



ภาควิชาครุศาสตร์วิศวกรรม
 คณะครุศาสตร์อุตสาหกรรม
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใบรับรองปริญญาโท

ชื่อหัวข้อ โครงการรวบรวมข้อมูลระยะไกล
 Data Logger

ชื่อนักศึกษา	1. นายพดล	ภูมิดิษฐ์	รหัสประจำตัว	42035305
	2. นายรัฐธรรมนูลู	สุขมีศรี	รหัสประจำตัว	42035315
	3. นายเอกภพ	แสนสวาท	รหัสประจำตัว	42035324
	4. นายเอกรัฐ	ปัญญาเทพ	รหัสประจำตัว	42035325

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์
 อาจารย์ที่ปรึกษา ดร.สุรสิทธิ์ ราตรี
 อาจารย์ที่ปรึกษาร่วม อาจารย์โกศล ตราชู

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. ดร.สุรสิทธิ์ ราตรี
2. อาจารย์โกศล ตราชู
3. อาจารย์สุชิน อจหาญ
4. อาจารย์ไพบูลย์ พวงวงศ์ตระกูล
5. อาจารย์สุระชัย พิมพ์สาลี

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 10.15 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
 ลงนาม.....
 (ผศ.วิสุทธิ อธิพรธรรม)
 หัวหน้าภาควิชาครุศาสตร์วิศวกรรม
 วันที่ 14 เดือน กพ. พ.ศ. 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปฏิญานิพนธ์

โครงข่ายรวบรวมข้อมูลระยะไกล

DATA LOGGER



นายนพดล	ภูมิดิษฐ์
นายรัฐธรรมนุญ	สุขมีศรี
นายเอกภพ	แสนสวาท
นายเอกรัฐ	ปัญญาเทพ

ปฏิญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

40161
1109.2233
2543

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

b. 1109.2233

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง โครงข่ายรวบรวมข้อมูลระยะไกล

Data Logger

วัตถุประสงค์

1. เพื่อศึกษาการนำอุปกรณ์อิเล็กทรอนิกส์มาประยุกต์ใช้ในการรับส่งข้อมูลระยะไกล
2. เพื่อออกแบบโครงข่ายรวบรวมข้อมูลระยะไกล
3. เพื่อสร้างโครงข่ายรวบรวมข้อมูลระยะไกล
4. เพื่อทดสอบประสิทธิภาพการทำงานของโครงข่ายรวบรวมข้อมูลระยะไกล
5. เพื่อนำโครงข่ายรวบรวมข้อมูลระยะไกลไปใช้ในงานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับการนำอุปกรณ์อิเล็กทรอนิกส์มาประยุกต์ใช้ในการรับส่งข้อมูลระยะไกล
2. ได้วางจรรยาบรรณของโครงข่ายรวบรวมข้อมูลระยะไกล
3. ได้เครื่องต้นแบบของโครงข่ายรวบรวมข้อมูลระยะไกล
4. ได้ผลการทดสอบประสิทธิภาพของโครงข่ายรวบรวมข้อมูลระยะไกล
5. สามารถนำเครื่องต้นแบบของโครงข่ายรวบรวมข้อมูลระยะไกลไปใช้งานได้จริง

ชื่อหัวข้อ

โครงข่ายรวบรวมข้อมูลระยะไกล

นักศึกษา

นายนพดล ภูมิดิษฐ์

นายรัฐธรรมนุญ สุขมีศรี

นายเอกภพ แสนสวาท

นายเอกรัฐ ปัญญาเทพ

อาจารย์ที่ปรึกษา

ดร.สุรสิทธิ์ ราตรี

อาจารย์ที่ปรึกษาร่วม

อาจารย์โกศล ตราชู

หลักสูตร

วิศวกรรมศาสตรบัณฑิต

สาขาวิชา

อิเล็กทรอนิกส์และคอมพิวเตอร์

ปีการศึกษา

2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการนำเสนอ โครงข่ายรวบรวมข้อมูลระยะไกล ซึ่งเป็นการเก็บรวบรวมข้อมูลด้วยระบบคอมพิวเตอร์อัตโนมัติ และบันทึกสถิติด้วยคอมพิวเตอร์ โครงข่ายรวบรวมข้อมูลระยะไกล ทำงานโดยการเขียนโปรแกรมวิซวลเบสิกให้ควบคุมวงจรที่ใช้รวบรวมข้อมูล โดยใช้การติดต่อสื่อสารผ่านโมเด็ม ไปยังระบบไมโครคอนโทรลเลอร์ ซึ่งทำหน้าที่เก็บรวบรวมข้อมูลที่อยู่ห่างไกลออกไป ซึ่งระบบไมโครคอนโทรลเลอร์จะทำการเก็บข้อมูลไว้ในหน่วยความจำเพื่อรอส่งให้กับระบบโดยอัตโนมัติ

II

Thesis Title	Data Logger
Student	Mr.Noppadol Pundist Mr.Rattaamnoon Sukmesri Mr.Eakkapop Sansawat Mr.Eakkarath Panyathep
Advisor	Dr.Surasit Ratre
Co-advisor	Mr.Koson Trachu
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	2000

ABSTRACT

This thesis is present to "Data Logger", this one is storage data and record static automatically. Data Logger working by using Visual Basic Program to manage terminal which have storage data function. Its communicate by using MODEM connect between central computer and micro controller. Function of micro controller is put the data in to memory before send to central computer. When in a time, micro controller will be send data in memory to central computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้ด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน และด้วยอนุเคราะห์ของท่านอาจารย์ที่ปรึกษาปริญญานิพนธ์ และอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุก ๆ ท่าน ที่กรุณาให้คำปรึกษา ข้อเสนอแนะ และแนวทางแก้ไขปัญหาต่าง ๆ ตลอดจน ความอนุเคราะห์ในด้าน เครื่องมือ อุปกรณ์ต่าง ๆ ขอขอบคุณ ห้องสมุด คณะครุศาสตร์อุตสาหกรรม ที่ช่วยอำนวยความสะดวกและเอื้อเฟื้อสถานที่ ในการค้นคว้าหาข้อมูล ขอขอบคุณเพื่อนนักศึกษาภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ให้กำลังใจและให้ความช่วยเหลือ

สุดท้ายที่ควรระลึกถึงอย่างยิ่ง บิดา มารดาที่คอยสนับสนุนด้านการศึกษาและเป็นผู้ให้กำลังใจด้วยดีตลอดมา



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มาของปัญญานิพนธ์	1
1.2 ชัดความสามารถของโครงการ	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	3
2.1 กล่าวนำ	3
2.2 ไมโครคอนโทรลเลอร์เบอร์ 8051	3
2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์เบอร์ 8051	3
2.2.2 โครงสร้างของไมโครคอนโทรลเลอร์เบอร์ 8051	4
2.2.3 สถาปัตยกรรมของ 8051	6
2.2.4 การจัดการหน่วยความจำของ 8051	9
2.2.5 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์	10
2.2.6 การทำงานของ 8051	11
2.2.7 การเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 กับหน่วยความจำภายนอก	13
2.2.8 ไทม์เมอร์/เคาน์เตอร์	17
2.2.9 การสื่อสารข้อมูลแบบอนุกรม	20
2.2.10 การอินเตอร์รัพต์	23
2.3 โปรแกรมวิชาลเวสิก	24
2.3.1 คุณสมบัติ	25
2.3.2 ส่วนประกอบของจอภาพ	25
2.3.3 แถบเครื่องมือ	26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.2.4 ก่อตั้งเครื่องมือ	29
2.2.5 ขั้นตอนในการพัฒนาโปรแกรม	29
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	31
3.1 กล่าวนำ	31
3.2 การออกแบบวงจร	31
3.2.1 การออกแบบวงจรชุดรวมข้อมูล	32
3.2.2 ส่วนวงจรสร้างฐานเวลา	33
3.2.3 วงจรการสื่อสารข้อมูลผ่านเทอร์มินัล	34
3.3 การออกแบบโปรแกรม	35
3.3.1 เครื่องมือที่ใช้ในการออกแบบสร้างโปรแกรม	35
3.3.2 ขั้นตอนในการออกแบบโปรแกรม	35
3.3.3 แนวคิดเกี่ยวกับ โครงสร้างของโปรแกรม	36
3.3.4 ผังงานโปรแกรมการติดต่อกับเครื่องลูกข่าย	37
3.3.5 ผังงานการรับข้อมูลจากเครื่องลูกข่าย	38
3.3.6 การสร้างโปรแกรม	39
บทที่ 4 การทดลอง ผลการทดลอง	42
4.1 กล่าวนำ	42
4.2 การทดลองการทำงานของวงจร โครงข่ายรวบรวมข้อมูลระยะไกล	42
4.2.1 การทดลองการเก็บรวบรวมข้อมูล	42
4.2.2 การทดลองการติดต่อสื่อสารผ่าน โมเด็ม	42
4.3 สรุปผลการทดลอง	45
บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไขและพัฒนา	46
5.1 บทสรุป	46
5.2 ปัญหาในการทำโครงงาน	46
5.3 การแก้ปัญหา	46
5.4 ประโยชน์ที่ได้รับจากการทำโครงงาน	47
5.5 แนวทางการพัฒนา	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
ภาคผนวก ก เครื่องต้นแบบ โครงข่ายรวบรวมข้อมูลระยะไกล	47
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	51
ภาคผนวก ค โปรแกรม	56
บรรณานุกรม	127
ประวัติผู้แต่ง	128



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051	4
ตารางที่ 2.2 ส่วนประกอบของจอภาพสี	25
ตารางที่ 2.3 แถบเครื่องมือของสี	27
ตารางที่ 3.1 หน้าที่ใช้สัญญาณของพอร์ตอนุกรม	34
ตารางที่ 4.1 ผลการทดลองเก็บรวบรวมข้อมูล	43
ตารางที่ 4.2 สรุปผลการทดลองส่งข้อมูล	44



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 โครงสร้างของ 8051	5
รูปที่ 2.2 สถาปัตยกรรมภายในของ 8051	7
รูปที่ 2.3 ลักษณะภายนอกของ 8051	8
รูปที่ 2.4 การจัดพื้นที่หน่วยความจำสำหรับ โปรแกรม	9
รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูล	10
รูปที่ 2.6 เวลาพื้นฐานของ 8051 และลำดับของช่วงเวลาใน 1 แมกซ์ซีไนซ์เกิด	11
รูปที่ 2.7 ลำดับสถานะการทำงานของไมโครคอนโทรลเลอร์ 8051	12
รูปที่ 2.8 สัญญาณการติดต่อกับหน่วยความจำโปรแกรม	14
รูปที่ 2.9 สัญญาณการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก	16
รูปที่ 2.10 การมัลติเพล็กซ์แอสแอดเดรสและบัสข้อมูลของพอร์ต 0	16
รูปที่ 2.11 การทำงานของไทม์เมอร์ โหมด 0	18
รูปที่ 2.12 การทำงานของไทม์เมอร์ โหมด 1	19
รูปที่ 2.13 การทำงานของไทม์เมอร์ โหมด 2	19
รูปที่ 2.14 การทำงานของไทม์เมอร์ โหมด 3	20
รูปที่ 2.15 การทำงานของพอร์ตอนุกรม โหมด 0	21
รูปที่ 2.16 การทำงานของพอร์ตอนุกรม โหมด 1	22
รูปที่ 2.17 การทำงานของพอร์ตอนุกรม โหมด 2	23
รูปที่ 2.18 ส่วนประกอบของจอภาพวีซวลเบสิก	26
รูปที่ 2.19 แถบเครื่องมือของวีซวลเบสิก	26
รูปที่ 2.20 กด่องเครื่องมือของวีซวลเบสิก	29
รูปที่ 3.1 แผนผังการทำงานของโครงข่ายรวบรวมข้อมูลระยะไกล	31
รูปที่ 3.2 การเชื่อมต่อกับหน่วยความจำสำหรับข้อมูล	32
รูปที่ 3.3 ส่วนวงจรสร้างฐานเวลา	33
รูปที่ 3.4 วงจรสื่อสารข้อมูลผ่านพอร์ตอนุกรม	35
รูปที่ 3.5 ผังงานของโปรแกรมเครื่องคอมพิวเตอร์แม่ข่าย	36
รูปที่ 3.6 ผังงานโปรแกรมย่อยที่เรียกไปยังเครื่องลูกข่าย	37
รูปที่ 3.7 ผังงานของการรับข้อมูลจากลูกข่าย	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.8 การสร้างฟอร์มหลักของโปรแกรม	39
รูปที่ 3.9 การสร้างหน้าต่างการกำหนดค่าเริ่มต้น	40
รูปที่ 3.10 การออกแบบฟอร์มการแสดงกราฟข้อมูล	41
รูปที่ 4.1 การเชื่อมต่อในการทดลองเก็บรวบรวมข้อมูล	42
รูปที่ 4.2 การทดลองติดต่อสื่อสารผ่านโมเด็ม	43
รูปที่ 4.3 วงจรโครงข่ายรวบรวมข้อมูลระยะไกล	44
รูปที่ ก.1 เครื่องต้นแบบโครงข่ายรวบรวมข้อมูลระยะไกล	49
รูปที่ ก.2 การเชื่อมต่อเครื่องต้นแบบกับโมเด็ม	49
รูปที่ ก.3 อุปกรณ์ตรวจจับ DS 1820	50
รูปที่ ข.1 วงจรโครงข่ายรวบรวมข้อมูลระยะไกล	52
รูปที่ ข.2 วงจรภาคจ่ายไฟ	53
รูปที่ ข.3 วงจรเรียลไทม์	53
รูปที่ ข.4 วงจรสื่อสารข้อมูล RS-232	53
รูปที่ ข.5 แผ่นวงจรพิมพ์ด้านบน	54
รูปที่ ข.6 แผ่นวงจรพิมพ์ด้านล่าง	54
รูปที่ ข.7 การลงอุปกรณ์	55

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญญาประดิษฐ์

ความต้องการที่สำคัญของมนุษย์ทุกวันนี้ ยังคงหลีกเลี่ยงไม่ได้ทั้งความต้องการความปลอดภัยในชีวิตและทรัพย์สินของตน มนุษย์ได้พยายามหาวิธีที่จะสนองตอบต่อความต้องการนั้นในหลายๆวิธี โดยพยายามหาทางป้องกันอันตรายทั้งที่เกิดจากน้ำมือของมนุษย์เองและอันตรายที่เกิดจากธรรมชาติ โดยเฉพาะภัยอันตรายที่เกิดจากธรรมชาตินี้เป็นสิ่งที่ยากจะคาดเดาได้ว่าจะเกิดขึ้นเมื่อใด ระดับความรุนแรงเพียงใด และจะเกิดขึ้นที่ใด ทั้งนี้การที่จะสามารถลดความสูญเสียจากภัยธรรมชาติให้ได้มากที่สุดนั้น ก็ต้องอาศัยข้อมูลจากวันเวลาที่ผ่านไป เพื่อนำมาใช้วิเคราะห์และคาดการณ์ภัยอันตรายที่จะเกิดขึ้นในอนาคต เป็นผลให้เกิดความปลอดภัยในชีวิตและทรัพย์สินสูงขึ้น หนึ่งในอดีตที่ผ่านมา มนุษย์ได้มีการเก็บรวบรวมข้อมูลต่างๆ ซึ่งเป็นมาตรการที่มนุษย์สร้างความปลอดภัยให้แก่ชีวิตของตนเองยังมีความผิดพลาดและความล่าช้า ทำให้ในบางครั้งไม่สามารถใช้งานได้ตามที่ ตามเหตุผลดังที่ได้กล่าวมาข้างต้น โดยใช้หลากหลายวิธีในการเก็บรวบรวมข้อมูล บางครั้งอาจยังมีการใช้มนุษย์เป็นผู้บันทึกผลดังกล่าวอยู่ ซึ่งยังคงเกิดความปลอดภัยในชีวิตและทรัพย์สินน้อย

คณะผู้จัดทำซึ่งมีความต้องการที่จะจัดทำโครงการสร้างอุปกรณ์เพื่อการสอน ได้เล็งเห็นความสำคัญของเหตุและผลดังกล่าวข้างต้น และต้องการให้โครงการที่จะจัดทำนั้นสามารถนำไปใช้ประโยชน์เพื่อเป็นการลดการใช้กำลังงานคนในการทำงานและลดความผิดพลาดในการเก็บรวบรวมข้อมูล หรือเป็นแนวทางให้กับผู้ที่ต้องการพัฒนา จึงได้คิดจัดทำโครงข่ายรวบรวมข้อมูลระยะไกลนี้ขึ้น

1.2 ขีดความสามารถของโครงการ

1. สามารถรับข้อมูลจากภาคตรวจจับใดๆ ที่เป็นสัญญาณดิจิทัลอสขนาด 8 บิต
2. มีการเชื่อมโยงเป็นเครือข่ายระหว่างลูกข่ายและศูนย์กลาง
3. สามารถรับส่งข้อมูลได้ในระยะไกล
4. สามารถกำหนดระยะเวลาในการส่งข้อมูลจากเครื่องลูกข่ายไปยังศูนย์กลาง
5. ลูกข่ายสามารถส่งข้อมูลไปยังศูนย์กลางตามการเปลี่ยนแปลงที่เกิดขึ้นอย่างกะทันหัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปฏิญญาพันธบัตรฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษา และทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบไปด้วยเนื้อหาในทางทฤษฎีที่เกี่ยวข้อง ซึ่งทำให้ผู้อ่าน มีความรู้ความเข้าใจที่เป็นพื้นฐานเสียก่อน อันจะเป็นประโยชน์ต่อการทำความเข้าใจงานจริงต่อไป

บทที่ 3 การออกแบบ การสร้างและการทำงาน โดยกล่าวถึงการสร้างการออกแบบและการทำงานของโครงข่ายรวบรวมข้อมูลระยะไกล การออกแบบวงจร รวมทั้งหลักการทำงานในส่วนต่างๆ ซึ่งจะช่วยให้ผู้อ่านเข้าใจการทำงานโดยรวมของโครงข่ายนี้

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงขั้นตอนการทดลอง การทดสอบประสิทธิภาพการทำงานของวงจรโครงข่ายรวบรวมข้อมูลระยะไกล เพื่อตรวจสอบว่าโครงข่ายนี้สามารถทำงานได้ตรงตามวัตถุประสงค์หรือไม่

บทที่ 5 สรุปปัญหา แนวทางการแก้ไขและพัฒนา โครงข่ายนี้ การสรุปผลการทำงานและได้เสนอแนวทางการแก้ไข แนวทางในการพัฒนาให้มีประสิทธิภาพและการใช้งานได้อย่างกว้างขวางมากขึ้น

ภาคผนวก ก รูปเครื่องต้นแบบ

ภาคผนวก ข วงจร และแผ่นวงจรพิมพ์

ภาคผนวก ค โปรแกรม

บทที่ 2

ทฤษฎีและหลักการ

2.1 กล่าวนำ

โครงข่ายรวบรวมข้อมูลระยะไกล เป็นอุปกรณ์ที่ใช้สำหรับรวบรวมข้อมูลที่อยู่ในที่ห่างไกลโดยใช้ ไมโครคอนโทรลเลอร์ 8051 เป็นส่วนที่ทำงานเก็บข้อมูลและ ส่งข้อมูลผ่านพอร์ตอนุกรมไปยังโมเด็มส่งผ่านสายโทรศัพท์ เพื่อจะนำข้อมูลที่ได้ไปใช้ประโยชน์ เพื่อให้สามารถจัดสร้างโครงข่ายรวบรวมข้อมูลระยะไกลได้ ต้องรู้หลักการทำงานของไมโครคอนโทรลเลอร์ 8051 และการเชื่อมต่อกับพอร์ตอนุกรม นอกจากนี้ให้สามารถสร้างโครงงานโครงข่ายรวบรวมข้อมูลระยะไกลได้แล้วยังเป็นแนวทางในการพัฒนาโครงงานโครงข่ายรวบรวมข้อมูลระยะไกล และเนื้อหาที่ได้กล่าวไว้ข้างต้นได้แสดงรายละเอียดตามหัวข้อต่อไปนี้

2.2 ไมโครคอนโทรลเลอร์ เบอร์ 8051

ไมโครคอนโทรลเลอร์ในตระกูลตระกูล MCS-51 ที่มีขนาด 8 บิต ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ต่างๆ ดังตารางที่ 2.1 ทุกๆเบอร์จะมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดของหน่วยความจำภายในและภายนอกที่แตกต่างกัน เพื่อความเหมาะสมในการนำไปใช้งานตามความต้องการต่างๆ แต่เดิม 8051 ถูกสร้างด้วยวิธี HMOS I แต่ในปัจจุบันได้สร้างด้วยวิธี HMOS II จึงมีชื่อเป็น 8051AH ไมโครคอนโทรลเลอร์ในตระกูล 51 นั้น ถึงแม้ว่าจะมีหลายเบอร์แต่เราก็จะเรียกว่าเป็น “8051” ซึ่งหมายถึงไมโครคอนโทรลเลอร์ตระกูล 51 ส่วนเบอร์ 8032 และ 8052 มีหน่วยความจำภายในเพิ่มขึ้นและมีวงจรรีบ/จับเวลา ขนาด 16 บิต เพิ่มขึ้นดังตารางที่ 2.1

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ เบอร์ 8051

- 1) หน่วยประมวลผลกลางขนาด 8 บิต
- 2) หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์
- 3) หน่วยความจำข้อมูลภายในขนาด 128 ไบต์
- 4) อี้าตำแหน่งของหน่วยความจำโปรแกรมได้ถึง 64 กิโลไบต์
- 5) อี้าตำแหน่งของหน่วยความจำข้อมูลได้ถึง 64 กิโลไบต์
- 6) หน่วยความจำโปรแกรมและข้อมูลที่อยู่ภายนอกไอซีแยกจากกันอย่างละ 64 กิโลไบต์
- 7) มีพอร์ตอินพุต/เอาต์พุต แบบขนานจำนวน 4 พอร์ต แยกกันอย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8) มีวงจรรีบ จับเวลา ขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
- 9) มีพอร์ตการสื่อสารอนุกรม รับส่งข้อมูลได้ในเวลาเดียวกัน สามารถเลือกรูปแบบการส่งได้ 4 รูปแบบ
- 10) รับส่งอินเทอร์รัพต์ได้ 6 แหล่งกระโดดไปทำงานตอบสนองได้ 5 ตำแหน่ง
- 11) มีวงจรออสซิลเลเตอร์ภายใน
- 12) นำข้อมูลมา AND, OR หรือทำการ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051

เบอร์	หน่วยความจำภายใน		จำนวนไทมเมอร์ / เคาน์เตอร์	อินเทอร์รัพต์ หมายเลข
	เก็บโปรแกรม	เก็บข้อมูล		
8052H	8K x 8 ROM	256 x 8 ROM	3 x 16 Bit	6
8051H	4K x 8ROM	128 x 8 ROM	2 x 16 Bit	5
8051	4K x 8ROM	128 x 8 ROM	2 x 16 Bit	5
8032AH	ไม่มี	256 x 8 ROM	3 x 16 Bit	6
8031AH	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8031	ไม่มี	128 x 8 ROM	2 x 16 Bit	5
8751H	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5
80751H-12	4K x 8 EPROM	128 x 8 ROM	2 x 16 Bit	5

2.2.2 โครงสร้างของไมโครคอนโทรลเลอร์ เบอร์ 8051

ภายใน 8051 จะประกอบด้วยเกตชนิดต่างๆ เช่น AND, OR, NOT ซึ่งแต่ละเกตเหล่านี้จะนำเอาออกมาแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่งม วงจรสร้างสัญญาณนาฬิกา เป็นต้น โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังรูปที่ 2.1

โครงสร้าง 8051 จะประกอบด้วย 3 ส่วนหลักๆ ดังนี้

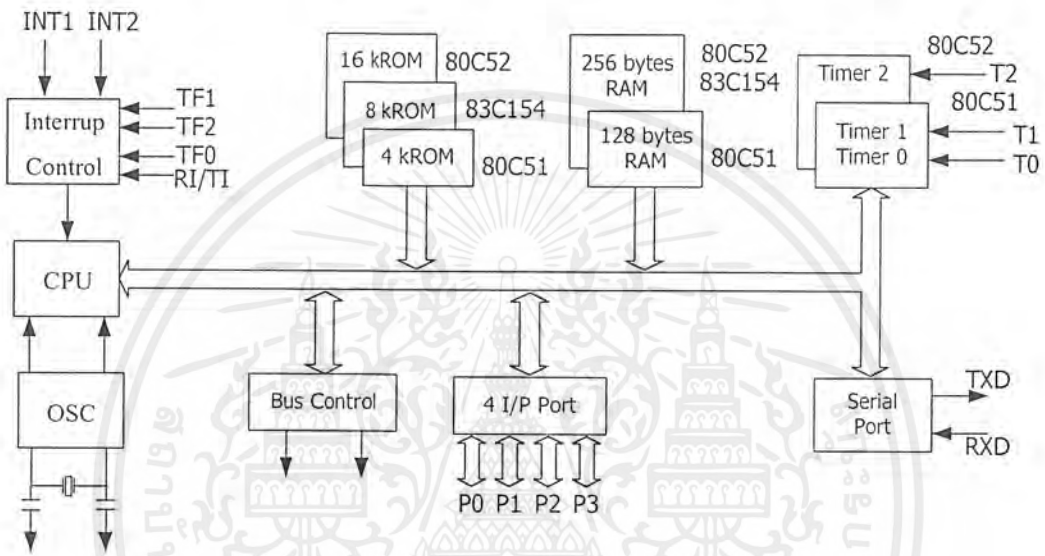
1) ซีพียู (Central Processing Unit)

ส่วนนี้จะทำหน้าที่สร้างสัญญาณควบคุมการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่ได้จากวงจรควบคุมได้แก่ สัญญาณสำหรับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออก ซึ่งส่วนควบคุมการจับจังหวะ และส่วนควบคุมบัสก็เป็นส่วนหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของวงจรควบคุมด้วย การสร้างสัญญาณวงจรควบคุมจากซีพียูนี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่งที่กำหนดไว้และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรรอสซิลเลเตอร์เพื่อให้ทุกๆ ส่วนทำงานประสานกันอย่างถูกต้อง

ในซีพียูยังประกอบด้วยส่วนประมวลผล (Arithmetic Logic Unit) ทำหน้าที่ประมวลผลข้อมูลเช่นการบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในหน่วยความจำที่ต้องการ



รูปที่ 2.1 โครงสร้างของ 8051

2) หน่วยความจำ (Memory)

มีไว้สำหรับจัดจำข้อมูล ซึ่งในการนำข้อมูลเข้าและออกจากหน่วยความจำ เราจำเป็นต้องรู้ตำแหน่งของหน่วยความจำ (Address) ในการนำข้อมูลไปเก็บไว้ในหน่วยความจำเรียกว่า การเขียนข้อมูลและการนำข้อมูลออกจากหน่วยความจำ เรียกว่า การอ่านข้อมูล ในไมโครคอนโทรลเลอร์ 8051 ข้อมูลในแต่ละตำแหน่งจะมีขนาด 8 บิต ดังนั้นในแต่ละตำแหน่งของหน่วยความจำจะสามารถเก็บข้อมูลมีค่าได้ระหว่าง 00000000₂ ถึง 11111111₂ หรือ 00H ถึง 0FFH ในการติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่ม คือ

2.1) ตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ซึ่งในไมโครคอนโทรลเลอร์ 8051 จะมีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลที่มีขนาดสูงสุดชนิดละ 65,536 ตำแหน่ง (64 กิโลไบต์) ดังนั้นการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐานสองทั้งหมด 16 เส้น (2¹⁶ เท่ากับ 65,536)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2) ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำในตำแหน่งที่เราต้องการ

2.3) สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำเพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล โดยวงจรถอดรหัสคำสั่งจะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

3) อุปกรณ์อินพุต / เอาต์พุต (Input / Output Device)

เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือนำข้อมูลออกจากไมโครคอนโทรลเลอร์ 8051 ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกได้ อุปกรณ์อินพุต / เอาต์พุต ได้แก่ อินพุต/เอาต์พุตพอร์ตแบบขนาน, วงจรนับ/จับเวลา 0, วงจรนับ/จับเวลา 1, พอร์ตสื่อสารอนุกรม

3.1) พอร์ตแบบขนาน เป็นที่สำหรับใช้รับส่งข้อมูลซึ่งเป็นสัญญาณดิจิทัลเข้าหรือออกจากตัวไมโครคอนโทรลเลอร์ 8051 มีทั้งหมด 4 พอร์ตโดยแต่ละพอร์ตจะรับส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตจะใช้งานได้มากกว่าหนึ่งอย่างก็ได้

3.2) วงจรนับ/จับเวลา 0 และวงจรนับ/จับเวลา 1 เป็นวงจรนับที่สามารถทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอกไมโครคอนโทรลเลอร์ 8051 หรือจำนวนของสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์ 8051 ก็ได้ สามารถตั้งค่าเริ่มต้นการนับและอ่านค่าการนับได้โดยซีพียู

3.3) พอร์ตอนุกรม ซีพียูจะอ่านและเขียนข้อมูลกับพอร์ตอนุกรมเป็นแบบ 8 บิต แต่ข้อมูลจะถูกส่งออกไปจากไมโครคอนโทรลเลอร์ 8051 เรียงไปที่ละบิตออกจากขา TXD และในการรับข้อมูลก็จะรับเข้ามาที่ละบิตทางขา RXD และจัดเรียงใหม่เป็น 8 บิตเพื่อให้ซีพียูอ่านไปใช้งานต่อไป ในไมโครคอนโทรลเลอร์ 8051 มีพอร์ตให้ใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่างๆ ได้มากมาย การนำพอร์ตไปใช้งานจะต้องเขียนโปรแกรมขึ้นมาควบคุม

2.2.3 สถาปัตยกรรมของ 8051

ในรูปที่ 2.2 เป็นรูปแสดงสถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ 8051 ซึ่งอธิบายถึงส่วนประกอบย่อยๆ ภายในตัวไมโครคอนโทรลเลอร์ 8051 และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขาของไมโครคอนโทรลเลอร์ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 2.3 ไมโครคอนโทรลเลอร์ 8051 บรรจุอยู่ในวงจรรวมแบบ DIP (Dual Inline Package) แบบ 40 ขา ดังนี้

