื่อนำไปต่อในระบบจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การออกแบบโปรแกรมสำหรับการบันทึกเสียง



รูปที่ 3.10 วงจรสำหรับการบันทึกเสียง

1. เริ่มแรกต้องให้ $P/\overline{R}$ เป็น 0 จากนั้นก็เลือก Address ที่จะบันทึกเสียงโดยการส่งค่า Address ให้กับ A0-A9 ของ ISD 2590

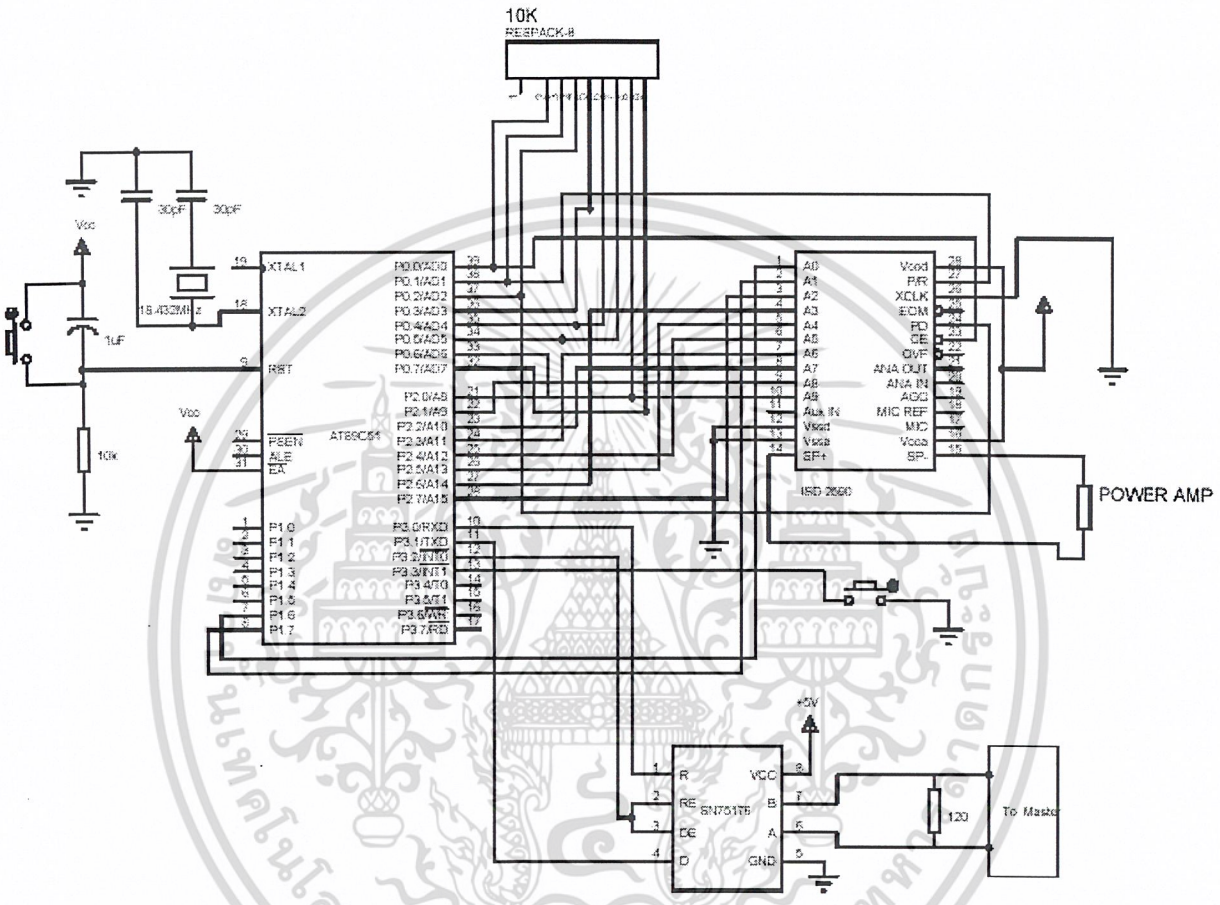
2. เปิดการทำงานของ ISD 2590 โดยที่ให้ PD เป็น 0 ประมาณ 37.5 ms จากนั้นเลือกให้ $\overline{CE}$ เป็น 0 เพื่อให้ ISD 2590 รับค่า Address เข้าไป แล้วก็เริ่มการบันทึกเสียง

3. เมื่อจบการบันทึกเสียงของแต่ละข้อความก็ให้ $\overline{CE}$ เป็น 1 และให้ PD เป็น 1 เช่นกันแล้ว Delay ประมาณ 100 ns เพื่อเข้าสู่โหมด Stand By

4. เริ่มต้นทำแบบเดิมเพื่อบันทึกเสียงที่ Address ที่กำหนดไว้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การออกแบบโปรแกรมสำหรับการเล่นเสียง



รูปที่ 3.11 วงจรการเชื่อมต่อกับ ISD 2590 สำหรับการใช้งานจริง

1. รอรับข้อมูลซึ่งก็คือหมายเลขของผู้ที่มาใช้บริการและหมายเลขของช่องให้บริการ(Counter) ที่ส่งมาจาก Master Controller

2. หลังจากนั้นก็นำข้อมูลที่ได้มาแยกเป็นหมายเลขของช่องให้บริการ(Counter) และหมายเลขของผู้ที่มาใช้บริการเป็นหลักหน่วย สิบ และร้อย

3. หลังจากได้หมายเลขมาเป็นเลขเดี่ยวๆแล้วก็นำมาเช็คว่าตัวเลขที่ได้นั้นอยู่ที่ Address ใดของ ISD 2590 แล้วก็เก็บค่านั้นเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. พอทราบ Address แล้วก็นำค่า Address มาส่งไปที่ ISD 2590 เพื่อเป็นตำแหน่งที่ต้องการให้ ISD 2590 ดึงข้อมูลออกมาเล่นโดยเริ่มจากให้ PD เป็น 0 เพื่อเปิดการทำงานของ ISD 2590 แล้ว Delay ประมาณ 37.5 ms จากนั้นก็ให้ $\overline{CE}$ เป็น 0 เพื่อให้ ISD 2590 รับค่า Address เข้าไปจากนั้น ISD 2590 ก็จะเล่นเสียงที่เก็บอยู่ใน Address นั้นออกมาช่วงเวลานั้นเราต้อง Delay ให้พอเหมาะกับเวลาของแต่ละข้อความ

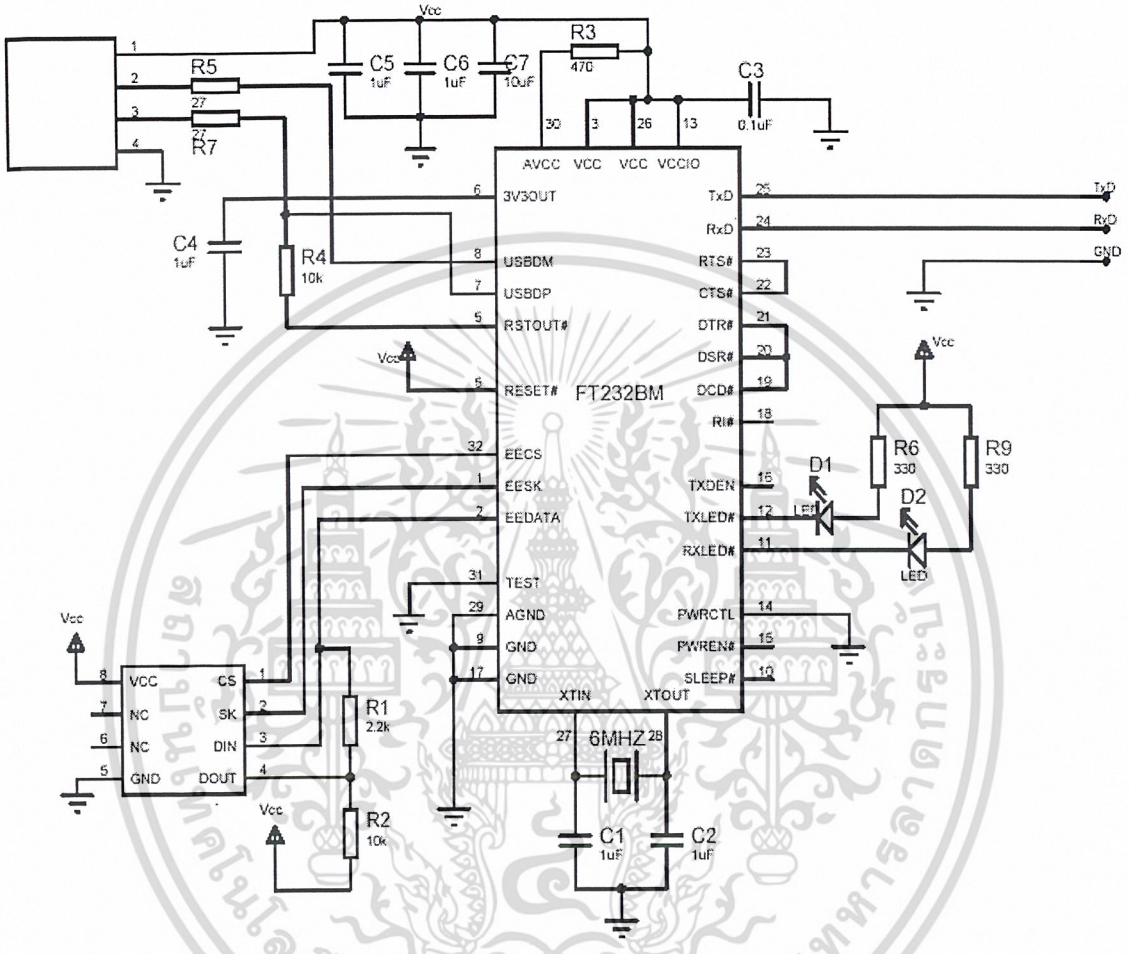
5. หลังจากที่ข้อความแรกเล่นจบไปแล้วก็จะให้ $\overline{CE}$ เป็น 1 และให้ PD เป็น 1 เช่นกันเพื่อรอการสั่งงานครั้งต่อไป และไมโครคอนโทรเลอร์ก็จะวนส่ง Address ให้กับ ISD 2590 จนหมดข้อมูลที่ได้รับมา ซึ่งสรุปแล้วเสียงที่เล่นออกมาก็จะมีดังนี้เช่น “เชิญ” , “หมายเลข” , “สอง” , “สาม” , “สี่” , “ที่ช่องบริการที่” , “หนึ่ง” , “ค่ะ”

