สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ซอฟต์แวร์สำหรับระบบตรวจสอบข้อมูลระยะไกลผ่านวิทยุสื่อสาร

Monitor program for

Long – rang Monitoring System using Armature Radio (LMSAR)

T104179

โดย

นายไชยา นะวงรัมย์

นายนภมณฑล อึ้งมณีภรณ์

นายมงคล จอมคำสิงห์

เลขหมู่.....
เลขทะเบียน 104179
วัน,เดือน,ปี 30 มี.ค. 2552

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Monitor program for

Long – rang Monitoring System using Armature Radio (LMSAR)

MR.CHAIYA NAWONGRAM

MR.NOPMONTOL UNGMANEEPORN

MR. MONGKOL JOMKAMSING

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KMIG MONGKUT'S INSTITUE OF TECHNOLOGY LADKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	ซอฟต์แวร์สำหรับระบบตรวจสอบข้อมูลระยะไกลผ่านวิทยุสื่อสาร Monitor program for Long – rang Monitoring System using Armature Radio (LMSAR)
นักศึกษา	นายไชยา นะวงรัมย์ รหัสนักศึกษา 49015447 นายนภมณฑล อึ้งมณีภรณ์ รหัสนักศึกษา 49015454 นายมงคล จอมคำสิงห์ รหัสนักศึกษา 49015460
อาจารย์ที่ปรึกษา	รศ.ดร.อรรถสิทธิ์ หล่าสกุล ผศ.สมภพ แก้วมีชัย
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2551

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

.....

(รศ.ดร.อรรถสิทธิ์ หล่าสกุล)

อาจารย์ที่ปรึกษา

.....

(ผศ.สมภพ แก้วมีชัย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	ซอฟต์แวร์สำหรับระบบตรวจสอบระยะไกลผ่านวิทยุสื่อสาร
นักศึกษา	นายไชยา นะวงรัมย์ รหัสนักศึกษา 49015447
	นายนภมณฑล อึ้งมณีภรณ์ รหัสนักศึกษา 49015454
	นายมงคล จอมคำสิงห์ รหัสนักศึกษา 49015460
อาจารย์ที่ปรึกษา	รศ.ดร.อรรถสิทธิ์ หล่าสกุล
	ผศ.สมภพ แก้วมีชัย
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2551

บทคัดย่อ

โครงงานนี้นำเสนอการออกแบบซอฟต์แวร์และสร้างระบบตรวจสอบวัดค่าระยะไกลแบบใหม่ที่อาศัยการรับส่งด้วยระบบวิทยุ ซึ่งทำให้สามารถติดตั้งเครื่องวัดของตัวลูก(Client) ได้ครอบคลุมพื้นที่มากขึ้น โดยไม่ต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุ โดยหลักการคือ ให้ตัวลูก(Client) ซึ่งมีอยู่หลายจุดนี้ มีการเชื่อมต่อกันคล้ายเครื่อข่าย(Network) โปรแกรมจะควบคุมการติดต่อกันอัตโนมัติ ตั้งแต่เริ่มส่งคำสั่งจากตัวแม่ (Main) ไปสู่ตัวลูก (Client) ที่ต้องการติดต่อด้วยเป็นทอดๆ โดยใช้ตัวลูก(Client)ที่อยู่ในพื้นที่กำลังส่งของตัวแม่ ทำหน้าที่เป็นตัวกลางในการรับและส่งข้อมูล และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถจะใช้ซอฟต์แวร์ที่ได้ออกแบบเป็นแบบ GUI ไว้บนคอมพิวเตอร์ทั่วไปเพื่อเข้าไปดูข้อมูลได้ทุกจุด ทุกเวลาตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Monitor program for Long – rang Monitoring System using Armature Radio

Student Mr.Chaiya Nawongram ID.49015447

Mr.Nopmontol Ungmaneeporn ID.49015454

Mr.Mongkol Jomkamsing ID.49015460

Adivisor Asst.Prof. Dr Attasit Lasakul

Asst.Prof. Sompob Kaewmeechai

Graduate Level Bachelor Degree of Information Engineering

Department Information Engineering

Academic Year 2008

ABSTRACT

This project presents designing software and build the system measures long distance new model value by use taking send with radio propagation which make can set up the measuring instrument of client get cover the area more and more by must not change or enhance the energy distributes the radio by use the principle is give client which exist many this dot has linking up are Network a program will control the connection automatic since begin send the command code from main to client that want to contact with by use client the location in energy area distributes main perform formed middle in taking and send the data and when the user wants to know value data measures that node any can use software that get design be GUI on general computer for enter can see the data every node at all times follow the requirement

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานเล่มนี้อาจไม่สำเร็จได้ด้วยดีหากไม่ได้รับความช่วยเหลือและร่วมมือจากหลาย ๆ ฝ่าย ด้วยกันบุคคลสำคัญที่มีส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้สำเร็จลงได้คือ อาจารย์อรรถสิทธิ์ หล่าสกุล และอาจารย์สมภพ แก้วมีชัย อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ให้ความเอาใจใส่ แนะนำและช่วยเหลืออย่างต่อเนื่อง ซึ่งต้องขอขอบพระคุณเป็นอย่างสูง

และต้องขอขอบคุณบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้คือ บิดา มารดา อันเป็นที่เคารพรักยิ่งซึ่งได้เลี้ยงดูพร้อมทั้งให้โอกาสทางการศึกษาอย่างเต็มที่ อีกทั้งยังให้กำลังใจและความเอาใจใส่ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ไชยา นะวงรัมย์
นภมณฑล อึ้งมณีภรณ์
มงคล จอมคำสิงห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ช
สารบัญตาราง	ฌ
บทที่ 1 บทนำ	**1**
1.1 ที่มาของปัญหาและแนวคิดเริ่มต้นในการทำโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตของโครงงาน	2
1.4 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**5**
2.1 ลักษณะโครงสร้างของไมโครคอนโทรลเลอร์ (AC89C51AC3)	5
2.1.1 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์	5
2.1.2 ขาสัญญาณและการเชื่อมต่อของไมโคโพรเซสเซอร์	6
2.1.3 การใช้ไทเมอร์เคาน์เตอร์	7
2.2 พอร์ตอนุกรม	9
2.2.1 การสื่อสารแบบอนุกรม	9
2.2.2 การสื่อสารข้อมูลแบบอะซิงโครนัส	9
2.2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232	10
2.2.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	11
2.2.5 UART	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.3 การอินเทอร์รัปต์	11
2.4 ภาษา C	14
2.4.1 คุณสมบัติที่น่าสนใจของภาษา C	14
2.4.2 โครงสร้างของภาษา C	15
2.5 Visual C# 2005	15
2.5.1 การเขียนโปรแกรมแบบ Event Driven Programming	16
2.5.2 รู้จักกับออบเจ็กต์	16
2.5.3 พร็อพเพอร์ตี้(Property) : คุณสมบัติของออบเจ็กต์	16
2.5.4 คอนโทรลกับคอมโพเนนต์	17
2.5.5 การกำหนดค่าใน Property Window	17
บทที่ 3 การออกแบบ	**18**
3.1 โครงสร้างการทำงานโดยรวม	18
3.2 ไดอะแกรม ซอฟแวร์ ควบคุมการรับส่งข้อมูล ของตัว Main และ Clients	19
3.3 รูปแบบการทำงานของซอฟต์แวร์	21
3.3.1 Command Format	21
3.3.2 Data Format	23
3.4 ส่วนประกอบของฮาร์ดแวร์	25
บทที่ 4 ผลการทดลอง	**27**
4.1 การทดลอง	27
4.2 การตั้งค่าก่อนใช้งาน	27
4.2.1 การตั้งค่าอุณหภูมิ	27
4.3 ขั้นตอนการทดลอง	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.4 ผลการทดลอง	34
4.4.1 ผลการอ่านค่าอุณหภูมิ	34
4.4.2 ผลการสั่งงานรีเลย์(Relay)	34
4.4.3 ผลการทดลอง สั่งงานและอ่านค่า Client ตัวอื่นๆ	35
4.5 สรุปผลการทดลอง	35
บทที่ 5 บทสรุป	**36**
5.1 บทสรุป	36
5.2 ปัญหาและอุปสรรคในการทำงาน	36
5.3 ข้อเสนอแนะ	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 1.1 การเชื่อมต่อกันโดยใช้คลื่นวิทยุ 2
รูปที่ 1.2 การเชื่อมต่อกันโดยใช้สายสัญญาณแทนคลื่นวิทยุ 3
รูปที่ 2.1 การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register) 6
รูปที่ 2.2 โครงสร้าง AT89C51AC3 7
รูปที่ 2.3 CPU MCS51 7
รูปที่ 2.4 การทำงานของไทเมอร์ โหมด 2 8 - bit Autoreload 8
รูปที่ 2.5 ลักษณะข้อมูลอนุกรมอย่างง่ายที่สุด 9
รูปที่ 2.6 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 10
รูปที่ 2.7 มาตรฐานระดับสัญญาณ RS232 10
รูปที่ 2.8 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 11
รูปที่ 2.9 การต่อใช้งานMAX232 13
รูปที่ 2.10 โครงสร้าง MAX232 13
รูปที่ 3.1 ตัวอย่างรูปแบบการเชื่อมต่อ 18
รูปที่ 3.2 โฟลว์ชาร์ทซอฟร์แวร์แสดงการควบคุมการรับส่งข้อมูลของตัว Main 19
รูปที่ 3.3 โฟลว์ชาร์ทซอฟร์แวร์แสดงการควบคุมการรับส่งข้อมูลของตัว Clients 20
รูปที่ 3.4 โฟลว์ชาร์ทการทำงานของซอฟต์แวร์ติดต่อใช้งานกับผู้ใช้ 21
รูปที่ 3.5 รูปแบบของ Command format 22
รูปที่ 3.6 รูปแบบของ Data format 21
รูปที่ 3.7 การเชื่อมต่อกันโดยใช้คลื่นวิทยุในการรับส่งข้อมูล 25
รูปที่ 3.8 การเชื่อมต่อกันโดยใช้สายสัญญาณแทนคลื่นวิทยุในการรับส่งข้อมูล 25
รูปที่ 3.9 ส่วนประกอบของฮาร์ดแวร์ชุดแม่ข่าย(Main) 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่ **หน้า**