VCC (ขา 40) ต่อไฟเลี้ยง +5 โวลต์

VSS (ขา 20) ต่อลงกราวด์

PORT 0 (ขา 29-32) มีทั้งหมด 8 บิต คือ P0.0-P0.7 ใช้งานเป็นอินพุต/เอาต์พุต

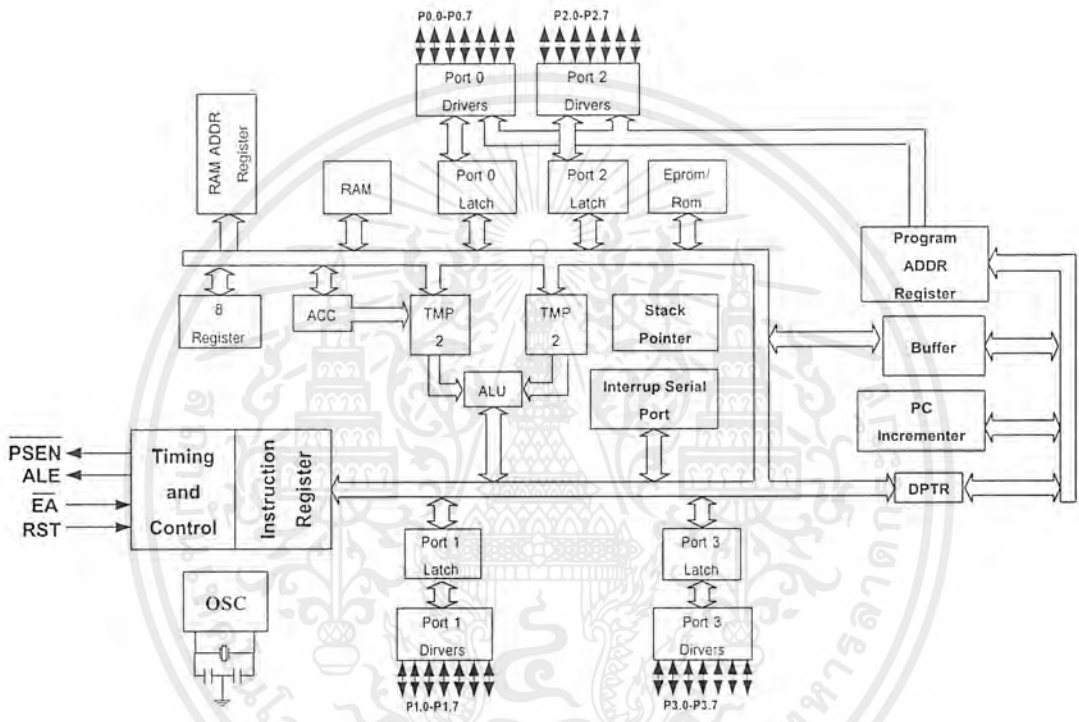
พอร์ต โดยทั่วไปใช้เป็นตัวส่งแอสเคลสไบต์ค่า (A0-A7) และรับส่งข้อมูล (D0-

D7) จากหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ P1.0-P1.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไป

PORT 2 (ขา 21-28) มีทั้งหมด 8 บิต คือ P2.0-P2.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไปและใช้เป็นตัวส่งแอสเคลสไบต์สูง (A8-A15) เพื่อใช้ติดต่อกับหน่วยความจำภายนอก



รูปที่ 2.2 สถาปัตยกรรมภายในของ 8051

PORT 3 (ขา 10-17) มีทั้งหมด 8 บิต คือ P3.0-P3.7 ใช้งานเป็นอินพุต/เอาต์พุตพอร์ตทั่วไปและใช้งานในหน้าที่พิเศษ ดังนี้

P3.0 /RXD (Serial Input Port) ใช้ส่งข้อมูลอนุกรม

P3.1 /TXD (Serial Output Port) ใช้ส่งข้อมูลแบบขนาน

P3.2 /INT0 (External Interrupt) ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.3 /INT1 (External Interrupt) ใช้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

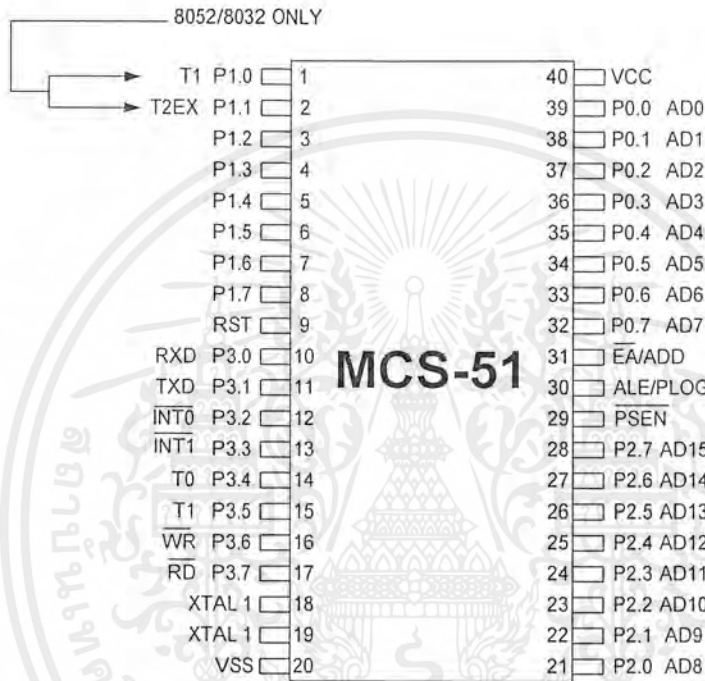
P3.4 /T0 (Timer/Counter 0 External Input) ใช้เป็นอินพุตให้วงจรรนับ/จับเวลาชุดที่ 0

P3.5 /T1 (Timer/Counter 1 External Input) ใช้เป็นอินพุตให้วงจรรนับ/จับเวลาชุดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.6 / $\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาคควบคุมการเขียนข้อมูลจากหน่วยความจำภายนอก

P3.7 / $\overline{\text{RD}}$ (External Data Memory Read Strobe) ขาคควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก



รูปที่ 2.3 ลักษณะขาภายนอกของ 8051

RST (ขา 9) Reset ใช้สำหรับรีเซ็ตวงจรทุกวงจรภายในชิพ เพื่อเริ่มต้นการทำงานใหม่ ในการรีเซ็ตต้องป้อนลอจิก “1” นานอย่างน้อย 2 แมกซ์อินไซเคิล

ALE (ขา 30) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอกเพื่อควบคุมการแลตช์ค่าแอสตเรสไปต์ค่าจากพอร์ต 0

PSEN (ขา 29) Program Strobe Enable เป็นขาสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขานี้ Active มีลอจิกเป็น “0” จะอ่านโปรแกรมจากหน่วยความจำภายนอก และถ้าเป็นการอ่านหน่วยความจำโปรแกรมภายในขานี้จะไม่ Active

EA (ขา 31) External Access เป็นขาที่ใช้สำหรับเลือกว่าให้ทำงานจากหน่วยความจำ

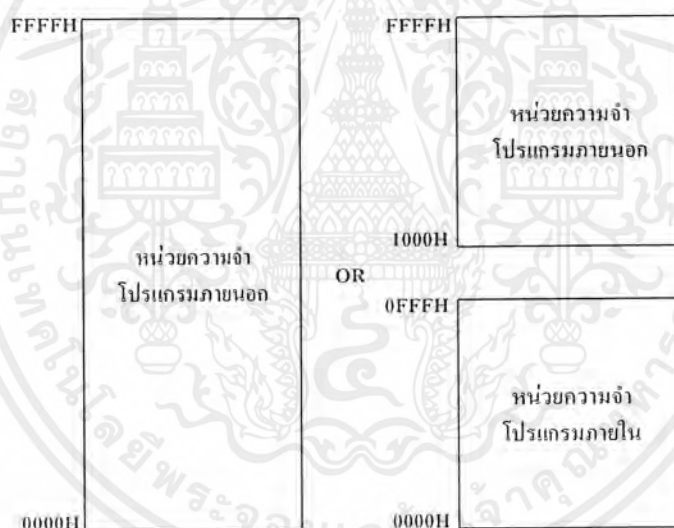
โปรแกรมภายในหรือหน่วยความจำโปรแกรมภายนอกซีพ เมื่อขานี้ Active ที่ลอจิก “0” จะเป็นการทำงานตามคำสั่งในหน่วยความจำโปรแกรมภายนอก
 XTAL 1 (ขา 19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจรรอสซิลเลเตอร์
 XTAL 2 (ขา 18) ใช้ต่อคริสตอลภายนอก โดยเป็นเอาต์พุตออกจากวงจรรอสซิลเลเตอร์

2.2.4 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกเป็น 2 แบบตามลักษณะการใช้งานดังนี้

1) หน่วยความจำโปรแกรม (Program Memory)

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมในรูปของภาษาเครื่องซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บไว้ในหน่วยความจำโปรแกรมไปทำการถอดรหัสแล้วสร้าง

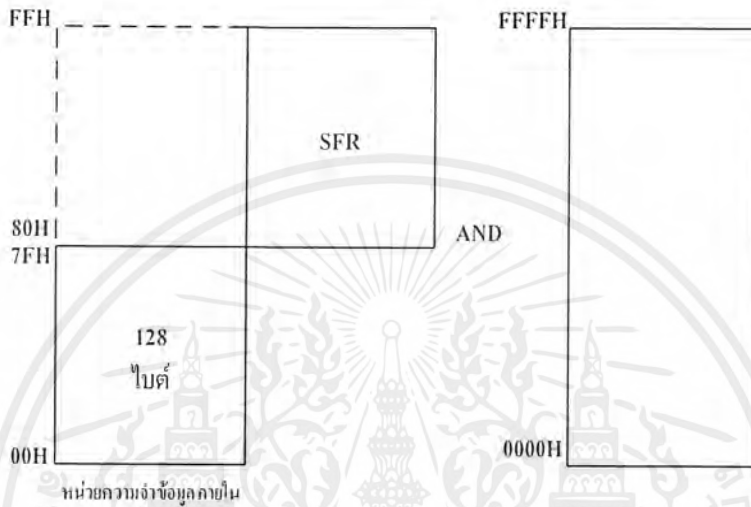


รูปที่ 2.4 การจัดพื้นที่หน่วยความจำสำหรับโปรแกรม

สัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้เป็นแบบรอมและผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการส่วนที่เป็นหน่วยความจำโปรแกรมก็คือแรมขนาด 4 กิโลไบต์ ดังรูปที่ 2.4

2) หน่วยความจำข้อมูล (Data Memory)

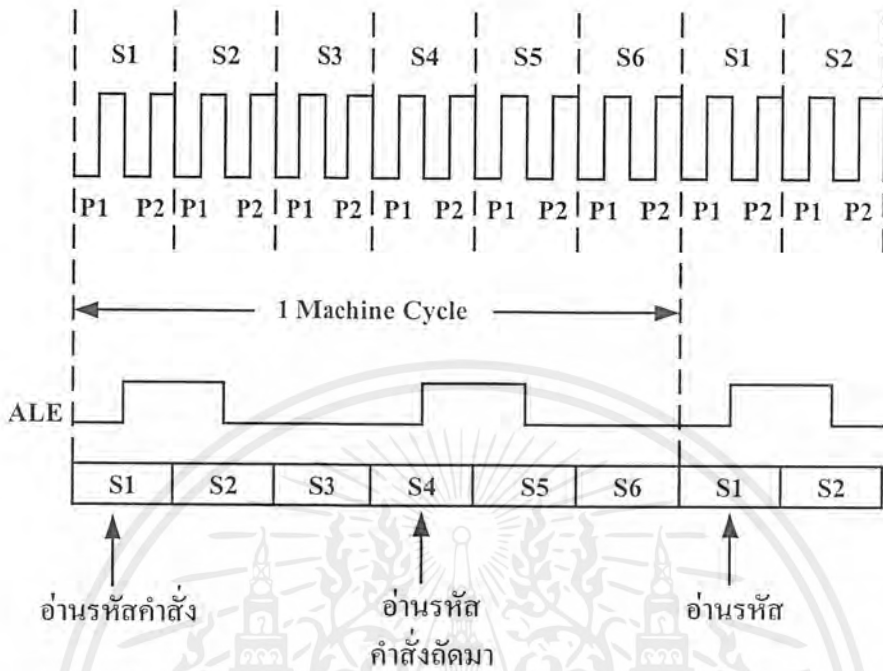
เป็นหน่วยความจำที่ใช้เก็บข้อมูล สามารถอ่านและเขียนข้อมูลได้ ซึ่งหน่วยความจำภายในมีขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำภายนอกไอซี มีขนาด 64 กิโลไบต์ ดังรูปที่ 2.5



รูปที่ 2.5 การจัดพื้นที่หน่วยความจำข้อมูล

2.2.5 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์

แมชชีนไซเคิล (Machine Cycle) คือรอบการทำงานของคำสั่ง เป็นค่าที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ถ้าเป็นคำสั่งที่ซับซ้อนมากก็จะต้องใช้เวลานาน 2-3 แมชชีนไซเคิล 1 แมชชีนไซเคิล จะประกอบด้วยสัญญาณนาฬิกา 12 ลูก โดยสัญญาณนาฬิกาแต่ละลูกเรียกว่าเฟส (Phase) สัญญาณนาฬิกา 2 เฟส รวมกันเรียกว่า 1 สเตท (State) เพราะฉะนั้นใน 1 แมชชีนไซเคิลจึงมีทั้งหมด 6 สเตท



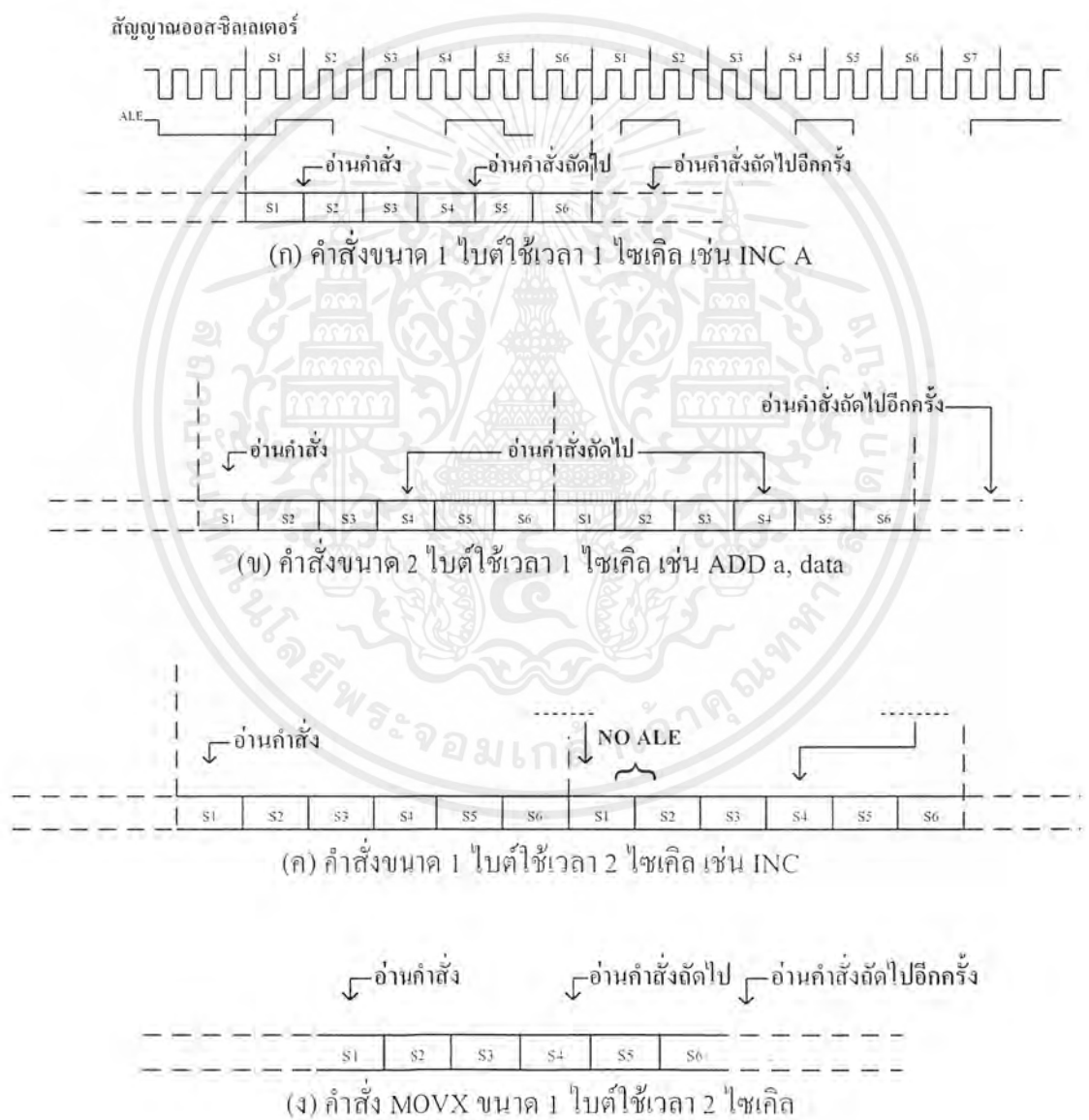
รูปที่ 2.6 เวลาพื้นฐานของ 8051 และลำดับของช่วงเวลาใน 1 แมชชีนไซเคิล

2.2.6 การทำงานของ 8051

จากรูปที่ 2.7 เมื่อป้อนไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์ เบอร์ 8051 ซึ่งมีวงจรรีเซ็ตเมื่อเปิดเครื่องจะเกิดการรีเซ็ต การทำงานภายในไมโครคอนโทรลเลอร์เบอร์ 8051 เริ่มจากภาคโปรแกรมเคาน์เตอร์ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนเส้นทางหมายเลข 1 เส้นทางนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR register ค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็นรอมภายใน หรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วนวงจรเวลาและควบคุม ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA เป็นการเลือกใช้รอมภายใน 8051 โดยที่วงจรเวลาและควบคุมจะสร้างสัญญาณไปยังรอมภายในให้ส่งข้อมูลที่คำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาทางเส้นทางหมายเลข 2 ข้อมูลจากรอม ถูกส่งไปยังเส้นทางหมายเลข 3 ที่เรียกว่า เส้นทางข้อมูลภายใน แล้วนำไปเก็บไว้ที่รีจิสเตอร์ IR (Instruction Register) เพื่อส่งไปให้กับวงจรเวลาและควบคุมทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆต่อไป ในกรณีที่เลือกรอมภายนอก โดยการป้อนสัญญาณลอจิก 1 เข้าไปที่ขา EA จะทำให้วงจรเวลาและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนเส้นทางหมายเลข 2 ออกไปใช้หน่วยความจำภายนอกจากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยังเส้นทางข้อมูลภายในแล้วไปเก็บที่รีจิสเตอร์ IR เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจากรอมภายใน การทำงานในช่วงส่งค่าตำแหน่งในหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่ง กลับเข้ามาเก็บไว้ในรีจิสเตอร์ IR เรียกว่าช่วงการเฟตช์ (Fetch) ช่วงต่อไปจะเป็นช่วงการทำงานตามคำสั่งที่เรียกว่า Execute Cycle



รูปที่ 2.7 ลำดับสถานะการทำงานของไมโครคอนโทรลเลอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งแต่ละคำสั่งของไมโครคอนโทรลเลอร์เบอร์ 8051 จะใช้เวลาทำงานเพียง 1, 2 หรือ 3 แมกซ์ซินไซเคิล แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 แมกซ์ซินไซเคิล จะใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของไมโครคอนโทรลเลอร์ เบอร์ 8051 จะใช้เวลา 12, 24 หรือ 36 ไซเคิลของสัญญาณนาฬิกา ลำดับการทำงานของไมโครคอนโทรลเลอร์ เบอร์ 8051 ดังแสดงในรูปที่ 2.7

2.2.7 การเชื่อมต่อไมโครคอนโทรลเลอร์ 8051 กับหน่วยความจำภายนอก

ไมโครคอนโทรลเลอร์ 8051 สามารถติดต่อหน่วยความจำภายนอกได้ 2 ส่วน คือ หน่วยความจำโปรแกรมภายนอกและหน่วยความจำข้อมูลภายนอก ในการใช้งานอาจจะต้องการใช้งานร่วมกันหรือต่อแยกการใช้งานก็ได้

1) สัญญาณในการเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก

ในการเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำโปรแกรมภายนอก ข้อที่สำคัญ คือ จะต้องต่อขาของสัญญาณควบคุมหน่วยความจำโปรแกรมให้ถูกต้อง สามารถตรวจสอบสัญญาณการเชื่อมต่อได้ดังรูปที่ 2.8 ซึ่งลักษณะการต่อขาสัญญาณควบคุม และการทำงานสามารถแบ่งลักษณะตามขาของสัญญาณได้ดังนี้

1.1) สัญญาณ \overline{CE} (Chip Enable)

สัญญาณนี้ทำหน้าที่เลือกให้อุปกรณ์ทำงาน โดยถ้าขานี้ แอคทีฟหรือเป็นระดับลอจิกต่ำจะยอมให้หน่วยความจำโปรแกรมทำงานตามปกติ แต่ถ้าเป็นระดับลอจิกสูงจะไม่มีการทำงานใดๆ ภายในหน่วยความจำโปรแกรม

1.2) สัญญาณ \overline{OE} (Output Enable)

เมื่อขาสัญญาณนี้ แอคทีฟหรือระดับลอจิกต่ำจะทำให้มีการนำข้อมูลภายในตำแหน่งที่ถูกระบุด้วย บัสแอดเดรส ส่งมายังบัสข้อมูลของหน่วยความจำโปรแกรม

1.3) สัญญาณ A0-A15

ขาสัญญาณนี้เป็นขาแอดเดรสที่ใช้เป็นตัวกำหนดตำแหน่งทั้งหมดของข้อมูลที่อยู่ในหน่วยความจำโปรแกรม

1.4) สัญญาณ D0-D7

ขาสัญญาณนี้เป็นขาแอดเดรสที่ใช้สำหรับการรับและส่งค่าของข้อมูลออกจากหน่วยความจำให้กับไมโครคอนโทรลเลอร์

1.5) สัญญาณ \overline{EA} (External Access)

ใช้ในการกำหนดว่าจะให้ไมโครคอนโทรลเลอร์อ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกหรือหน่วยความจำโปรแกรมภายในของตัวไมโครคอนโทรลเลอร์เอง ซึ่งหากเป็นระดับลอจิกต่ำจะอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก และถ้าเป็นลอจิกสูงจะอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลจากหน่วยความจำโปรแกรมภายใน และมีการใช้งานแอดเดรสที่อยู่ในช่วงที่สูงเกินค่าสูงสุดของหน่วยความจำโปรแกรมภายใน ไมโครคอนโทรลเลอร์จะทำการอ่านคำสั่งหรือข้อมูลตำแหน่งที่สูงเกินกว่าค่าสูงสุดมาจากหน่วยความจำโปรแกรมภายนอกโดยอัตโนมัติ

1.6) สัญญาณ PSEN (Program Strobe Enable)

ขาสัญญาณนี้จะใช้ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก จะแอกทิฟที่ระดับลอจิกต่ำ

1.7) สัญญาณ P0.0-P0.7

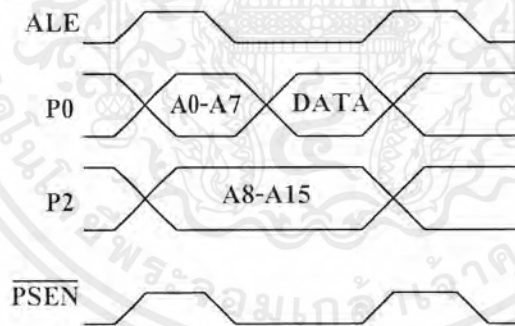
ขาสัญญาณจะใช้ในการกำหนดค่าของแอดเดรสไบต์ต่ำ (A0-A7) ของหน่วยความจำและใช้เป็นบัสข้อมูล ในการรับและส่งข้อมูล

1.8) สัญญาณ P2.0-P2.7

ขาสัญญาณจะใช้ในการกำหนดค่าของแอดเดรสไบต์สูง (A8-A15) ของหน่วยความจำโปรแกรม

1.9) ขาสัญญาณ ALE (Address Latch Enable)

ขาสัญญาณนี้จะใช้ในการควบคุมการแลตช์ค่าของแอดเดรสไบต์ต่ำ



รูปที่ 2.8 สัญญาณการติดต่อกับหน่วยความจำโปรแกรม

2) สัญญาณในการเชื่อมต่อกับหน่วยความจำข้อมูลภายนอก

ในการเชื่อมต่อไมโครคอนโทรลเลอร์กับหน่วยความจำข้อมูลภายนอก ข้อที่สำคัญคือ จะต้องขอขาสัญญาณควบคุมหน่วยความจำข้อมูลให้ถูกต้องสามารถตรวจสอบดูสัญญาณการเชื่อมต่อขาสัญญาณได้ดังรูปที่ 2.9 ซึ่งเป็นลักษณะของขาสัญญาณควบคุม และการทำงาน โดยสามารถแบ่งลักษณะตามขาของสัญญาณได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1) สัญญาณ $\overline{\text{CE}}$ (Chip Enable)

สัญญาณนี้ทำหน้าที่เลือกในการเลือกให้แรมทำงาน โดยถ้าขานี้แอกทีฟหรือเป็นระดับลอจิกต่ำจะยอมให้แรมทำงานโดยปกติ แต่ถ้าเป็นระดับลอจิกสูงจะไม่มีการทำงานใดๆ ภายในหน่วยความจำข้อมูล

1.2) สัญญาณ $\overline{\text{EA}}$ (External Access)

เมื่อขาสัญญาณนี้แอกทีฟหรือเป็นระดับลอจิกต่ำ จะทำให้มีการนำข้อมูลภายในตำแหน่งที่ถูกระบุด้วยแอดเดรส ส่งมายังบััสข้อมูลของหน่วยความจำข้อมูล

1.3) สัญญาณ A0-A15

ขาสัญญาณนี้จะเป็นขาแอดเดรสที่ใช้ในการกำหนดตำแหน่งของ ข้อมูลภายในหน่วยความจำข้อมูล

1.4) สัญญาณ D0-D7

ขานี้เป็นขานำข้อมูลเข้าและส่งออกจากหน่วยความจำข้อมูล

1.5) สัญญาณ P0.0-P0.7

ขาสัญญาณนี้ใช้เป็นตัวกำหนดแอดเดรสไบต์ต่ำ (A0-A7) ของหน่วยความจำและใช้เป็นบััสข้อมูล

1.6) สัญญาณ P2.0-P2.7

ขาสัญญาณนี้ใช้กำหนดแอดเดรสไบต์สูง (A8-A15) ของหน่วยความจำข้อมูล

1.7) สัญญาณ ALE (Address Latch Enable)

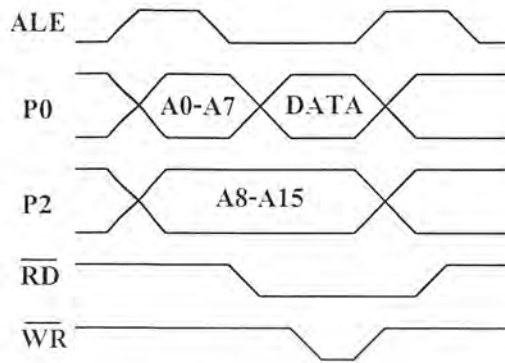
ขาสัญญาณนี้ใช้สำหรับควบคุมการแลตช์ค่าแอดเดรสไบต์ต่ำ

1.8) สัญญาณ P3.6 ($\overline{\text{RD}}$)

ขาสัญญาณนี้ใช้สำหรับควบคุมการอ่านข้อมูลออกมาจากหน่วยความจำข้อมูล

1.9) สัญญาณ P3.7 ($\overline{\text{WR}}$)

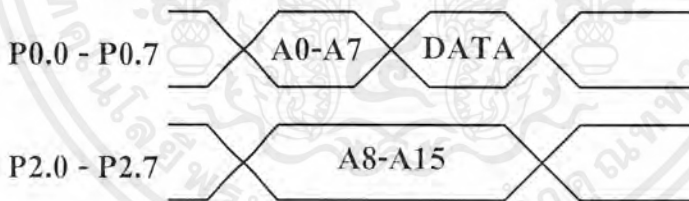
ขาสัญญาณนี้ใช้สำหรับควบคุมการเขียนข้อมูลลงในหน่วยความจำข้อมูล



รูปที่ 2.9 สัญญาณการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

3) การเชื่อมต่อกับหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลภายนอก

ในไมโครคอนโทรลเลอร์ระบบบัสแอดเดรสและบัสข้อมูลของ 8051 เป็นลักษณะของการมัลติเพล็กซ์จากพอร์ตเดียวกัน คือ ในระยะเวลาเริ่มต้น เส้นสัญญาณของพอร์ตจะใช้ส่งค่าของ แอดเดรสตำแหน่งที่ต้องการติดต่อด้วย ในช่วงเวลาต่อมาจึงเปลี่ยนเป็นสภาวะอิมพีแดนซ์สูงเพื่อใช้งานในฐานะของบัสข้อมูล ซึ่งจะมีลักษณะของสัญญาณ ดังรูปที่ 2.10



รูปที่ 2.10 การมัลติเพล็กซ์กับแอดเดรสและบัสข้อมูลของพอร์ต 0

แต่เนื่องจากว่า หน่วยความจำโปรแกรม ที่ใช้งานกันทั่วไปนั้นไม่ใช้การมัลติเพล็กซ์และมีขาสัญญาณบัสแอดเดรสและบัสข้อมูลแยกออกจากกันโดยชัดเจน ดังนั้นการเชื่อมต่อกับหน่วยความจำเพื่อทำหน้าที่เป็นหน่วยความจำโปรแกรม จึงจำเป็นต้องมีวงจรมัลติเพล็กซ์สัญญาณ (Latch) ประกอบเพิ่มเติมขึ้น เพื่อทำการค้างค่าของแอดเดรสที่ส่งออกมาจากชิพของ 8051 ในช่วงแรกให้กับขาสัญญาณแอดเดรสของหน่วยความจำโปรแกรมต่อไป