6. จากนั้นไมโครคอนโทรเลอร์ก็จะรอรับข้อมูลใหม่จาก Master Controller ต่อไป

3.7 การเชื่อมต่อส่วนประมวลผลกลางไปยังคอมพิวเตอร์ผ่าน USB Port

การเชื่อมต่อจะใช้ IC FT232BM ซึ่งเป็นโมดูลที่สร้างโปรโตคอลการเชื่อมต่อ USB โดยจะทำการแปลงข้อมูลที่ถูกส่งมาจากหน่วยประมวลผลกลางให้เป็นมาตรฐานของการเชื่อมต่อแบบ USB เพื่อส่งไปยังคอมพิวเตอร์และในทางกลับกันก็แปลงข้อมูลที่ส่งมาจากคอมพิวเตอร์เป็นมาตรฐานแบบ UART และส่งไปยังหน่วยประมวลผลกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 รูปแบบวงจรการเชื่อมต่อ USB

จากวงจรในรูปที่ 3.12 นี้ใช้กำลังงานในการทำงานจากบัสของ USB และการเชื่อมต่อ USB นี้ก็เป็นการเชื่อมต่อโดยใช้มาตรฐาน USB 2.0 โดยการกำหนดคุณสมบัติการทำงานของวงจรสามารถกำหนดได้จากการใส่ค่าพารามิเตอร์ต่างๆของ USB ไว้ใน EEPROM 93C46 โดย EEPROM นี้สามารถโปรแกรมได้จากคอมพิวเตอร์ได้โดยตรงผ่านโมดูลตัวนี้ได้เลย โดยสามารถแก้ไขค่าสตริงดีสคริปเตอร์ , ซีเรียลนัมเบอร์ , เวนเดอร์ไอดี(PID)และโพรดักซ์ไอดี(PID) และการติดต่อระหว่าง IC กับส่วนประมวลผลกลางจะเป็นการเชื่อมต่อแบบ 3 wires loop back

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

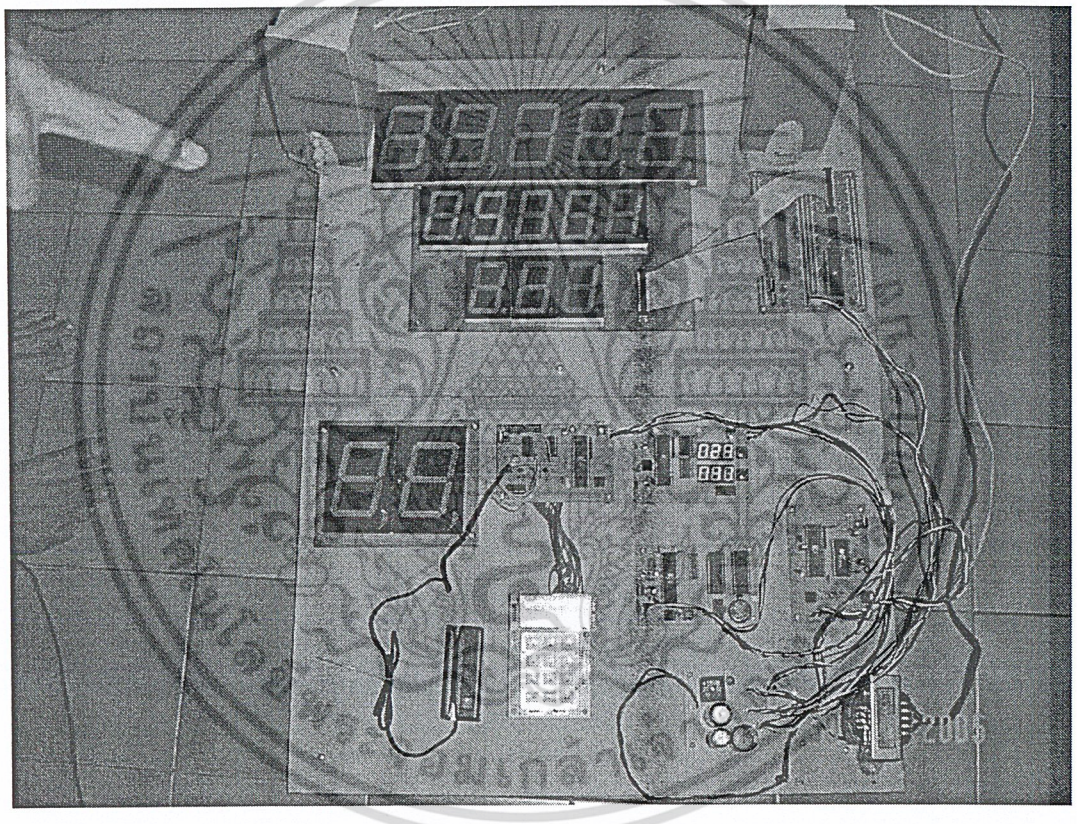
บทที่ 4

ผลการทดสอบ

1. หน่วยประมวลผลหลัก
   - สามารถควบคุมการติดต่อสื่อสารแบบ Multiprocessor ได้อย่างดี
2. ส่วนแจกคิว
   - สามารถแจกคิวโดยแบ่งตามหน้าที่ได้ 2 หน้าที่คือ
   1. หมายเลข 0-100 ทำหน้าที่ที่ 1
   2. หมายเลข 101-200 ทำหน้าที่ที่ 2
   - สามารถแสดงหมายเลขคิวของผู้ที่มาใช้บริการรวมทั้งแสดงจำนวนคนที่รอด้วย
3. ส่วนให้บริการ
   - สามารถตั้งค่าหมายเลข Counter ได้
   - สามารถตั้งค่าหน้าที่ที่ให้บริการได้
   - สามารถเรียกหมายเลขคิวได้โดยจะแบ่ง Counter ออกเป็นหน้าที่ตามการตั้งค่าตอนเริ่มต้นเปิดระบบ
   - สามารถขอผ่านคิวที่กำลังให้บริการอยู่ได้
   - สามารถเรียกคิวที่ผ่านไปแล้วใหม่ได้
4. ส่วนแสดงผลหลัก
   - สามารถแสดงผลหมายเลขที่กำลังเรียกอยู่ได้
   - สามารถแสดงผลหมายเลขที่เรียกก่อนหน้านี้ได้
   - สามารถแสดงจำนวนผู้ที่มาใช้บริการที่รอกำลังรออยู่ได้
5. ส่วนแสดงเสียงพูด
   - สามารถใช้เสียงเรียกผู้ที่มาใช้บริการได้โดยที่หมายเลขที่เรียกจะเป็นหมายเลขเดียวกันกับส่วนแสดงผลหลัก
6. ส่วนคอมพิวเตอร์
   - นำข้อมูลการทำงานของระบบมาเก็บใน Database ของคอมพิวเตอร์ได้ โดยเก็บเป็น Database ของโปรแกรม Microsoft Access