รูปที่ 3.10 ส่วนประกอบของฮาร์ดแวร์ชุดลูกข่าย(Client) 26
รูปที่ 4.1 ปุ่มกดเพื่อตั้งค่าอุณหภูมิ 27
รูปที่ 4.2 แป้นกด(Keypad) ที่ใช้กดตั้งค่าอุณหภูมิ 28
รูปที่ 4.3 ค่าอุณหภูมิที่ตั้งไว้ 28
รูปที่ 4.4 รูปแบบการเชื่อต่อ 29
รูปที่ 4.5 หน้าจอของโปรแกรม 30
รูปที่ 4.6 การตั้งค่าในการรับส่งข้อมูล 31
รูปที่ 4.7 กำหนดคำสั่งที่จะส่ง 32
รูปที่ 4.8 กำหนดเส้นทางที่จะส่ง 33
รูปที่ 4.9 ชุดคำสั่งที่ถูกส่งออกไป 33
รูปที่ 4.10 ค่าอุณหภูมิที่อ่านค่าและตอบกลับมา 34
รูปที่ 4.11 Relay 1 ON 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 4.1 ผลการทดลอง สั่งงาน และ อ่านค่า Client ตัวอื่นๆ 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของปัญหาและแนวคิดเริ่มต้นในการทำโครงงาน

ในการตรวจสอบบันทึกค่าการเปลี่ยนแปลงต่างๆ ติดต่อกันในช่วงเวลานานๆ เพื่อการคาดการณ์ หรือวิเคราะห์เหตุการณ์ อันอาจเกิดขึ้นในอนาคต เป็นสิ่งจำเป็นมากต่อการพิจารณาเพื่อหาแนวทางแก้ไขหรือบรรเทาปัญหาต่างๆที่อาจเกิดขึ้นได้ ตัวอย่างเช่น การติดตามค่าอุณหภูมิ,ค่าระดับน้ำ,ค่าความชื้น เป็นต้น ซึ่งหากได้มีการตรวจสอบอย่างต่อเนื่องไปเป็นระยะนานๆ ก็สามารถนำค่าเหล่านั้นมาหาสถิติความน่าจะเป็นต่างๆ ซึ่งนับว่าเป็นประโยชน์ในการพิจารณาแก้ไขปัญหาต่อไปในอนาคต ก่อนหน้านี้ ได้มีการสร้างโครงงานเครื่องตรวจวัดเพื่อทำหน้าที่ดังกล่าว ซึ่งจะทำให้ผู้ใช้งานสามารถดึงข้อมูล ผลการตรวจวัดจากตัวเซนเซอร์เหล่านั้นได้โดยวิธีการรีโมทจากเครื่องคอมพิวเตอร์ของผู้ใช้งานเอง โดยไม่ต้องเข้าไปถึงตัวตำแหน่งที่วางเซนเซอร์นั้น ด้วยการใช้วิธีส่งผ่านคลื่นวิทยุ แต่มีข้อจำกัดคือตัวลูก(Client) ต้องอยู่ในย่านกำลังรับ-ส่ง ของเครื่องแม่(Main) เท่านั้น ซึ่งเป็นการจำกัดพื้นที่การติดตั้งเครื่องลูก ที่ใช้ในการตรวจวัด เพราะหากต้องการพื้นที่กว้างมากขึ้น จะต้องใช้ระบบวิทยุที่มีกำลังส่งสูงมาก ส่งผลให้มีค่าใช้จ่ายเพิ่มมากขึ้นตามไปด้วย ดังนั้นด้วยข้อด้อยจุดนี้ โครงงานนี้จึงนำเสนอการออกแบบและสร้างระบบเครื่องตรวจสอบวัดค่าระยะไกลแบบใหม่โดยเน้นที่ออกแบบซอฟต์แวร์สำหรับใช้งานกับระบบฮาร์ดแวร์เดิม ซึ่งทำให้สามารถติดตั้งเครื่องวัดของตัวลูกได้ครอบคลุมพื้นที่มากขึ้น โดยไม่จำเป็นต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุอีก โดยการให้ตัวลูกซึ่งมีอยู่หลายจุดนี้ มีการเชื่อมต่อกันคล้ายเครือข่าย(Network) โปรแกรมจะควบคุมการติดต่อกันอัตโนมัติตั้งแต่เริ่มส่งคำสั่งจากตัวแม่ ไปสู่ตัวลูก ที่ต้องการติดต่อด้วยเป็นทอดๆ จึงไม่ต้องเพิ่มกำลังส่งวิทยุของตัวแม่ เพียงแต่จะต้องมีตัวลูกตัวอื่นที่อยู่ในพื้นที่กำลังส่งของตัวแม่ ทำหน้าที่เป็นตัวกลางรับ-ส่งข้อมูลแทน ดังนั้นในการใช้งานจึงสามารถติดตั้งตัวลูก ในสถานที่ต่างๆหลายจุด ให้ครอบคลุมบริเวณที่ผู้ใช้งานต้องการได้มากขึ้น แต่ต้องคำนึงถึงว่าต้องอยู่ในพื้นที่รับส่งต่อเนื่องกันได้ของตัวลูกเหล่านั้น และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถจะใช้ซอฟต์แวร์ที่ได้ออกแบบเป็นแบบ User Interface ไว้บนคอมพิวเตอร์ทั่วไปเพื่อเข้าไปดูข้อมูลได้ทุกจุดทุกเวลาได้ตามความต้องการ ซึ่งวิธีนี้เป็นการเพิ่มประสิทธิภาพการนำไปใช้งานได้มากยิ่งขึ้นนั่นเอง

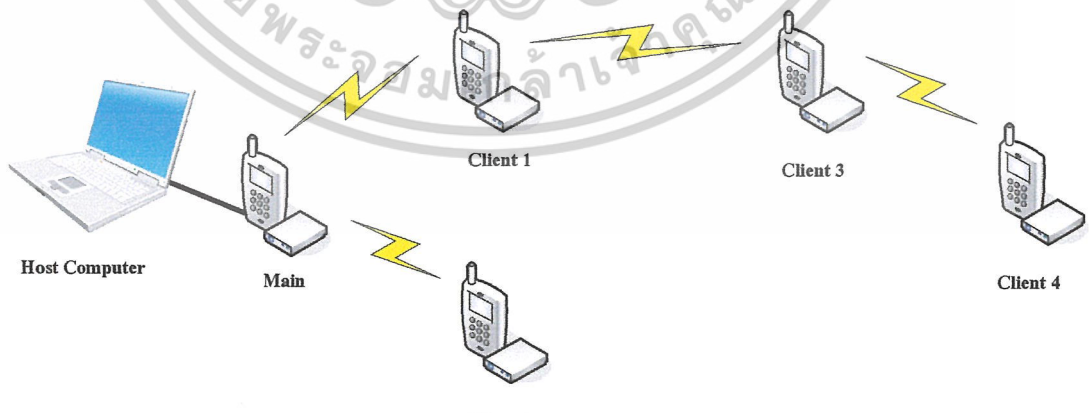
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1. ให้มีความรู้ในด้านฮาร์ดแวร์และซอฟต์แวร์ของไมโครคอนโทรลเลอร์เอ็มซีเอส 51
2. เพื่อศึกษาการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ด้วยภาษาซี
3. เพื่อศึกษาการเขียนโปรแกรมหน้าจอติดต่อใช้งาน (Graphic User Interface) บนวินโดว์
4. เพื่อเพิ่มประสิทธิภาพการนำไปใช้งานของระบบตรวจสอบระยะไกลผ่านวิทยุสื่อสาร

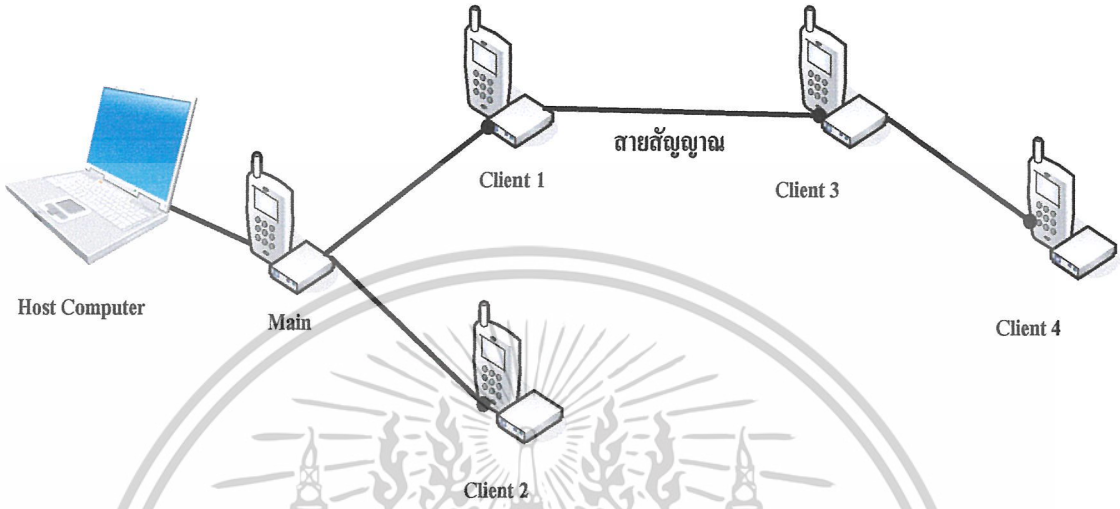
1.3 ขอบเขตของโครงงาน

1. โปรแกรมสามารถติดต่อสื่อสารกันระหว่างคอมพิวเตอร์ (Host Computer) กับ ตัวตรวจสอบหลัก หรือ เครื่องแม่(Main) ได้
2. เครื่องลูกข่าย (Client) แต่ละตัวที่อยู่ในย่านการรับส่งของคลื่นวิทยุ สามารถที่จะรับ-ส่งข้อมูลระหว่างกันได้ สามารถรับ-ส่งข้อมูลในแบบเดียวกันได้กับเครื่องแม่(Main)
3. สามารถรับและส่งข้อมูลต่อกันเป็นทอดๆได้
4. ในการทดลองการทำงานของระบบ จำเป็นต้องใช้เครื่องลูกข่ายจำนวนหลายๆชุดซึ่งมีราคาค่อนข้างสูง จึงทำการจำลองเครื่องลูกข่ายขึ้นมา ซึ่งเครื่องลูกข่ายที่สามารถติดต่อรับ-ส่งข้อมูลกันได้นั้น ต้องอยู่ในย่านกำลังส่งของวิทยุสื่อสารที่ใช้ โดยใช้สายสัญญาณเป็นสื่อรับ-ส่งข้อมูลแทนคลื่นวิทยุ