การติดต่อกับหน่วยความจำโปรแกรมภายนอกชิพ 8051 จะสังเกตได้ว่าในช่วงเวลาของแมชชีนไซเคิลหนึ่ง ๆ นั้นไมโครคอนโทรลเลอร์ 8051 ซึ่งจะมีการนำข้อมูลมาถึงสองครั้งด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นการทำงานของไมโครคอนโทรลเลอร์ 8051 จึงจะไปอ่านข้อมูลโปรแกรมมาเป็นสองเท่าเสมอในช่วงเวลาเริ่มต้นของการติดต่อกับหน่วยความจำภายนอก ที่พอร์ต 0 จะเป็นค่าแอดเดรสไบต์ต่ำ (A0-A7) และช่วงเวลาต่อมาจึงจะเป็นบัสข้อมูล การส่งค่าแอดเดรสไบต์นี้จะอยู่ในช่วงเวลาของลอจิกต่ำดังนั้นในการออกแบบวงจร ALE ในการทำให้ไอซีแลตซ์ภายนอกข้างระดับสัญญาณของแอดเดรสเหล่านี้ไว้ ส่วนสัญญาณ \overline{PSEN} จะใช้ในการเลือกให้หน่วยความจำโปรแกรมโปรแกรมทำงานและอ่านค่าข้อมูลกลับมา

เมื่อสัญญาณ \overline{PSEN} เป็นระดับลอจิกต่ำหน่วยความจำโปรแกรมก็จะทำการถอดรหัสค่าแอดเดรสและส่งข้อมูลที่อยู่ในตำแหน่งนั้นออกมาโดยสัญญาณ \overline{PSEN} นี้จะค้างสถานะการเป็นลอจิกต่ำไว้ช่วงเวลาหนึ่งเพื่อให้ข้อมูลถูกส่งออกมาจากหน่วยความจำโปรแกรมเรียบร้อยแล้วจึงกลับไปเป็นลอจิกสูงตามเดิม และในช่วงจังหวะขาขึ้นของสัญญาณ \overline{PSEN} ชิป 8051 จะทำการอ่านข้อมูลเข้ามาสำหรับข้อมูลทางพอร์ต 2 ซึ่งจะเป็นค่าแอดเดรสไบต์สูง (A8-A15) นั้นจะถูกส่งออกมาในช่วงเวลาที่กลางระหว่างที่สัญญาณ ALE เป็นลอจิกสูง ซึ่งจะเป็นเวลาใกล้เคียงที่ใช้กับการส่งค่าของแอดเดรสไบต์ต่ำออกมาทางพอร์ต 0 สำหรับค่าแอดเดรสพอร์ต 2 นั้นจะค้างค่าอยู่ตลอดช่วงรอบเวลาของการติดต่อกับหน่วยความจำโปรแกรม

2.2.8 ไทม์เมอร์/เคาน์เตอร์ (Timer/Counter)

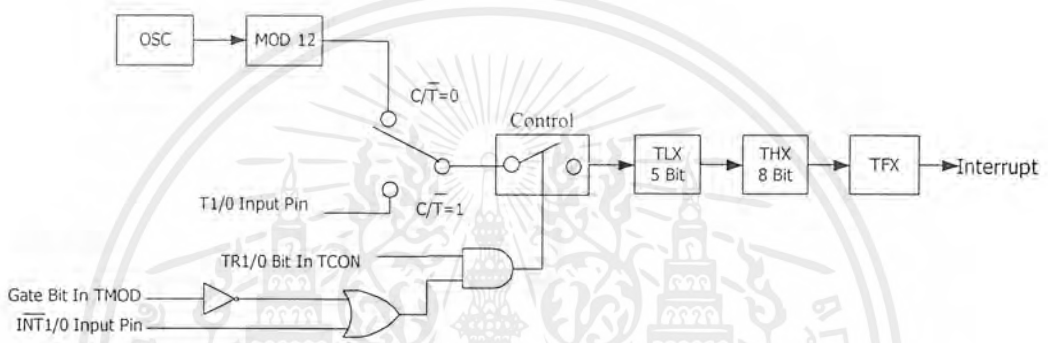
ภายในชิพไมโครคอนโทรลเลอร์ 8051 จะมีไทม์เมอร์/เคาน์เตอร์ 2 ตัว ขนาด 16 บิต คือ ไทม์เมอร์/เคาน์เตอร์ 0 และไทม์เมอร์/เคาน์เตอร์ 1 ส่วนเบอร์ 8032/8052 จะมีเพิ่มอีกหนึ่งชุด คือ ไทม์เมอร์/เคาน์เตอร์ 2 ขณะที่แต่ละ ไทม์เมอร์/เคาน์เตอร์ สามารถที่จะติดตั้งให้ทำงานได้เป็นตัวจับเวลาหรือตัวนับก็ได้ โดยวิธีการเซตหรือการเคลียร์บิตตัวที่ควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR ในฟังก์ชันไทม์เมอร์ ตัวรีจิสเตอร์จะเพ็่งค่าทุก ๆ รอบแมชชีนไซเคิล ดังนั้นค่าของตัวเลขในรีจิสเตอร์จะเป็นจำนวนรอบของแมชชีนไซเคิล และเนื่องจากแต่ละรอบแมชชีนไซเคิลประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะใช้เวลา 1/12 ของความถี่ออสซิลเลเตอร์

ในฟังก์ชันตัวนับรีจิสเตอร์จะมีการเพิ่มค่าทุกครั้งที่มีการเปลี่ยนแปลงสถานะจาก “1” เป็น “0” ที่เข้ามาที่ขา T0 หรือ T1 ในฟังก์ชันนี้สัญญาณภายนอกที่เข้ามาจะถูกสุ่ม (Sampling) ระหว่างช่วง S5P2 ของทุกรอบแมชชีนไซเคิล โดยถ้าสุ่มสัญญาณเข้ามาเป็นระดับสูงในหนึ่งรอบ ดังนั้น ถ้าในหนึ่งรอบต่อมาของสัญญาณเข้ามาเป็นระดับต่ำ รีจิสเตอร์จะนับเพ็่งหนึ่งค่า โดยที่ค่าใหม่ของตัวนับจะปรากฏที่รีจิสเตอร์ช่วง S3P1 ของรอบแมชชีนไซเคิล ซึ่งค่าหนึ่ง ๆ ที่รับเข้าไปจะต้องใช้เวลา 2 แมชชีนไซเคิล (เท่ากับ 24 คาบ) ในการรับค่าช่วงการเปลี่ยน 1 เป็น 0 ดังนั้น ค่าสูงสุดในการนับจะมีอัตรา 1/24 ของความถี่ออสซิลเลเตอร์ และสัญญาณอินพุตที่นับนั้นจะไม่มีช่วงระยะห่างที่แน่นอนของคิวต์ไซเคิล แต่จะถูกนับเมื่อระดับแรงดันที่ถูกสุ่มในแต่ละครั้งจะต้องมีช่วงคงที่อย่างน้อย 1

รอบแมชชีนไซเคิลก่อนที่จะเปลี่ยนค่าระดับแรงดันใหม่ในการเลือกทำงานระหว่างไทม์เมอร์กับ เกลาน์เตอร์จะเลือกได้ 4 โหมด คือ โหมด 0, โหมด 1, โหมด 2 และโหมด 3

1) โหมด 0

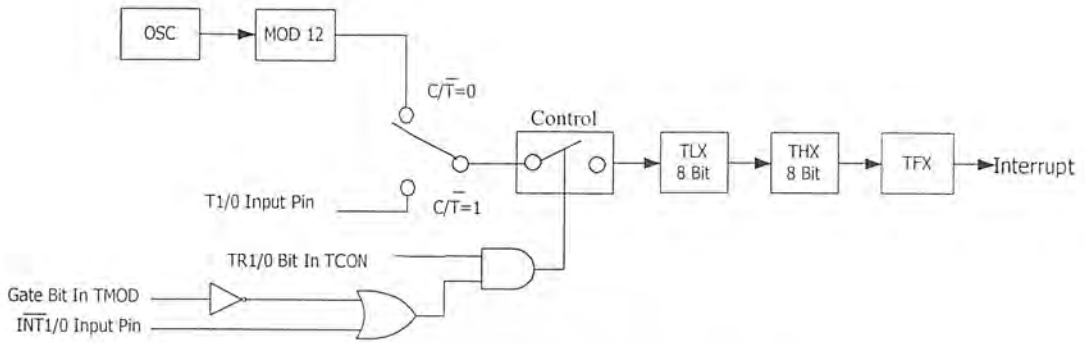
ในโหมด 0 จะเป็นการใช้งานไทม์เมอร์ขนาด 13 บิต โดยจะใช้ไทม์เมอร์ไบต์สูง (THX) ขนาด 8 บิต ต่ออนุกรมกับไทม์เมอร์ไบต์ต่ำ (TLX) อีก 5 บิต (บิต 0 – บิต 4) รวมเป็น 13 บิตส่วน บิตที่ 5 ของไทม์เมอร์ต่ำจะไม่ถูกนำมาใช้งาน ลักษณะการใช้งานของไทม์เมอร์ โหมด 0 ดังรูปที่ 2.11



รูปที่ 2.11 การทำงานของไทม์เมอร์โหมด 0

2) โหมด 1

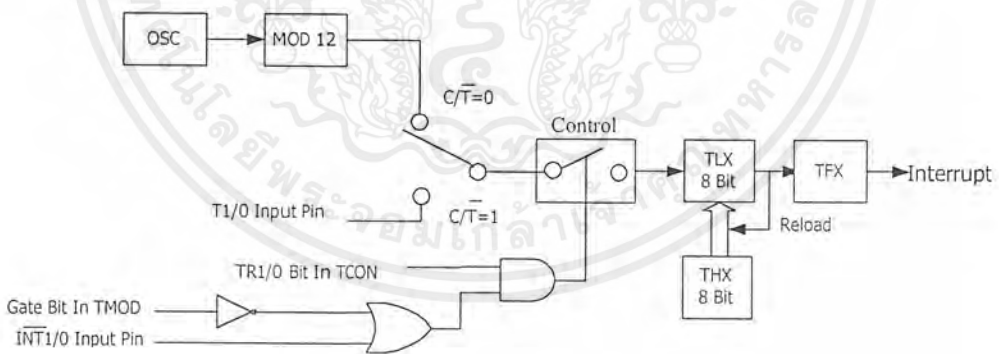
ในโหมด 1 เป็นการใช้งานไทม์เมอร์ขนาด 16 บิต ซึ่งจะทำงานเหมือนกับโหมด 0 (ต่างกันที่ โหมด 1 จะใช้งานทั้ง 16 บิต) สัญญาณนาฬิกาจะป้อนให้รีจิสเตอร์ไทม์เมอร์ที่เกิดจาก TLX และ THX ต่อกันเมื่อได้รับสัญญาณนาฬิกา ไทม์เมอร์จะเริ่มนับจาก 0000H ไปเรื่อย ๆ จนถึง FFFFH ใน รีจิสเตอร์ไทม์เมอร์ (THX และ TLX) สามารถที่จะทำการอ่านหรือเขียนได้โดยซอฟต์แวร์เช่นเดียวกัน ซึ่งลักษณะการทำงานของไทม์เมอร์ โหมด 1 ดังรูปที่ 2.12



รูปที่ 2.12 การทำงานของไทม์เมอร์ โหมด 1

3) โหมด 2

ในโหมด 2 จะให้ไทม์เมอร์ทำงานเป็นแบบ 8 บิต โดยจะใช้ไบต์สูงเก็บค่าที่ต้องการให้ไหลตติมาใหม่เอาไว้เมื่อเกิดการโอเวอร์โฟลว์ซึ่งจะเกิดขึ้นเมื่อการนับเปลี่ยนจาก FFH เป็น 00H แฟล็กของไทม์เมอร์จะเซต และค่าใน THX จะถูกไหลตติมาเข้าไปเก็บไว้ใน TLX อัตโนมัตติ ซึ่งลักษณะการทำงานของไทม์เมอร์ โหมด 2 ดังรูป 2.13

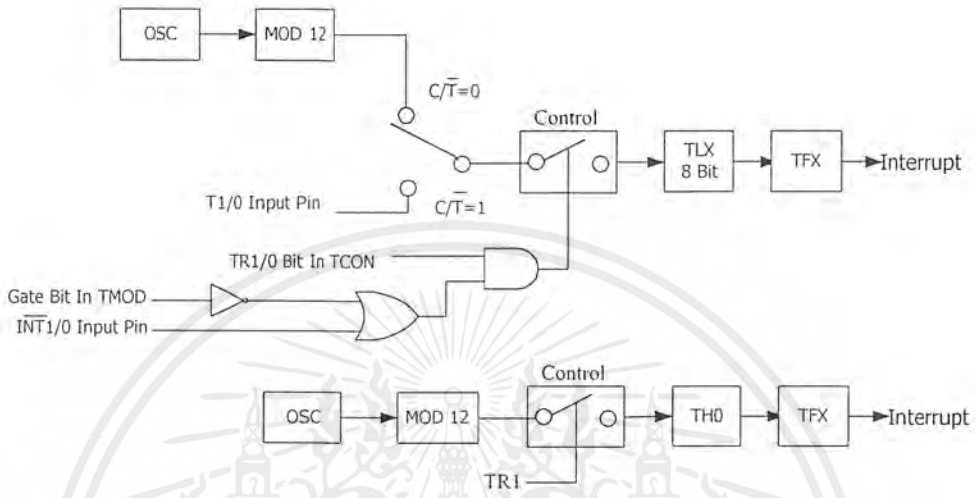


รูปที่ 2.13 การทำงานของไทม์เมอร์โหมด 2

4) โหมด 3

ในโหมด 3 จะเป็นการแยกไทม์เมอร์ให้เป็นอิสระจากกันโดยในโหมด 3 นี้ไทม์เมอร์ 0 จะถูกแยกเป็นไทม์เมอร์เวลาขนาด 8 บิต 2 ชุด คือ TLO และ TH0 ไทม์เมอร์ทั้ง 2 ชุดจะทำงานแยกจาก

กันอย่างอิสระ โดยแฟล็ก TF0 จะเซตเมื่อ TLO เกิดการโอเวอร์โพล์ และแฟล็ก TF1 จะเซตเมื่อ TH0 เกิดการโอเวอร์โพล์ เช่นกัน ลักษณะการทำงานของไทม์เมอร์ โหมด 3 ดังรูปที่ 2.14



รูปที่ 2.14 การทำงานของไทม์เมอร์โหมด 3

หมายเหตุ โหมด 3 นี้ไทม์เมอร์ 1 จะหยุดการทำงานแต่ยังสามารถใช้งานได้โดยสั่งให้ทำงานในโหมดอื่น แต่แฟล็กโอเวอร์โพล์ของไทม์เมอร์ 1 จะไม่มีผลต่อไทม์เมอร์ 1 เมื่อเกิดการโอเวอร์โพล์ขึ้น (เนื่องจากเมื่ออยู่ในโหมด 3 แฟล็กโอเวอร์โพล์ของตัวจับเวลา 1 จะต่ออยู่กับ TH0) การใช้งานแบบนี้จะเหมือนกับมีไทม์เมอร์ขนาด 8 บิต ให้ใช้งานถึง 3 ชุด โดยจะใช้งานไทม์เมอร์ 0 ในโหมด 3 ทำงานส่วนที่ต้องการ ส่วนไทม์เมอร์ 1 อาจใช้เป็นตัวกำเนิดอัตราบอด (Baud Rate Generator) ในการใช้งานพอร์ตอนุกรมได้หรือจะใช้งานอื่น ๆ ที่ไม่เกี่ยวข้องกับอินเทอร์รัพต์ได้

2.2.9 การสื่อสารข้อมูลแบบอนุกรม

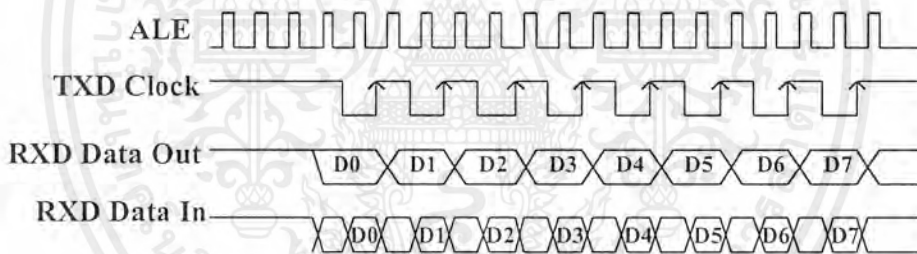
ไมโครคอนโทรลเลอร์ 8051 มีพอร์ตอนุกรม (Serial Port) ที่สามารถสั่งให้ทำงานได้หลายโหมดอยู่ภายในชิพ ซึ่งการทำงานจะเป็น การรับส่งข้อมูลแบบ 2 ทิศทาง (Full Duplex) สามารถที่จะรับและส่งข้อมูลได้พร้อมกัน ในการทำงานจะมีรีจิสเตอร์ที่ทำหน้าที่ในการรับข้อมูล มาเก็บไว้ในบัฟเฟอร์ ในขณะที่กำลังรับข้อมูลตัวต่อไป เมื่อหน่วยประมวลผลกลางอ่านข้อมูลตัวแรกออกไปก่อนที่จะรับข้อมูลตัวต่อไปจนเสร็จ ส่วนบัฟเฟอร์อีกตัวหนึ่งจะใช้สำหรับเก็บข้อมูลก่อนที่จะส่งออกไป บัฟเฟอร์ทั้งด้านรับและด้านส่ง คือรีจิสเตอร์ SBUF (Serial Port Buffer) ซึ่งจะอยู่ในตำแหน่งแอดเดรสที่ 99H ในรีจิสเตอร์ SFR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของารรับส่งข้อมูลทางพอร์ตอนุกรมในชิพไมโครคอนโทรลเลอร์ 8051 จะมีความสะดวก โดยผู้ใช้สามารถกำหนดการทำงานได้ ซึ่งมีการทำงานด้วยกัน 4 ประเภท ดังนี้

1) โหมด 0

ในการใช้งานโหมด 0 จะต้องกำหนดให้บิต SM0 และ SM1 ที่อยู่ในรีจิสเตอร์ SCON มีค่าเป็น 0 ทั้งคู่ ซึ่งจะทำให้รีจิสเตอร์ SBUF รับหรือส่งข้อมูลได้ขนาด 8 บิต โดยใช้ขา RXD ต่อกับสายสัญญาณที่จะใช้ส่งข้อมูลภายนอก ส่วนขา TXD จะใช้ต่อกับวงจรถ่ายสัญญาณนาฬิกา ที่อยู่ในชิพ 8051 โดยใช้ส่งสัญญาณนาฬิกาที่กำหนดความถี่อ้างอิงในการส่งข้อมูลออกมาภายนอก ซึ่งความถี่ที่ส่งออกมาทางขา TXD จะมีค่าเท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ เมื่อทำการส่งข้อมูลออกข้อมูลจะถูกเลื่อนออกมาทีละบิตผ่านขา RXD ในเวลาเดียวกัน สัญญาณพัลส์จะถูกส่งออกมาทางขา TXD ด้วย และในเวลาเดียวกันเมื่อรับข้อมูลเข้าก็จะรับมาทางขา RXD การรับข้อมูลจะเกิดขึ้นเมื่อเซตให้ REN=1 และ RI=0 เมื่อบิต RI ถูกเคลียร์ สัญญาณพัลส์จะถูกส่งออกมาที่ขา TXD เพื่อทำการซิงโครไนซ์กับข้อมูลที่รับเข้ามาที่ขอบด้านบวกของพัลส์ลักษณะดังรูปที่ 2.15



รูปที่ 2.15 การทำงานของพอร์ตอนุกรมโหมด 0

2) โหมด 1

ในการใช้งานโหมด 1 จะต้องกำหนดให้บิต SM0 และบิต SM1 ในรีจิสเตอร์ SCON มีค่าเป็น 0 และ 1 ซึ่งเป็นการกำหนดให้รีจิสเตอร์ SBUF สามารถมีการรับส่งข้อมูลได้ขนาด 10 บิต เป็นรูปแบบการรับส่งข้อมูล 2 ทิศทาง ซึ่งสามารถรับและส่วนข้อมูลได้ในเวลาเดียวกัน (ต่างกับโหมด 0 ก็จะสามารถรับหรือส่งในเวลาหนึ่ง ๆ ได้เพียงอย่างเดียวเท่านั้น) โดยจะใช้ขา RXD รับสัญญาณอนุกรมที่เข้ามา และ TXD ส่งข้อมูลแบบอนุกรมออกไปภายนอก

การส่งข้อมูลจะเริ่มต้นด้วยการส่งบิตเริ่มต้น ออกไปก่อน แล้วตามด้วยบิตข้อมูลอีก 8 บิต จากนั้นจึงทำการส่งบิตหยุด ส่วนแฟล็ก TI จะเซตเมื่อส่งข้อมูลครบทั้ง 10 บิต

การรับข้อมูลจะเริ่มจากขอบขาขึ้นของบิตเริ่มต้น เมื่อครบทั้ง 8 บิตแล้วจะเกิดสถานะดังนี้

- บิตที่ 9 (Stop bit) จะถูกเก็บไว้ใน RB8 ของ SCON
- SBUF จะทำการ โหลดข้อมูลทั้ง 8 บิต ไปเก็บไว้
- แฟล็ก RI จะเซตเป็น "1" ซึ่ง สิ่งที่กำลังกล่าวมาทั้งหมดนี้จะเป็นจริงเมื่อ
- แฟล็ก RI = 0
- SM2 = 1 และ Stop bit ที่รับเข้ามาเป็น 1 หรืออีกกรณีหนึ่งคือ SM2 = 0

ถ้าไม่เกิดทั้ง 2 กรณี การรับข้อมูลก็จะสมบูรณ์ ถ้าเกิดทั้งสองกรณี ตัวบิตหยุดจะไปเก็บที่ RB8 และข้อมูลจะเข้า SBUF และ RI จะแอกทีฟสูง ช่วงเวลานี้ไม่ว่าจะเกิดขึ้นทั้ง 2 กรณีหรือไม่หน่วยควบคุมการรับข้อมูลจะกลับไปตรวจการเปลี่ยนแปลงสถานะการส่งข้อมูลของ RXD ใหม่ ซึ่งลักษณะการทำงานของพอร์ตอนุกรมในโหมด 1 ดังรูปที่ 2.16

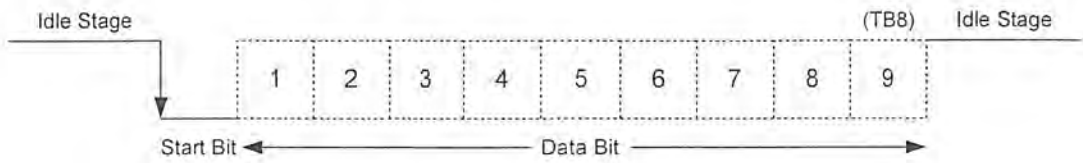


รูปที่ 2.16 การทำงานของพอร์ตอนุกรมโหมด 1

3) โหมด 2

ในการทำงานของโหมด 2 มีการทำงานคล้ายกับโหมด 1 มีความต่างที่โหมด 2 จะเป็นการรับส่งข้อมูลแบบ 11 บิต คือ บิตเริ่มต้น 1บิตม บิตข้อมูล และบิตหยุดอีก 1 บิตเมื่อการรับส่งข้อมูลที่บิตที่ 9จะได้จากบิต TB8 ในรีจิสเตอร์ SCON ส่วนเวลารับข้อมูลบิตที่ 9 จะถูกเก็บไว้ในบิต RB8

ข้อกำหนดในการเซตบิต RI ในโหมด 2 คือ บิต RI จะต้องเป็น 0 ก่อนบิตสุดท้ายที่จะเข้ามาภายในชิพ 8051 และบิต SM2 จะต้องเป็น 0 (หรือบิตที่ 9 ของข้อมูลจะต้องเป็น 1) การรับส่งข้อมูลในโหมดนี้ส่วนมากจะใช้ในการรับส่งข้อมูลระหว่างไมโครโปรเซสเซอร์ด้วยกัน (ซึ่งจะทำการต่อสายสัญญาณเข้ากัน โดยตรง) เนื่องจากอัตราการรับส่งข้อมูลจะสูงมากกว่าโหมด 1 โดยลักษณะการทำงานของพอร์ตอนุกรมในโหมด 2 ดังรูปที่ 2.17



รูปที่ 2.17 การทำงานของพอร์ตอนุกรมโหมด 2

4) โหมด 3

การทำงานในโหมด 3 จะมีการทำงานเหมือนในโหมด 2 ยกเว้นอัตราบอดเท่านั้น โดยจะมีความแน่นอนมากกว่าโหมด 2 โดยซึ่งใช้การคำนวณเหมือนโหมด 1 ทำการสร้างความถี่สำหรับการส่งสัญญาณแทน

2.2.10 การอินเตอร์รัพต์

โปรแกรมคอมพิวเตอร์มีเพียง 2 วิธีเท่านั้นที่จะหาเงื่อนไขที่เป็นจริงของวงจรภายในและภายนอก วิธีแรกคือ ใช้คำสั่งซอฟต์แวร์ กระโดดบนสถานะแฟล็ก และพอร์ตพิน วิธีที่สอง คือ สัญญาณฮาร์ดแวร์ที่เรียกว่า อินเตอร์รัพต์ ซึ่งทำให้โปรแกรมไปเรียกโปรแกรมย่อย เทคนิคของซอฟต์แวร์ใช้เวลาของโปรเซสเซอร์ ซึ่งสามารถไปใช้งานอื่น ๆ อินเตอร์รัพต์จะใช้เวลาของโปรเซสเซอร์เมื่อถูกทำ เมื่อต้องการใช้โปรแกรมประยุกต์การใช้งานเกือบทั้งหมดของไมโครคอนโทรลเลอร์เกี่ยวข้องกับเหตุการณ์ ที่เร็วพอจะควบคุมภาวะแวดล้อมที่สร้างเหตุการณ์

ในชิพ 8051 มีอินเตอร์รัพต์ 5 ชนิด มี 3 ชนิดถูกสร้างโดยฮาร์ดแวร์ โดยโหมดแฟล็ก 0, โหมดแฟล็ก 1, และอินเตอร์รัพต์พอร์ตอนุกรม (RI หรือ TI) อินเตอร์รัพต์อีก 2 ตัวถูกทริกโดยสัญญาณภายนอกโดยวงจรที่ต่อกับขา $\overline{INT0}$ และ $\overline{INT1}$ หน้าที่ของการอินเตอร์รัพต์จะอยู่ภายใต้การควบคุมของโปรแกรมเมอร์ ซึ่งสามารถที่จะเปลี่ยนบิตการควบคุมในรีจิสเตอร์ IE, รีจิสเตอร์ IP และรีจิสเตอร์ TCON โปรแกรมเมอร์สามารถหยุดการอินเตอร์รัพต์ทั้งหมดหรือบางส่วนจากโปรแกรมโดยการรีเซต หรือ การเคลียร์บิตในรีจิสเตอร์ เหล่านี้หลังจากอินเตอร์รัพต์ถูกจัดการโดยโปรแกรมย่อย ซึ่งทำโดยผู้เขียนโปรแกรมลงในหน่วยความจำโปรแกรม โปรแกรมการอินเตอร์รัพต์ต้องทำงานที่คำสั่งที่เกิดการอินเตอร์รัพต์โปรแกรมถูกทำโดยเก็บค่า PC ไปบนสแตคในหน่วยความจำข้อมูลก่อนเปลี่ยนรีจิสเตอร์ PC ซึ่งเป็นค่าตำแหน่งของการอินเตอร์รัพต์ในหน่วยความจำโปรแกรม คำรีจิสเตอร์ PC จะได้จากสแตคหลังจากคำสั่ง RETI ได้ทำแล้วที่ส่วนท้ายของโปรแกรมย่อย การอินเตอร์รัพต์สามารถแบ่งตามลักษณะการเกิดได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) อินเทอร์รัพต์แฟลกไทม์เมอร์

เมื่อไทม์เมอร์/เคาน์เตอร์ เกิดโอเวอร์โฟลว์ ผลคือ ไทม์เมอร์แฟลก (TF0 และ TF1) จะเซตเป็น 1 แฟลกถูกเคลียร์เป็น 0 เมื่ออินเทอร์รัพต์ทำให้โปรแกรมเรียกโปรแกรมย่อยของไทม์เมอร์ในหน่วยความจำโปรแกรม

2) การอินเทอร์รัพต์พอร์ตอนุกรม

เมื่อทำการรับข้อมูลแล้ว บิตอินเทอร์รัพต์ (RI) ในรีจิสเตอร์ SCON จะต้องเซตเป็น 1 เมื่อทำการส่งข้อมูลเรียบร้อยแล้ว บิตอินเทอร์รัพต์ (TI) ตั้งเซตและจะนำมา OR กันเพื่อหาตัวอินเทอร์รัพต์แก่หน่วยประมวลผลกลาง ซึ่งเป็นการอินเทอร์รัพต์พอร์ตอนุกรมบิตเหล่านี้จะไม่ถูกเคลียร์เมื่อมีการเรียกโปรแกรมอินเทอร์รัพต์ถูกทำโดยหน่วยประมวลผลกลาง โปรแกรมที่จัดการสื่อสารข้อมูลอนุกรมต้องรีเซตบิต RI หรือ TI เพื่อจะทำข้อมูลถัดไป

3) การอินเทอร์รัพต์ภายนอก

ขา $\overline{INT0}$ และ $\overline{INT1}$ จะใช้โดยวงจรภายนอก การอินพุตบนขาเหล่านี้จะต้องเซตแฟลก IEO และแฟลก IEI ในรีจิสเตอร์ SCON เป็น 1 โดยมี 2 วิธี แฟลก \overline{INTX} บิต IT0 และบิต IT1 จะต้องโปรแกรมขา \overline{INTX} สำหรับการอินเทอร์รัพต์ที่ 0 และเมื่อเซตเป็น 0 จะต้องทำการโปรแกรมขา \overline{INTX} สำหรับอินเทอร์รัพต์ที่การเปลี่ยนลอจิกเมื่อเซตเป็น 1

แฟลก IEX จะทำการรีเซตเมื่อการอินเทอร์รัพต์ที่เกิดจากการเปลี่ยนลอจิก พบโดยการโปรแกรมและเข้าถึงโปรแกรมย่อย วงจรภายนอกต้องเป็น 1 ก่อนที่บิต RETI จะถูกจัดความผิดพลาดในเรื่องนี้จะทำให้เกิดการอินเทอร์รัพต์ทันทีหลังจากบิต RETI ด้วยแหล่งอินเทอร์รัพต์เดียวกัน