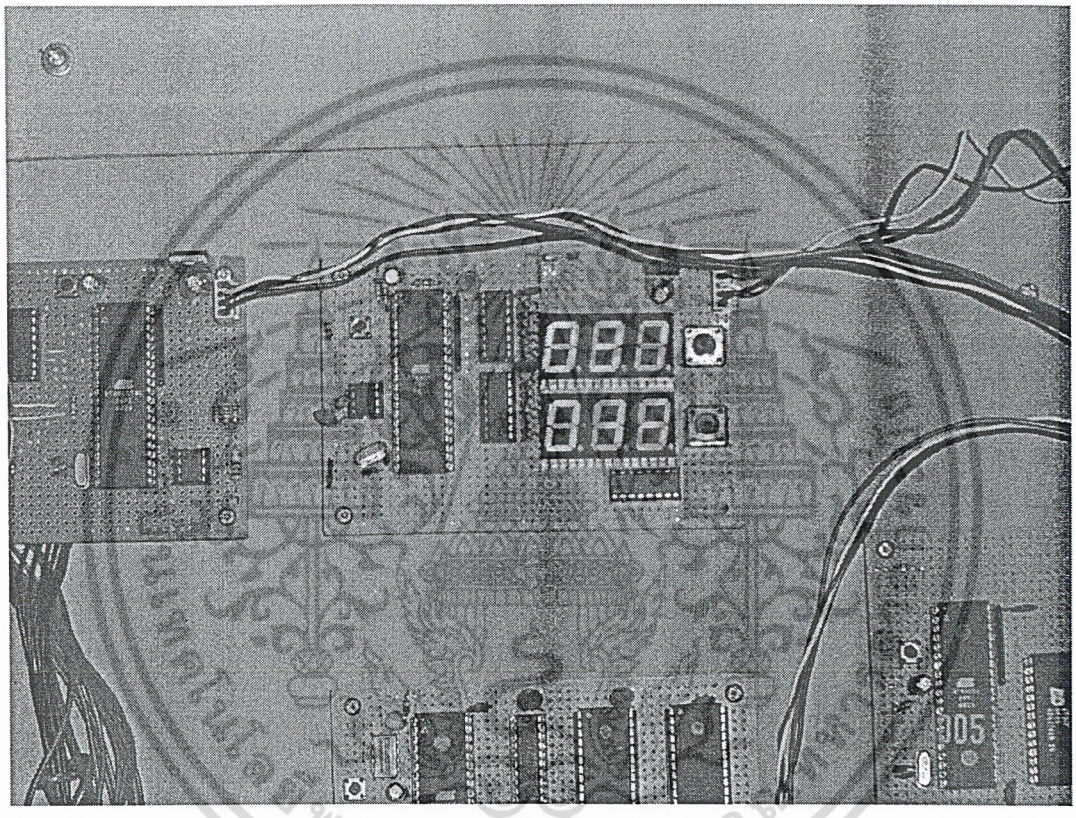
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถตั้งค่าช่วงของหน้าที่ของการให้บริการได้



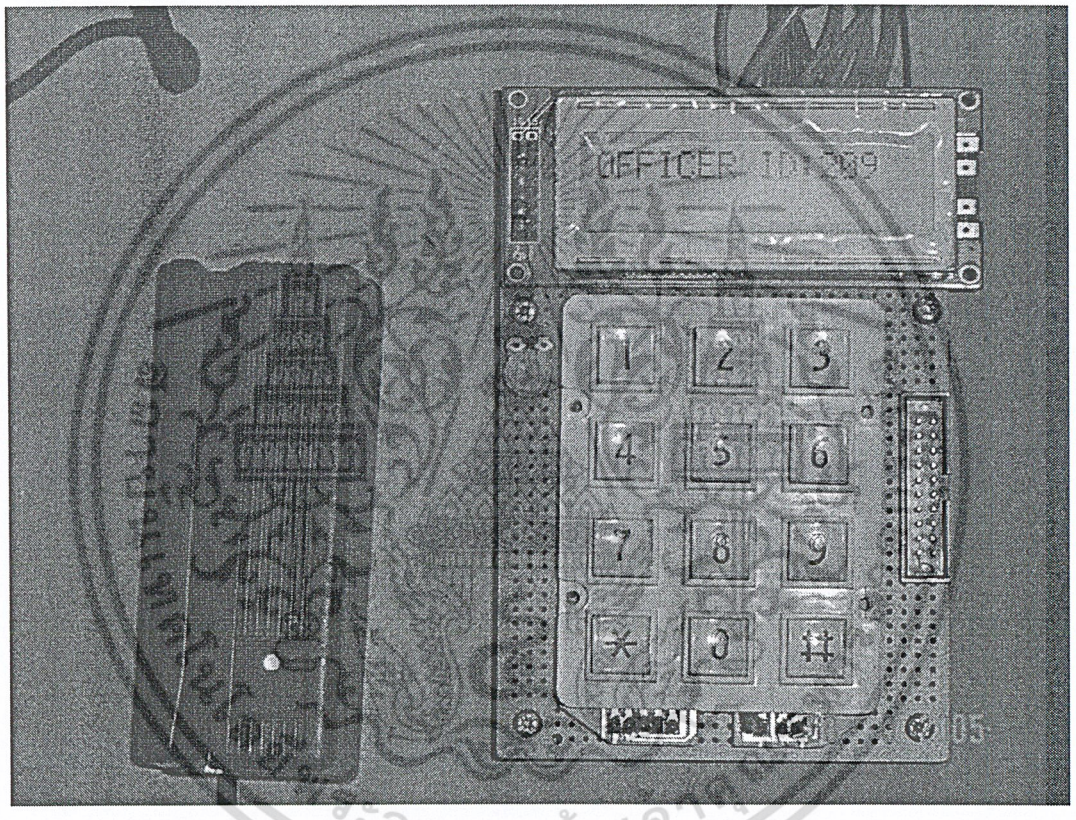
รูปที่ 4.1 การเชื่อมต่อของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



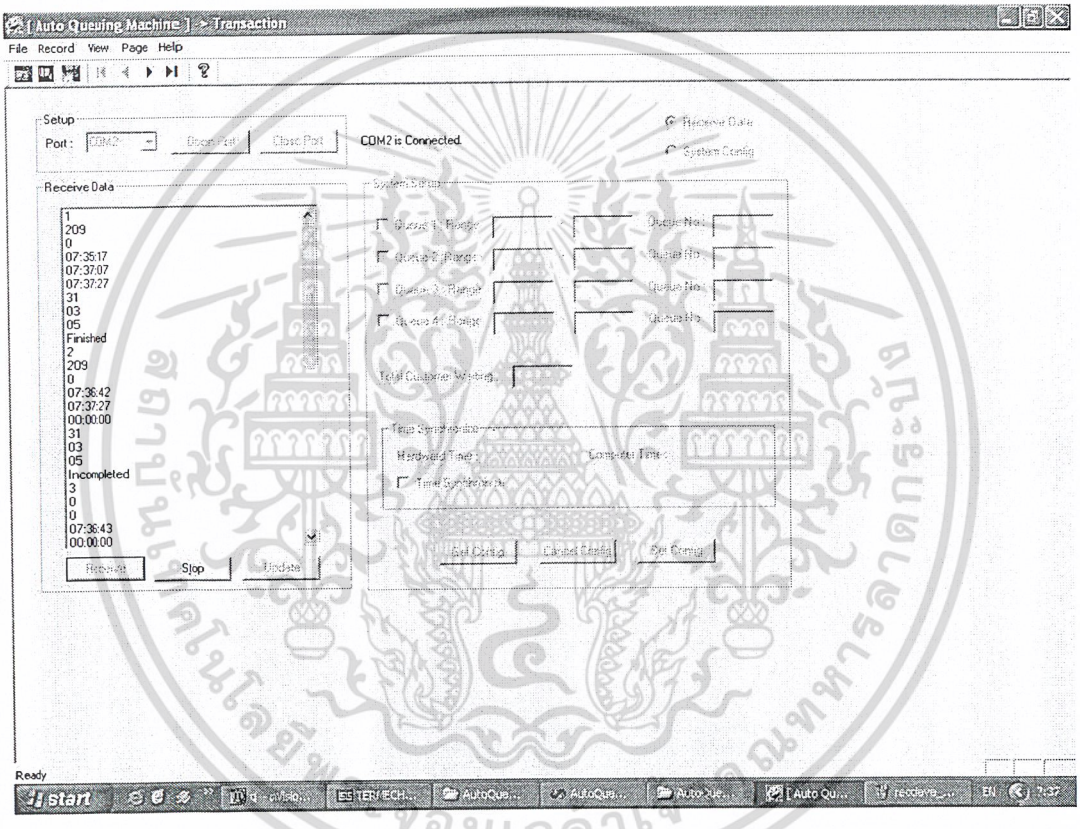
รูปที่ 4.2 การทำงานของส่วนแจกคิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