รูปที่ 1.1 การเชื่อมต่อกันโดยใช้คลื่นวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.2 การเชื่อมต่อกันโดยใช้สายสัญญาณแทนคลื่นวิทยุ

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาข้อมูลเกี่ยวกับไมโครคอนโทรลเลอร์เอ็มซีเอส 51
2. ขั้นตอนทำฮาร์ดแวร์โดยทำบอร์ดจำลองเอ็มซีเอส 51เอซี 3 เป็นตัวลูกเครือข่าย (Client)
3. ขั้นตอนการเขียนโปรแกรม Visual C# แสดงผลหน้าจอติดต่อใช้งาน (User Interface)
4. ขั้นตอนการเขียนโปรแกรมระหว่างคอมพิวเตอร์กับตัวโหนดแม่ (Main)
5. เขียนโปรแกรมควบคุม MCS51 ทั้งระบบ
6. จัดส่วนอุปกรณ์ทั้งหมดเข้าด้วยกัน
7. ทดสอบการทำงาน
8. ปรับปรุงแก้ไขและทดลอง
9. เขียนปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอน	พ.ศ. 2551							พ.ศ. 2552	
	มิ.ย	ก.ค	ส.ค	ก.ย	ต.ค	พ.ย	ธ.ค	ม.ค	ก.พ
1. ศึกษาข้อมูลและออกแบบ	←	→							
2. จัดทำวงจร MCS51 ตัวลูก (ฮาร์ดแวร์)		←	→	→	→	→			
3. เขียนโปรแกรมแสดงหน้าจอติดต่อใช้งาน		←	→	→					
4. เขียนโปรแกรมติดต่อคอมพิวเตอร์กับโหนด		←	→	→					
5. เขียนโปรแกรมควบคุม MCS51 ทั้งระบบ				←	→	→	→		
6. จัดส่วนของอุปกรณ์ทั้งหมดเข้าด้วยกัน						←	→	→	
7. ทดสอบการทำงาน						←	→	→	
8. ปรับปรุงแก้ไขและทดลอง								←	→
9.เขียนปริญญานิพนธ์		←	→	→	→	→	→	→	→

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ลักษณะโครงสร้างของไมโครคอนโทรลเลอร์ (AC89C51AC3)

ไมโครคอนโทรลเลอร์AT89C51AC3 เป็นไมโครคอนโทรเลอร์ในตระกูลMCS – 51 ขนาด 52 pin ของ บริษัท ATMEL บรรจุอยู่ภายในตัวถังแบบ 52 pin PLCC โดยไมโครคอนโทรลเลอร์ตัวนี้จะมีจุดเด่น คือ เรื่องของความเร็วในการประมวลผล มีแหล่งกำเนิดสัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งสามารถกำหนดการทำงานของ MCU ให้ทำงานในโหมดความเร็ว 2 เท่า (×2 Mode) ได้ ทำให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 58.9824 MHz และคุณสมบัติที่เด่นๆ ของ ไมโครคอนโทรลเลอร์เบอร์นี้ ได้แก่

มีหน่วยความจำ Flash สำหรับเขียนโปรแกรมขนาด 64 KByte
มีหน่วยความจำสำหรับเก็บข้อมูล แบบ EEPROM ขนาด 2 KByte
หน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 2304 Byte (2048+256)
วงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
พอร์ต I/O ขนาด 8 บิต จำนวน 5 พอร์ต (P0, P1, P2, P3 และ P4 (5 Bit))
มีวงจร Timer/Counter ขนาด 16 บิต จำนวน 3 ชุด
มีวงจร ADC ขนาด 10 บิต จำนวน 8 ช่อง (ใช้ Port – P1 โดยกำหนดจากโปรแกรม)
มีวงจร Watchdog, Power – ON Reset, Capture/Compare, PWM

2.1.1 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ จะแบ่งหน่วยความจำออกเป็น 3 กลุ่มคือ

1. หน่วยความจำสำหรับเก็บโปรแกรม(Program Memory) หรือ (Code Memory) จะทำหน้าที่เก็บชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ปฏิบัติตามคำสั่งนั้นๆ หน่วยความจำจะเป็นลักษณะแบบแฟลช(Flash) ที่มีคุณสมบัติในการใช้งานโดยสามารถจะทำการลบข้อมูลด้วยไฟฟ้าและเก็บข้อมูลเข้าเก็บไว้ในตัวไมโครคอนโทรลเลอร์ ได้กว่า 1ล้านครั้ง โดยใช้เครื่องโปรแกรม

2. หน่วยความจำสำหรับเก็บข้อมูล(Data Memory) หน่วยความจำข้อมูล (RAM) จะทำหน้าที่เก็บรักษาข้อมูล โดยข้อมูลอาจจะเป็นค่าหลังจากไมโครคอนโทรลเลอร์ ทำการการประมวลผล หรือเก็บค่าข้อมูลที่จะให้กับไมโครคอนโทรลเลอร์ประมวลผลในขณะนั้น

3. รีจิสเตอร์ที่ทำหน้าที่เฉพาะ(Special Function) รีจิสเตอร์เฉพาะหรือรีจิสเตอร์พิเศษ (Special Function Register) ที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ จะอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

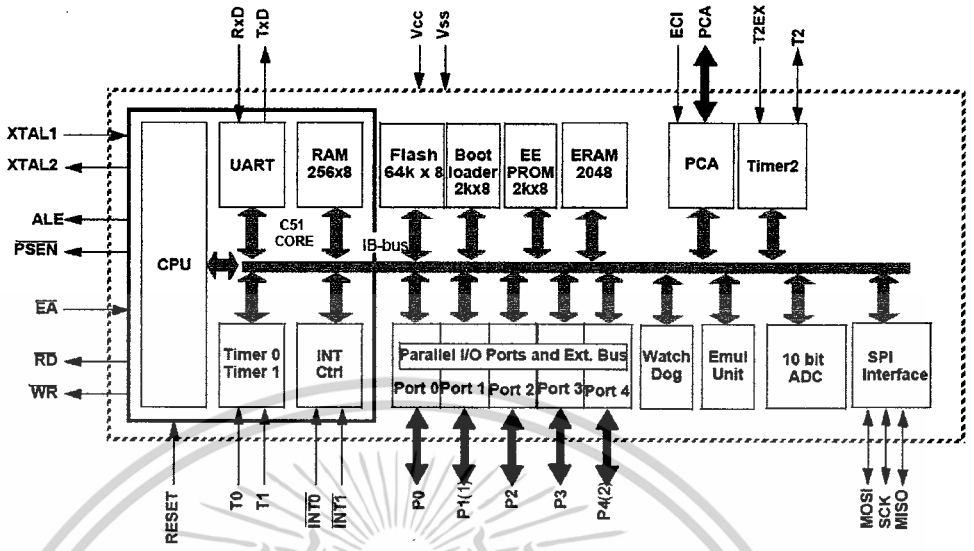
ในหน่วยความจำตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งสามารถจะเรียกใช้ชื่อของรีจิสเตอร์ได้โดยตรง หรืออาจจะเรียกชื่อตามตำแหน่งแอดเดรสก็ได้ รีจิสเตอร์เฉพาะจะประกอบด้วย

รูปที่ 2.1 การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ(Special Function Register)

2.1.2 ขาสัญญาณและการเชื่อมต่อของไมโครโพรเซสเซอร์

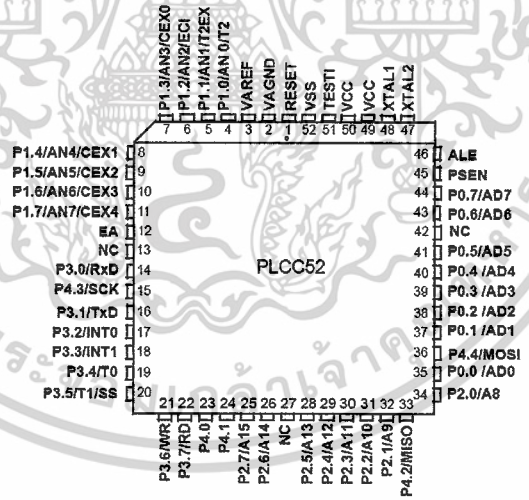
ขาสัญญาณของ MCS-51 โดยทั่วไปจะบรรจุมาบนตัวถังพลาสติกแบบ DIP ขนาด 40 ขา สำหรับไมโครคอนโทรลเลอร์ AT89C51AC3 จะมี 52 ขาตัวถังแบบ PLCC52 ซึ่งมีขาใช้งานแบ่งออกเป็นหลายกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Notes: 1. 8 analog Inputs/8 Digital I/O
2. 5-Bit I/O Port

รูปที่ 2.2 โครงสร้าง AT89C51AC3



รูปที่ 2.3 CPU MCS51

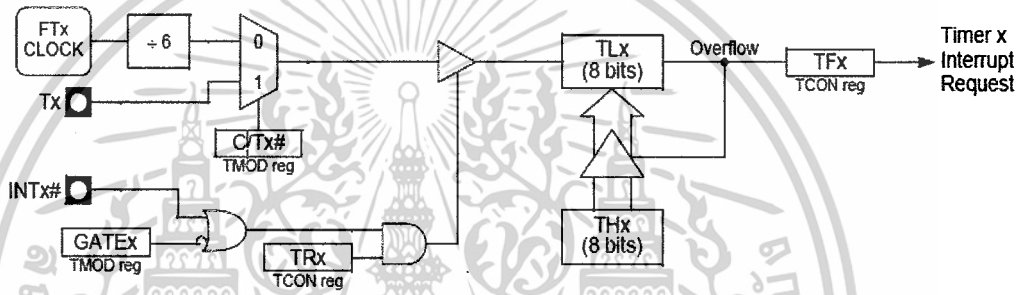
2.1.3 การใช้ไทเมอร์เคาน์เตอร์

ใน MCS – 51 ทุกตัวจะมีไทเมอร์เคาน์เตอร์อยู่ภายใน บางเบอร์จะมีสามตัว แต่ส่วนใหญ่แล้วจะมีสำหรับ AT89C51AC3 มีสองตัวคือไทเมอร์ 0 (Timer 0) และไทเมอร์ 1 (Timer 1) โดยเป็นรีจีสเตอร์ขนาด 16 บิต(Timer 0 แบ่งเป็น TH0 และ TL0 ส่วน Timer 1 แบ่งเป็น TH1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ TL1) และจะสามารถโปรแกรมให้เป็นตัวนับหรือตัวจับเวลาก็ได้ การใช้งานเป็นไทเมอร์จะเป็นการตั้งเวลาด้วยการโปรแกรม เมื่อเวลาถึงค่าที่กำหนดไทเมอร์จะแสดงผลออกมาทางบิตแฟลก (TF) การใช้งานไทเมอร์สามารถโปรแกรมได้หลายโหมด หลายรูปแบบ