4) การควบคุมการอินเทอร์รัพต์

โปรแกรมต้องสามารถหยุดการอินเทอร์รัพต์ทั้งหมดหรือบางตัวเพื่อให้งานที่สำคัญได้ทำงานเสร็จ รีจิสเตอร์ IE จะเก็บค่าบิตที่โปรแกรมได้ เพื่อให้ทำการอินเทอร์รัพต์ได้ตามต้องการ และเมื่อเลือกอินเทอร์รัพต์แล้ว แหล่งอินเทอร์รัพต์แต่ละแหล่งอาจถูกเลือกหรือไม่ก็ได้

ในการที่ต้องการอินเทอร์รัพต์ที่สำคัญกว่าโดยทันที บิตของรีจิสเตอร์ IP อาจถูกเซตโดยโปรแกรมเพื่อกำหนดความสำคัญของแต่ละแหล่งอินเทอร์รัพต์ เพื่อให้อินเทอร์รัพต์ที่สำคัญกว่าทำการอินเทอร์รัพต์ก่อนเมื่อมีการอินเทอร์รัพต์พร้อมกัน 2 แหล่งขึ้นไป

2.3 โปรแกรมวิซวลเบสิก (Visual Basic)

โปรแกรมวิซวลเบสิก เป็นภาษาคอมพิวเตอร์ (Programming Language) ที่พัฒนาโดย บริษัท ไมโครซอฟท์ โดยตัวภาษามีรากฐานมาจากภาษาเบสิก (Basic) ซึ่งย่อมาจาก Beginner's All-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Purpose Symbolic Instruction ถ้าแปลให้ได้ความหมายคือ “ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น” ภาษาเบสิกมีจุดเด่นคือ สามารถเรียนรู้และนำไปใช้งานได้โดยง่ายและรวดเร็ว เมื่อเทียบกับการเรียนภาษาคอมพิวเตอร์อื่นๆ

2.3.1 คุณสมบัติ

โปรแกรมวิซวลเบสิกสามารถสร้างโปรแกรมบนวินโดว์ (Windows) โดยอาศัยการออกแบบโปรแกรม ในการกำหนดตำแหน่งของวัตถุ (Object) ลงบนจอภาพเพื่อติดต่อกับผู้ใช้โดยตรง วัตถุเหล่านี้จะเปลี่ยนไปตามเหตุการณ์ (Event) ต่างๆที่เกิดขึ้น เช่นการเคลื่อนเมาส์ หรือการรับข้อมูลจากคีย์บอร์ด ในการกำหนดขั้นตอนการทำงานให้กับวัตถุ ภายใต้เหตุการณ์ใดๆ จะใช้ภาษาเบสิกเข้ามาช่วยในการเขียนโปรแกรม ดังนั้นอาจกล่าวได้ว่า การพัฒนาโปรแกรมบนวินโดว์โดยใช้โปรแกรมวิซวลเบสิก มีความง่าย และสะดวกในการใช้งาน โดยการเลือกฟอร์ม (Form) และคอนโทรล (Control) ที่เหมาะสมแล้ววางลงบนจอภาพเพื่อใช้ติดต่อกับผู้ใช้ จากนั้นจึงทำการเขียนภาษาเบสิกเพื่อควบคุม

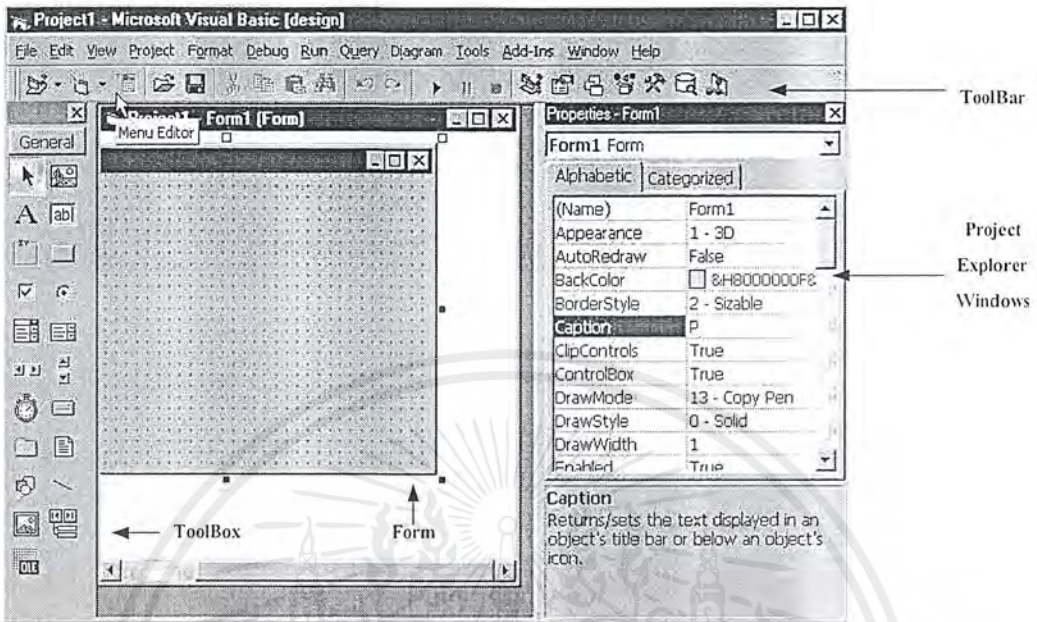
2.3.2 ส่วนประกอบของจอภาพ

ส่วนประกอบของจอภาพ โปรแกรมวิซวลเบสิก แสดงดังรูปที่ 2.18 ซึ่งสามารถอธิบายรายละเอียดได้ ดังตารางที่ 2.2

ตารางที่ 2.2 ส่วนประกอบของจอภาพวิซวลเบสิก

ส่วนประกอบ	รายละเอียด
Form	เป็นจอภาพสำหรับพัฒนาโปรแกรมขึ้นใช้งาน
Toolbox	ประกอบด้วยไอคอนที่จะนำไปวางบนฟอร์ม
Toolbar	ประกอบด้วยไอคอนที่ใช้ร่วมในการพัฒนาโปรแกรม
Project Exploere Windows	ใช้ดูองค์ประกอบต่างๆของโปรเจกต์ (Project)
Properties Windows	ใช้กำหนดคุณสมบัติให้ฟอร์มและวัตถุที่อยู่บนฟอร์ม
Form Layout Windows	ใช้ดูและเลื่อนตำแหน่งของฟอร์มบนจอภาพ

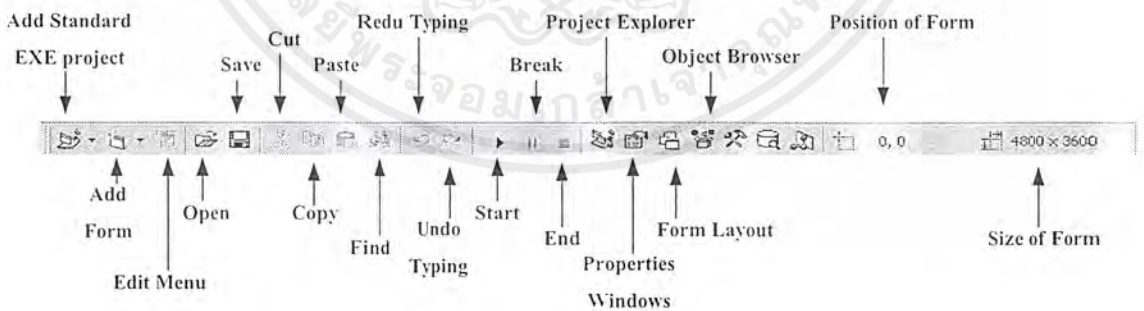
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ส่วนประกอบของจอภาพวิซวลเบสิก

2.3.3 แถบเครื่องมือ (Toolbar)

แถบเครื่องมือจะประกอบด้วยไอคอนต่างๆดังรูปที่ 2.19



รูปที่ 2.19 แถบเครื่องมือของวิซวลเบสิก

แถบเครื่องมือ ทำหน้าที่เป็นผู้ช่วยในการพัฒนาโปรแกรม เมื่อทำการเลื่อนเมาส์ไปชี้ยังไอคอนใด ก็ จะปรากฏชื่ออยู่ใต้ไอคอนนั้น แต่ละไอคอนจะมีหน้าที่แตกต่างกันไป ดังตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แถบเครื่องมือของวิซวลเบสิก

ชื่อไอคอน	รายละเอียดการใช้งาน
Add Standard EXE-Project	ใช้เปิดโปรเจกต์ใหม่ ในกรณีที่มีหลายโปรเจกต์อยู่บนจอภาพ สามารถสลับไปมาระหว่างโปรเจกต์ได้โดยเลือกในเมนูวินโดว์
Add Form	ใช้เพิ่มฟอร์มให้กับโปรเจกต์ ซึ่งอาจเรียกจากเมนูโปรเจกต์ และเพิ่มฟอร์มตามลำดับ
Menu Editor	ใช้เรียกเมนูอิดิเตอร์(Menu Editor) ขึ้นมาเพื่อใช้สร้างเมนูให้กับฟอร์ม อาจกดปุ่ม Ctrl+E หรือเลือกจากเมนู Tools และ Editor ตามลำดับ
Open Project	ใช้เปิดโปรเจกต์ซึ่งอาจกดปุ่ม Ctrl+O หรือเรียกจากเมนู File และ Editor ตามลำดับ
Save Project	ใช้บันทึกโปรเจกต์และฟอร์มอาจเลือกจากเมนู File และ Save Project หรือ Save Project As ตามลำดับ ในกรณีที่ต้องการระบุชื่อของโปรเจกต์ใหม่
Cut	ใช้ตัดวัตถุต่างๆบนฟอร์มอาจใช้ Ctrl+X หรืออาจเลือกจากเมนู Edit และ Cut ตามลำดับ
Copy	ใช้คัดลอกวัตถุบนฟอร์ม ซึ่งอาจใช้ Ctrl+C หรือเลือกจากเมนู Edit และ Copy ตามลำดับ
Paste	ใช้วางวัตถุที่ตัด หรือคัดลอกไว้ อาจจะใช้ Ctrl+V หรือเลือกจากเมนู Edit และ Paste ตามลำดับ
Find	ใช้ค้นหาค่าในอิดิเตอร์ ที่ใช้เขียนโปรแกรม อาจใช้ Ctrl+F หรือเลือกจากเมนู Edit และ Find ตามลำดับ
Undo Typing	ใช้ยกเลิกคำที่พิมพ์ในอิดิเตอร์ อาจใช้ Ctrl+Z หรือเลือกจากเมนู Edit และ Undo Typing ตามลำดับ ในกรณีที่ไม่ได้อยู่ในจอภาพ จะปรากฏข้อความ Con't Undo แทนทั้งในส่วนของผู้ใช้ไอคอนและในเมนู Edit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

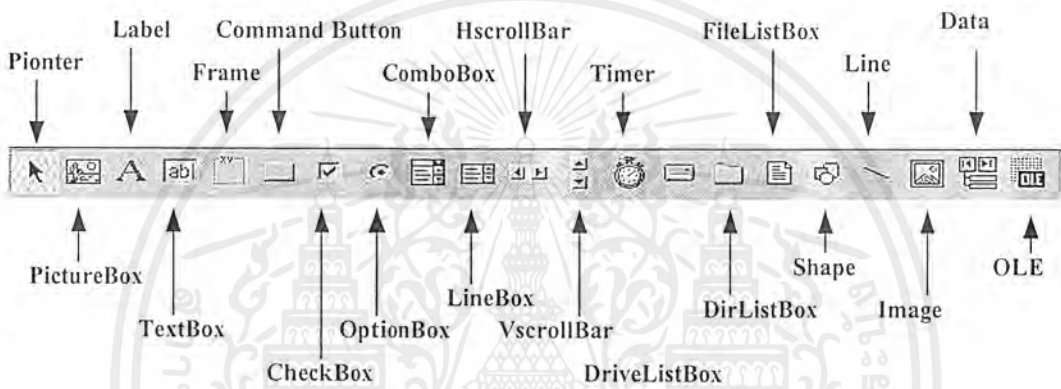
ตารางที่ 2.3 (ต่อ) แถบเครื่องมือของ Visual Basic

ชื่อไอคอน	รายละเอียดการใช้งาน
Redo Typing	ใช้ซ้ำคำที่พิมพ์ในอีดิเตอร์ ซึ่งอาจเลือกจากเมนู Edit และ Redo Typing ตามลำดับ และเช่นเดียวกับ Undo Typing ในกรณีที่ไม่ได้อยู่ในจอภาพอีดิเตอร์ จะปรากฏข้อความ Can't Redo แทน
Start	ใช้เริ่มต้นการทำงานของโปรเจกต์ ที่จัดทำขึ้น อาจใช้ F5 หรือเลือกจากเมนู Run และ Start ตามลำดับ
Break	ใช้หยุดการทำงานของโปรเจกต์ชั่วคราว อาจใช้ Ctrl+Break หรืออาจเลือกจากเมนู Run และ Break ตามลำดับ
End	ใช้หยุดการทำงานของโปรเจกต์ ซึ่งอาจเลือกจากเมนู Run และ End ตามลำดับ
Project Explorer	ใช้แสดงว่าโปรเจกต์นั้นประกอบไปด้วยฟอร์มและโมดูลใดบ้างขึ้นมาบนจอภาพ อาจใช้ Ctrl+R หรือเลือกจากเมนู View และ Project Explorer ตามลำดับ
Properties Windows	ใช้กำหนดคุณสมบัติ(Properties) ของวัตถุและฟอร์ม ซึ่งอาจจะใช้ F4 หรือเลือกจากเมนู View และ Properties Windows ตามลำดับ
Form Layout Windows	ใช้เรียกจอภาพ Form Layout ซึ่งใช้แสดงตำแหน่งของฟอร์มบนจอภาพ อาจจะเลือกจากเมนู View และ Form Layout Windows ตามลำดับ
Object Browser	ใช้แสดงถึงชั้น (Class) และสมาชิกของแต่ละชั้นอาจใช้ F2 หรือเลือกจากเมนู View และ Object Browser ตามลำดับ
Toolbox	ใช้เรียกกล่องเครื่องมือ ขึ้นมาบนจอภาพ อาจเลือกจากเมนู View และ Toolbox ตามลำดับ
ตำแหน่งของฟอร์ม	ใช้บอกตำแหน่งในแกน X และ Y ของฟอร์ม
ขนาดของฟอร์ม	ใช้บอกขนาดของฟอร์ม ตามแนวแกน X และ Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 กล่องเครื่องมือ (Toolbox)

กล่องเครื่องมือของโปรแกรมวิซวลเบสิก ดังรูปที่ 2.20 จะประกอบด้วยไอคอน (Icon) ต่างๆ ที่เรียกว่า “คอนโทรล” โดยจะนำมาใช้ร่วมกับฟอร์ม เพื่อสร้างจอภาพของโปรแกรมที่แต่ละคอนโทรลจะใช้เครื่องมือในการสร้างส่วนที่ติดต่อกับผู้ใช้ หรือที่เรียกว่า “User Interface” เช่นข้อความต่างๆ ช่องว่างสำหรับรับข้อมูลจากคีย์บอร์ด ปุ่มต่างๆ เป็นต้น โดยการนำคอนโทรลที่ต้องการไปวางลงบนฟอร์ม



รูปที่ 2.20 กล่องเครื่องมือของวิซวลเบสิก

2.3.5 ขั้นตอนในการพัฒนาโปรแกรม

ขั้นตอนในการพัฒนาโปรแกรม ประกอบไปด้วยขั้นตอนหลัก 2 ขั้นตอน ดังนี้

1) ขั้นตอนที่ 1 สร้างจอภาพโปรแกรม

ในขั้นตอนนี้ จะทำการออกแบบฟอร์มเพื่อใช้ในการติดต่อกับผู้ใช้ ในการพัฒนาโปรแกรมแบบเดิมขั้นตอนนี้จะใช้เวลาและค่าใช้จ่ายสูง เนื่องจากจะต้องเขียนโปรแกรม เพื่อสร้างจอภาพต่างๆ จากนั้นต้องคอมไพล์ (Compile) โปรแกรมแล้วรัน จึงจะมองเห็นภาพที่จัดทำขึ้น แต่สำหรับโปรแกรมวิซวลเบสิก ปัญหาในลักษณะนี้ได้ถูกแก้ไขให้สามารถทำได้ง่ายขึ้น โดยการนำเอาคอนโทรลต่างๆ ในกล่องเครื่องมือที่ต้องการใช้งานมาวางไว้บนฟอร์มซึ่งทำให้ประหยัดเวลาและสามารถเห็นลักษณะจอภาพที่ออกแบบได้ในขณะนั้นเลย

2) ขั้นตอนที่ 2 เขียนโปรแกรม

เมื่อทำการวางคอนโทรลต่างๆ ลงบนฟอร์มเป็นที่เรียบร้อยแล้ว (คอนโทรลต่างๆ เมื่อถูกวางบนฟอร์มจะเรียกว่า “ออปเจ็กต์”) ขั้นตอนต่อมาคือ การเขียนโปรแกรมเพื่อกำหนดการทำงานให้กับแต่ละออปเจ็กต์ภายใต้เหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับจอภาพนั้นๆ



บทที่ 3

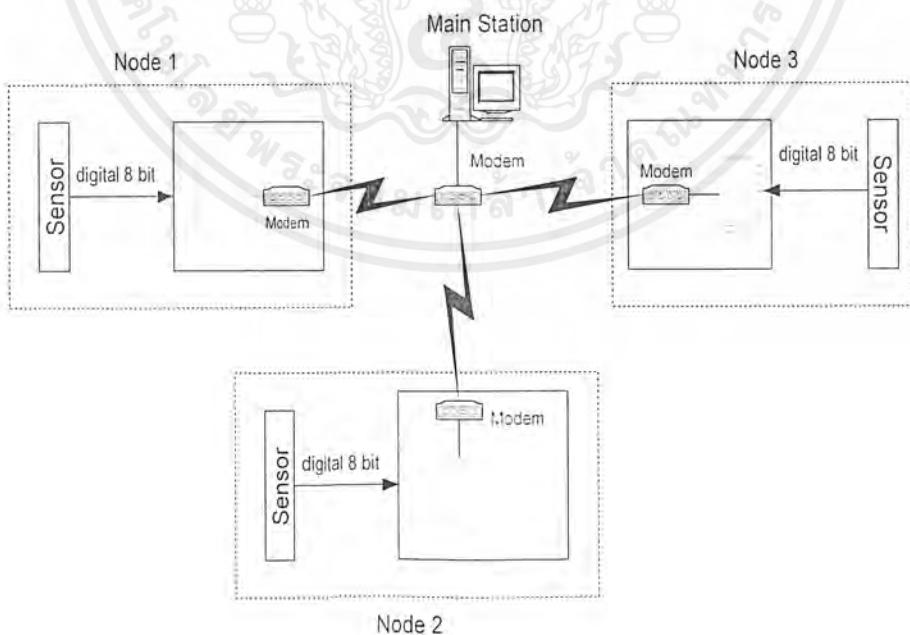
การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการออกแบบและการสร้าง ส่วนของฮาร์ดแวร์และส่วนของซอฟต์แวร์ว่ามีขั้นตอนการออกแบบอย่างไรรวมถึงการทำงานในส่วนของ วงจรที่ออกแบบด้วย

3.2 การออกแบบวงจร

โครงข่ายรวบรวมข้อมูลระยะไกลนี้ออกแบบให้มีการทำงานแบบอัตโนมัติ ซึ่งรวบรวมข้อมูลเป็นวงจรที่ทำหน้าที่เก็บข้อมูลต่างๆ ที่เราต้องการมาพักไว้ในหน่วยความจำเพื่อรอส่งข้อมูลดังกล่าวไปยังไมโครคอมพิวเตอร์ต่อไป โดยในส่วนของวงจรนี้ได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล MCS - 51 เบอร์ #AT89S52 เป็นอุปกรณ์ที่ใช้ควบคุมวงจรส่วนนี้จากแผนผังการทำงานของระบบในรูปที่ 3.1 ประกอบไปด้วย วงจรส่วนรวบรวมข้อมูลซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ MCS - 51 และส่วนของเครื่องคอมพิวเตอร์ที่ทำหน้าที่รับข้อมูลที่ส่งมาจากส่วน รวบรวมข้อมูล



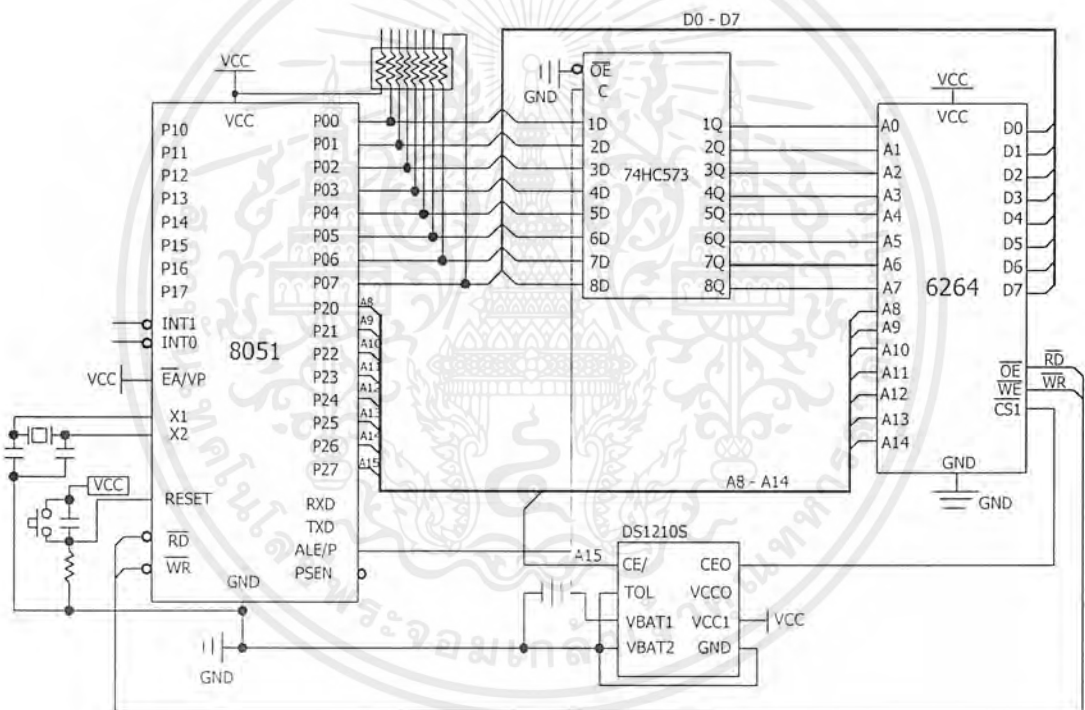
รูปที่ 3.1 แผนผังการทำงานของโครงข่ายรวบรวมข้อมูลระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การออกแบบวงจรชุดรวบรวมข้อมูล

ในส่วนของวงจร ชุดรวบรวมข้อมูลจะถูกควบคุมโดยไมโครคอนโทรลเลอร์ โดยชุดคำสั่งที่บันทึกอยู่ในหน่วยความจำสำหรับโปรแกรม หรืออาจจะควบคุมได้โดยสัญญาณ ผ่านทางไมโครคอมพิวเตอร์โดยใช้โมเด็มเป็นตัวกลางในการติดต่อสื่อสารกันซึ่งสามารถที่จะสั่งให้วงจรส่วนรวบรวมข้อมูลส่งข้อมูลที่ต้องการมาให้ได้ ซึ่งได้ออกแบบวงจรชุดรวบรวมข้อมูลไว้ดังนี้

ในการออกแบบวงจรเบื้องต้นเริ่มจากการนำเอาตัวไอซีไมโครคอนโทรลเลอร์มาต่อร่วมกับไอซีหน่วยความจำสำหรับข้อมูลภายนอก เบอร์ 6264 ซึ่งมีวิธีการเชื่อมต่อกับหน่วยความจำสำหรับข้อมูลแสดงดังรูปที่ 3.2



รูปที่ 3.2 การเชื่อมต่อกับหน่วยความจำสำหรับข้อมูล

จากแผนผังการเชื่อมต่อหน่วยความจำสำหรับข้อมูลกับไมโครคอนโทรลเลอร์นี้จะประกอบด้วยอุปกรณ์และมีหน้าที่การทำงานดังนี้

- 1) ไอซี ไมโครคอนโทรลเลอร์ ทำหน้าที่ประมวลผลตามโปรแกรมที่เราป้อนลงหน่วยความจำสำหรับโปรแกรมซึ่งสามารถสั่งงานได้จากพอร์คโดยตรง โดยมีคริสตอลเป็นตัวกำเนิดความถี่ป้อนให้กับวงจร

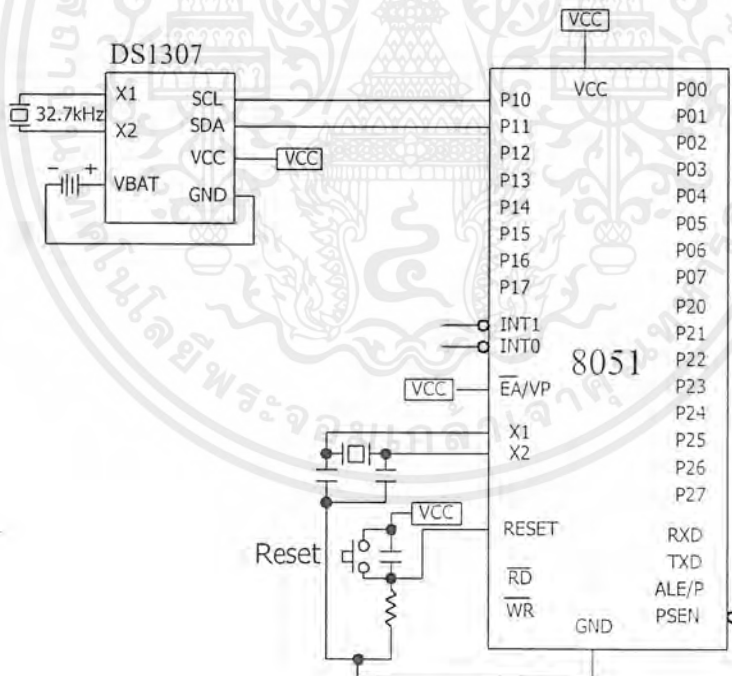
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ไอซี 74HC573 เป็นไอซีที่ทำหน้าที่ Late ข้อมูลโดยจะค้างสถานะการทำงานไว้ที่ตัวมัน แล้วส่งต่อไปยัง Port Address ของหน่วยความจำที่ขา A0-A7

3) ไอซี 6264 เป็นไอซีที่ทำหน้าที่ขยายหน่วยความจำในกรณีที่ตัวไมโครคอนโทรลเลอร์มีหน่วยความจำไม่พอในการประมวลผล สามารถเขียนข้อมูลลงไปหน่วยความจำได้

3.2.2 ส่วนของวงจรสร้างฐานเวลา

ส่วนของวงจรฐานเวลาในวงจรรวบรวมข้อมูล ถูกออกแบบให้ใช้ไอซีเบอร์ DS1307 ซึ่งเป็นไอซีที่ทำหน้าที่สร้างฐานเวลาจริงให้กับระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลที่เกี่ยวกับเวลาทั้งหมด โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง ส่วนของวงจรถา SDA ซึ่งเป็นขา DATA ของ DS1307 จะต่อเข้ากับขา P2.6 ของไมโครคอนโทรลเลอร์ และขา SCL ซึ่งเป็นขา CLOCK ของ DS1307 จะถูกต่อเข้ากับ P2.5 ของ ไมโครคอนโทรลเลอร์ และวงจรจะต้องมี BATTERY 3V ในการทำหน้าที่การทำงาน อยู่ตลอดเวลาเมื่อไม่มีไฟเลี้ยงวงจร สำหรับวงจรแสดงดังรูปที่ 3.3



รูปที่ 3.3 ส่วนวงจรสร้างฐานเวลา

3.2.3 วงจรการสื่อสารข้อมูลผ่านพอร์ตอนุกรม

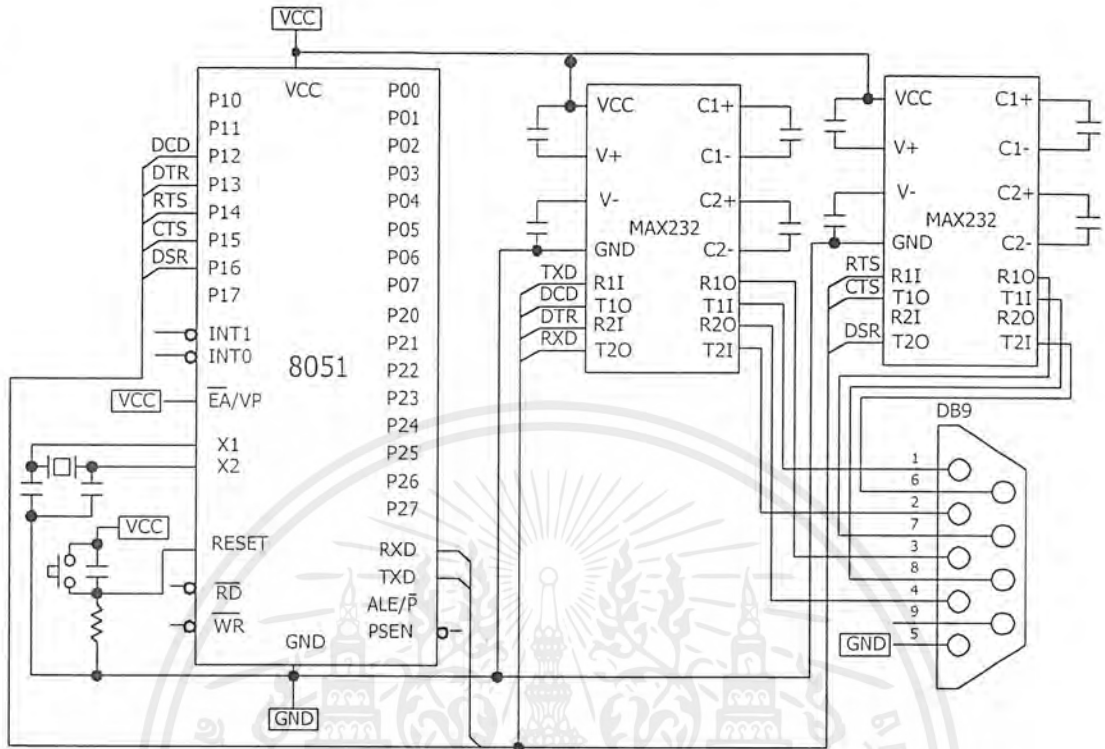
วงจรการสื่อสารข้อมูลผ่านทางพอร์ตอนุกรมของไมโครคอนโทรลเลอร์เป็นการสื่อสารข้อมูลขนาด 8 บิตการออกแบบวงจร จะเชื่อมต่อขา TXD และ RXD ของไมโครคอนโทรลเลอร์ซึ่งเป็นขาซึ่งไมโครคอนโทรลเลอร์ใช้ติดต่อสื่อสารแบบอนุกรม กับไอซี MAX232 ซึ่งเป็นไอซีที่ทำหน้าที่ปรับระดับแรงดัน ที่ไมโครคอนโทรลเลอร์ให้เป็นมาตรฐานเดียวกันกับไมโครคอมพิวเตอร์เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 จะมีระดับสัญญาณตั้งแต่ ± 3 ถึง ± 12 V ในขณะที่สัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับที่ทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรงจึงต้องอาศัยการเชื่อมต่อผ่านไอซี MAX232 ซึ่งทำหน้าที่ในการแปลงระดับสัญญาณ