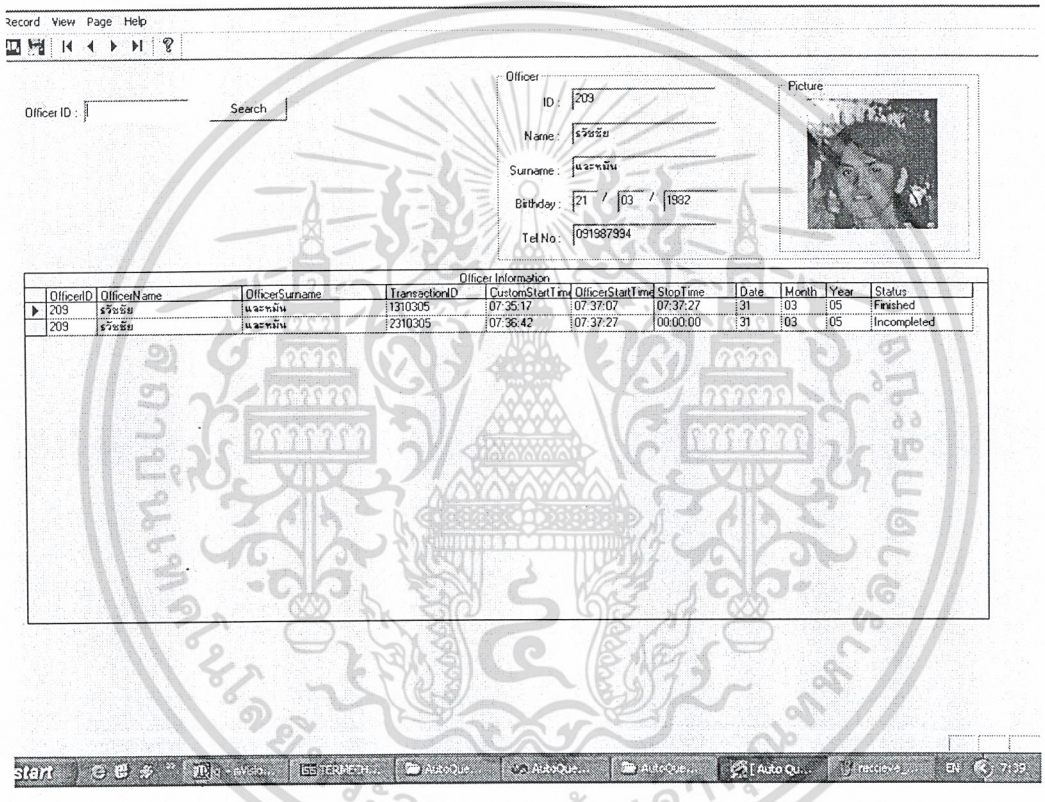
รูปที่ 4.3 การทำงานของส่วนรับบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



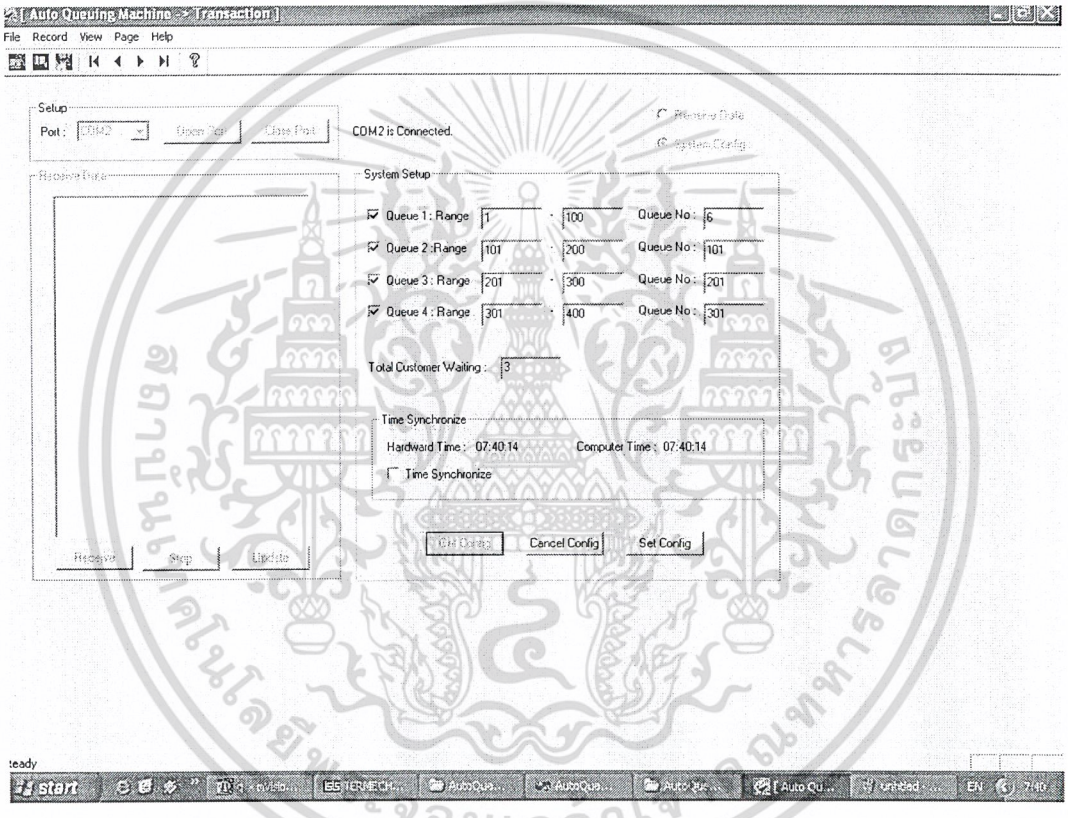
รูปที่ 4.4 แสดงการรับข้อมูลจากระบบของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



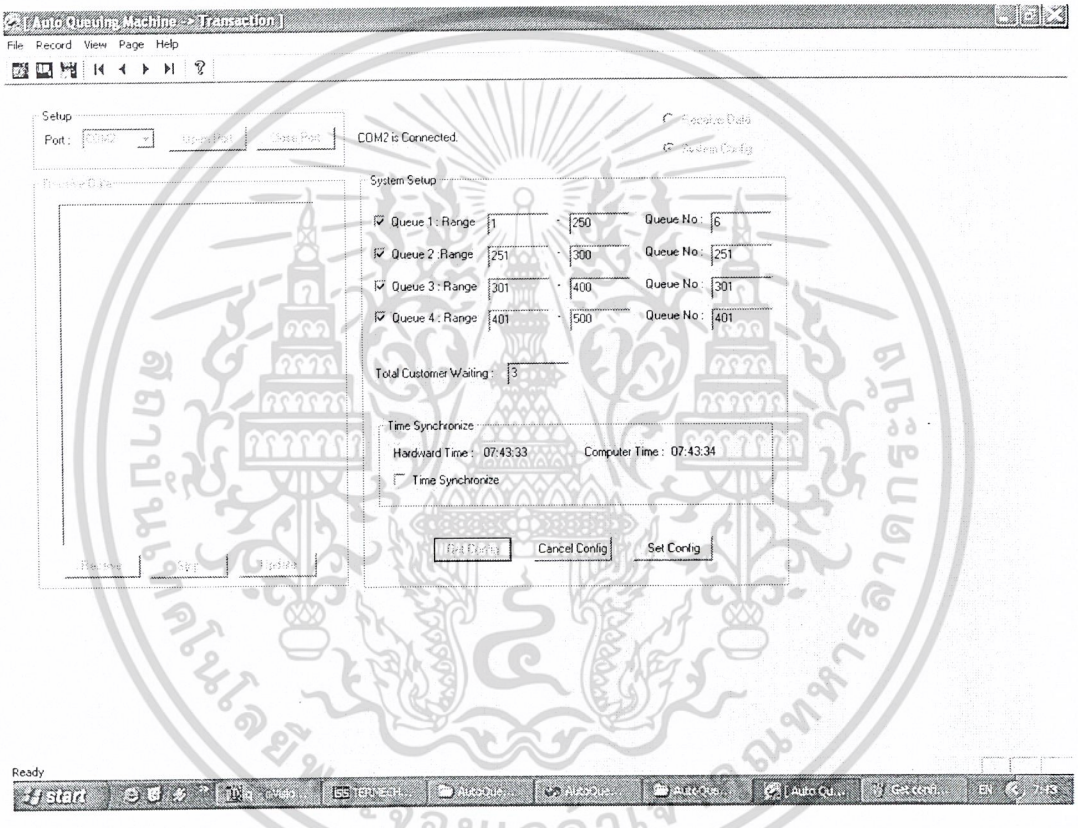
รูปที่ 4.5 แสดงผลการเก็บค่าข้อมูลที่ได้ไปไว้ที่ฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงการเรียกดูค่าเริ่มต้นของระบบผ่านคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงค่าเริ่มต้นของระบบที่ถูกตั้งค่าใหม่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุป ประเมิน และวิจารณ์ผลการทำโครงงาน

5.1 สรุปการทำโครงงาน

จากการทำโครงงานเครื่องจัดลำดับอัตโนมัติ สามารถสรุปโครงสร้างของการทำงานในส่วนต่างๆได้ดังนี้