การใช้งานไทเมอร์จะมีรีจิสเตอร์ที่เกี่ยวข้องอยู่สามตัวคือ รีจิสเตอร์ TMOD สำหรับกำหนดโหมดการทำงาน รีจิสเตอร์ TCON สำหรับควบคุมการปิด-เปิดไทเมอร์และควบคุมการอินเทอร์รัปต์ และรีจิสเตอร์ไทเมอร์ ถ้าหากต้องการให้ไทเมอร์ทำงานในโหลด 1 จะมีรูปแบบการทำงาน



รูปที่ 2.4 การทำงานของไทเมอร์ โหมด 2 8 - bit Autoreload

การทำงานของ Timer แต่ละตัวจะทำงานเป็นอิสระจากกันการทำงานในโหมดของตัวจับเวลา จะใช้สัญญาณนาฬิกาที่อยู่ภายในตัวไมโครคอนโทรลเลอร์ ซึ่งทราบความถี่ ค่าของเวลาเริ่มต้นที่อยู่ในรีจีสเตอร์ TH และ TL จะเพิ่มขึ้น 1 ในทุกแมชชีนไซเคิล (ความถี่สัญญาณนาฬิกาหารด้วย 12 (เมื่อค่าของรีจีสเตอร์ TH และ TL เพิ่มขึ้นเป็น 1 ทุกบิตแล้วจะวนกลับไปเป็น 0 ใหม่อีกช่วงนี้เรียกว่าเป็นโอเวอร์โฟลว (Overflow) ซึ่งเป็นช่วงของการครบกำหนด เวลาที่ตั้งไว้ ค่าของเวลาสามารถคำนวนได้จากค่าเริ่มต้นใน TH และ TL ไปจนถึงค่าสูงสุดคือ 0000H (หรือ 65536) รีจีสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของตัวตั้งเวลา/ตัวนับมี 2 ตัวคือรีจีสเตอร์ TCON และ รีจีสเตอร์ TMOD

รีจีสเตอร์ TCON (Timer/Counter Control Register)

เป็นรีจีสเตอร์ที่ทำหน้าที่เฉพาะ(อยู่ในตำแหน่ง 89H สามารถอ้างตำแหน่งแบบบิตได้) สำหรับควบคุมการทำงานของ Timer และเลือกลักษณะสัญญาณอินเตอร์รัพท์จากภายนอก โดยรีจีสเตอร์นี้จะทำหน้าที่เป็นแฟลกแสดงสถานะการทำงาน 4 บิตที่เซต และเคลียด้วยการทำงานของฮาร์ดแวร์ ส่วนที่เหลืออีก 4 บิตเป็นบิตควบคุมการทำงานของ Timer ซึ่งเราสามารถเซตและเคลีย

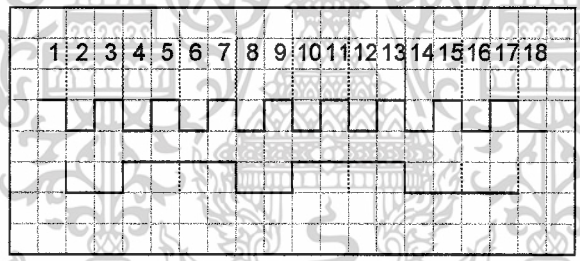
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยคำสั่งทางซอฟต์แวร์จากการใช้คำสั่งเซตบิตหรือการ โอนย้ายข้อมูลก็ได้ บิตต่างๆ ของรีจีสเตอร์ TCON

TMOD (อยู่ในตำแหน่ง 89H) เป็นรีจีสเตอร์ที่ทำหน้าที่ควบคุม โมดการทำงาน และสัญญาณควบคุมการทำงานของ Timer0 และ Timer1 รีจีสเตอร์ตัวนี้ไม่สามารถอ้างตำแหน่งแบบบิต การกำหนดค่าทำได้โดย ใช้คำสั่งโอนย้ายข้อมูล 8 บิต

2.2 พอร์ตอนุกรม

2.2.1 การสื่อสารแบบอนุกรม การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2แบบ คือ การสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ดังนั้นการติดต่อสื่อสารกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมอย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกาข้อมูล และกราวด์ รูปที่ 2.5 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.5 ลักษณะข้อมูลอนุกรมอย่างง่ายที่สุด

2.2.2 การสื่อสารข้อมูลแบบอะซิงโครนัส การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที(bit per second: bps) รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4ส่วนด้วยกัน คือ

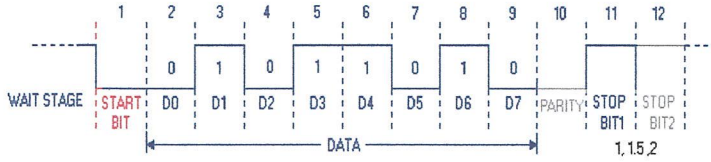
1.บิตเริ่มต้น(Start Bit) ซึ่งจะมีขนาด 1บิต

2.บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6,7หรือ 8บิต

3.บิตตรวจสอบพาริตี้(Parity Bit) จะมีขนาด 1บิตหรือไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

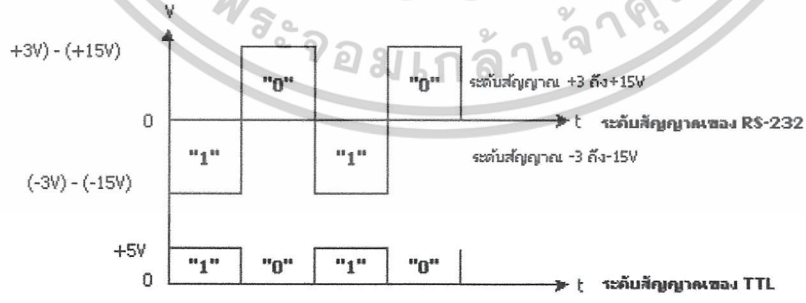
4.บิตปิดท้าย(Stop Bit) จะมีขนาด 1,1.5หรือ 2บิต



รูปที่ 2.6 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

จากรูปที่ 2.6 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่งขา DATA จะมีสถานะลอจิก " 1 " วงจะเรียกสถานะนี้ว่าสถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มให้จากการให้ขา DATA มีลอจิก " 0 " ด้วยช่วงระยะเวลา 1บิต เรียกบิตนี้ว่า บิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5, 6,7 หรือ 8บิต ก็ได้ จากนั้นก็ตามด้วย บิตพาริตี้ ซึ่งใช้ตรวจสอบความผิดพลาดที่อาจจะเกิดขึ้นกับการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตปิดท้าย ซึ่งจะให้สถานะขาดาต้ามีสถานะลอจิก " 1 " อีกครั้ง ด้วยระยะเวลาอย่างน้อย 1บิต, 1.5บิต หรือ 2บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

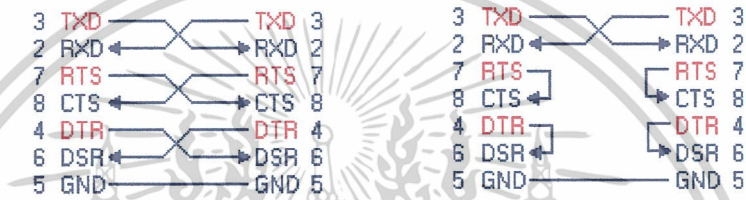
2.2.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232 ที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ความยาวสูงสุดของสายสัญญาณประมาณ 50 ฟุต มีระดับสัญญาณตั้งแต่ 3-ถึง 12-V แสดงว่ามีข้อมูล(Mark) และ +3ถึง + 12V แสดงว่าเป็นช่องว่าง (Space)



รูปที่ 2.7 มาตรฐานระดับสัญญาณ RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB 25 ตัวผู้ หรือ DB 9 ตัวผู้ซึ่งคอนเน็กเตอร์แบบ DB 25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB 9 เนื่องจากขาอื่นๆที่ใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาดังรูป



ก แบบ Null modem ข แบบ 3 เส้น

รูปที่ 2.8 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

2.2.5 UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรม

ไมโครคอนโทรลเลอร์ตระกูล MCS-51จะมีวงจรสื่อสารอนุกรม UART (Universal Asynchronous Receiver Transmitter) แบบ Full Duplex คือ สามารถรับและส่งข้อมูล ได้พร้อมกันคือขา P3.0 เป็นขารับข้อมูล (RX) และ P3.1 เป็นขา ส่งข้อมูล (TX) มีรีจิสเตอร์ที่เกี่ยวข้องคือ SCON (Serial Port Control Register), SUBF (Serial Data Buffer) และ PCON (Power Control Register)

2.3 การอินเทอร์รัปต์

อุปกรณ์ภายนอกสามารถติดต่อกับไมโครคอนโทรลเลอร์ได้สองวิธีคือการติดต่อแบบ Polling Method ที่ CPU ต้องคอยตรวจสอบอุปกรณ์อินพุตเอาต์พุตตลอดเวลา และการอินเทอร์รัปต์ (interrupt) ที่ CPU ไม่ต้องคอยไปตรวจสอบอุปกรณ์อินพุตเอาต์พุตถ้าอุปกรณ์ตัวใดต้องการติดต่อกับ CPU หรือไมโครคอลโทรเลอร์ลงได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ MCS-51 จะมีการอินเทอร์รัปต์อยู่สองประเภทคือ สัญญาณอินเทอร์รัปต์จากภายนอกที่เกิดจากสัญญาณที่เข้ามาทางขา INT0 และ INT1 และสัญญาณอินเทอร์รัปต์ภายในเกิดจากไทเมอร์ 0 ไทเมอร์ 1 และพอร์ตอนุกรม เมื่อไมโครคอนโทรลเลอร์ถูกอินเทอร์รัปต์ เมื่อทำโปรแกรมตอบสนองการอินเทอร์รัปต์จบแล้วจะทำโปรแกรมหลักที่ทำงานค้างอยู่ต่อไป การใช้อินเทอร์รัปต์ของไมโครคอนโทรเลอร์ MCS-51 จะมีรีจิสเตอร์ที่เกี่ยวข้องอยู่ 3 ตัว คือ