คอนเนกเตอร์สำหรับมาตรฐานการเชื่อมต่อแบบ RS-232 เลือกใช้คอนเนกเตอร์แบบ DB-9 ซึ่งจะมีหน้าที่ ของขาสัญญาณของพอร์ตอนุกรมแบบ DB-9 แสดงไว้ในตารางที่ 3.1

ตารางที่ 3.1 หน้าที่ขาสัญญาณของพอร์ตอนุกรม

ขา	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	Data Carrier Detect : DCD	อินพุต
2	Received Data : RxD	อินพุต
3	Transmitted Data : TxD	เอาต์พุต
4	Data Terminal Ready : DTR	เอาต์พุต
5	Signal Ground : Gnd	-
6	Data Set Ready : DSR	อินพุต
7	Request To Send : RTS	เอาต์พุต
8	Clear To Send : CTS	อินพุต
9	Ring Indicator : Ri	อินพุต

สำหรับวงจรของการต่อไมโครคอนโทรลเลอร์ MCS-51 แสดงดังรูปที่ 3.4



รูปที่ 3.4 วงจรสื่อสารข้อมูลผ่านพอร์ตอนุกรม

3.3 การออกแบบโปรแกรม

3.3.1 เครื่องมือที่ใช้ในการออกแบบสร้างโปรแกรม

1) ทางด้านฮาร์ดแวร์(HARD WARE)

- เครื่องคอมพิวเตอร์

2) ทางด้านซอฟต์แวร์

- Microsoft Visual Basic 6.0

3.3.2 ขั้นตอนการออกแบบโปรแกรม

1) ออกแบบการทำงาน Flow Chart

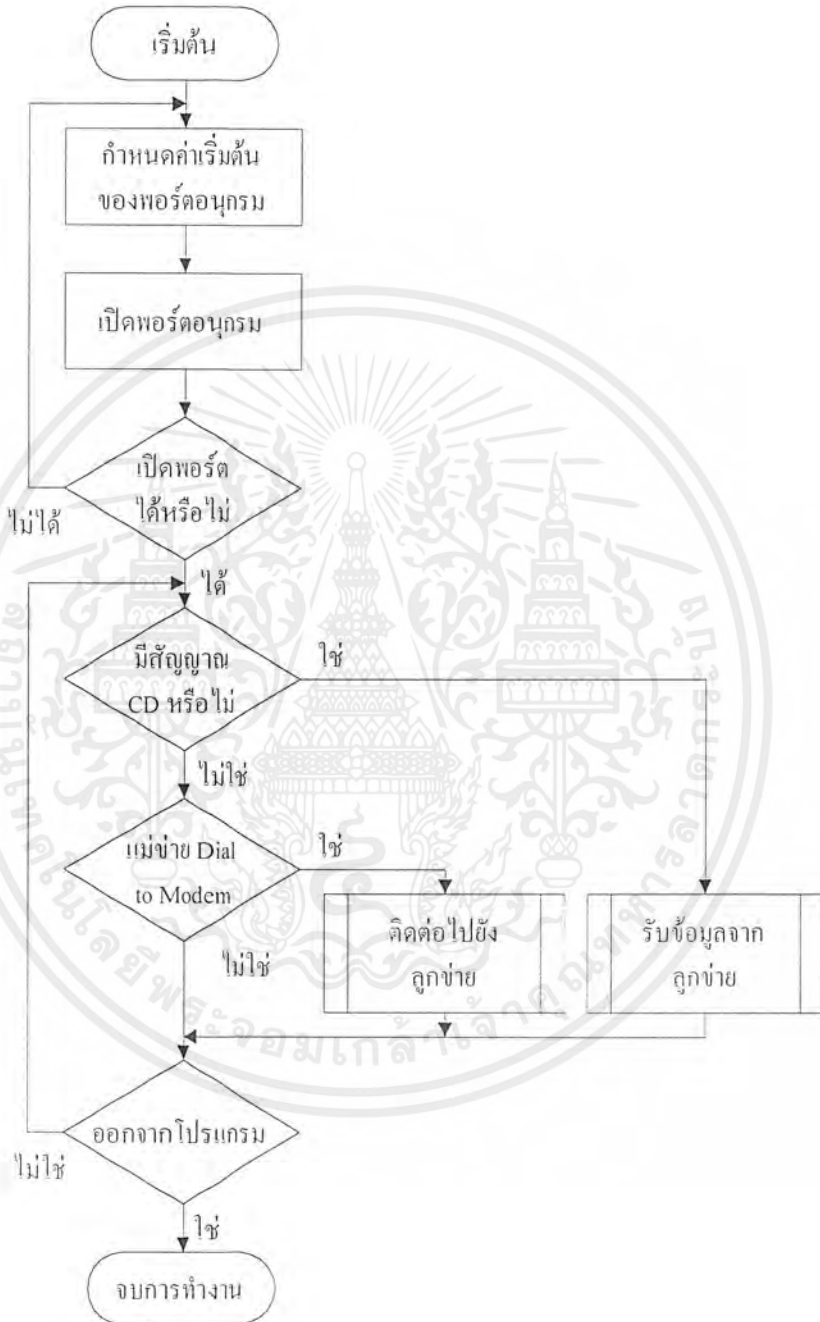
2) ทดสอบโปรแกรมการใช้งาน

3) แก้ไขปรับปรุงโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 แนวคิดเกี่ยวกับโครงสร้างของโปรแกรม

ผังงานของโปรแกรมหลักที่เครื่องไมโครคอมพิวเตอร์



รูปที่ 3.5 ผังงานของโปรแกรมเครื่องคอมพิวเตอร์เม้าส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 ผังงานโปรแกรมการติดต่อกับเครื่องลูกข่าย

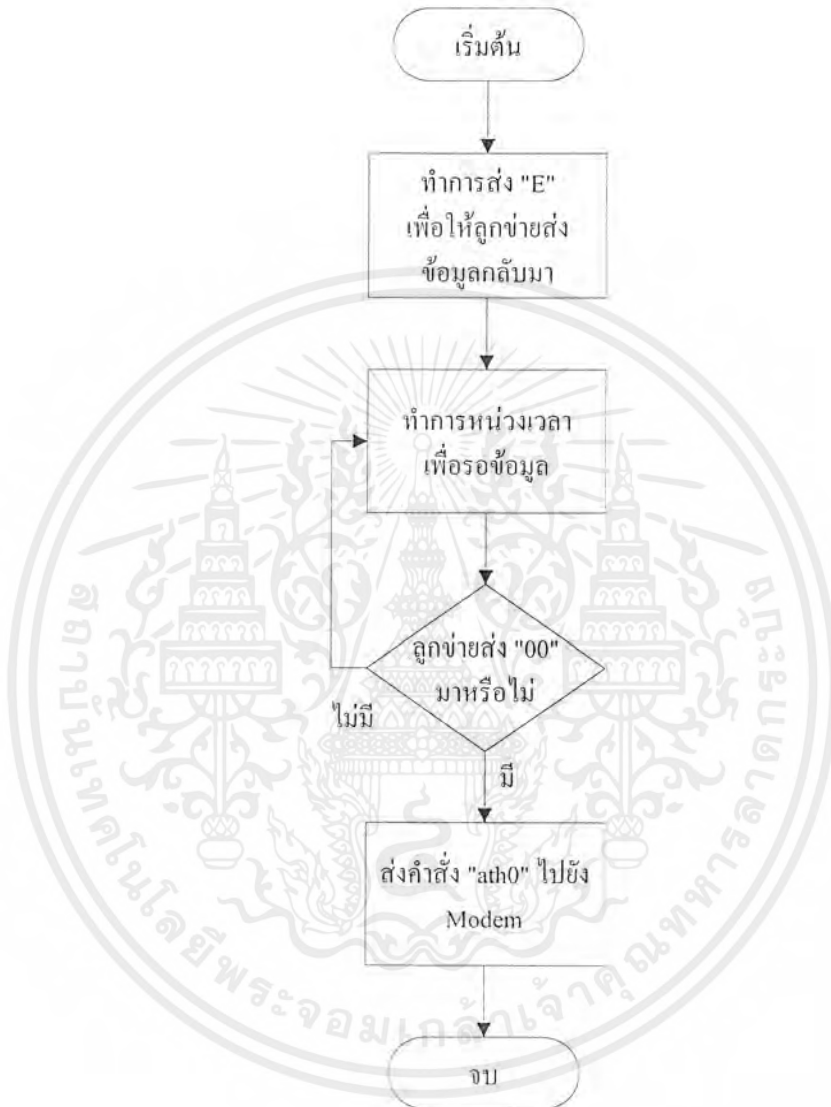
ผังงาน โปรแกรมย่อยที่เรียกไปยังเครื่องลูกข่าย



รูปที่ 3.6 ผังงาน โปรแกรมย่อยที่เรียกไปยังเครื่องลูกข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 ฟังก์ชันการรับข้อมูลจากเครื่องลูกข่าย



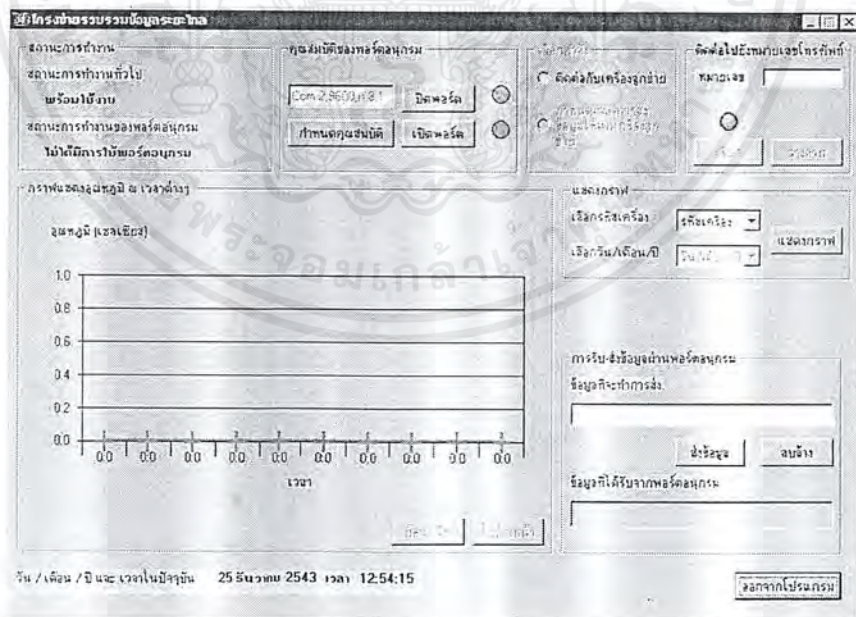
รูปที่ 3.7 ฟังก์ชันของการรับข้อมูลจากเครื่องลูกข่าย

ในส่วน of เครื่องคอมพิวเตอร์แม่ข่าย จะต้องทำหน้าที่การทำงานคือสามารถรับการเรียกเข้ามาจากวงจรรวบรวมข้อมูลเมื่อส่วนรวบรวมข้อมูลต้องการส่งข้อมูลเข้ามา และยังสามารถเรียกไปยังส่วนรวบรวมข้อมูลได้เมื่อเครื่องแม่ข่ายต้องการข้อมูลจากวงจรส่วนรวบรวมข้อมูล จากฟังก์ชันดังกล่าวสามารถอธิบายการทำงานได้คือ เมื่อโปรแกรมเริ่มทำงาน ผู้ใช้จะต้องกำหนดค่าเริ่มต้นให้กับโปรแกรม เช่นการ กำหนดพอร์ตอนุกรมที่ใช้ในการสื่อสาร การกำหนดอัตราบอด(Baud Rate) การเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดพาริตี (Parity) การกำหนดบิตจอบ และการกำหนดบิตข้อมูลที่ใช้ในการสื่อสารข้อมูลระหว่างเครื่องแม่ข่ายและวงจรส่วนรวบรวมข้อมูล ซึ่งในส่วนนี้ผู้ใช้อาจจะข้ามผ่านขั้นตอนนี้ไปได้ จากนั้นก่อนที่ผู้ใช้จะสามารถติดต่อสื่อสารกับวงจรส่วนรวบรวมข้อมูลได้นั้นจะต้องทำการเปิดพอร์ตสื่อสารอนุกรมก่อนไม่เช่นนั้นจะไม่สามารถติดต่อสื่อสารกับวงจรส่วนรวบรวมข้อมูลได้ ถ้าทำการเปิดพอร์ตแล้วมีปัญหาโปรแกรมไม่ยอมเปิดพอร์ตให้ใช้งานจะต้องไปทำการกำหนดค่าเริ่มต้นใหม่ก่อนซึ่งอาจจะเป็นการเลือกใช้พอร์ตอนุกรมผิด แต่ถ้าไม่มีปัญหาในการเปิดพอร์ตใช้งานโปรแกรมจะเช็คว่ามีการเรียกเข้ามาจากวงจรส่วนรวบรวมข้อมูลหรือไม่ถ้ามีก็จะไปทำงานโปรแกรมย่อยที่ใช้ในการรับข้อมูลจากวงจรส่วนรวบรวมข้อมูล หรือว่าเครื่องแม่ข่ายต้องการจะติดต่อกับวงจรส่วนรวบรวมข้อมูลหรือไม่ถ้าต้องการจะติดต่อกับวงจรส่วนรวบรวมข้อมูลซึ่งอาจจะเป็นการเรียกให้ส่งข้อมูลเข้ามา โปรแกรมจะไปทำงานในโปรแกรมย่อยที่ใช้ในการติดต่อไปยังวงจรส่วนรวบรวมข้อมูล

3.3.6 การสร้างโปรแกรม

ในการสร้างโปรแกรมใช้งานได้เลือกใช้โปรแกรม Microsoft Visual Basic 6.0 ในการสร้างซึ่งในขั้นแรกได้ออกแบบสร้างฟอร์มหลักในการทำงานดังรูปที่ 3.6 ซึ่งจะประกอบไปด้วยปุ่มคำสั่งต่างที่ออกแบบไว้ดังนี้



รูปที่ 3.8 แสดงการสร้างฟอร์มหลักของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) สถานะการทำงาน

1.1) สถานะการทำงานทั่วไป ใช้แสดงสถานะการทำงานของโปรแกรมว่าใช้งานได้หรือไม่

1.2) สถานะการทำงานของพอร์ตอนุกรม ใช้แสดงว่าพอร์ตอนุกรมที่เราเลือกสามารถเปิดใช้งานได้หรือไม่

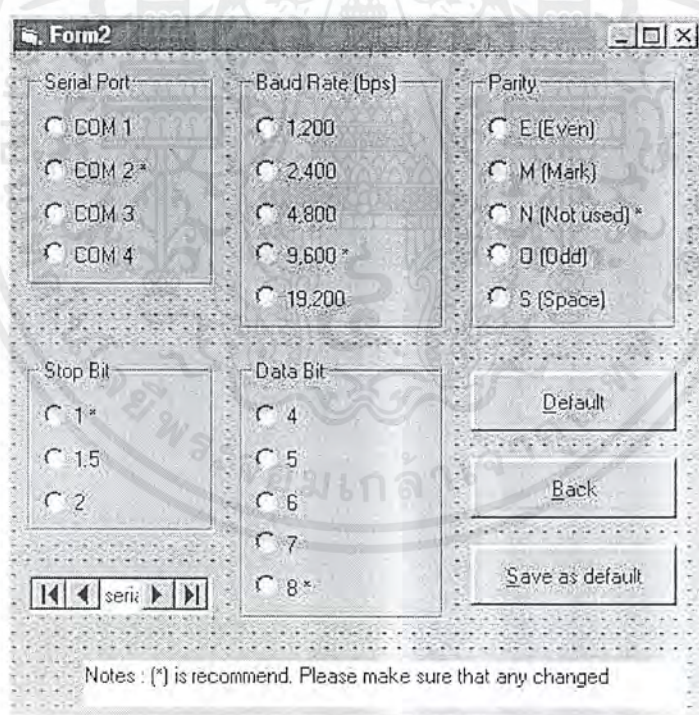
2) คุณสมบัติของพอร์ตอนุกรม

3) การติดต่อกับเครื่องลูกข่าย

4) การแสดงกราฟ

5) การรับส่งข้อมูล

จากรูปที่ 3.6 เป็นการออกแบบหน้าต่างการทำงานหลักของโปรแกรม ซึ่งจะต้องมีฟอร์มสำหรับกำหนด Properties ซึ่งได้สร้างขึ้นเพื่อใช้สำหรับการกำหนดค่าเริ่มต้นให้กับโปรแกรม เมื่อคลิกเมาส์ที่ปุ่ม กำหนดคุณสมบัติ ดังรูปที่ 3.7



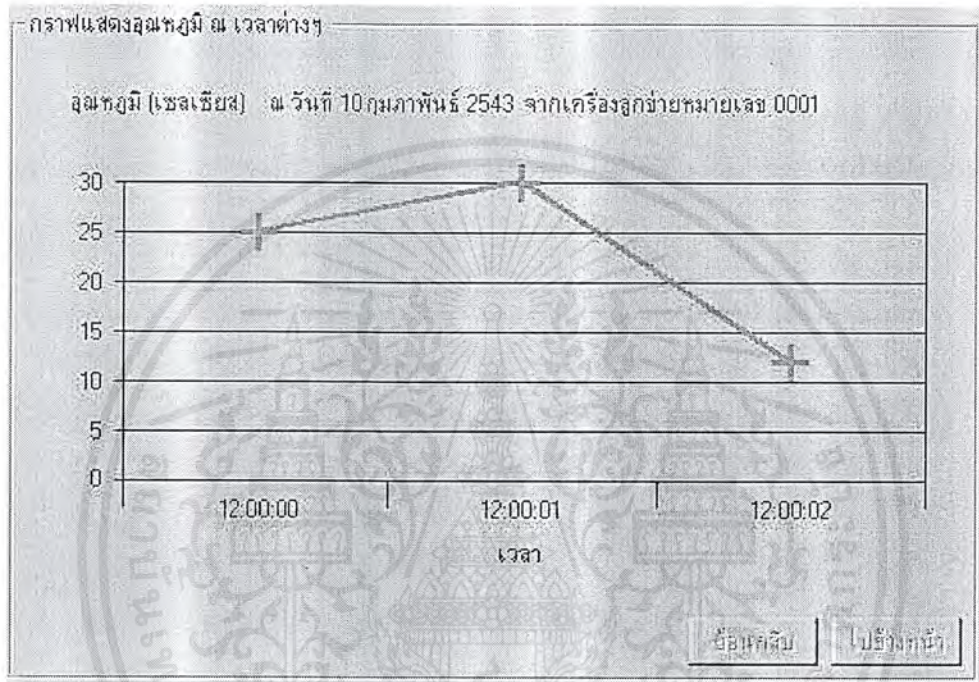
รูปที่ 3.9 การสร้างหน้าต่างการกำหนดค่าเริ่มต้น

ในฟอร์มของ Properties นี้ ได้กำหนดให้ผู้ใช้งานสามารถเลือกใช้งานพอร์ตอนุกรมได้ สามารถกำหนดอัตราบอดได้ สามารถ กำหนดพาริตีได้ สามารถกำหนดบิตหยุด และสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดจำนวนบิตในการรับส่งข้อมูลได้อีกทั้งยังสามารถ บันทึกการกำหนดค่าของ Properties ได้อีกด้วย

ในรูปที่ 3.8 แสดงการออกแบบฟอร์มเพื่อใช้ในการแสดงกราฟของข้อมูลที่ได้รับจากลูกข่ายและจากฐานข้อมูลที่เก็บไว้ดังรูป



รูปที่ 3.10 การออกแบบฟอร์มการแสดงผลกราฟข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

4.1 กล่าวนำ

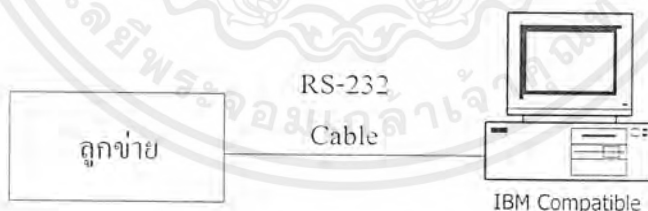
ในส่วนของการทดลองการทำงานของระบบโครงข่ายรวบรวมข้อมูลระยะไกลได้ทำการแบ่งการทดลองออกเป็นสองส่วนคือ การทดลองการทำงานของวงจรโครงข่ายรวบรวมข้อมูลระยะไกลที่สร้างขึ้น และการทดลองการทำงานของโปรแกรมคอมพิวเตอร์ที่คอมพิวเตอร์แม่ข่าย แล้วนำทั้งสองส่วนมาทำงานร่วมกันเพื่อทดสอบอีกครั้งหนึ่ง

4.2 การทดลองการทำงานของวงจรโครงข่ายรวบรวมข้อมูล

4.2.1 การทดลองการเก็บรวบรวมข้อมูล

1) ลำดับขั้นการทดลอง

- 1.1) ต่อวงจรโครงข่ายรวบรวมข้อมูลที่สร้างขึ้นกับเครื่องคอมพิวเตอร์ตามรูปที่ 4.1
- 1.2) ป้อนแรงดันไฟฟ้ากระแสตรงให้กับวงจรโครงข่ายรวบรวมข้อมูล
- 1.3) เปรียบเทียบการเก็บข้อมูลจากวงจรและจากวิธีปกติ



รูปที่ 4.1 การเชื่อมต่อในการทดลองเก็บรวบรวมข้อมูล

2) ผลการทดลอง

เมื่อทำการป้อนแรงดันไฟฟ้ากระแสตรงให้กับวงจรผลการทดลอง ปรากฏว่า วงจรโครงข่ายรวบรวมข้อมูลที่สร้างขึ้นมีการเก็บรวบรวมข้อมูลได้ใกล้เคียงกับการเก็บข้อมูลด้วยวิธีปกติ สำหรับผลการทดลองแสดงดังตารางที่ 4.1

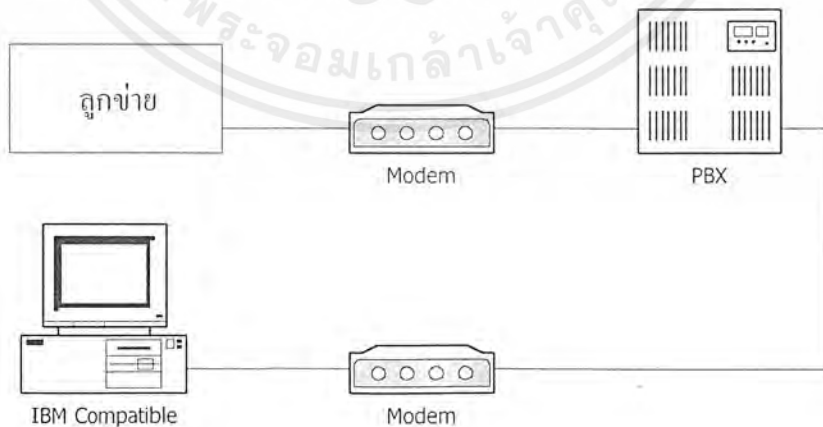
ตารางที่ 4.1 ผลการทดลองเก็บรวบรวมข้อมูล

ครั้งที่	ผลการเก็บข้อมูลจาก โครงข่ายฯ	ผลการเก็บข้อมูลด้วย วิธีปกติ
1	27.5	28
2	29	29
3	30	30
4	25	26
5	26.5	25
6	25.5	25

4.2.2 การทดลองการติดต่อสื่อสารผ่านโมเด็ม

1) ลำดับขั้นตอนการทดลอง

- 1.1) ต่อโครงข่ายรวบรวมข้อมูลกับเครื่องคอมพิวเตอร์แม่ข่ายดังรูปที่ 4.2
- 1.2) ป้อนแรงดันไฟฟ้ากระแสตรงให้กับวงจรโครงข่ายรวบรวมข้อมูลระยะไกล
- 1.3) อ่านข้อมูลที่โครงข่ายรวบรวมข้อมูลส่งมาที่เครื่องคอมพิวเตอร์แม่ข่ายแล้วเปรียบเทียบข้อมูลที่ได้อกับการเก็บข้อมูลแบบปกติ

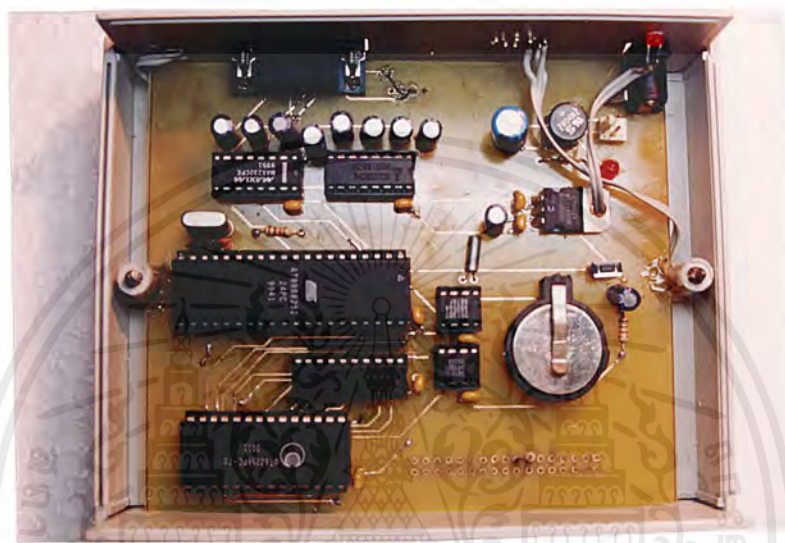


รูปที่ 4.2 การทดลองติดต่อสื่อสารผ่าน โมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ผลการทดลอง

เมื่อทำการทดลองเชื่อมต่อวงจรโครงข่ายรวบรวมข้อมูลเข้ากับโมเด็ม และทดลองเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ติดต่อไปยังเครื่องคอมพิวเตอร์แม่ข่าย ผลการทดลอง ปรากฏว่า ข้อมูลที่ได้จากวงจรโครงข่ายรวบรวมข้อมูลมีค่าใกล้เคียงกับวิธีการปกติ



รูปที่ 4.3 วงจร โครงข่ายรวบรวมข้อมูลระยะไกล

4.3 สรุปผลการทดลอง

การทดลองในบทนี้เป็นการทดลองการทำงานของระบบ โดยแยกการทำงานออกเป็น ส่วน ๆ แล้วจึงนำแต่ละส่วนมาทดลองการทำงานทั้งหมดของระบบ ซึ่งจากการทดลองที่ผ่านมา ผลการทดลองแสดงให้เห็นว่า โครงข่ายรวบรวมข้อมูลระยะไกลสามารถทำงานได้ผลเป็นที่น่าพอใจ ดังตาราง 4.2 ซึ่งแสดงผลการทดลองส่งข้อมูลในแต่ละครั้งซึ่งมีความผิดพลาดของข้อมูลลดลง

ตารางที่ 4.2 สรุปผลการทดลองส่งข้อมูล

ครั้งที่	จำนวนข้อมูล	จำนวนข้อมูลที่ผิดพลาด	จำนวนข้อมูลที่ถูก ต้อง
1	100	69	31
2	100	50	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 (ต่อ) สรุปผลการทดลองส่งข้อมูล

ครั้งที่	จำนวนข้อมูล	จำนวนข้อมูลที่ผิดพลาด	จำนวนข้อมูลที่ถูกต้อง
3	100	55	45
4	100	66	34
5	100	82	18
6	100	94	6

ในการทดลองครั้งที่ 1 ได้ทดลองส่งข้อมูลจำนวน 100 บิตโดยผ่าน โครงข่ายรวบรวมข้อมูลระยะไกล ปรากฏว่าการรับส่งข้อมูลมีความผิดพลาดมากซึ่งมีจำนวนบิต ที่ส่งข้อมูลผิดพลาดไปเป็นจำนวน 69 บิต ซึ่งมากกว่า 50 เปอร์เซ็นต์ของข้อมูลทั้งหมดจึงได้ทำการแก้ไขปรับปรุงให้ดีขึ้นตามลำดับโดยจะเห็นว่าในการทดลองครั้งที่ 6 ซึ่งเป็นครั้งสุดท้ายมีจำนวนบิตข้อมูลที่รับส่งผิดพลาดไป 6 บิต ถือว่าอยู่ในเกณฑ์ที่สามารถยอมรับได้

บทที่ 5

บทสรุป ปัญหาแนวทางการแก้ไข และการพัฒนา

5.1 บทสรุป

โครงข่ายรวบรวมข้อมูลระยะไกลนี้ได้จัดทำขึ้นเพื่อเป็นแนวทางในการพัฒนาและปรับปรุง ระบบการเก็บรวบรวมข้อมูลแบบเดิมที่ใช้สมุดจดบันทึก ให้มีความสะดวกรวดเร็วและได้ข้อมูลที่มีความถูกต้องมากที่สุด ซึ่งเป็นจุดประสงค์หลักที่ต้องการจะนำไปใช้ แทนการเก็บรวบรวมข้อมูลแบบเดิม จากการทำงานของระบบโครงข่ายรวบรวมข้อมูลระยะไกลพบว่า การเก็บรวบรวมข้อมูลโดยใช้วิธีการนี้ จะมีความรวดเร็วในการเก็บข้อมูลมากกว่าระบบเดิมและสามารถที่จะลดการใช้แรงงานคนลงไปได้มากอีกทั้งยังสามารถ รับส่งข้อมูลจากระยะไกลได้อย่างรวดเร็วซึ่งจะเป็นการประหยัดเวลาลงไปได้มาก