การทำงานของระบบจะเป็นแบบ Multiprocessing ที่มีส่วนประกอบคือไมโครคอนโทรเลอร์ที่ทำหน้าที่เป็น Master 1 ตัว ที่เหลือจะเป็นไมโครคอนโทรเลอร์ที่ทำหน้าที่เป็น Slave และอุปกรณ์ต่างๆ ที่เชื่อมต่ออยู่กับ Slave เช่นส่วนแสดงผลหลักอีกทั้งยังรวมไปถึงคอมพิวเตอร์ที่เชื่อมต่ออยู่กับตัว Master ด้วยเช่นกัน โดยเริ่มแรก Master จะทำการวนส่ง Address ของ Slave ทุกตัวเพื่อถามว่ามีความต้องการข้อมูลชนิดใดหรือไม่ถ้ามี Master ก็จะส่งข้อมูลนั้นไปให้ แต่ถ้าไม่มี Master ก็จะส่ง Address ของตัวต่อๆไป ทำซ้ำอย่างนี้ไปเรื่อยๆ ในส่วนของ Slave ก็จะรอการส่ง Address มาจาก Master เมื่อได้ Address มาแล้วก็จะนำมาตรวจดูว่าใช่ Address ของตัวเองหรือไม่ ถ้าใช่ก็จะส่งคำขอกลับไปที่ Master เพื่อขอข้อมูลจากนั้นก็จะรอรับข้อมูลที่จะส่งมาจาก Master ต่อไป แต่ถ้าไม่ใช่ Address ของตัวเองก็จะไม่ส่งอะไรออกไปเลยและจะไม่รับข้อมูลของ Slave ตัวอื่นที่ถูกส่งมาจาก Master ด้วยแต่จะรอจนกว่าจะมีการส่ง Address ครั้งใหม่มาจึงจะนำมาตรวจอีกทีว่าใช่ Address ของตัวเองหรือไม่ หลังจากที่ Slave ได้รับข้อมูลจาก Master มาแล้วก็จะนำข้อมูลที่ได้นั้นมาประมวลผลตามหน้าที่ของตัวเองเช่นนำมาประมวลผลเพื่อแสดงเป็นส่วนแสดงผลเรียกผู้ที่มาใช้บริการ

การรับส่งระหว่าง Master กับ Slave จะมีตลอดเวลา เมื่อปริมาณข้อมูลที่มีการส่งมากจึงมีโอกาสที่จะผิดพลาดได้มาก ดังนั้นผู้จัดทำจึงออกแบบระบบรักษาความปลอดภัยของข้อมูลที่ส่งโดยให้มีการส่งข้อมูลเดียวกัน 3 ครั้งไม่ว่าจะเป็นการส่งข้อมูลจาก Master ไปหา Slave หรือการส่งข้อมูลจาก Slave ไปหา Master จากนั้นนำข้อมูลที่ได้มาเปรียบเทียบกันทั้งหมด ถ้าเท่ากันทั้งหมดถือว่าข้อมูลที่ได้รับถูกต้องหรือถ้าเหมือนกัน 2 ตัวก็จะถือว่าเป็นข้อมูลที่ถูกต้องเช่นกัน และจากการทดสอบพบว่าการรับส่งเป็นไปได้ด้วยดี

5.2 วิจารณ์การทำงาน

ในการทำโครงงานนี้ผู้ทำได้รับความรู้ต่างๆมากมายเกี่ยวกับระบบการทำงานของไมโครคอนโทรเลอร์หลายตัว การติดต่อสื่อสารด้วยมาตรฐาน RS-485 และทราบว่าสัญญาณรบกวนมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลอย่างมากต่อการส่งสัญญาณซึ่งทำให้การสื่อสารผิดพลาดหรืออาจถึงขั้นล้มเหลวได้ ได้เรียนรู้ปัญหาในการส่งสัญญาณผ่านสายสัญญาณและการแก้ปัญหาในการรับส่งข้อมูล

ในส่วนของการเขียนโปรแกรมใช้งานบนไมโครคอนโทรเลอร์ยังมีข้อเสียอยู่ที่การเขียนโปรแกรมยังไม่มีการวางแผนระบบที่ครอบคลุมทั้งหมดดังนั้นเมื่อเขียนโปรแกรมขึ้นมา 1 Label จะไม่สามารถนำไปใช้กับโปรแกรมอื่นๆได้ จึงควรจะมีการวางแผนภาพรวมของระบบทั้งหมดโดยละเอียดก่อนที่จะมีการเขียนโปรแกรมโดยที่ควรจะกำหนดว่าตัวแปรชื่ออะไรจะใช้ทำอะไร , หน่วยความจำที่ตำแหน่งไหนจะใช้เก็บข้อมูลใด และในแต่ละตัวจะมีการใช้งานฟังก์ชันใดของไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ธีรวัฒน์ ประกอบผล. **การพัฒนาไมโครคอนโทรเลอร์ด้วยภาษาซี** กรุงเทพมหานคร : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , 2545.

วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51** กรุงเทพมหานคร : บริษัทอินโนเวตีฟ เอ็กเพอรริเมนต์ จำกัด , 2545.

วรเทพ ไพบูลรัตนากรณ์ และ บุญอนันต์ เกียงเอีย. **สัมผัสโลก USB ด้วย USB MODULE** : บริษัท แอสทรอน ลอจิก รีเสิร์ชแอนด์ดิวิลอปเมนต์ จำกัด.

ยุธทนา ลีลาศวัฒนกุล. **คู่มือการเขียนโปรแกรมและใช้งาน Visual C++. NET ฉบับสมบูรณ์ :** Infopress Developer Book.

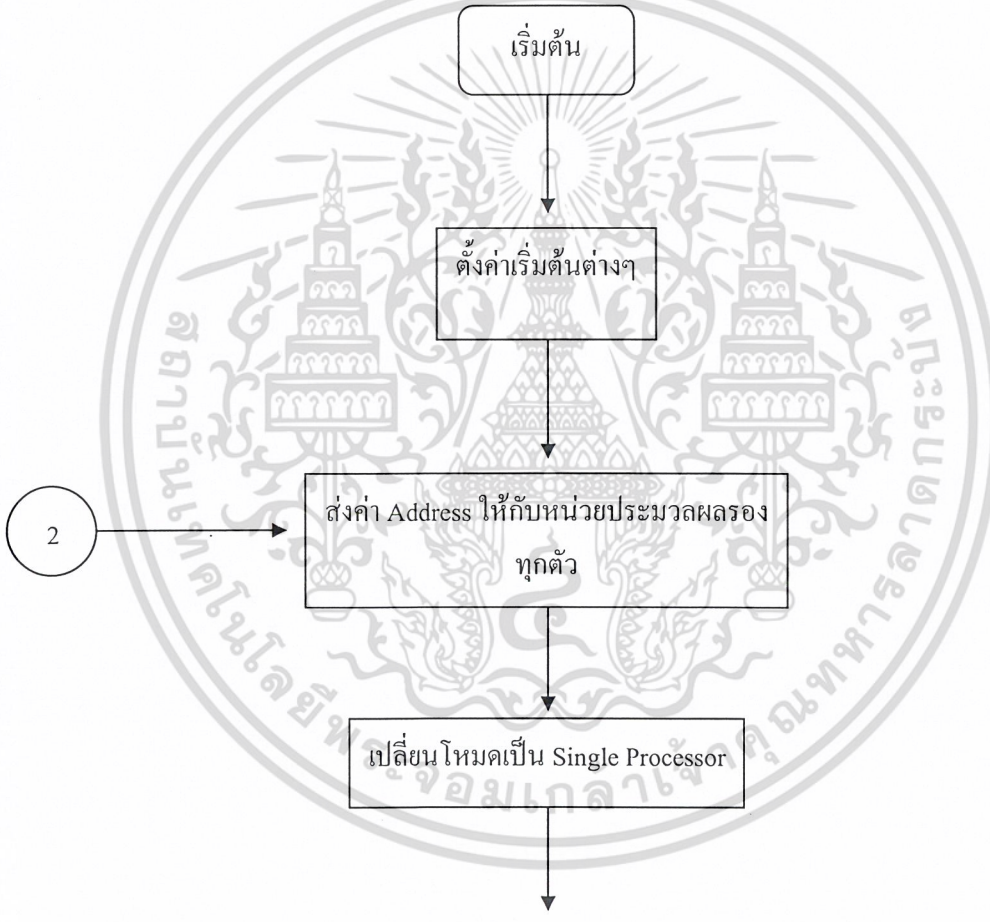
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