- รีจิสเตอร์ IE (Interrupt Enable) ใช้ในการกำหนดว่าจะยอมให้อินเทอร์รัปต์จากแหล่งใดได้บ้าง

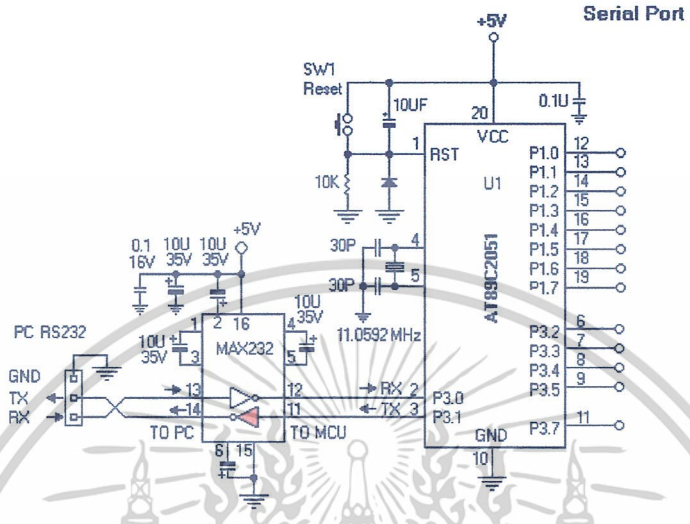
- รีจิสเตอร์ IP (Interrupt Priority) ใช้กำหนดลำดับของการอิเทอร์รัปต์ กรณีที่เกิดการอินเทอร์รัปต์จากหลายแหล่งพร้อมๆ กัน

- รีจิสเตอร์ TCON (Timer Control) รีจิสเตอร์ตัวนี้นอกจากจะใช้ควบคุมไทเมอร์แล้วยังใช้ในการอินเทอร์รัปต์อีกด้วย

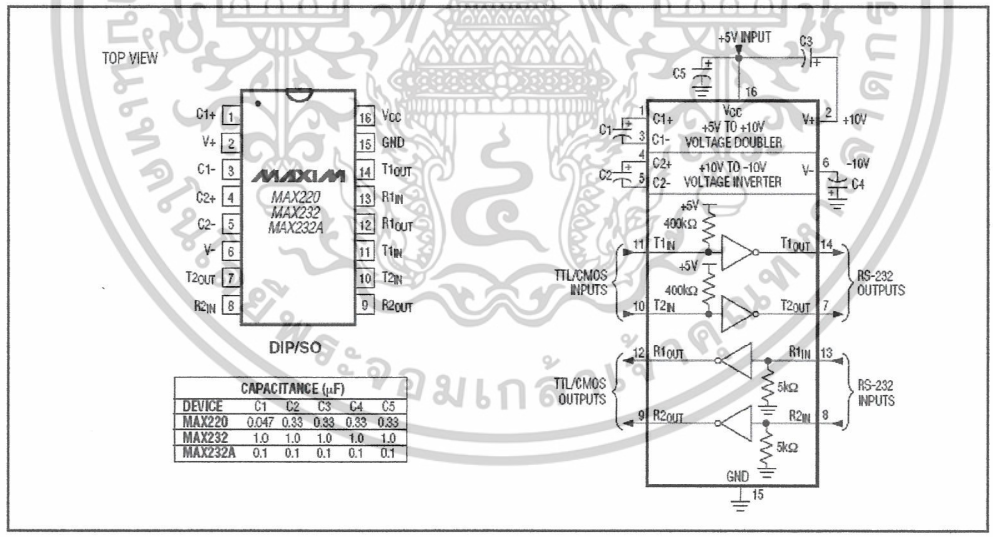
ถ้าหากเขียนโปรแกรมสำหรับการอินเทอร์รัปต์ด้วยภาษาแอสเซมบลี โปรแกรมตอบสนองการอินเทอร์รัปต์ที่เกิดจากสัญญาณอินเทอร์รัปต์จากแหล่งใด จะต้องนำโปรแกรมไปเก็บในตำแหน่งนั้นที่เรียกว่าอินเทอร์รัปต์เวกเตอร์

สัญญาณที่ออกมาจากพอร์ตของไมโครคอนโทรลเลอร์นั้นเป็นสัญญาณระดับ TTL ซึ่งมีระดับสัญญาณอยู่ที่ 0-5V แต่ตามมาตรฐานการสื่อสารแบบอนุกรม RS-232 นั้นสัญญาณ logic “0” ต้องมีระดับสัญญาณอยู่ที่ 3-15 V และ logic “1” ต้องมีระดับสัญญาณอยู่ที่ (-3)-(-15V) ดังนั้นเราจึงต้องใช้วงจรแปลงระดับสัญญาณซึ่งสามารถใช้ IC MAX232 เป็นตัวปรับระดับสัญญาณจากระดับ TTL ให้เป็นไปตามมาตรฐานของ RS-232 โดยสามารถต่อวงจรได้ตามรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การต่อใช้งาน MAX232



รูปที่ 2.10 โครงสร้าง MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษา C

2.4.1 คุณสมบัติที่ของภาษา C

- เป็นภาษาที่มีลักษณะเป็นโครงสร้างจึงเขียนโปรแกรมง่าย โปรแกรมที่เขียนขึ้นจะทำงานได้อย่างมีประสิทธิภาพสูง สั่งงานคอมพิวเตอร์ได้รวดเร็วกว่าภาษาระดับสูงอื่น ๆ

- สั่งงานอุปกรณ์ในระบบคอมพิวเตอร์ได้เกือบทุกส่วนของฮาร์ดแวร์ซึ่งภาษาระดับสูงภาษาอื่นทำงานดังกล่าวได้น้อยกว่า

- คอมไพเลอร์ภาษาซีทุกโปรแกรมในท้องตลาดจะทำงานอ้างอิงมาตรฐาน (ANSI= American National Standards Institute) เกือบทั้งหมด จึงทำให้โปรแกรมที่เขียนขึ้นด้วยภาษาซีสามารถนำไปใช้กับคอมพิวเตอร์ได้ทุกรุ่นที่มาตรฐาน ANSI รับรอง

- โปรแกรมที่เขียนขึ้นด้วยภาษาซีสามารถนำไปใช้กับเครื่องคอมพิวเตอร์ที่ใช้ซีพียูต่างเบอร์กันได้ หรือกล่าวได้ว่าโปรแกรมมีความยืดหยุ่น (portabiliy) สูง

- สามารถนำภาษาซีไปใช้ในการเขียนโปรแกรมประยุกต์ได้หลายระดับเช่น เขียน โปรแกรมจัดระบบงาน)OS) คอมไพเลอร์ของภาษาอื่น โปรแกรมสื่อสารข้อมูล โปรแกรมจัดฐานข้อมูล โปรแกรมปัญญาประดิษฐ์ (AI = Artificial Inteeligent) รวมทั้งโปรแกรมคำนวณงานทางด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ เป็นต้น

- มีโปรแกรมช่วย (tool box) ที่ช่วยในการเขียนโปรแกรมมาก และราคาไม่แพงหาซื้อได้ง่าย เช่น vitanin c หรืออื่น ๆ

- สามารถประกาศข้อมูลได้หลายชนิดและหลายรูปแบบ ทำให้สะดวกรวดเร็วต่อการพัฒนาโปรแกรมตามวัตถุประสงค์ของผู้ใช้

- ประยุกต์ใช้ในงานสื่อสารข้อมูล และงานควบคุมที่ต้องการความแม่นยำในเรื่องเวลา)real time application) ได้ดีกว่าภาษาระดับสูงอื่น ๆ หลาย ๆภาษา

- สามารถเขียนโปรแกรมด้วยเทคนิคแบบโอโอพี (OOP = Object OrientedProgramming) ได้หากใช้ภาษาซีรุ่น TURBO C++ ขึ้นไป ทำให้สามารถพัฒนาโปรแกรมประยุกต์เพื่อใช้งานได้กว้างขวางมากยิ่งขึ้นกว่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 โครงสร้างของภาษา C

ส่วนประกอบของโปรแกรมภาษาซีอาจแบ่งออกได้ดังต่อไปนี้

1. คอมไพเลอร์ไดเรกทีฟ (Compiler Directive) เป็นส่วนของโปรแกรมที่ใช้สำหรับเป็นตัวบอกคอมไพเลอร์ว่าให้รวมไฟล์ต่าง ๆ ที่กำหนดไว้ในส่วนนี้กับตัวโปรแกรมที่เขียนขึ้น โดยต้องเขียนตามรูปแบบที่กำหนดไว้ ดังนี้

#include <ชื่อไฟล์.h>

เช่น #include <stdio.h>

2. ตัวโปรแกรม (body) เป็นส่วนที่ผู้ใช้ต้องเขียนขึ้นเองโดยนำเอาคำสั่งหรือฟังก์ชันมาตรฐานต่าง ๆ มาเรียบเรียงกันขึ้นเป็นโปรแกรมเพื่อสั่งให้คอมพิวเตอร์รับข้อมูลจากอุปกรณ์อินพุตประมวลผล จะขึ้นต้นด้วย main()

{ } ซึ่งภายในวงเล็บประกอบด้วยคำสั่ง ต่าง ๆ ดังนี้

ส่วนของการประกาศตัวแปร (declaration)

ส่วนนำเข้าข้อมูล (input)

ส่วนกำหนดค่า/หรือคำนวณ (assignment or computation)

ส่วนแสดงผลลัพธ์ output

3. ส่วนคำอธิบายโปรแกรม (comment lines) ส่วนนี้ใช้ในการอธิบายโปรแกรมอาจพิมพ์ข้อความ สูตร หรืออื่น ๆ ที่เกี่ยวข้องกับโปรแกรมโดยอธิบายส่วนต่าง ๆ ของโปรแกรม ว่าแต่ละส่วนทำหน้าที่อะไร ตัวแปรแต่ละตัวใช้แทนค่าอะไร ส่วนของโปรแกรมส่วนนี้จะมีหรือไม่มีก็ได้ ถ้าต้องเขียนให้เริ่มต้นด้วยเครื่องหมาย */ตามด้วยข้อความที่ต้องการ แล้วปิดท้ายด้วย /*(เครื่องหมาย / และ *ต้องเขียนติดกันเสมอ)