ข้อดีของโครงข่ายรวบรวมข้อมูลระยะไกลนี้คือ สามารถทำงานได้ดีใกล้เคียงกับระบบรวบรวมข้อมูลแบบเก่า แต่สามารถประหยัดทรัพยากร และเวลาลงไปได้อย่างมาก

ข้อเสียของ โครงข่ายรวบรวมข้อมูลระยะไกลคือในการรับส่งข้อมูลทางสายโทรศัพท์ยังไม่ค่อยมีความปลอดภัยในการรับ/ส่งข้อมูลมากนัก

5.2 ปัญหาในการทำโครงการ

ในการจัดทำโครงการโครงข่ายรวบรวมข้อมูลระยะไกลนี้ ได้มีปัญหาก่อเกิดขึ้นหลายประการด้วยกัน โดยมีรายละเอียดของปัญหาและแนวทางการแก้ไข ดังนี้

1. เนื่องจากโครงข่ายรวบรวมข้อมูลระยะไกล มีการทำงานโดยใช้การติดต่อสื่อสารระหว่างโปรแกรมวิซวลเบสิกกับชุดวงจรไมโครคอนโทรลเลอร์ผ่านโครงข่ายสายโทรศัพท์ ซึ่งข้อมูลที่กล่าวถึงรายละเอียดในส่วนนี้ยังมีน้อย ทำให้การศึกษาและพัฒนาจึงเป็นไปอย่างล่าช้า

2. การรับส่งข้อมูลจาก วงจรไมโครคอนโทรลเลอร์มีความล่าช้าและบางครั้งก็เกิดความผิดพลาด

5.3 การแก้ปัญหา

1. สอบถามจากผู้รู้ ศึกษาค้นคว้าจากปริญญาณิพนธ์ต่างๆ และค้นหาข้อมูลจากอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แก้ไขโปรแกรมแอสเซมบลี ของไมโครคอนโทรลเลอร์ให้มีประสิทธิภาพในการรับส่งข้อมูลมากขึ้น

5.6 แนวทางในการพัฒนา

1. ออกแบบวงจรให้ใช้พลังงานให้ต่ำลงเพื่อประหยัดกำลังงาน
2. ออกแบบวงจรให้มีการติดต่อสื่อสารกันแบบไร้สาย
3. พัฒนาโปรแกรมแอสเซมบลี ของไมโครคอนโทรลเลอร์และโปรแกรมวิซวลเบสิกให้มีความกระชับและมีประสิทธิภาพมากขึ้น



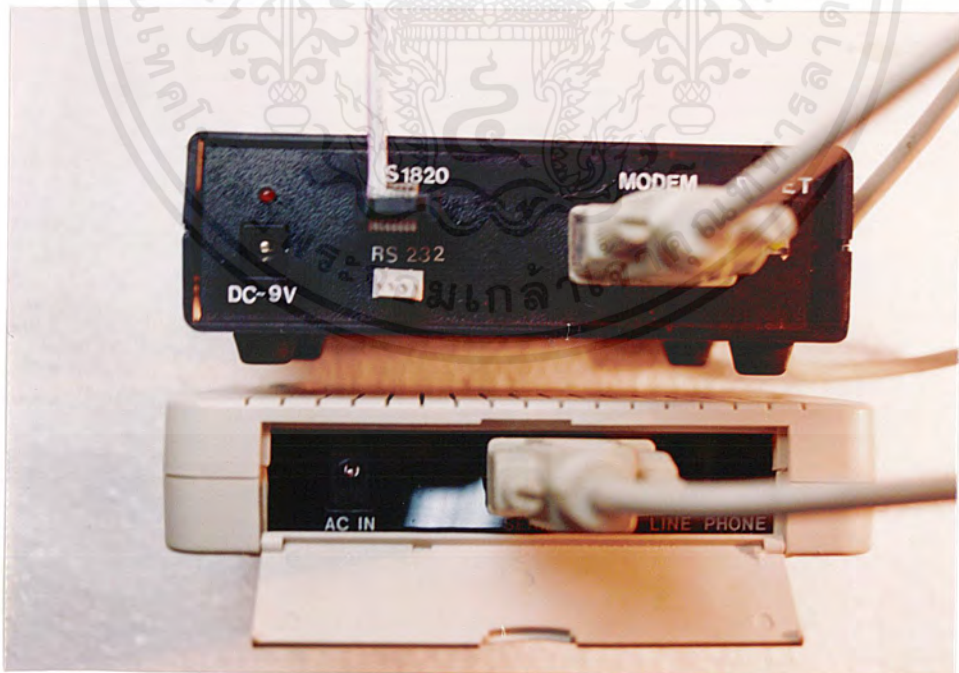
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

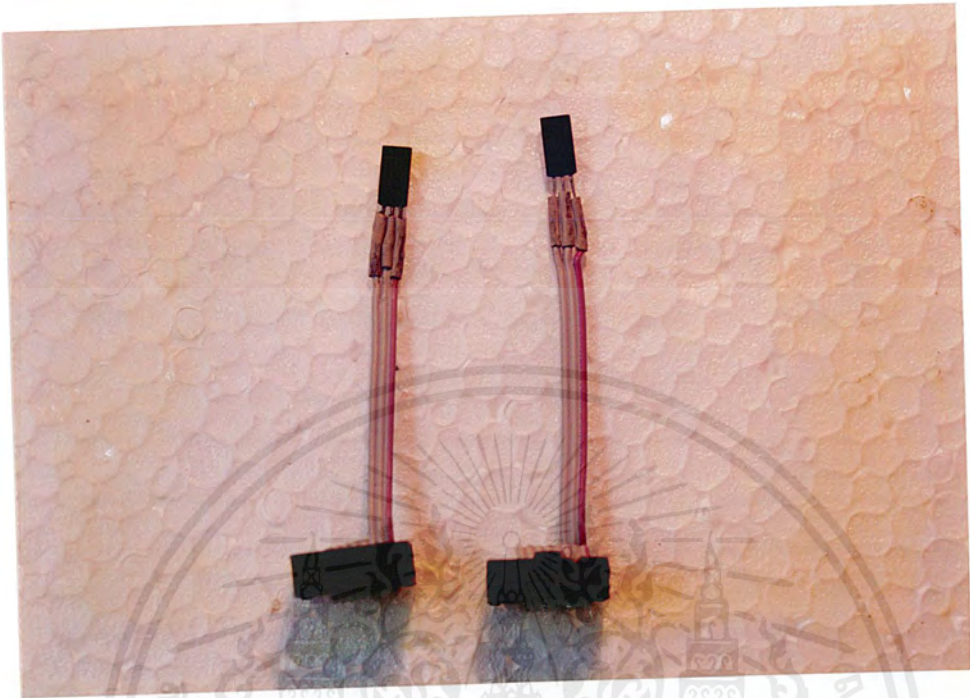


รูปที่ ก.1 เครื่องต้นแบบโครงข่ายรวบรวมข้อมูลระยะไกล



รูปที่ ก.2 การเชื่อมต่อเครื่องต้นแบบกับโมเด็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



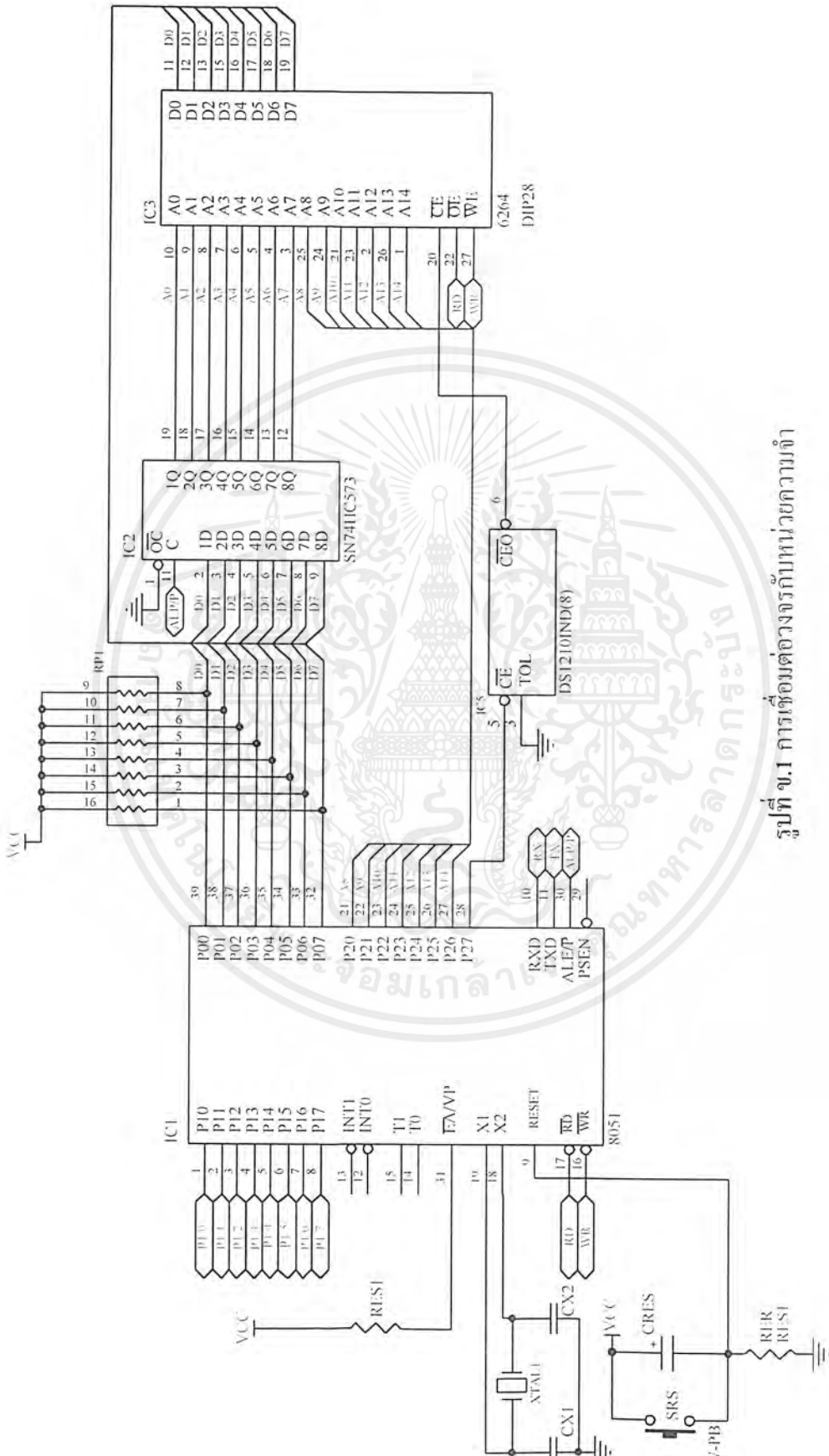
รูปที่ ก.3 อุปกรณ์ตรวจจับ DS 1820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



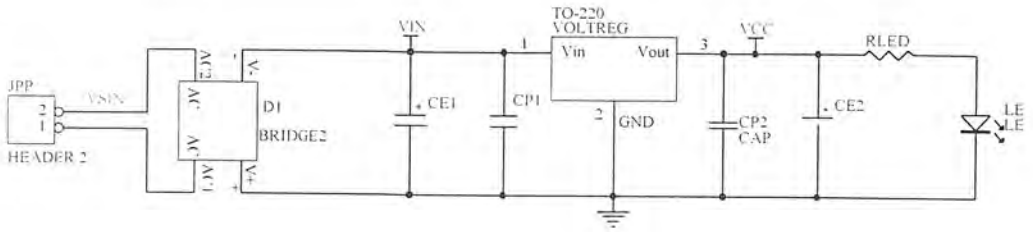
ภาคผนวก ข
วงจร และแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

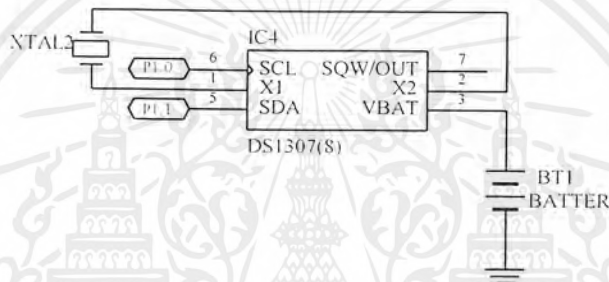


รูปที่ ข.1 การเชื่อมต่อวงจรกับหน่วยความจำ

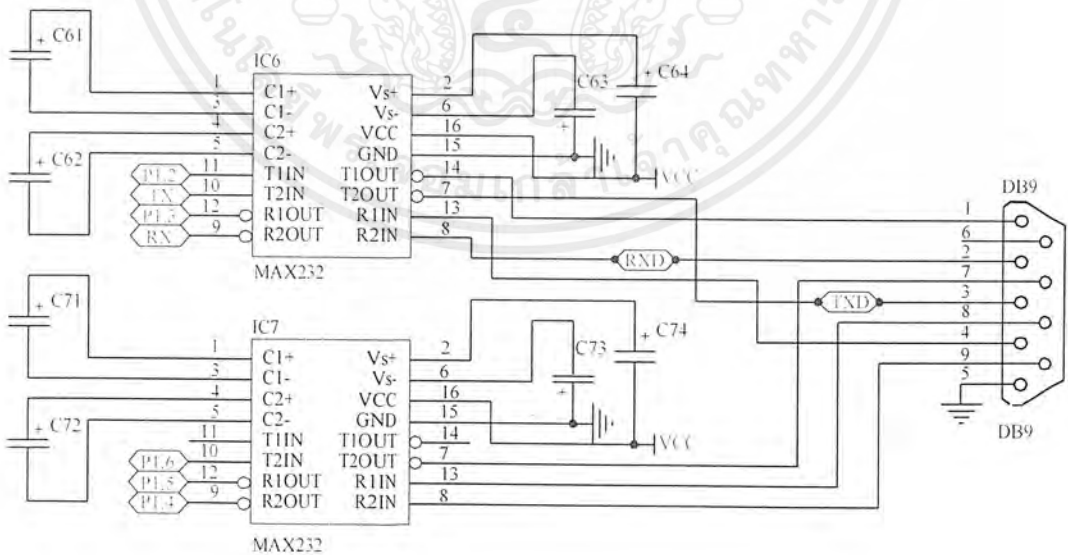
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขหรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.2 วงจรภาคจ่ายไฟ

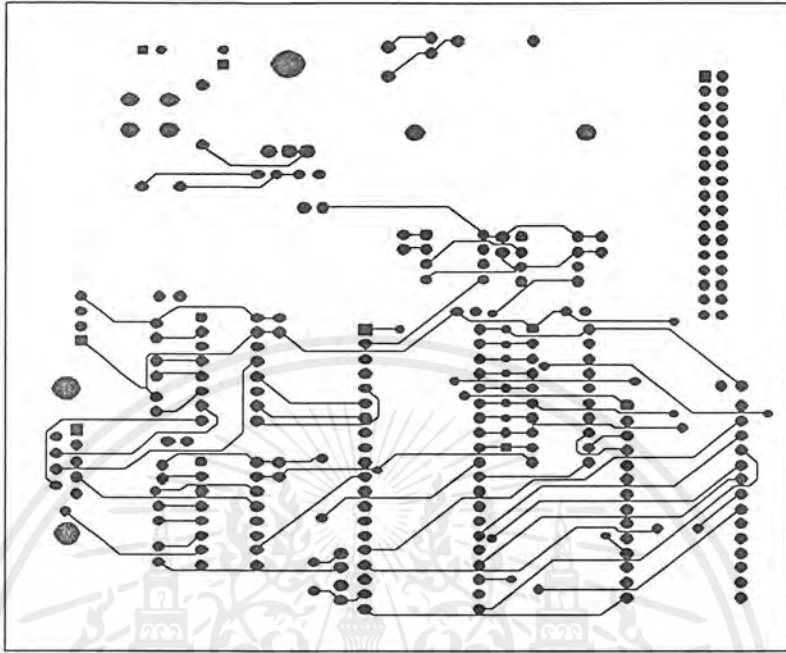


รูปที่ ข.3 วงจรเรียลไทม์

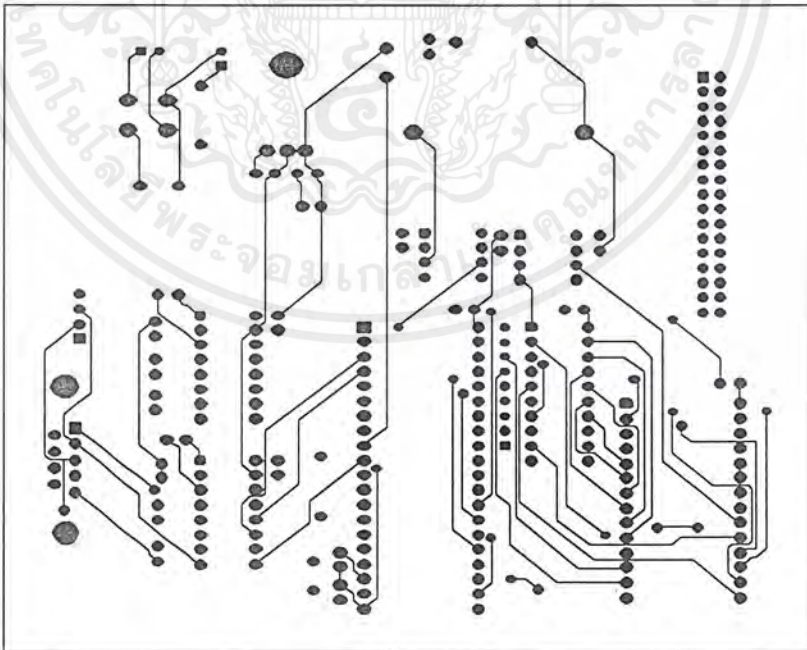


รูปที่ ข.4 วงจรสื่อสารข้อมูล RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

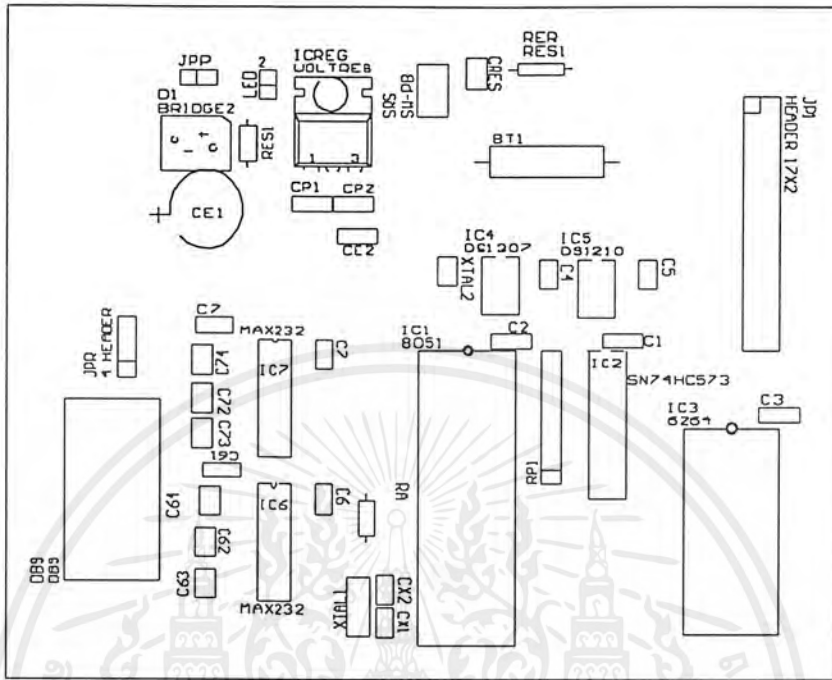


รูปที่ ข.5 แผ่นวงจรพิมพ์ด้านบน



รูปที่ ข.6 แผ่นวงจรพิมพ์ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.7 การวางอุปกรณ์ลงบนแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Option Explicit
Dim page As Integer
Dim rest_ten As Integer
Dim more_ten As Integer
Dim temperature As Integer
Dim binary_data As String
Dim binary_rrc As String
Dim decimal_rrc As Byte
Dim nipple_bit As String
Dim lsb, msb As Integer
Dim temp_keep(1 To 10) As Variant
Dim carry_flag As String
Dim date_time As String
Dim time_in As String
Dim called As Boolean
Dim f2_b As String
Dim f2_p As String
Dim f2_s As String
Dim f2_d As String
Dim f2_c As String
Dim FSO As New FileSystemObject
Dim PROJ_FOLDER As Folder
Dim f_number As Integer
Dim com_def, baud_def, parity_def, data_def, stop_def As String
Dim m As String
Dim send_day As String, send_d As String, send_m As String, send_y As String, send_h As
String, send_mi As String, send_s As String
Dim date_keep As Variant, hour_keep As Variant, minute_keep As Variant, sec_keep As Variant
Dim month_now As String
Dim year_now As Integer

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim buff_send As Variant
Dim cr_return As String
Dim numuric As String
Dim command_stat As Integer
Dim login As Integer
Dim MasterCall As Boolean
Dim code_id As String

Private Sub btn_GraphLeft_Click()
    Dim i As Integer
    Dim j As Integer
    If rest_ten <> 0 And page > more_ten Then
        cht_graph.RowCount = 10
        j = 10 + rest_ten - 1
    Else
        cht_graph.RowCount = 10
        j = 19
    End If
    data.Recordset.MovePrevious
    i = 0
    btn_GraphLeft.Enabled = False

    Do While i <> j
        If data.Recordset.Fields("date_input") = cmb_GraphDate.Text Then
            i = i + 1
            data.Recordset.MovePrevious
        End If
    Loop
    If data.Recordset.BOF Then
        data.Refresh
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data.Recordset.MoveFirst
End If
i = 0
Do While i < 11
    For i = 1 To 10

        cht_graph.Row = i
        Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
            data.Recordset.MoveNext
        Loop
        cht_graph.data = data.Recordset.Fields("temp_input")
        cht_graph.RowLabel = data.Recordset.Fields("time_input")
        data.Recordset.MoveNext
        If data.Recordset.EOF = True Then
            Exit Do
        End If
    Next i
    Loop
    btn_GraphRight.Enabled = True
    btn_GraphLeft.Enabled = True
    page = page - 1
    If page = 1 Then
        btn_GraphLeft.Enabled = False
    End If
    If rest_ten <> 0 Then
        lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & " หน้า"
    Else
        lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten & " หน้า"
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub btn_GraphRight_Click()
    Dim i As Integer
    Dim i_rest As Integer
    If page = more_ten Then
        i_rest = rest_ten
    Else
        i_rest = 10
    End If
    Do While data.Recordset.EOF <> True And i < i_rest
        cht_graph.RowCount = i_rest
        For i = 1 To i_rest
            cht_graph.Row = i
            Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
                data.Recordset.MoveNext
            Loop
            cht_graph.data = data.Recordset.Fields("temp_input")
            cht_graph.RowLabel = data.Recordset.Fields("time_input")
            data.Recordset.MoveNext
            If data.Recordset.EOF = True Then
                Exit Do
            End If
        Next i
    Loop
    page = page + 1
    If page = more_ten And rest_ten = 0 Then
        btn_GraphRight.Enabled = False
        btn_GraphLeft.Enabled = True
    ElseIf page < more_ten And rest_ten <> 0 Then
        btn_GraphLeft.Enabled = True
        btn_GraphRight.Enabled = True
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf page < more_ten And rest_ten = 0 Then
    btn_GraphLeft.Enabled = True
    btn_GraphRight.Enabled = True
ElseIf page > more_ten And rest_ten <> 0 Then
    btn_GraphRight.Enabled = False
    btn_GraphLeft.Enabled = True
End If
If rest_ten <> 0 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & " หน้า"
Else
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten & " หน้า"
End If
End Sub

Private Sub btn_HangUp_Click()
    Dim HangCode As String
    called = False
    If MSComm1.PortOpen Then
        MSComm1.DTREnable = False
        Call delay_
        Call delay_
        MSComm1.DTREnable = True
    End If
    login = 0
    Frame1.Enabled = True
End Sub

Private Sub btn_SendData_Click()
    Dim DataToModem As Variant
    If MSComm1.CDHolding Then
        DataToModem = send_text.Text & vbCrLf
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.OutBufferCount = 0
MSComm1.Output = DataToModem
send_text.Text = ""
send_text.SetFocus
Else
lbl_ReceiveData.Caption = "ไม่มีสัญญาณ Carrier"
send_text.Text = ""
send_text.SetFocus
End If
End Sub
Private Sub cancel_comm_Click()
'Frame2.Enabled = False
Frame1.Enabled = True
Option1.Value = False
Option2.Value = False
'cancel_comm.Enabled = False
'ok_comm.Enabled = False
btn_calling.Enabled = False
send_text.Text = ""
End Sub

Private Sub clear_comm_Click()
Dim i As Integer
lbl_StatusGen.Caption = "พร้อมใช้งาน"
Shape3.BackColor = 255
send_text.Text = ""
lbl_ReceiveData.Caption = ""
Option1.Value = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Option2.Value = False
'btn_calling.Enabled = False
'Frame1.Enabled = True
'Frame2.Enabled = False
'cancle_comm.Enabled = False
'ok_comm.Enabled = False
cht_graph.ColumnCount = 1
cht_graph.RowCount = 10
For i = 1 To 10
    cht_graph.Row = i
    cht_graph.data = 0
    cht_graph.RowLabel = "0:0"
Next i
cht_graph.Title = "อุณหภูมิ (เซลเซียส)"
btn_GraphRight.Enabled = False
btn_GraphLeft.Enabled = False
cmb_GraphDate.ListIndex = 0
cmb_CodeId.ListIndex = 0
End Sub
Private Sub close_comm_Click()
    btn_calling.Caption = "เรียก"
    Frame1.Enabled = False
    Option1.Value = False
    Option2.Value = False
    send_text.Text = ""
    prop_comm.Enabled = True
    open_comm.Enabled = True
    If MSComm1.PortOpen = False Then
        lbl_StatusGen.Caption = "ไม่ได้เปิดพอร์ตอนุกรม"
        Shape1.BackColor = QBColor(10)
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Shape2.BackColor = 255
btn_calling.Enabled = False
Shape3.BackColor = 255
status_label.Caption = "ไม่มีการใช้งานพอร์ตอนุกรม"
lbl_ReceiveData.Caption = ""
Else
MSComm1.PortOpen = False
lbl_StatusGen.Caption = "ไม่ได้เปิดพอร์ตอนุกรม"
Shape1.BackColor = QBColor(10)
Shape2.BackColor = 255
btn_calling.Enabled = False
Shape3.BackColor = 255
status_label.Caption = "พอร์ตอนุกรมไม่ได้ใช้งาน"
lbl_ReceiveData.Caption = ""
End If
End Sub
Private Sub btn_graph_Click()
Dim today As String
Dim i As Integer
Dim p_slash As Integer
Dim row_count As Integer
Dim year_conv As Integer
Dim slash As String
Dim year_BD As Integer
Dim slash_t As String
Dim g_date As String
Dim g_month As String

today = Day(Date) & "/" & Month(Date) & "/" & Year(Date)
btn_GraphRight.Enabled = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

btn_GraphLeft.Enabled = False
i = 0
page = 0
more_ten = 0
rest_ten = 0

If cmb_Codeld.Text <> "ระบุวันที่" And cmb_GraphDate.Text <> "วัน/เดือน/ปี" Then
    On Error GoTo error_graph
    year_BD = 2543
    year_conv = Right(cmb_GraphDate.Text, 4)
    year_conv = year_conv - 2000
    year_conv = year_BD + year_conv
    p_slash = 1
    slash_t = ""
    slash = ""
    Do While slash <> "/"
        slash = Mid$(cmb_GraphDate.Text, p_slash, 1)
        If slash <> "/" Then
            slash_t = slash_t + slash
        End If
        p_slash = p_slash + 1
    Loop
    g_date = slash_t
    slash_t = ""
    slash = ""
    Do While slash <> "/"
        slash = Mid$(cmb_GraphDate.Text, p_slash, 1)
        If slash <> "/" Then
            slash_t = slash_t + slash
        End If
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    p_slash = p_slash + 1
Loop
g_month = slash_t
Select Case (g_month)
    Case "1" Or "01"
        g_month = "มกราคม"
    Case "2" Or "02"
        g_month = "กุมภาพันธ์"
    Case "3" Or "03"
        g_month = "มีนาคม"
    Case "4" Or "04"
        g_month = "เมษายน"
    Case "5" Or "05"
        g_month = "พฤษภาคม"
    Case "6" Or "06"
        g_month = "มิถุนายน"
    Case "7" Or "07"
        g_month = "กรกฎาคม"
    Case "8" Or "08"
        g_month = "สิงหาคม"
    Case "9" Or "09"
        g_month = "กันยายน"
    Case "10" Or "10"
        g_month = "ตุลาคม"
    Case "11" Or "11"
        g_month = "พฤศจิกายน"
    Case "12" Or "12"
        g_month = "ธันวาคม"
    Case "01"
        g_month = "มกราคม"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case "02"
    g_month = "กุมภาพันธ์"
Case "03"
    g_month = "มีนาคม"
Case "04"
    g_month = "เมษายน"
Case "05"
    g_month = "พฤษภาคม"
Case "06"
    g_month = "มิถุนายน"
Case "07"
    g_month = "กรกฎาคม"
Case "08"
    g_month = "สิงหาคม"
Case "09"
    g_month = "กันยายน"
End Select
cht_graph.Title = "อุณหภูมิจากเครื่องวัดอุณหภูมิ (เซลเซียส) ณ วันที่ " & g_date & " " & g_month & " " &
year_conv & " จากเครื่องวัดอุณหภูมิหมายเลข " & cmb_CodId
If cmb_CodId.Text = "0000" Then
    data.RecordSource = "t_0000"
    data.Refresh
If data.Recordset.EOF <> True Then
    data.Recordset.MoveFirst
    btn_GraphLeft.Enabled = False
Else
    MsgBox ("ไม่มีข้อมูลในวันนี้")
End If
data.Refresh
data.Recordset.MoveFirst