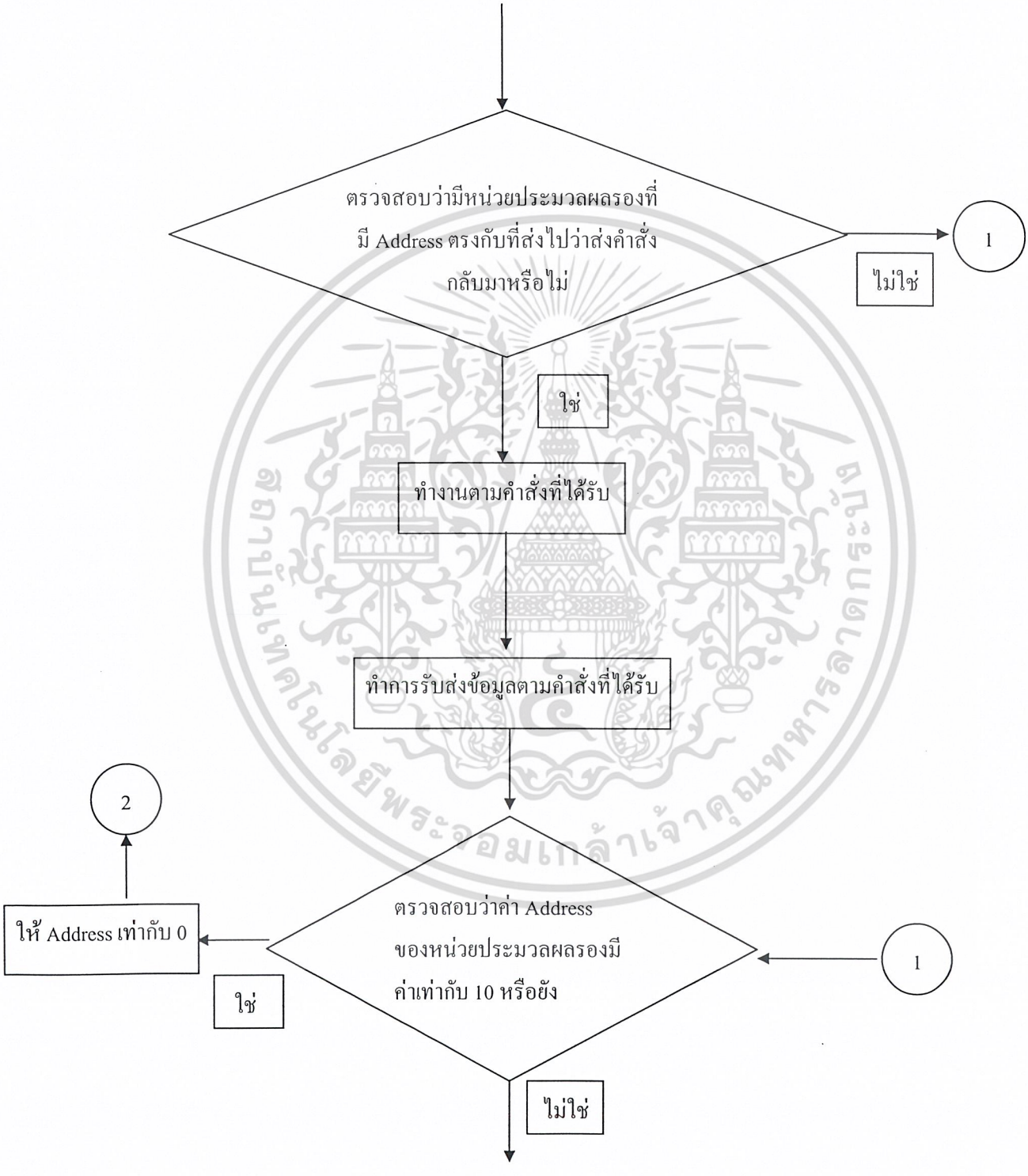
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOW CHART

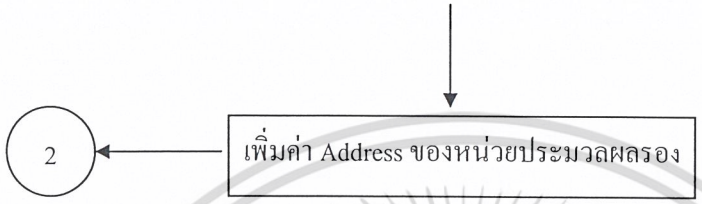
การออกแบบโปรแกรมหน่วยประมวลผลหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