2.5 Visual C# 2005

Visual C# 2005 เป็นเครื่องมือที่ช่วยในการเขียนโปรแกรมในภาษา C# ซึ่งภาษา C# เป็นภาษาโปรแกรมยุคใหม่ที่ถูกสร้างขึ้นมาสำหรับการพัฒนาซอฟต์แวร์ภายใต้เทคโนโลยี Microsoft .NET ซึ่งเป็นเทคโนโลยีที่ได้รับความนิยมสูงที่สุดในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Visual C# นั้นถูกแนะนำครั้งแรกพร้อมกับ Visual Studio .NET ในชื่อของ Visual C# .NET ซึ่งก็ได้รับการต้อนรับจากนักพัฒนาซอฟต์แวร์เป็นอย่างดี จนกระทั่งไมโครซอฟท์ได้พัฒนา Visual Studio 2005 ขึ้นมาทำให้ Visual C# ก็ได้รับการพัฒนาเป็น Visual C# 2005

2.5.1 การเขียนโปรแกรมแบบ Event Driven Programming

ในการเขียนโปรแกรมเพื่อสร้างแอพพลิเคชันเพื่อทำงานบน Windows นั้นจะมีวิธีคิดในการเขียนโปรแกรมที่แตกต่างจากรูปแบบเดิมที่เราคุ้นเคย หากเราต้องเขียนโปรแกรมด้วยภาษา C หรือ Pascal นั้น เราจะต้องมองขั้นตอนการทำงานให้ทะลุปรุโปร่งเสียก่อน แล้วค่อยเขียนโปรแกรมให้ครอบคลุมการทำงานเหล่านั้นให้หมด ซึ่งถือว่าใช้ได้ดีถ้าโปรแกรมนั้นมีขนาดเล็กหรือไม่ซับซ้อนมากนัก เราเรียกวิธีคิดนี้ว่า การเขียนโปรแกรมแบบ Structure Programming

แต่ถ้าโปรแกรมที่มีขนาดใหญ่ขึ้น ซับซ้อนขึ้น (เช่น โปรแกรม MS Word) หรือกลายเป็นระบบงานที่มีความหลากหลาย (เช่นโปรแกรมระบบบัญชี) หากใช้วิธีคิดแบบ Stucture Programming แล้วจะทำให้การเขียนโปรแกรมยุ่งยาก แก้ไขก็ยา โดยเฉพาะอย่างยิ่งเมื่อโปรแกรมนั้นทำงานบน Windows ดังนั้น ไมโครซอฟท์จึงมีวิธีคิดที่เรียกว่า Event Driven Programming ขึ้นมาสำหรับการเขียนโปรแกรมบน Windows

Event Driven Programming นั้นจะคิดว่า ถ้าเกิดเหตุการณ์ขึ้นมาแต่ละอย่างแล้วจะจัดการกับมันอย่างไร (โดยไม่สนใจว่าจะมีอะไรเกิดขึ้นกับโปรแกรมบ้าง)

2.5.2 รู้จักกับออบเจ็กต์

สิ่งที่ควรรู้เมื่อต้องเขียนโปรแกรม หรือพัฒนาแอพพลิเคชันเพื่อใช้งานบน Windows ก็คือเรื่องของออบเจ็กต์ (Object) ซึ่งจะสอดคล้องกับแนวคิดของ Event Driven Programming ซึ่งที่ผ่านมาเราจะเห็นว่าแอพพลิเคชันนั้นจะประกอบมาจากสิ่งต่าง ๆ เช่น ปุ่มกด,วินโดว์ ซึ่งเราเรียกแต่ละสิ่งที่ประกอบเป็นแอพพลิเคชันว่า ออบเจ็กต์ (Object)

2.5.3 พร็อพเพอร์ตี้ (Property) : คุณสมบัติของออบเจ็กต์

ออบเจ็กต์แต่ละชนิดย่อมมีคุณสมบัติประจำตังของมัน เช่น ออบเจ็กต์รถยนต์ก็จะมีคุณสมบัติ เช่น ยี่ห้อ,รุ่น,ขนาดของเครื่องยนต์,สี เป็นต้น ซึ่งเราเรียกคุณสมบตินี้ว่าพร็อพเพอร์ตี้

ในการใช้งาน Visual C# 2005 นั้นการกำหนดพร็อพเพอร์ตี้ให้กับออบเจ็กต์ชนิดต่างๆ จะสามารถทำได้ทั้งในช่วงของการออกแบบ และในขณะที่แอพพลิเคชันนั้นทำงานจะใช้การเขียนโปรแกรมกำกับการทำงานเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.5.4 คอนโทรลกับคอมโพเนนต์

ในการสร้างแอพพลิเคชันด้วย Visual C# 2005 นั้นมีออบเจ็กต์ให้เลือกใช้งานได้หลายตัว ออบเจ็กต์ที่สามารถมองเห็นได้เราจะเรียกว่า คอนโทรล (Control) เช่น คอนโทรล Button,คอนโทรล Textbox เป็นต้น และยังมีออบเจ็กต์อีกประเภทหนึ่งที่มองไม่เห็นเวลาที่แอพพลิเคชันทำงาน (ทำงานอยู่หลังฉาก) เราจะเรียกว่า คอมโพเนนต์ (Component)

เราสามารถแยกชนิดของคอนโทรลกับคอมโพเนนต์ได้อย่างชัดเจน โดยที่จะมีการแบ่งประเภทใน Toolbox เอาไว้แล้ว

2.5.5 การกำหนดค่าใน Property Window

ในการกำหนดค่าให้กับพร็อพเพอร์ตี้ต่างๆ นั้นเราจะกำหนดผ่านหน้าต่าง Property Window ซึ่งจะสามารถกำหนดได้ 3 รูปแบบ

1. การกำหนดค่าอย่างอิสระ : เราสามารถป้อนค่าใน Property Window ได้อย่างอิสระ เช่น พร็อพเพอร์ตี้ Text,พร็อพเพอร์ตี้ Name เป็นต้น
2. การกำหนดค่าจากรายการตัวเลือก : เป็นการกำหนดค่าโดยเลือกจากตัวเลือกในรายการที่มีให้ของ Prooerty Window เช่น BorderStyle, BackColor เป็นต้น
3. การกำหนดค่าจากไดอะล็อกบ็อกซ์ : เป็นการกำหนดค่าโดยเลือกจากตัวเลือกที่มีให้ในไดอะล็อกบ็อกซ์ เช่น Font, PictureFile เป็นต้น

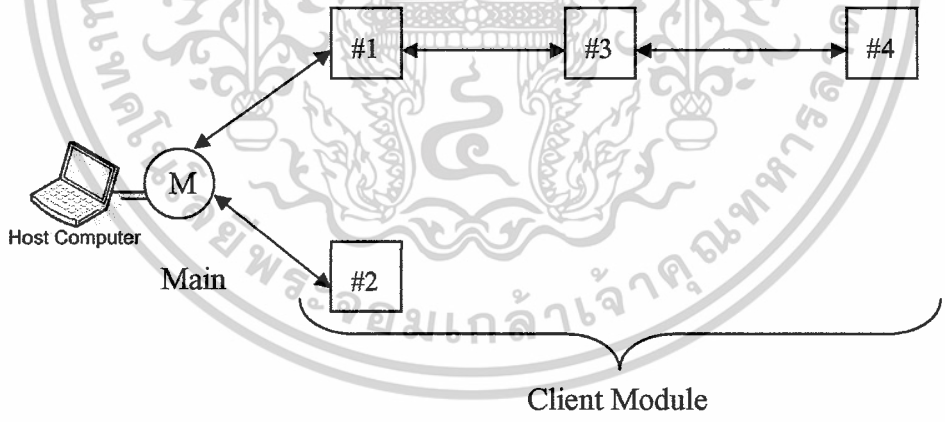
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104179

บทที่ 3

การออกแบบ

3.1 โครงสร้างการทำงานโดยรวม

เนื่องจากการเล็งเห็นถึงประโยชน์ของวิทยุที่สามารถสร้างระบบตรวจสอบวัดค่าระยะไกลแบบใหม่ ซึ่งทำให้สามารถติดตั้งเครื่องวัดของเครื่องลูกข่าย(Client) ได้ครอบคลุมพื้นที่มากขึ้นโดยไม่ต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุ โดยหลักการคือ ให้เครื่องลูกข่าย(Client) ซึ่งมีอยู่หลายจุดนี้ มีการเชื่อมต่อกันคล้ายเครือข่าย(Network) โปรแกรมจะควบคุมการติดต่อกันอัตโนมัติ ตั้งแต่เริ่มส่งคำสั่งจากเครื่องแม่ข่าย(Main) ไปสู่เครื่องลูกข่าย(Client) ที่ต้องการติดต่อด้วยเป็นทอดๆ โดยใช้เครื่องลูกข่าย(Client)ตัว ที่อยู่ในพื้นที่กำลังส่งของเครื่องแม่ข่าย(Main) ทำหน้าที่เป็นตัวกลางในการรับและส่งข้อมูลไปยังเครื่องลูกข่าย(Client) ตัวอื่นๆ และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถจะใช้ซอฟต์แวร์ที่ได้ออกแบบไว้บนคอมพิวเตอร์ทั่วไปเพื่อเข้าไปดูข้อมูลได้ทุกจุดทุกเวลาตามความต้องการ