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

row_count = 0

Do While data.Recordset.EOF <> True
    If data.Recordset.Fields("date_input") = cmb_GraphDate.Text Then
        row_count = row_count + 1
    End If
    data.Recordset.MoveNext

Loop

If row_count < 10 Then
    cht_graph.RowCount = row_count
Else
    cht_graph.RowCount = 10
    more_ten = row_count \ 10
    rest_ten = row_count Mod 10
End If

data.Refresh
data.Recordset.MoveFirst

Select Case (row_count)
    Case Is > 10
        Do While data.Recordset.EOF <> True And i < 10
            For i = 1 To 10
                cht_graph.Row = i
                Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
                    data.Recordset.MoveNext
                Loop
                cht_graph.data = data.Recordset.Fields("temp_input")
                cht_graph.RowLabel = data.Recordset.Fields("time_input")
                data.Recordset.MoveNext
            If data.Recordset.EOF = True Or i = 10 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Exit Do
End If
Next i
Loop
btn_GraphRight.Enabled = True
page = page + 1
Case Is <= 10
Do While data.Recordset.EOF <> True And i < row_count
For i = 1 To row_count
cht_graph.Row = i
Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
data.Recordset.MoveNext
Loop
cht_graph.data = data.Recordset.Fields("temp_input")
cht_graph.RowLabel = data.Recordset.Fields("time_input")
data.Recordset.MoveNext
If data.Recordset.EOF = True Or i = row_count Then
page = page + 1
Exit Do
End If
Next i
Loop
End Select
If rest_ten <> 0 And more_ten > 10 Then
lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
หน้า"
ElseIf rest_ten <> 0 And more_ten < 10 Then
lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
หน้า"
ElseIf rest_ten = 0 And more_ten > 10 Then

```

```

lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten & " หน้า"
ElseIf rest_ten = 0 And more_ten < 10 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
    หน้า"
End If
Else
If cmb_CodeId.Text = "0001" Then
    data.RecordSource = "t_0001"
    data.Refresh
    If data.Recordset.EOF <> True Then
        data.Recordset.MoveFirst
        btn_GraphLeft.Enabled = False
    Else
        MsgBox ("ไม่มีข้อมูลในวันนี้")
    End If
    data.Refresh
    data.Recordset.MoveFirst
    row_count = 0
    Do While data.Recordset.EOF <> True
        If data.Recordset.Fields("date_input") = cmb_GraphDate.Text Then
            row_count = row_count + 1
        End If
        data.Recordset.MoveNext
    Loop
    If row_count < 10 Then
        cht_graph.RowCount = row_count
    Else
        cht_graph.RowCount = 10
        more_ten = row_count \ 10
        rest_ten = row_count Mod 10

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End If

data.Refresh

data.Recordset.MoveFirst

Select Case (row_count)
Case Is > 10
    Do While data.Recordset.EOF <> True And i < 10
        For i = 1 To 10
            cht_graph.Row = i
            Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
                data.Recordset.MoveNext
            Loop
            cht_graph.data = data.Recordset.Fields("temp_input")
            cht_graph.RowLabel = data.Recordset.Fields("time_input")
            data.Recordset.MoveNext
            If data.Recordset.EOF = True Then
                Exit Do
            End If
        Next i
    Loop
    btn_GraphRight.Enabled = True
    page = page + 1

Case Is <= 10
    Do While data.Recordset.EOF <> True And i < row_count
        For i = 1 To row_count
            cht_graph.Row = i
            Do While data.Recordset.Fields("date_input") <> cmb_GraphDate.Text
                data.Recordset.MoveNext

```

```

    Loop
    cht_graph.data = data.Recordset.Fields("temp_input")
    cht_graph.RowLabel = data.Recordset.Fields("time_input")
    data.Recordset.MoveNext
    If data.Recordset.EOF = True Or i = row_count Then
        page = page + 1
        Exit Do
    End If
    Next i
Loop
End Select
End If
If rest_ten <> 0 And more_ten > 10 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
    หน้า"
Elseif rest_ten <> 0 And more_ten < 10 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
    หน้า"
Elseif rest_ten = 0 And more_ten > 10 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten & " หน้า"
Elseif rest_ten = 0 And more_ten < 10 Then
    lbl_StatusGen.Caption = "หน้าที่ " & page & " จากทั้งหมด " & more_ten + 1 & "
    หน้า"
End If
End If
Else
    If cmb_Codeld.Text = "รหัสเครื่อง" And cmb_GraphDate.Text = "วัน เดือน/ปี" Then
        MsgBox ("กรุณาเลือก รหัสเครื่อง และ วัน เดือน/ปี ที่ต้องการให้แสดงผล")
    End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    Select Case (cmb_Codeld.Text)
        Case "รหัสเครื่อง"
            MsgBox ("กรุณาเลือกรหัสเครื่องที่ต้องการให้แสดงกราฟ")
        End Select
    Select Case (cmb_GraphDate.Text)
        Case "วัน/เดือน/ปี"
            MsgBox ("กรุณาเลือกวัน/เดือน/ปีที่ต้องการให้แสดงกราฟ")
        End Select
    End If
End If
Exit Sub
error_graph: MsgBox ("ไม่สามารถอ่านฐานข้อมูลได้, กรุณาตรวจสอบที่ตั้งของฐานข้อมูล")
End Sub
Private Sub cmb_Codeld_Click()
    Dim tmp_selectdate As String
    Dim ListIndex_Point As Integer
    Dim CheckEqual As Integer
    Dim duplicate_item As Boolean

    btn_GraphLeft.Enabled = False
    btn_GraphRight.Enabled = False

    If cmb_GraphDate.ListCount > 1 Then
        ListIndex_Point = cmb_GraphDate.ListCount - 1
        cmb_GraphDate.ListIndex = ListIndex_Point
        Do While cmb_GraphDate.ListCount <> 1
            cmb_GraphDate.RemoveItem (cmb_GraphDate.ListCount - 1)

```

```

Loop
    cmb_GraphDate.ListIndex = 0
End If

If cmb_CodeId.Text <> "รหัสเครื่อง" Then

    cmb_GraphDate.Visible = False
    tmp_selectdate = ""
    Select Case (cmb_CodeId.Text)
        Case "0000"
            data.RecordSource = "t_0000"
            data.Refresh
            If data.Recordset.EOF Then
                lbl_StatusGen.Caption = "ไม่มีข้อมูลในฐานข้อมูล"
            Else
                lbl_StatusGen.Caption = "พบข้อมูลในฐานข้อมูล"
                duplicate_item = False
                data.Recordset.MoveFirst
            Do While data.Recordset.EOF <> True
                ListIndex_Point = 0
                tmp_selectdate = data.Recordset.Fields("date_input")
                Do While ListIndex_Point <> cmb_GraphDate.ListCount And duplicate_item <>
True
                    cmb_GraphDate.ListIndex = ListIndex_Point
                    If tmp_selectdate = cmb_GraphDate.Text Then
                        duplicate_item = True
                    End If
                    ListIndex_Point = ListIndex_Point + 1
                Loop

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Select Case duplicate_item
    Case False

        cmb_GraphDate.AddItem (tmp_selectdate)
    End Select

    duplicate_item = False
    data.Recordset.MoveNext

Loop

cmb_GraphDate.ListIndex = 0
End If
Case "0001"
    tmp_selectdate = ""
    data.RecordSource = "t_0001"
    data.Refresh
    If data.Recordset.EOF Then
        lbl_StatusGen.Caption = "ไม่มีข้อมูลในฐานข้อมูล"
    Else
        lbl_StatusGen.Caption = "พบข้อมูลในฐานข้อมูล"
        duplicate_item = False
        data.Recordset.MoveFirst
        Do While data.Recordset.EOF <> True

            ListIndex_Point = 0
            tmp_selectdate = data.Recordset.Fields("date_input")
            Do While ListIndex_Point <> cmb_GraphDate.ListCount And duplicate_item <>
True

                cmb_GraphDate.ListIndex = ListIndex_Point
                If tmp_selectdate = cmb_GraphDate.Text Then
                    duplicate_item = True
                End If
                ListIndex_Point = ListIndex_Point + 1
            End Do
        End Do
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Loop
Select Case duplicate_item
    Case False
        cmb_GraphDate.AddItem (tmp_selectdate)
End Select
duplicate_item = False
data.Recordset.MoveNext

Loop
cmb_GraphDate.ListIndex = 0
End If
End Select

cmb_GraphDate.Visible = True
cmb_GraphDate.Enabled = True
Else
If cmb_GraphDate.ListCount > 1 Then
    ListIndex_Point = cmb_GraphDate.ListCount - 1
    cmb_GraphDate.ListIndex = ListIndex_Point
Do While cmb_GraphDate.ListCount <> 1
    cmb_GraphDate.RemoveItem (cmb_GraphDate.ListCount - 1)
Loop
cmb_GraphDate.ListIndex = 0
End If

If MSComm1.PortOpen = True Then
    lbl_StatusGen.Caption = "พอร์ตอนุกรมถูกใช้งาน"
Else
    lbl_StatusGen.Caption = "พร้อมใช้งาน"
End If

cmb_GraphDate.Enabled = False

End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmb_GraphDate_Click()
```

```
    btn_GraphLeft.Enabled = False
```

```
    btn_GraphRight.Enabled = False
```

```
End Sub
```

```
Private Sub cmb_Graph_Change()
```

```
End Sub
```

```
Private Sub exit_comm_Click()
```

```
    End
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Dim i As Integer
```

```
    Form1.Icon = LoadPicture("c:\visual\loger.ico")
```

```
    called = False
```

```
    Option2.Enabled = False
```

```
    cmb_GraphDate.Enabled = False
```

```
    btn_GraphLeft.Enabled = False
```

```
    btn_GraphRight.Enabled = False
```

```
    cht_graph.ColumnCount = 1
```

```
    cht_graph.RowCount = 10
```

```
    cht_graph.Enabled = False
```

```
    For i = 1 To 10
```

```
        cht_graph.Row = i
```

```
        cht_graph.data = 0
```

```
        cht_graph.RowLabel = "0:0"
```

```
    Next i
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

If lbl_NumDial.Caption = "" Then
    btn_calling.Enabled = False
End If
MasterCall = False
Date = FormatDateTime(Date, vbShortDate)
Call dateformat
btn_HangUp.Enabled = False
command_stat = 0
cr_return = Chr$(&HD)
Form1.Caption = "โครงข่ายรวบรวมข้อมูลระยะไกล"
Frame1.Caption = "เลือกคำสั่ง"
'Frame2.Caption = "การตั้งค่าเวลาสำหรับเครื่องดูข่าย"
'Frame2.Enabled = False
btn_calling.Caption = "เรียก"
'ok_comm.Enabled = False
'cancle_comm.Enabled = False
login = 0
lbl_NumDial.Caption = Form4.txt_DialNum.Text
If Form4.txt_DialNum.Text <> "" Then
    btn_calling.Enabled = True
Else
    btn_calling.Enabled = False
End If

Call check_db
'For i = 1 To 7 Step 1
'    Select Case i
'        Case 2
'            mcs_day.AddItem "วันจันทร์"
'        Case 3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

' mcs_day.AddItem "วันอังคาร"
' Case 4
' mcs_day.AddItem "วันพุธ"
' Case 5
' mcs_day.AddItem "วันพฤหัสบดี"
' Case 6
' mcs_day.AddItem "วันศุกร์"
' Case 7
' mcs_day.AddItem "วันเสาร์"
' Case 8
' mcs_day.AddItem "วันอาทิตย์"
' End Select
' Next
' For i = 1 To 31 Step 1
' mcs_d.AddItem CStr(i)
' Next
' For i = 1 To 12 Step 1
' mcs_m.AddItem CStr(i)
' Next
' For i = 2543 To 2550 Step 1
' mcs_y.AddItem CStr(i)
' Next
' For i = 0 To 23 Step 1
' mcs_h.AddItem CStr(i)
' Next
' For i = 0 To 59 Step 1
' mcs_mi.AddItem CStr(i)
' mcs_s.AddItem CStr(i)
' Next

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

On Error GoTo check_cont

'mcs_day.ListIndex = Weekday(Date) - 1

'mcs_d.Text = CStr(Day(Date))

'mcs_m.Text = CStr(Month(Date))

'mcs_y.Text = CStr(Year(Date))

'mcs_h.Text = CStr(Hour(time))

'mcs_mi.Text = CStr(Minute(time))

'mcs_s.Text = CStr(Second(time))

lbl_datetime.Caption = Day(Date) & " " & month_now & " " & year_now & " เวลา " &
Format(Now, "hh:mm:ss")

Frame1.Enabled = False

Call default_sr

If MSComm1.PortOpen = False Then

    lbl_StatusGen.Caption = "พอร์ตอนุกรมปิด"

    Shape1.BackColor = QBColor(10)

    If MSComm1.CDHolding Then

        btn_calling.Enabled = False

    End If

    Shape3.BackColor = 255

    status_label.Caption = "ไม่ได้มีการใช้พอร์ตอนุกรม"

Else

    lbl_StatusGen.Caption = "พอร์ตอนุกรมถูกเปิดอยู่แล้ว"

    Shape1.BackColor = 255

    Shape2.BackColor = QBColor(10)

    If MSComm1.CDHolding Then

        btn_calling.Enabled = False

    Else

        btn_calling.Enabled = True

        Frame1.Enabled = True

    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Shape3.BackColor = 255
status_label.Caption = "พอร์ตอนุกรม COM : " & MSComm1.CommPort & " ถูกใช้งาน"
End If
cmb_CodeId.AddItem "รหัสเครื่อง"
cmb_CodeId.AddItem "0000"
cmb_CodeId.AddItem "0001"
cmb_CodeId.ListIndex = 0
cmb_GraphDate.AddItem "วัน/เดือน/ปี"
cmb_GraphDate.ListIndex = 0
Timer1.Interval = 1000
Timer1.Enabled = True

Exit Sub
check_cont: MsgBox ("กรุณาตรวจสอบรูปแบบวัน/เดือน/ปี ที่ Control Panel")
'mcs_day.ListIndex = 1
'mcs_d.ListIndex = 1
'mcs_m.ListIndex = 1
'mcs_y.ListIndex = 1
'mcs_h.ListIndex = 1
'mcs_mi.ListIndex = 1
'mcs_s.ListIndex = 1

End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub menu_suggest_Click()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End Sub

```

Private Sub MSComm1_OnComm()
    Dim buff As Variant
    Dim waiting1 As Integer
    Dim waiting2 As Integer
    cr_return = Chr$(&HD)
    MSComm1.InputLen = 0
    Select Case MSComm1.CommEvent
        Case comEvReceive
            send_text.Text = MSComm1.Input
        Case comEvCD
            btn_HangUp.Enabled = True
            btn_calling.Enabled = False
            MSComm1.InBufferCount = 0
            If MSComm1.CDHolding Then
                btn_graph.Enabled = False
                send_text.Text = "มีสัญญาณ Carrier"
                btn_calling.Enabled = False
                Frame1.Enabled = False
                btn_HangUp.Enabled = True
                login = 1
                For waiting1 = 0 To 6000 Step 1
                For waiting2 = 0 To 8000 Step 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Next waiting2
    Next waiting1
    Call ACCEPT
    MSComm1.InBufferCount = 0
    cht_graph.Title.Text = " ณ วันที่ " & date_keep & month_now & year_now & " จาก
เครื่อง #" & code_id
Else
    For waiting1 = 0 To 100 Step 1
    For waiting2 = 0 To 200 Step 1
        Next waiting2
        Next waiting1
    Call ACCEPT
    MSComm1.InBufferCount = 0
    send_text.Text = " ไม่มีสัญญาณ Carrier"
    btn_graph.Enabled = True
    btn_calling.Enabled = True
    Frame1.Enabled = True
    btn_HangUp.Enabled = False
    send_text.Text = ""
    lbl_NumDial.Caption = ""
End If
' If MasterCall = True Then
'   MSComm1.Output = "T"
' Else
'   MSComm1.Output = "T"
'   Call delay_
' End If
Case comEvCTS
    send_text.Text = "Clear to send"
Case comEvDSR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    send_text.Text = "Data setting ready"
Case comEvSend
    send_text.Text = "Send data"
Case comEventCDTO
    send_text.Text = "ออกจากการทำงาน"
    MSComm1.Output = "+++ath0" & vbCrLf
    login = 0
    Call delay_
Case comInputModeText
    send_text.Text = "text in"
Case comRTSXOnXOff
    send_text.Text = "Hand checking"
End Select
End Sub

Private Sub open_comm_Click()
    Dim buff As Variant
    Dim s_buff As Variant
    Dim numeric As String

    On Error GoTo error_open
    prop_comm.Enabled = False
    MSComm1.InBufferCount = 0

    'mcs_day.ListIndex = Weekday(Date) - 1
    'mcs_d.Text = CStr(Day(Date))
    'mcs_m.Text = CStr(Month(Date))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'mcs_y.Text = CStr(Year(Date))
'mcs_h.Text = CStr(Hour(time))
'mcs_mi.Text = CStr(Minute(time))
'mcs_s.Text = CStr(Second(time))

If MSComm1.PortOpen = False Then
    MSComm1.PortOpen = True
    lbl_StatusGen.Caption = "เลือกใช้พอร์ตอนุกรม COM : " & MSComm1.CommPort
    status_label.Caption = "พอร์ตอนุกรม COM : " & MSComm1.CommPort & " ถูกใช้งาน"
    lbl_ReceiveData.Caption = ""
    Shape2.BackColor = QBColor(10)
    Shape1.BackColor = 255

Else
    lbl_StatusGen.Caption = "พอร์ตอนุกรมถูกเปิดใช้งานอยู่แล้ว"
    status_label.Caption = "พอร์ตอนุกรม COM : " & MSComm1.CommPort & " ถูกใช้งาน"
    lbl_ReceiveData.Caption = ""
End If

open_comm.Enabled = False
s_buff = 2
cr_return = ChrS(&HD)
buff = "atq1" & vbCrLf
Call delay_
'Call delay_
MSComm1.Output = buff
buff = "ats0=" & s_buff & vbCrLf
'Call delay_
Call delay_
MSComm1.Output = buff

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'Call delay_
's_buff = 95
buff = "ats11=" & s_buff & vbCrLf
MSComm1.Output = buff
Call delay_
s_buff = 14
buff = "ats10=" & s_buff & vbCrLf
MSComm1.Output = buff
Call delay_
MSComm1.InputLen = 0
MSComm1.InBufferCount = 0
'lbl_ReceiveData.Caption = MSComm1.Input
'MSComm1.Handshaking = comRTSXOnXOff
MSComm1.EOFEnable = True
'lbl_ReceiveData.Caption = ""
Frame1.Enabled = True
btn_calling.Enabled = False

Exit Sub

error_open: MsgBox ("ไม่สามารถใช้งานพอร์ตอนุกรม COM : " & MSComm1.CommPort)
    Frame1.Enabled = False
    send_text.Text = ""
    lbl_StatusGen.Caption = ""
    prop_comm.Enabled = True

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Option1_Click()
    'Frame2.Enabled = False
    'cancle_comm.Enabled = False
    'ok_comm.Enabled = False
    send_text.Enabled = True
    Shape3.BackColor = 255
    Form1.Enabled = False
    Form4.Show
    command_stat = 1

```

```
End Sub
```

```

Private Sub Option2_Click()
    'ok_comm.Enabled = True
    'cancle_comm.Enabled = True
    'Frame2.Enabled = True
    Frame1.Enabled = False
    lbl_ReceiveData.Caption = ""
    Shape3.BackColor = 255
    send_text.Text = ""
    command_stat = 2

```

```
End Sub
```

```

Private Sub prop_comm_Click()
    Form1.Enabled = False
    Form2.Show
    Form2.Enabled = True

```

```
End Sub
```

```

Private Sub btn_calling_Click()
    Dim BUFFER As Variant
    Dim cr_return As String

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim dial_number As String
Dim HangUp_no As Integer
Dim temp As Variant

called = True
MasterCall = True
If command_stat = 2 Then
    BUFFER = send_day
    MSComm1.Output = Chr$(BUFFER)
    Call delay_
    MSComm1.OutBufferCount = 0
    BUFFER = send_d
    MSComm1.Output = Chr$(BUFFER)
    Call delay_
    Option2.Value = False
    command_stat = 0
Else
    If command_stat = 1 Then
        If MSComm1.CTSHolding Then
            dial_number = lbl_NumDial.Caption
            cr_return = Chr$(&HD)
            MSComm1.DTREnable = True
            MSComm1.RTSEnable = True
            MSComm1.InBufferCount = 0
            MSComm1.OutBufferCount = 0
            BUFFER = "atdt " & dial_number & vbCrLf
            MSComm1.Output = BUFFER
            btn_calling.Enabled = False
            Shape3.BackColor = QBColor(10)
        Else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MsgBox ("ไม่มีมีการต่อโมเด็ม")
    Option1.Value = False
    btn_calling.Enabled = False
    Frame1.Enabled = True
End If
Else
    If lbl_NumDial.Caption = "" Then
        MsgBox ("ไม่ได้ระบุหมายเลขโทรศัพท์")
        Shape3.BackColor = QBColor(12)
    Else
        Shape3.BackColor = QBColor(12)
    End If
    Option1.Value = False
End If
End If
If MSComm1.CTSHolding = False And command_stat = 1 Then
    Frame1.Enabled = True
Else
    Frame1.Enabled = False
End If

MSComm1.OutBufferCount = 0
'Shape3.BackColor = QBColor(10)
' For i = 0 To 2000 Step 1
'   For j = 0 To 1000 Step 1
'     Next j
'   Next i
' Call ACCEPT(time)
End Sub
Private Sub ACCEPT()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim a As Variant
Dim in_buff As Variant
Dim temp_dat As Variant
Dim i, j, k As Integer
Dim waiting1, waiting2 As Integer

```

```

MSComm1.InputMode = comInputModeText
MSComm1.InputLen = 0
If MSComm1.InBufferCount Then
in_buff = MSComm1.Input
For i = 1 To 10
temp_keep(i) = 0
Next i
On Error GoTo data_fail
a = in_buff
lbl_ReceiveData.Caption = in_buff
j = Len(a)
If Asc(in_buff) <> 0 Then
code_id = Left$(a, 4)
date_keep = Asc(Mid$(a, 6, 1))
date_keep = Hex(date_keep)
hour_keep = Mid$(a, 8, 1)
hour_keep = Asc(hour_keep)
hour_keep = Hex(hour_keep)
minute_keep = Asc(Mid(a, 10, 1))
minute_keep = Hex(minute_keep)

sec_keep = Asc(Mid(a, 12, 1))
sec_keep = Hex(sec_keep)

j = 14

```

```

k = 0
For i = 1 To 10
temp_keep(i) = Asc(Mid(a, j, 1))
j = j + 2
Next i

MSComm1.InBufferCount = 0
For i = 1 To 10
modem2temp (temp_keep(i))
temp_keep(i) = decimal_rc
If carry_flag = "1" Then
temp_keep(i) = temp_keep(i) + 0.5
Else
temp_keep(i) = temp_keep(i) + ".0"
End If
cht_graph.Row = i
cht_graph.data = temp_keep(i)

If code_id = "0000" Then
data.RecordSource = "t_0000"
Else
data.RecordSource = "t_0001"
End If

data.Refresh

data.Recordset.AddNew

data.Recordset.Fields("date_input") = date_keep
data.Recordset.Fields("month_input") = Month(Date)
data.Recordset.Fields("year_input") = Year(Date)
data.Recordset.Fields("time_input") = hour_keep & ":" & minute_keep & ":" &
sec_keep

data.Recordset.Fields("temp_input") = temp_keep(i)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data.Recordset.Update
carry_flag = "0"
cht_graph.RowLabel = hour_keep & ":" & minute_keep & ":" & sec_keep
If sec_keep = 59 Then
    sec_keep = 0
    If minute_keep = 59 Then
        If hour_keep = 23 Then
            hour_keep = 0
        Else
            hour_keep = hour_keep + 1
        End If
        minute_keep = 0
    Else
        minute_keep = minute_keep + 1
    End If
Else
    sec_keep = sec_keep + 1
End If
Next i
k = 0
'temp_keep = LeftS(a, 4)
'date_keep = MidS(a, 6, 8)
'time_keep = RightS(a, 8)
'temp = Val(temp_keep)

'tbl_ReceiveData.Caption = temp_keep & " At " & date_keep & " " & time_keep

' Shape4.BackColor = QBColor(10)
' MSComm1.InBufferCount = 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'Else
  ' MSComm1.InBufferCount = 0
  ' lbl_ReceiveData.Caption = ""
  'lbl_ReceiveData.Caption = "Buffer is empty"
  ' Shape4.BackColor = 255
'End If
Else
  'lbl_ReceiveData.Caption = "Buffer is empty"
  MSComm1.InBufferCount = 0
End If
For waiting1 = 0 To 4000 Step 1
  For waiting2 = 0 To 5000 Step 1
  Next waiting2
  Next waiting1
  MSComm1.InBufferCount = 0
Exit Sub
data_fail:  MsgBox ("Workstation sent bad data")
  lbl_ReceiveData.Caption = ""
  Option1.Value = False
  Option2.Value = False
  send_text.Text = ""
End Sub

'Private Sub time_comm_Click()
  ' Form1.Enabled = False

```

```
' Form3.Show
' Form3.Enabled = True
' Form3.SetFocus
'End Sub
```

```
Private Sub Timer1_Timer()
```

```
Dim modem_return As Variant
```

```
Dim y_num As Integer
```

```
Dim bud_year As Integer
```

```
Select Case (Month(Date))
```

```
Case "1"
```

```
month_now = "มกราคม"
```

```
Case "2"
```

```
month_now = "กุมภาพันธ์"
```

```
Case "3"
```

```
month_now = "มีนาคม"
```

```
Case "4"
```

```
month_now = "เมษายน"
```

```
Case "5"
```

```
month_now = "พฤษภาคม"
```

```
Case "6"
```

```
month_now = "มิถุนายน"
```

```
Case "7"
```

```
month_now = "กรกฎาคม"
```

```
Case "8"
```

```
month_now = "สิงหาคม"
```

```
Case "9"
```

```
month_now = "กันยายน"
```

```
Case "10"
```

```
month_now = "ตุลาคม"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Case "11"
    month_now = "พฤศจิกายน"
Case "12"
    month_now = "ธันวาคม"
End Select
year_now = 2543
y_num = Val(Year(Date))
y_num = y_num - 2000
year_now = year_now + y_num
'If MSComm1.CD Holding And login = 0 Then
' btn_HangUp.Enabled = True
' login = 1
' btn_calling.Enabled = False
' Frame1.Enabled = False
' ElseIf MSComm1.CD Holding = False Then
' btn_HangUp.Enabled = False
' If open_comm.Value = True Then
'     Frame1.Enabled = True
' End If
' If Form4.txt_DialNum.Text <> "" Then
'     btn_calling.Enabled = True
' End If
' End If
lbl_datetime.Caption = Day(Date) & " " & month_now & " " & year_now & " เวลา " &
Format(Now, "hh:mm:ss")
' If MSComm1.PortOpen = True Then
'     If MSComm1.InBufferCount Then
'         modem_return = MSComm1.Input
'         modem_return = LTrim$(RTrim$(modem_return))
'         lbl_ReceiveData.Caption = modem_return

```

```

' If modem_return = "0" Then
'   btn_HangUp.Enabled = True
'   MSComm1.Output = "Prompt to Receive"
'   Call delay_
' End If
' End If
' End If

```

End Sub

Private Sub default_sr()

```

com_def = "2"
baud_def = "9600"
parity_def = "N"
data_def = "8"
stop_def = "1"
m = ","
If FSO.FolderExists("c:\visual") = False Then
    FSO.CreateFolder("c:\visual")
    FSO.CreateFolder("c:\visual\config")
    f_number = FreeFile(2)
    Open ("c:\visual\config\sr_log.ini") For Output As #f_number
    Print #f_number, com_def; m; baud_def; m; parity_def; m; data_def; m; stop_def
    Close (f_number)
Else
    If FSO.FolderExists("c:\visual\config") = False Then
        FSO.CreateFolder("c:\visual\config")
        f_number = FreeFile(2)
        Open ("c:\visual\config\sr_log.ini") For Output As #f_number
        Print #f_number, com_def; m; baud_def; m; parity_def; m; data_def; m; stop_def
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Close (f_number)
Else
    If FSO.FileExists("c:\visual\config\sr_log.ini") = False Then
        f_number = FreeFile(2)
        Open ("c:\visual\config\sr_log.ini") For Output As #f_number
        Print #f_number, com_def; m; baud_def; m; parity_def; m; data_def; m; stop_def
        Close (f_number)
    End If
End If
End If

f_number = FreeFile(2)
Open ("c:\visual\config\sr_log.ini") For Input As #f_number
Input #f_number, f2_c, f2_b, f2_p, f2_d, f2_s
Close (f_number)

Select Case (f2_c)
    Case "1"
        MSComm1.CommPort = 1
    Case "2"
        MSComm1.CommPort = 2
    Case "3"
        MSComm1.CommPort = 3
    Case "4"
        MSComm1.CommPort = 4
    Case Else
        MSComm1.CommPort = 2
End Select

MSComm1.Settings = f2_b & "," & f2_p & "." & f2_d & "," & f2_s