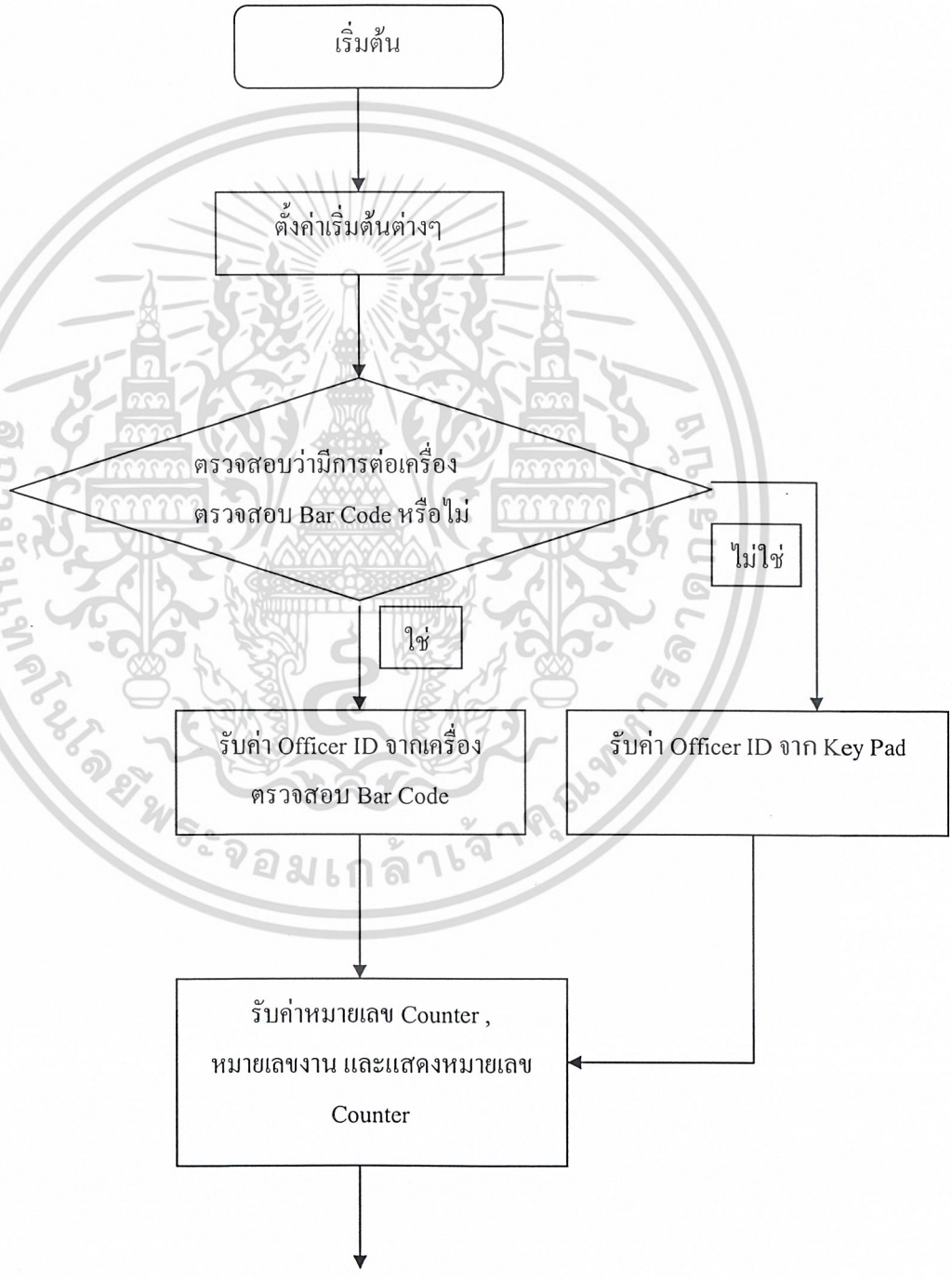


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

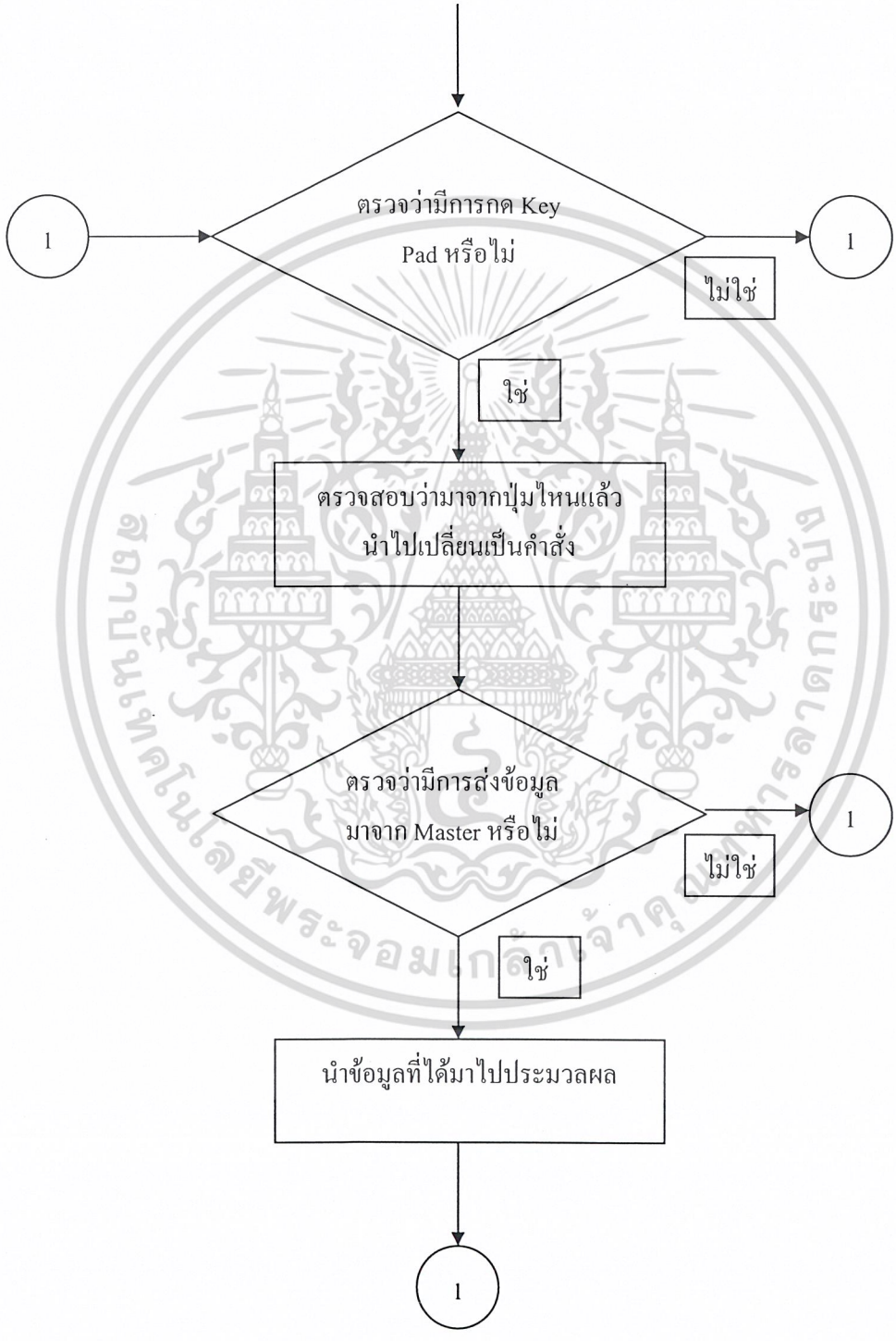


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมส่วนให้บริการ

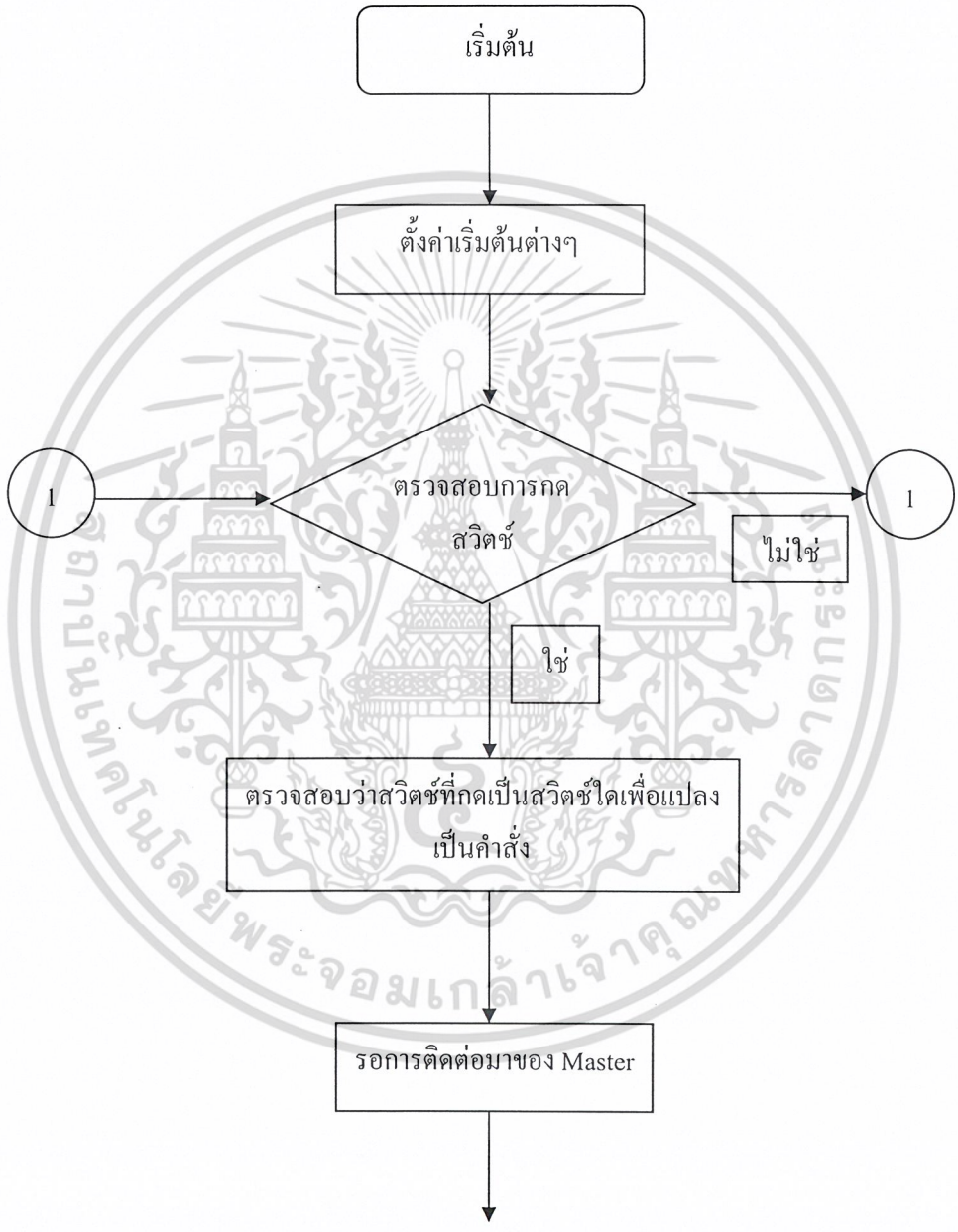


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

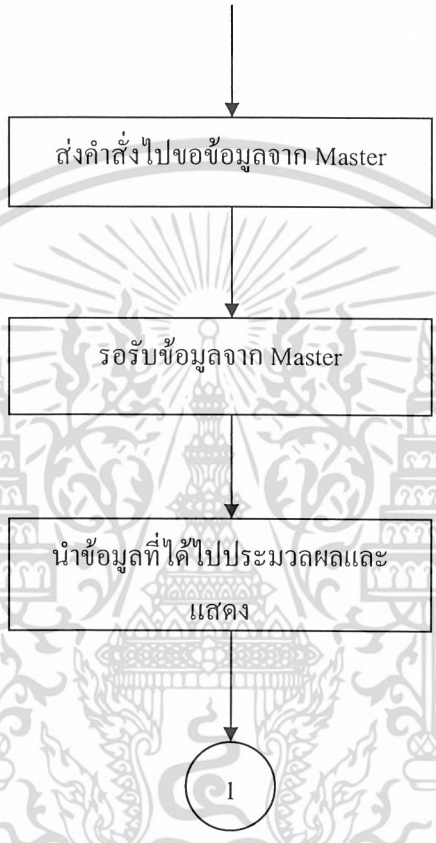


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมส่วนแจกคิว

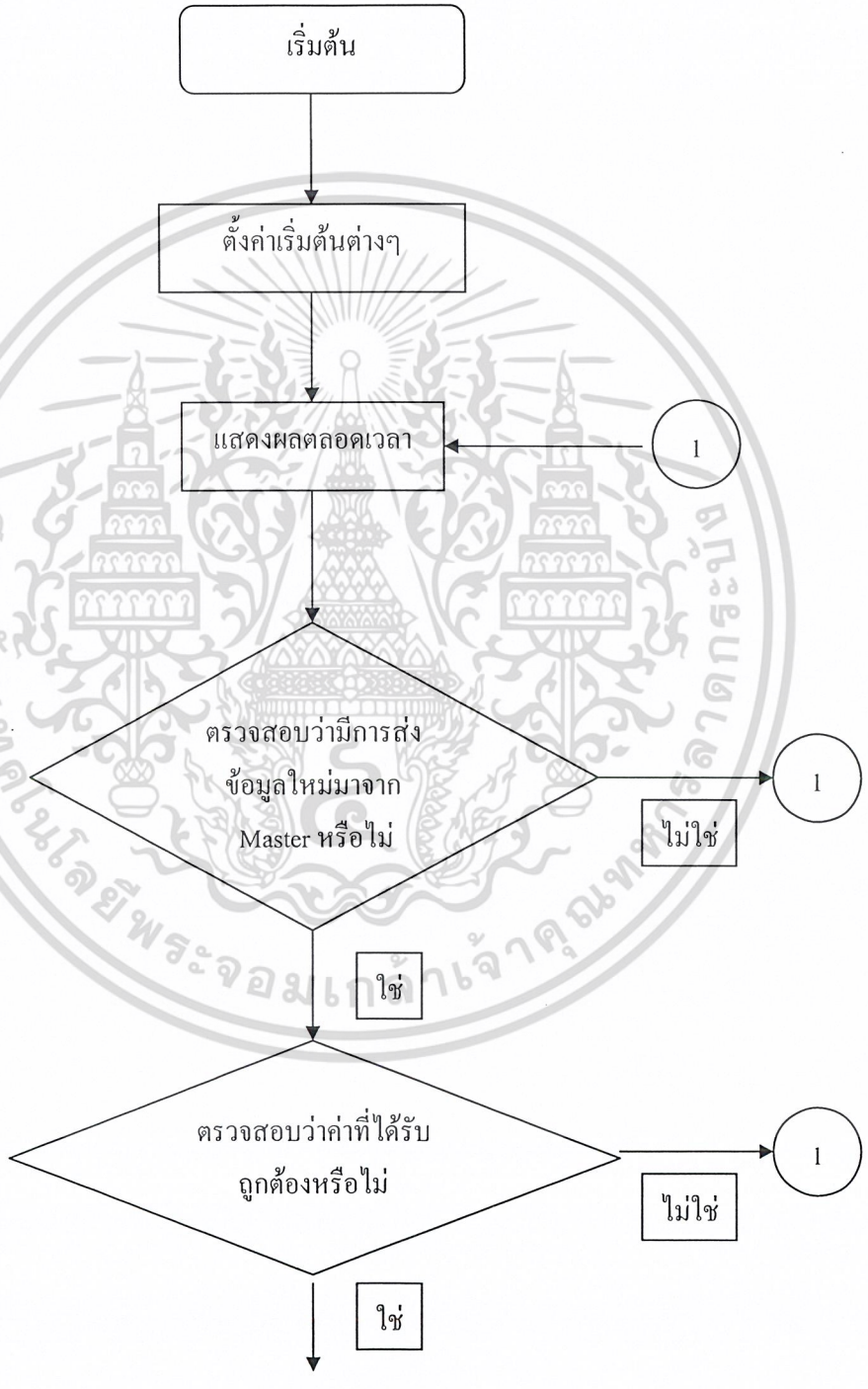