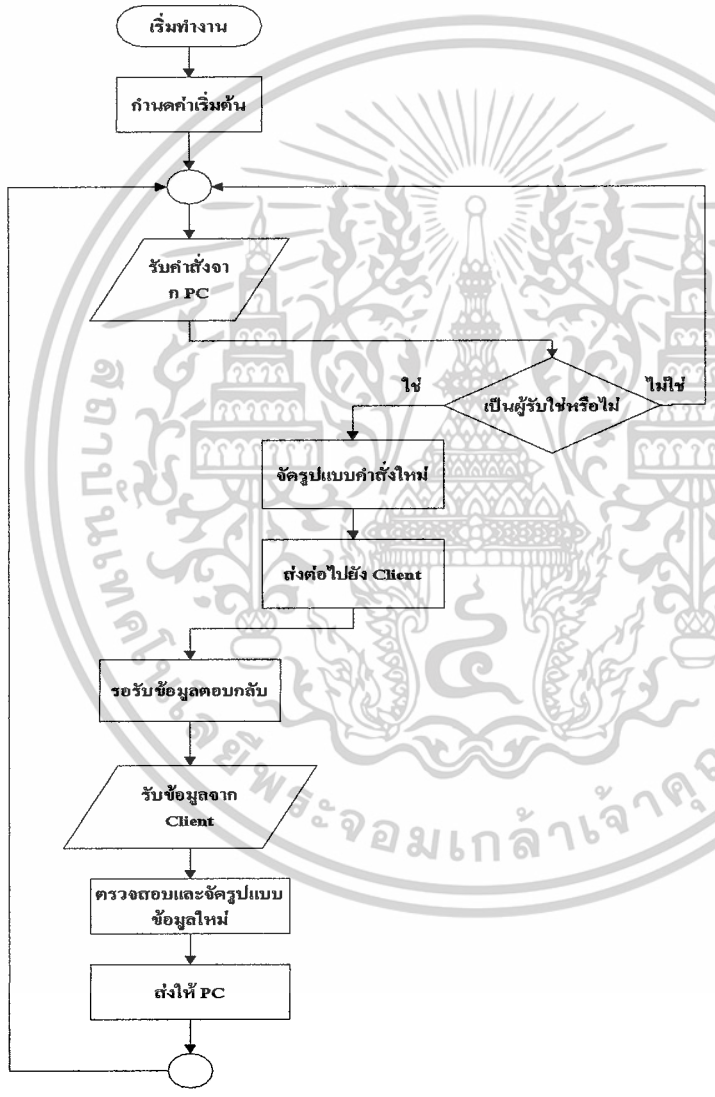
รูปที่ 3.1 ตัวอย่างรูปแบบการเชื่อมต่อ

สำหรับโครงงานนี้ ในส่วนของการติดต่อรับส่งข้อมูลกันระหว่างเครื่องแม่ข่าย(Main) กับเครื่องลูกข่าย (Client) หรือแม้แต่เครื่องลูกข่าย (Client) ด้วยกันเอง จะใช้สายสัญญาณจำลองแทนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

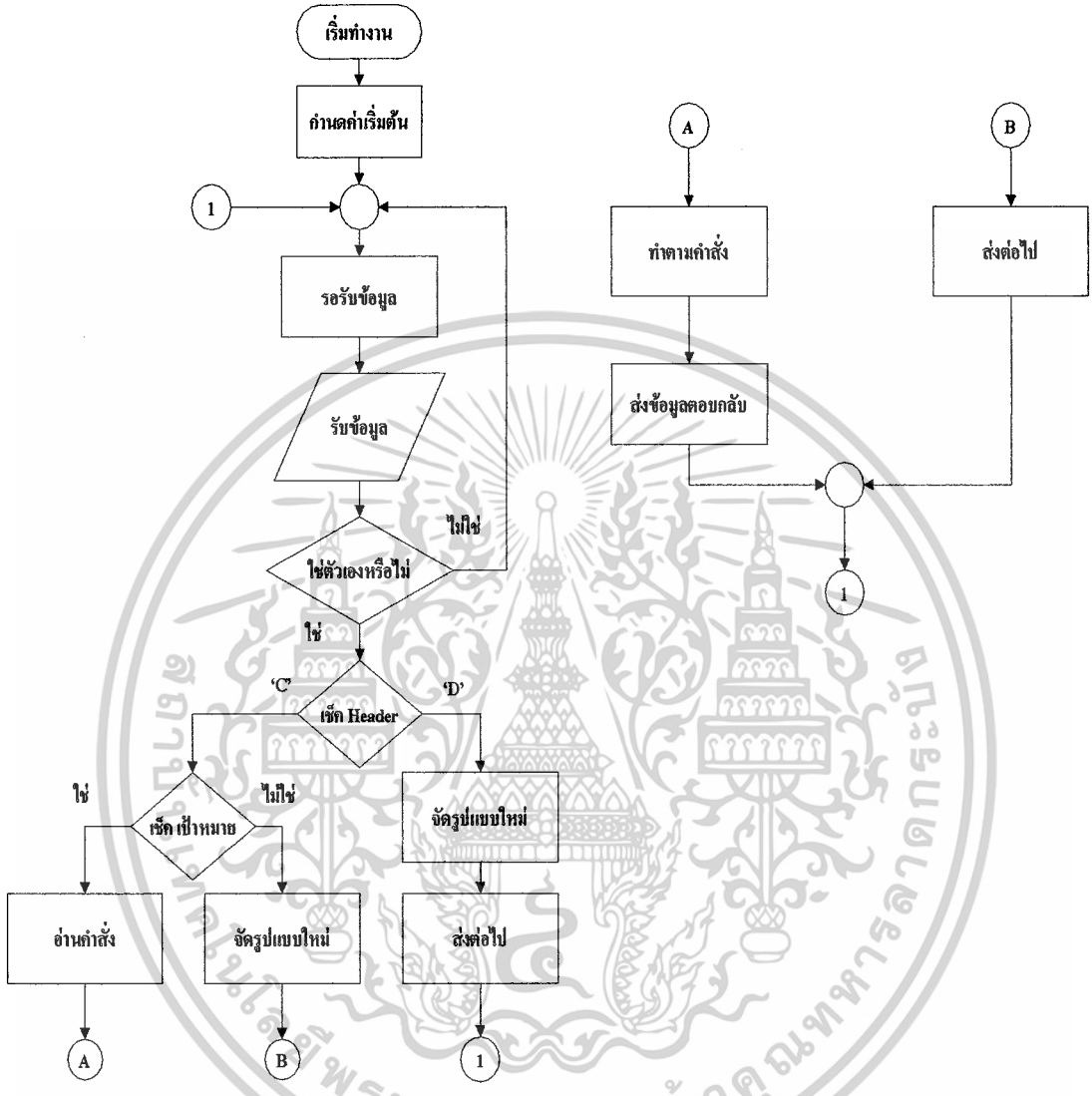
รับส่งข้อมูลกันผ่านคลื่นวิทยุ โดยเชื่อมต่อผ่านพอร์ตอนุกรมของตัวไมโครคอนโทรลเลอร์ เพื่อแสดงให้เห็นว่าซอฟต์แวร์ใช้งานได้จริง

3.2 ไดอะแกรม ซอฟแวร์ ควบคุมการรับส่งข้อมูล ของตัว Main และ Clients



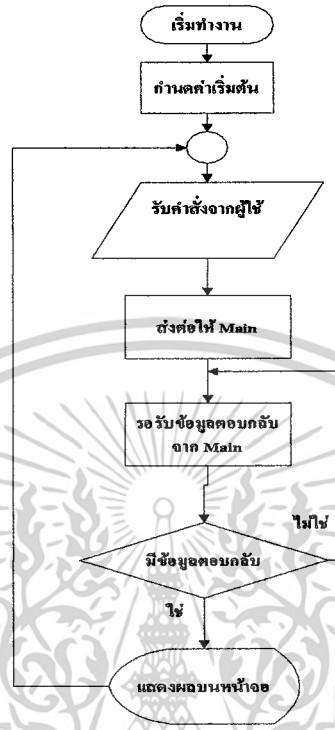
รูปที่ 3.2 โฟลว์ชาร์ทซอฟต์แวร์แสดงการควบคุมการรับส่งข้อมูลของตัว Main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 โฟลว์ชาร์ทซอฟต์แวร์แสดงการควบคุมการรับส่งข้อมูลของตัว Clients

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 โฟลว์ชาร์ทการทำงานของซอฟต์แวร์ติดต่อใช้งานกับผู้ใช้

3.3 รูปแบบการทำงานของซอฟต์แวร์

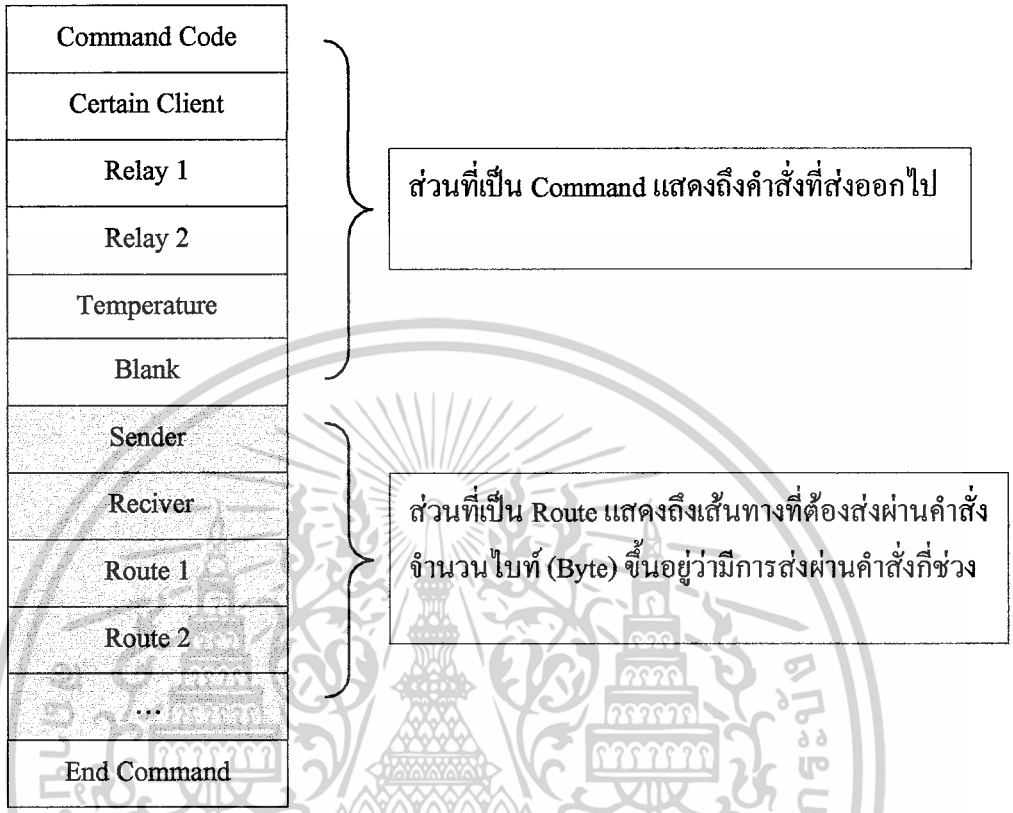
โปรโตคอล(Protocal) ในการรับส่งข้อมูลมี 2 ส่วน คือ Command Format เป็นโปรโตคอล (Protocal) ในการส่งคำสั่งออกไปเพื่อสั่งงานหรืออ่านข้อมูล และ Data Format เป็นโปรโตคอล (Protocal) ในการส่งข้อมูลกลับไปสู่คอมพิวเตอร์