```

```

setting_port.Text = "Com " & MSComm1.CommPort & "," & MSComm1.Settings
End Sub

Private Sub delay_()
    Dim i As Integer, j As Integer

    For i = 1 To 2000 Step 1
        For j = 1 To 1000 Step 2
            Next j
        Next i
    End Sub

Private Sub check_db()
    If FSO.FolderExists("c:\visual") = False Then
        FSO.CreateFolder ("c:\visual")
        FSO.CreateFolder ("c:\visual\storage")
        MsgBox ("Please make sure that file TEMP.MDB is in c:\visual\storage")
    End
Else
    If FSO.FolderExists("c:\visual\storage") = False Then
        FSO.CreateFolder ("c:\visual\storage")
        MsgBox ("Please make sure that file TEMP.MDB is in c:\visual\storage")
    End
Else
    If FSO.FileExists("c:\visual\storage\temp.mdb") = False Then
        MsgBox ("ไม่สามารถตรวจพบฐานข้อมูลที่ใช้เก็บค่าอุณหภูมิชื่อ TEMP.MDB ในไดเรกทอรี c:\visual\storage" & vbCrLf & "ออกจากโปรแกรมเพื่อสร้างฐานข้อมูลในไดเรกทอรีดังกล่าว")
        btn_graph.Enabled = False
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    open_comm.Enabled = False
    close_comm.Enabled = False
    prop_comm.Enabled = False
    clear_comm.Enabled = False
    send_text.Enabled = False
    btn_SendData.Enabled = False

End
End If
End If
End If
End Sub

Private Sub RxCommand()
    Dim RxInbuff As Variant

    MSComm1.InputLen = 0
    RxInbuff = MSComm1.Input
    Call delay_
    If RxInbuff = Chr$(0) Then
        btn_HangUp.Enabled = True
    End If
    MSComm1.Output = Chr$(0)
    Call delay_
    MSComm1.OutBufferCount = 0
End Sub

Private Sub dateformat()
    Dim y_num As Integer
    Dim bud_year As Integer
    Date = FormatDateTime(Date, vbShortDate)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Select Case (Month(Date))
```

```
Case "1"
```

```
month_now = "มกราคม"
```

```
Case "2"
```

```
month_now = "กุมภาพันธ์"
```

```
Case "3"
```

```
month_now = "มีนาคม"
```

```
Case "4"
```

```
month_now = "เมษายน"
```

```
Case "5"
```

```
month_now = "พฤษภาคม"
```

```
Case "6"
```

```
month_now = "มิถุนายน"
```

```
Case "7"
```

```
month_now = "กรกฎาคม"
```

```
Case "8"
```

```
month_now = "สิงหาคม"
```

```
Case "9"
```

```
month_now = "กันยายน"
```

```
Case "10"
```

```
month_now = "ตุลาคม"
```

```
Case "11"
```

```
month_now = "พฤศจิกายน"
```

```
Case "12"
```

```
month_now = "ธันวาคม"
```

```
End Select
```

```
year_now = 2543
```

```
y_num = Val(Year(Date))
```

```
y_num = y_num - 2000
```

```
year_now = year_now + y_num
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

End Sub
Function moderm2temp(real_temp As Variant) As Variant
    Dim i As Integer
    Dim temp_bin As String
    Dim bit As String
    value2hex (real_temp)
    temp_bin = Left(binary_rrc, 4)
    temp_bin = bin2hex(temp_bin)
    temp_bin = temp_bin + bin2hex(Right(binary_rrc, 4))
    nipple_bit = Left(temp_bin, 1)
    Call decimal_comp(nipple_bit)
    decimal_rrc = msb
    nipple_bit = Right(temp_bin, 1)
    Call decimal_comp(nipple_bit)
    decimal_rrc = decimal_rrc + lsb
End Function
Function hex2bin(temp As String) As String
    Dim temp As String
    temp$ = test
    Select Case (temp)
        Case "0"
            hex2bin = "0000"
        Case "1"
            hex2bin = "0001"
        Case "2"
            hex2bin = "0010"
        Case "3"
            hex2bin = "0011"
    
```

Case "4"
hex2bin = "0100"

Case "5"
hex2bin = "0101"

Case "6"
hex2bin = "0110"

Case "7"
hex2bin = "0111"

Case "8"
hex2bin = "1000"

Case "9"
hex2bin = "1001"

Case "A"
hex2bin = "1010"

Case "B"
hex2bin = "1011"

Case "C"
hex2bin = "1100"

Case "D"
hex2bin = "1101"

Case "E"
hex2bin = "1110"

Case "F"
hex2bin = "1111"

End Select

End Function

Function rrc(binary As String) As String

Dim c As String

carry_flag = Right(binary, 1)

```

    rrc = "0" + Left(binary, 1) + Mid(binary, 2, 1) + Mid(binary, 3, 1) + Mid(binary, 4, 1) + Mid
(binary, 5, 1) + Mid(binary, 6, 1) + Mid(binary, 7, 1)

```

```
End Function
```

```
Function bin2hex(bin_value As String) As String
```

```
    Select Case (bin_value)
```

```
        Case "0000"
```

```
            bin2hex = "0"
```

```
        Case "0001"
```

```
            bin2hex = "1"
```

```
        Case "0010"
```

```
            bin2hex = "2"
```

```
        Case "0011"
```

```
            bin2hex = "3"
```

```
        Case "0100"
```

```
            bin2hex = "4"
```

```
        Case "0101"
```

```
            bin2hex = "5"
```

```
        Case "0110"
```

```
            bin2hex = "6"
```

```
        Case "0111"
```

```
            bin2hex = "7"
```

```
        Case "1000"
```

```
            bin2hex = "8"
```

```
        Case "1001"
```

```
            bin2hex = "9"
```

```
        Case "1010"
```

```
            bin2hex = "A"
```

```
        Case "1011"
```

```
            bin2hex = "B"
```

```
        Case "1100"
```

```

    bin2hex = "C"
Case "1101"
    bin2hex = "D"
Case "1110"
    bin2hex = "E"
Case "1111"
    bin2hex = "F"
End Select
End Function
Function value2hex(dec As Integer) As String
    Dim stss As String
    Dim i, j As Integer
    Dim result As String
    Dim bin_rrc As String

    stss = Hex(dec)
    j = Len(stss)
    If j = 1 Then
        result = "0000" + hex2bin(stss)
    Else
        result = ""
        For i = 1 To 2
            result = result + hex2bin(Right(Left(stss, i), 1))
        Next i
    End If
    bin_rrc = rrc(result)
    binary_data = result
    binary_rrc = bin_rrc
End Function
Sub decimal_comp(n_bit As String)

```

Select Case (n_bit)

Case "0"

lsb = 0

msb = 0

Case "1"

lsb = 1

msb = 16

Case "2"

lsb = 2

msb = 32

Case "3"

lsb = 3

msb = 48

Case "4"

lsb = 4

msb = 64

Case "5"

lsb = 5

msb = 80

Case "6"

lsb = 6

msb = 96

Case "7"

lsb = 7

msb = 112

Case "8"

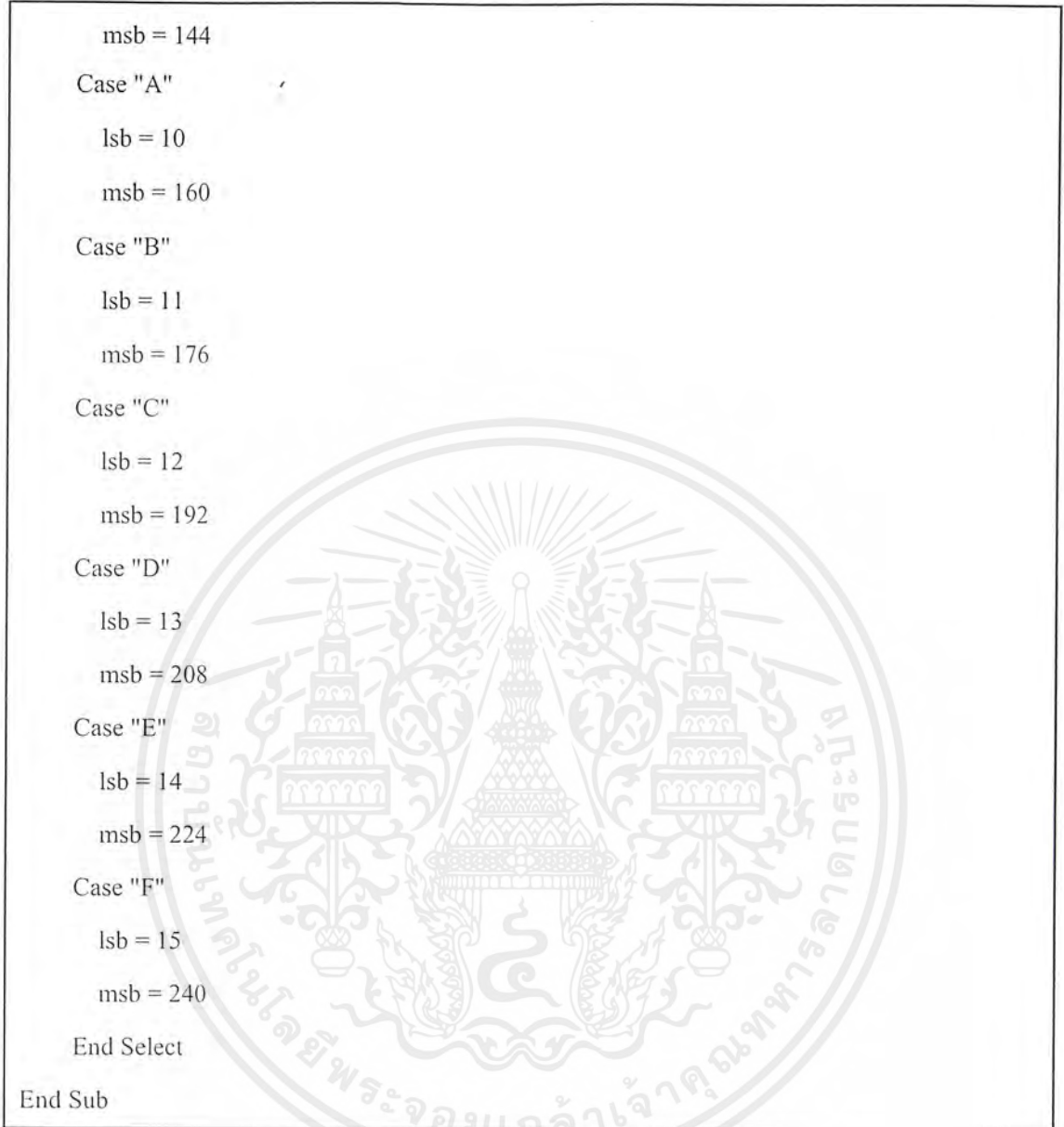
lsb = 8

msb = 128

Case "9"

lsb = 9





โปรแกรมที่ ค.1 โปรแกรมการทำงานหลัก

```
Private Sub Form_Load()
```

```
    comdef = "1"
```

```
    bauddef = "9600"
```

```
    paritydef = "N"
```

```
    datadef = "8"
```

```
    stopdef = "1"
```

```
    c = "."
```

```
    If FSO2.FolderExists("c:\visual") = False Then
```

```
        FSO2.CreateFolder ("c:\visual")
```

```
        FSO2.CreateFolder ("c:\visual\config")
```

```
        f2_number = FreeFile(2)
```

```
        Open ("c:\visual\config\sr_log.ini") For Output As #f2_number
```

```
        Print #f2_number, comdef; c; bauddef; c; paritydef; c; datadef; c; stopdef
```

```
        Close (f2_number)
```

```
    Else
```

```
        If FSO2.FolderExists("c:\visual\config") = False Then
```

```
            FSO2.CreateFolder ("c:\visual\config")
```

```
            f2_number = FreeFile(2)
```

```
            Open ("c:\visual\config\sr_log.ini") For Output As #f2_number
```

```
            Print #f2_number, comdef; c; bauddef; c; paritydef; c; datadef; c; stopdef
```

```
            Close (f2_number)
```

```
        Else
```

```
            If FSO2.FileExists("c:\visual\config\sr_log.ini") = False Then
```

```
                f2_number = FreeFile(2)
```

```
                Open ("c:\visual\config\sr_log.ini") For Output As #f2_number
```

```
                Print #f2_number, comdef; c; bauddef; c; paritydef; c; datadef; c; stopdef
```

```
                Close (f2_number)
```

```
            End If
```

```
    End If
```

```

End If
f2_number = FreeFile(2)
Open ("c:\visual\config\sr_log.ini") For Input As #f2_number
Input #f2_number, com_text, baud_text, parity_text, data_text, stop_text
Close (f2_number)

Select Case com_text
Case "1"
    c1_option.Value = True
Case "2"
    c2_option.Value = True
Case "3"
    c3_option.Value = True
Case "4"
    c4_option.Value = True
Case Else
    c2_option.Value = True
End Select

Select Case baud_text
Case "1200"
    br12_option.Value = True
Case "2400"
    br24_option.Value = True
Case "4800"
    br48_option.Value = True
Case "9600"
    br96_option.Value = True
Case "19200"
    br192_option.Value = True

```

```
Case Else  
    br96_option.Value = True  
End Select
```

```
Select Case parity_text
```

```
Case "E"  
    e_option.Value = True
```

```
Case "M"  
    m_option.Value = True
```

```
Case "N"  
    n_option.Value = True
```

```
Case "O"  
    o_option.Value = True
```

```
Case "S"  
    s_option.Value = True
```

```
Case Else  
    n_option.Value = True
```

```
End Select
```

```
Select Case data_text
```

```
Case "4"  
    db4_option.Value = True
```

```
Case "5"  
    db5_option.Value = True
```

```
Case "6"  
    db6_option.Value = True
```

```
Case "7"  
    db7_option.Value = True
```

```
Case "8"  
    db8_option.Value = True
```

```

Case Else
    db8_option.Value = True
End Select

Select Case stop_text
    Case "1"
        sb1_option.Value = True
    Case "1.5"
        sb15_option.Value = True
    Case "2"
        sb2_option.Value = True
    Case Else
        sb1_option.Value = True
End Select

End Sub

Private Sub Form_Unload(Cancel As Integer)
    Form1.Enabled = True
    Form1.Show
End Sub

Private Sub save_comm_Click()
    Call comm_select
    Call baud_select
    Call parity_select
    Call data_select
    Call stop_select
    c = ","
    f2_number = FreeFile(2)

```

```

Open "c:\visual\config\sr_log.ini" For Output As #f2_number
Print #f2_number, com_text; c; baud_text; c; parity_text; c; data_text; c; stop_text
Close f2_number

MsgBox ("Com " & com_text & "," & baud_text & "," & parity_text & "," & data_text & "," &
stop_text & " Saved.")

End Sub

Private Sub baud_select()
    If br12_option.Value Then
        baud_text = "1200"
    ElseIf br24_option.Value Then
        baud_text = "2400"
    ElseIf br48_option.Value Then
        baud_text = "4800"
    ElseIf br96_option.Value Then
        baud_text = "9600"
    ElseIf br192_option.Value Then
        baud_text = "19200"
    End If
End Sub

Private Sub parity_select()
    If e_option.Value Then
        parity_text = "E"
    ElseIf m_option.Value Then
        parity_text = "M"
    ElseIf n_option.Value Then
        parity_text = "N"
    ElseIf o_option.Value Then
        parity_text = "O"
    End If
End Sub

```

```

ElseIf s_option.Value Then
    parity_text = "S"
End If
End Sub

```

```

Private Sub data_select()
    If db4_option.Value Then
        data_text = "4"
    ElseIf db5_option.Value Then
        data_text = "5"
    ElseIf db6_option.Value Then
        data_text = "6"
    ElseIf db7_option.Value Then
        data_text = "7"
    ElseIf db8_option.Value Then
        data_text = "8"
    End If
End Sub

```

```

Private Sub stop_select()
    If sb1_option.Value Then
        stop_text = "1"
    ElseIf sb15_option.Value Then
        stop_text = "1.5"
    ElseIf sb2_option.Value Then
        stop_text = "2"
    End If
End Sub

```

โปรแกรมที่ ค.2 โปรแกรมการกำหนดค่าเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ORG 0000H
```

```
LJMP RST_VEC
```

```
CONTROL EQU 046H
```

```
RTC_ID EQU 11010000B
```

```
SCL EQU P1.0
```

```
SDA EQU P1.1
```

```
CONT_BYTE_W EQU 0D0H
```

```
CONT_BYTE_R EQU 0D1H
```

```
ADD_LOW: DS 1
```

```
DATA: DS 1
```

```
CAP_DATA EQU 032H
```

```
TEMP EQU 033H
```

```
BUFFER EQU 036H
```

```
FLAG EQU 02FH
```

```
BUSY BIT FLAG.0
```

```
HALF BIT FLAG.1
```

```
CAPTURE BIT P1.7
```

```
RST_VEC: MOV SP,#07FH-032H
```

```
INIT_SR: MOV TMOD,#020H
```

```
MOV TH1,#0FBH
```

```

MOV  A,#00H
MOV  SCON,#040H
MOV  PCON,A
MOV  IE,#10010000B
SETB TR1
MOV  DPTR,#MD_COMM_REACT
LCALL SEND_MODEM
LCALL DELAY100MS
MOV  DPTR,#MD_COMM_ACCEPT
LCALL SEND_MODEM
SETB REN
CLR  P1.2
CLR  RI
L:   CLR  P1.2

MOV  DPTR,#0000H
MOV  R5,#0AH
PUSH ACC
SETB SCL
SETB SDA

MOV  ADD_LOW,#04H
LCALL READ_BYTE
MOVX @DPTR,A
INC  DPTR

MOV  ADD_LOW,#02H
LCALL READ_BYTE
MOVX @DPTR,A

```

```

INC DPTR

MOV ADD_LOW,#01H
LCALL READ_BYTE
MOVX @DPTR,A
INC DPTR

```

```

MOV ADD_LOW,#00H
LCALL READ_BYTE
MOVX @DPTR,A
INC DPTR

```

```

L1:  LCALL DS1820_T_RD
     LCALL DS1820_T_RD
     MOV  A,TEMP
     MOVX @DPTR,A
     INC DPTR
     PUSH 05H
     LCALL DELAY1S
     POP 05H
     DJNZ R5,L1
     MOV DPTR,#MD_COMM_CALL
     LCALL SEND_MODEM
     JB  P1.3,$
     LCALL RAM_PC
     POP ACC
     SETB P1.2
     LCALL DELAY10MS
     SJMP L

```

```

RAM_PC:   PUSH  05H
          PUSH  04H
          MOV   DPTR,#CODE_ID
          LCALL SEND_DATA
          MOV   A,#' '
          LCALL TX
          MOV   DPTR,#0000H
          MOV   R4,#14

```

```

RAM_PC1:  MOVX  A,@DPTR

```

```

P_0:     LCALL TX
          MOV   A,#' '
          LCALL TX
          INC  DPTR
          DJNZ R4,RAM_PC1
          POP  04H
          POP  05H
          RET

```

```

SEND_DATA: MOV  A,#00H

```

```

LABEL2:   PUSH ACC
          MOVC A,@A+DPTR
          CJNE A,#00H,LABEL
          POP  ACC
          LJMP OUT_SEND

```

```

LABEL:    LCALL TX
          POP  ACC
          INC  A
          SJMP LABEL2

```

```

OUT_SEND:  RET

SEND_MODEM:  MOV  A,#00H
MODEM2:     PUSH  ACC
             MOVC A,@A+DPTR
             CJNE A,#0FFH,MODEM1
             POP  ACC
             LJMP OUT_MD
MODEM1:     LCALL TX
             POP  ACC
             INC  A
             SJMP MODEM2
OUT_MD:     RET

TX:         CLR  TI
             MOV  SBUF,A
             JNB  TI,S
             CLR  TI
             RET

STORE_DATA:  LCALL DS1820_T_RD
             MOV  A,=TEMP
             LCALL TX
             RET

DS1820_T_RD:  LCALL DS1820_RST
             LCALL DS1820_RES
             MOV  CAP_DATA,#0CCH

```

```

LCALL DS1820_WR
MOV CAP_DATA,#044H
LCALL DS1820_WR
SETB BUSY

```

```

PRES_CHK_LP: LCALL DS1820_RST

```

```

LCALL DS1820_RES
JB BUSY,PRES_CHK_LP
NOP
NOP
NOP
NOP
LCALL DS1820_RST
LCALL DS1820_RES
MOV CAP_DATA,#0CCH
LCALL DS1820_WR
MOV CAP_DATA,#0BEH
LCALL DS1820_WR
LCALL DS1820_RD
MOV TEMP,CAP_DATA
LCALL DS1820_RST
LCALL DS1820_RES
RET

```

```

DS1820_RST: CLR CAPTURE

```

```

LCALL DELAY1MS
SETB CAPTURE
MOV R4,#8
DJNZ R4,$
RET

```

```

DS1820_RES:  MOV  R4,#8
DS1820_R1:  MOV  R3,#0
DS1820_R2:  JNB  CAPTURE,DS1820_R3
             DJNZ R3,DS1820_R2
             DJNZ R4,DS1820_R1
             RET

DS1820_R3:  JNB  CAPTURE,$
             MOV  R4,#8
             DJNZ R4,$
             CLR  BUSY
             RET

DS1820_RD:  MOV  R4,#8
             CLR  A
DS_RD_LOOP: CLR  CAPTURE
             NOP
             NOP
             SETB CAPTURE
             NOP
             NOP
             NOP
             NOP
             MOV  C,CAPTURE
             ACALL CAP_DELAY
             RRC  A
             DJNZ R4,DS_RD_LOOP
             MOV  CAP_DATA,A
             RET

```

```

DS1820_WR:  MOV  R4,#8
             MOV  A,CAP_DATA
DS_WR_LOOP:  RRC  A
             JNC  DS_WR_L
             CLR  CAPTURE
             NOP
             NOP
             NOP
             NOP
             SETB CAPTURE
             LCALL CAP_DELAY
             AJMP DS_WR_NX
DS_WR_L:     CLR  CAPTURE
             LCALL CAP_DELAY
             SETB CAPTURE
             NOP
             NOP
             NOP
             NOP
DS_WR_NX:    DJNZ R4,DS_WR_LOOP
             RET

;*****
;*      READ DATA FROM EEPROM 1 BYTE      *
;* INPUT : ADD_HIGH                        *
;*      : ADD_LOW                          *
;* OUTPUT : DATA                          *
;*****
READ_BYTE:   CLR  SDA           ;start bit
             CLR  SCL
             MOV  A,#CONT_BYTE_W   ;send control byte

```

```

LCALL LOOP_BYTE
SETB SDA
SETB SCL

JB SDA,READ_BYTE ;loop until busy
CLR SCL
MOV A,ADD_LOW ;send address low
LCALL LOOP_BYTE
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
SETB SCL
SETB SDA
CLR SDA ;start bit
CLR SCL
MOV A,#CONT_BYTE_R ;send control byte
LCALL LOOP_BYTE
SETB SDA
SETB SCL
JB SDA,READ_BYTE ;loop until busy
CLR SCL
LCALL LOOP_READ
SETB SDA
SETB SCL
CLR SCL

SETB SCL ;stop bit
SETB SDA
RET

```

```

*****
;*          WRITE          *
;* INPUT: ACC              *
*****

LOOP_BYTE:  PUSH  02H
             MOV   R2,#08H
LOOP_SEND:  RLC   A
             MOV   SDA,C
             SETB  SCL
             CLR   SCL
             DJNZ  R2,LOOP_SEND
             POP   02H
             RET

*****
;*          READ           *
;* OUTPUT: ACC            *
*****

LOOP_READ:  PUSH  02H
             MOV   R2,#08H
LOOP_READ1: SETB  SCL
             MOV   C,SDA
             CLR   SCL
             RLC   A
             DJNZ  R2,LOOP_READ1
             MOV   DATA,A
             POP   02H
             RET

```

```

DELAY4M:  PUSH DPH
          PUSH DPL
          MOV  DPTR,#0000H

```

```

DEL4:    INC  DPTR
          MOV  A,DPL
          ORL  A,DPH
          JNZ  DEL4
          POP  DPL
          POP  DPH
          RET

```

```

;*****
;

```

```

;*      Send 1-Byte to SERIAL      *

```

```

;*      Input  : ACC              *

```

```

;*      Output : Serial port     *
;*****

```

```

TX_BYTE:  PUSH IE

```

```

          CLR  TI

```

```

          MOV  SBUF,A

```

```

          JNB  TI,$

```

```

          CLR  TI

```

```

          POP  IE

```

```

          RET

```

```

;*****
;

```

```

;*      Receive Data From SERIAL  *

```

```

;*      Input  : Serial Port     *

```

```

;*      Output : ACC            *
;*****

```

```

RX_BYTE:  PUSH IE

```

```

JNB RI,$      ; Wait data
CLR RI
MOV A,SBUF
LCALL TX_BYTE
POP IE
RET

```

```

CAP_DELAY:  MOV R6,#012H
CAP_D1:     NOP
            NOP
            DJNZ R6,CAP_D1
            RET

```

```

DELAY100MS: MOV R7,#100
D100MS_1:   MOV R6,#0FFH
D100MS_2:   NOP
            NOP
            NOP
            DJNZ R6,D100MS_2
            DJNZ R7,D100MS_1
            RET

```

```

DELAY10MS:  MOV R7,#010
D10MS_1:    MOV R6,#0FFH
D10MS_2:    NOP
            NOP
            NOP

```

```

NOP
DJNZ R6,D10MS_2
DJNZ R7,D10MS_1
RET

DELAY1MS: MOV R6,#0FFH
D1MS_1: NOP
NOP
NOP
DJNZ R6,D1MS_1
RET

DELAY1S: SETB RS0
MOV R5,#100
DIS_1: LCALL DELAY10MS
DJNZ R5,DIS_1
CLR RS0
RET

DELAY2S: SETB RS0
MOV R5,#100
D2S_1: LCALL DELAY10MS
DJNZ R5,D2S_1
CLR RS0
RET

MD_COMM_ACCEPT: DB 'at s0=1 s11=95'.0DH,0FFH
MD_COMM_CALL: DB 'atdt=16',0DH,0FFH

```

```
MD_COMM_REACT: DB 'atq1',0DH,0FFH
```

```
CODE_ID: DB 30H,30H,30H,30H,00H
```

```
END
```

โปรแกรมที่ ค.4 โปรแกรมควบคุมการทำงานของโครงข่ายรวบรวมข้อมูลระยะไกล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

'ไกรวุฒิ วิจารณ์ประเสริฐสุด. เข้าใจ/สร้าง/เล่น 'ไมโครโปรเซสเซอร์ 2. พิมพ์ครั้งที่ 1 .กรุงเทพฯ
:ซีเอ็ดยูเคชั่น .2539

ธีรวัฒน์ ประกอบผล. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์. พิมพ์ครั้งที่ 3. กรุงเทพฯ
: ประชาชน . 2542

ธีรวัฒน์ ประกอบผล. ปฏิบัติการทดลองไมโครคอนโทรลเลอร์ด้วย JASS-31.เล่มที่ 2 . กรุงเทพฯ
: สติลารีเสิร์ช . 2540



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาโท	นายพอล ภูมิดิษฐ์
วันเดือนปีเกิด	11 มิถุนายน พ.ศ.2522
สถานที่เกิด	อำเภอบางคล้า จังหวัดฉะเชิงเทรา
ภูมิลำเนาเดิม	83 หมู่ 13 ตำบลวังน้ำเย็น อำเภอบางคล้า จังหวัดฉะเชิงเทรา 27210
ที่อยู่ปัจจุบัน	83 หมู่ 13 ตำบลวังน้ำเย็น อำเภอบางคล้า จังหวัดฉะเชิงเทรา 27210
โทรศัพท์	(037) 251844
ประวัติการศึกษา	โรงเรียนวัดวังสีทอง
ระดับประถมศึกษา	โรงเรียนวังน้ำเย็นวิทยาคม
ระดับมัธยมศึกษาตอนต้น	วิทยาลัยเทคนิคฉะเชิงเทรา
ระดับปวช.	วิทยาลัยเทคนิคฉะเชิงเทรา
ระดับปวส.	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ปริญญาตรี	ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	ทุนยกเว้นหน่วยกิต
คติพจน์	ผู้ยิ่งใหญ่ย่อมไม่กลัวต่อความผิดที่ตนตัดสินใจทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญาโท	นายรัฐธรรมนุญ สุขมีศรี
วันเดือนปีเกิด	10 ธันวาคม พ.ศ. 2520
สถานที่เกิด	จังหวัดสุพรรณบุรี
ภูมิลำเนาเดิม	53 หมู่ 2 ตำบลบางตะเคียน อำเภอสองพี่น้อง จังหวัดสุพรรณบุรี 72110
ที่อยู่ปัจจุบัน	136/1 ถนนเชื่อมสัมพันธ์ เขตหนองจอก กรุงเทพมหานคร 10530
โทรศัพท์	(035) 428083 . 428047
ประวัติการศึกษา	โรงเรียนวัดสำเภาทอง
ระดับประถมศึกษา	โรงเรียนกรรณสูตศึกษาลัย
ระดับมัธยมศึกษาตอนต้น	วิทยาลัยเทคนิคสุพรรณบุรี
ระดับปวช.	วิทยาลัยเทคนิคสุพรรณบุรี
ระดับปวส.	วิทยาลัยเทคนิคสุพรรณบุรี
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	สิ่งที่ผิดพลาด เป็นบทเรียนและประสบการณ์ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายเอกภพ แสนสวาท
วันเดือนปีเกิด	7 เมษายน พ.ศ. 2522
สถานที่เกิด	จังหวัดนครสวรรค์
ภูมิลำเนาเดิม	51 หมู่ 1 ตำบลนครสวรรค์ออก
ที่อยู่ปัจจุบัน	อำเภอเมืองฯ จังหวัดนครสวรรค์ 60000 556/1 หมู่ 1 ซอยหมู่บ้านริมสวน ถนนอ่อนนุช แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร 10520
โทรศัพท์	01-3436455
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนวัดจอมคีรีนาคพรต
ระดับมัธยมศึกษาตอนต้น	โรงเรียนวิสุทธิศึกษา
ระดับปวช.	วิทยาลัยการอาชีพนครสวรรค์
ระดับปวส.	วิทยาลัยเทคนิคนครสวรรค์
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	เราเลือกเกิดไม่ได้ แต่เราเลือกที่จะเป็นคนดีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายเอกรัฐ ปัญญาเทพ
วันเดือนปีเกิด	30 ธันวาคม พ.ศ. 2522
สถานที่เกิด	โรงพยาบาลจังหวัดลำปาง
ภูมิลำเนาเดิม	95/26 ถนนสมักรสรพการ ตำบลแม่สอด อำเภอแม่สอด จังหวัดตาก
ที่อยู่ปัจจุบัน	300/5 หมู่บ้านรุ่งอรุณ 1 ถนนคลองกรุง แขวงลำปลาทิว เขตตลาดกระบี่ กรุงเทพมหานคร 10520
โทรศัพท์	01-4425487
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนภัทรวิทยา
ระดับมัธยมศึกษาตอนต้น	โรงเรียนสรรพวิทยาคม
ระดับปวช.	สถาบันเทคโนโลยีราชมงคล วิทยาเขตตาก
ระดับปวส.	สถาบันเทคโนโลยีราชมงคล วิทยาเขตตาก
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาวิศวกรรมวิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	ชีวิต คือตัวเราและเพื่อนคู่ใจ จงใช้สองสิ่งนี้เพื่อก้าวไปข้างหน้าอย่างเข้มแข็ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้