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

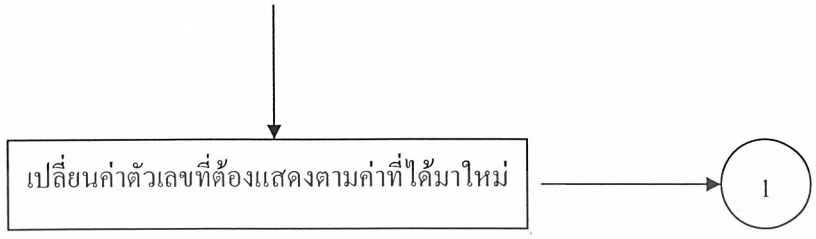


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมหน่วยแสดงผลหลัก

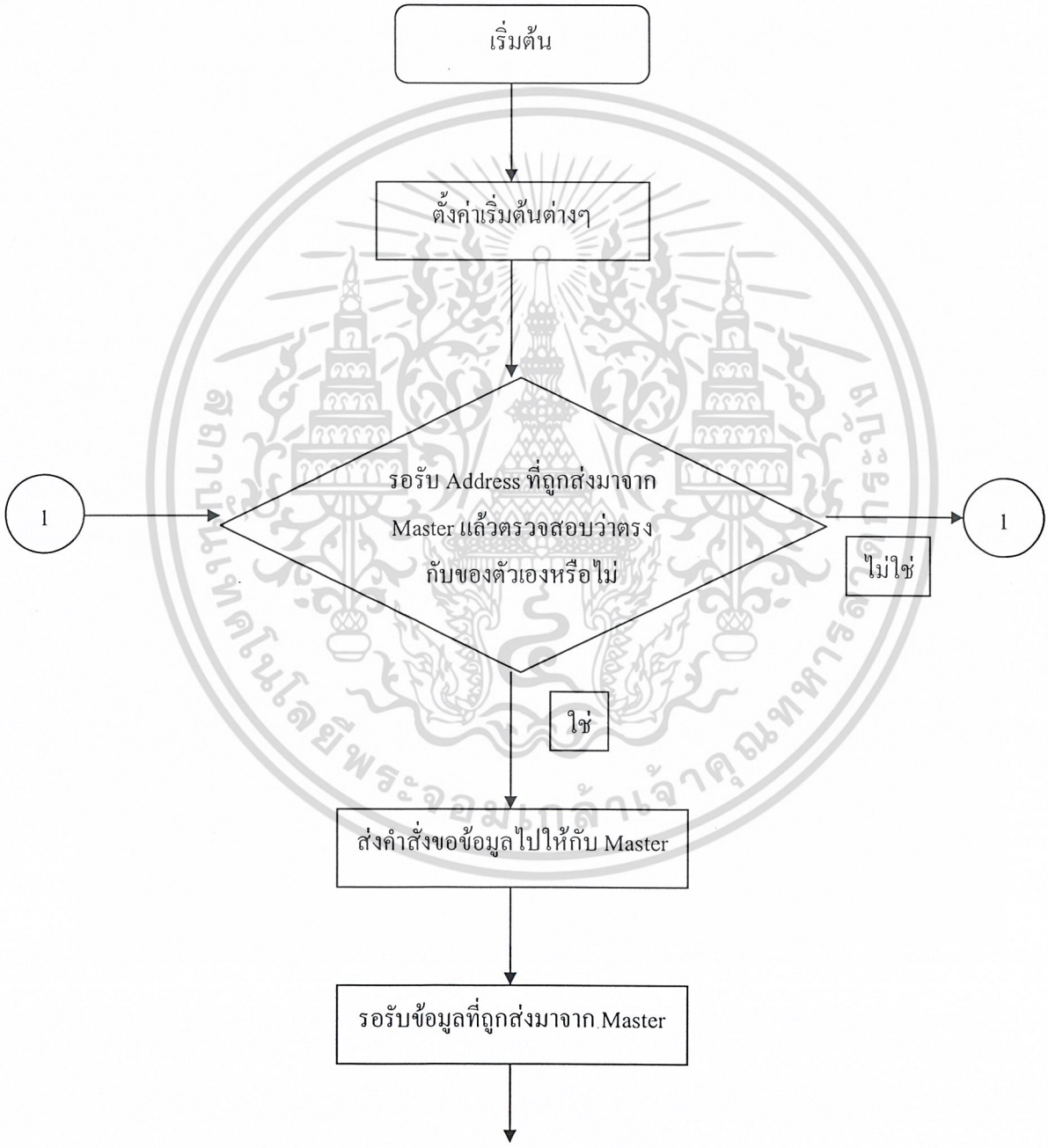


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

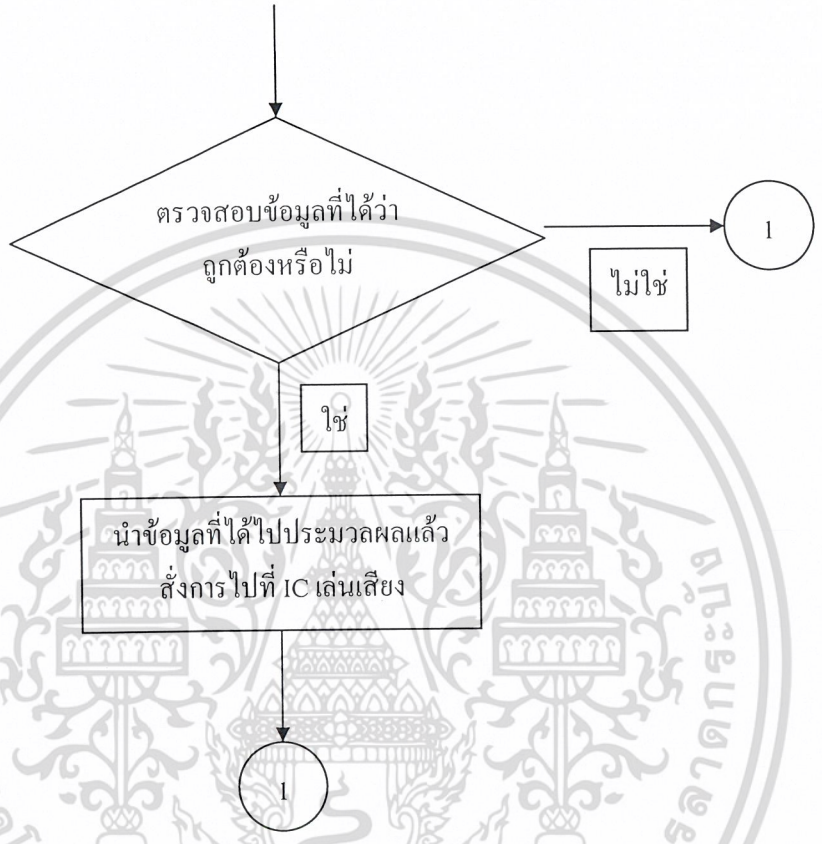


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบโปรแกรมส่วนกระจายเสียง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้