3.3.1 Command Format

เป็นฟอร์แมทที่ใช้ในการส่งคำสั่งจาก ตัวแม่ข่าย(Main) ไปสู่แต่ละช่วงของตัวลูกข่าย(Clients) ที่เป็นตัวส่งผ่านไปจนถึงตัวตัวลูกข่าย(Clients) ที่เป็นตัวที่ถูกระบุให้ทำงานหรืออ่านข้อมูล เพื่อส่งกลับต่อไป

ฟอร์แมทคำสั่ง (Command Format) จะประกอบไปด้วยสองส่วนหลักๆ คือ ส่วนที่เป็นคำสั่ง (Command) และส่วนที่เป็นเส้นทาง(Route) ในการส่งคำสั่ง โดยทั้งสองส่วนจะเป็นไบท์(Byte)ที่เรียงกัน ซึ่งแต่ละไบท์(Byte) ก็จะมีความหมายของตัวเอง โดยมีจำนวนมากหรือน้อยขึ้นอยู่กับจำนวนของการส่งต่อข้อมูล ฟอร์แมทคำสั่ง (Command Format) สามารถแสดงได้ ดังรูปที่ 3.2 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 รูปแบบของ Command format

Command Code ไบท์(Byte) นี้จะใช้เก็บค่าที่บอกว่าเป็น คำสั่ง (Command) จะมีค่าเป็น 'C'

Certain Client ไบท์(Byte) นี้จะเก็บค่าที่บอกถึงหมายเลขของตัวลูก(Client) ตัวเป้าหมายที่จะให้ทำงานหรืออ่านค่าข้อมูลที่ต้องการ

Relay 1, Relay 2 สองไบท์(Byte) นี้ จะเป็นค่าที่ใช้สั่งงาน รีเลย์(Relay) ตัวที่หนึ่งและตัวที่สองตามลำดับ ให้เปิดหรือปิด โดยถ้าให้เปิดจะเป็น '1' และปิด จะเป็น '0'

Temperature และ Blank ไบท์(Byte)นี้จะเป็นค่าที่ใช้อ่านค่าอุณหภูมิ โดย Temperature จะมีค่าเป็น 'T' เมื่อต้องการอ่านค่าอุณหภูมิ และเป็น '0' เมื่อไม่ต้องการอ่านค่า ส่วน Blank จะเป็นค่าที่ว่างเอาไว้เพื่อใช้บรรจุค่าข้อมูลของอุณหภูมิที่จะส่งกลับ

Sender ไบท์(Byte) นี้จะเป็นค่าที่แสดงถึงผู้ที่ส่งคำสั่ง(Command) โดยจะเป็นหมายเลขของตัวลูกข่าย(Client) แต่ละตัว ส่วนตัวแม่ข่าย(Main) จะเป็น 'M' แต่ถ้าเป็นส่งคำสั่งระหว่าง คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Commputer) กับตัวแม่ข่าย(Main) ไบท์(Byte) นี้จะมีค่าเป็นหมายเลขพอร์ต(Port) ที่คอมพิวเตอร์ (Commputer) ใช้เชื่อมต่อกับตัวแม่ข่าย(Main)

Reciver ไบท์(Byte) นี้จะเป็นค่าที่บอกถึงผู้ ที่จะรับคำสั่ง โดยจะเป็นหมายเลขของตัวลูกข่าย (Client) แต่ละตัว ส่วนตัวแม่ข่าย(Main) จะเป็น 'M'

Route 1, Route 2, ... ตั้งแต่ไบท์(Byte) นี้เป็นต้นไปจะเป็นค่าที่บอกถึงเส้นทางว่าจะต้องส่งผ่านตัวลูก(Client) ตัวใดบ้างเพื่อจะไปถึงตัวลูก(Client) ที่ต้องการ(Certain Client) โดยตัวสุดท้ายของส่วนนี้จะเป็นหมายเลขตัวลูก(Client) ที่เป็นเป้าหมาย (Certain Client)

End Command เป็นไบท์(Byte) ที่ใส่ค่า 'E' แสดงถึงการจบของรหัสคำสั่ง

ในการส่ง Command Format นี้ไปในแต่ละช่วงของการส่งต่อ ตัวลูกข่าย(Client) ที่ต้องส่งคำสั่งต่อจะต้องมีการทำการปรับ Command Format ส่วนของเส้นทาง(Route) นี้ไปทุกช่วงจนถึงตัวเป้าหมาย ที่ต้องการสั่งงานหรืออ่านข้อมูล

3.3.2 Data Format

เมื่อตัวตัวลูก(Client) ที่เป็นเป้าหมาย(Certain Client) ได้ทำการอ่านคำสั่งและทำงานตามคำสั่งที่ได้รับแล้ว ขั้นตอนต่อไปก็จะทำการส่งข้อมูลกลับสู่เครื่องคอมพิวเตอร์ของผู้ใช้ โดยส่งตามเส้นทางเดิมที่ได้รับคำสั่งนั้นมา

ฟอร์แมทข้อมูล (Data Format) จะประกอบไปด้วยสองส่วนเช่นกัน คือ ส่วนที่เป็นข้อมูล (Data) และส่วนที่เป็นเส้นทาง(Route) ในการส่งข้อมูล โดยทั้งสองส่วนจะเป็นไบท์(Byte)ที่เรียงกัน ซึ่งแต่ละไบท์(Byte) ก็จะมีความหมายของตัวเอง โดยมีจำนวนมากหรือน้อยขึ้นอยู่กับจำนวนของการส่งต่อข้อมูล ฟอร์แมทข้อมูล (Command Format) สามารถแสดงได้ ดังรูปที่ 3.2 ดังนี้

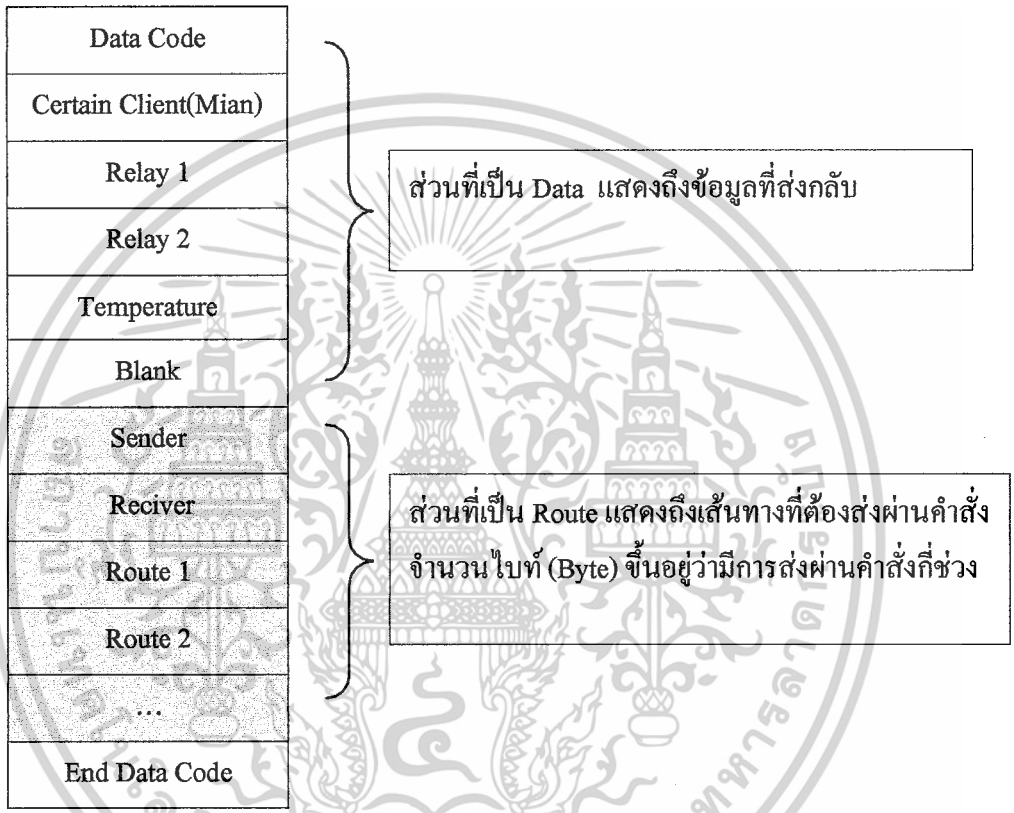
Data Code ไบท์(Byte)นี้จะใช้เก็บค่าที่บอกว่าเป็น ข้อมูล (Data) จะเป็นตัว 'D'

Certain Client(Main) ไบท์(Byte) นี้จะเก็บค่าที่บอกถึงหมายเลขของเป้าหมายที่จะให้รับข้อมูลที่ตอบกลับ ซึ่งจะกำหนดไว้ให้เป็น 'M' หมายถึงส่งข้อมูลไปยังตัวแม่(Main)

Relay 1, Relay 2 สองไบท์(Byte) นี้ จะเป็นค่าสถานะของ รีเลย์(Relay) ตัวที่หนึ่งและตัวที่สองตามลำดับ โดยสถานะเปิดจะเป็น '1' และปิด จะเป็น '0'

Temperature และ Blank สองไบท์(Byte)นี้จะเป็นค่าที่ใช้เก็บค่าอุณหภูมิที่ตอบกลับ โดยค่าอุณหภูมิจะมีอยู่สองหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดง Format ของ Data Format

Sender ไบท์(Byte) นี้จะเป็นค่าที่แสดงถึงผู้ที่ส่งข้อมูล(Data) โดยจะเป็นหมายเลขของตัวลูกข่าย(Client) แต่ละตัว ส่วนตัวแม่ข่าย(Main) จะเป็น 'M' แต่ถ้าเป็นการส่งระหว่าง คอมพิวเตอร์(Commputer) กับตัวแม่ข่าย(Main) ไบท์(Byte) นี้จะมีค่าเป็นหมายเลขพอร์ต(Port) ที่คอมพิวเตอร์(Commputer) ใช้เชื่อมต่อกับตัวแม่ข่าย(Main)

Reciver ไบท์(Byte) นี้จะเป็นค่าที่บอกถึงผู้ ที่จะรับข้อมูลต่อไป โดยจะเป็นหมายเลขของตัวลูกข่าย(Client) แต่ละตัว ส่วนตัวแม่ข่าย(Main) จะเป็น 'M'

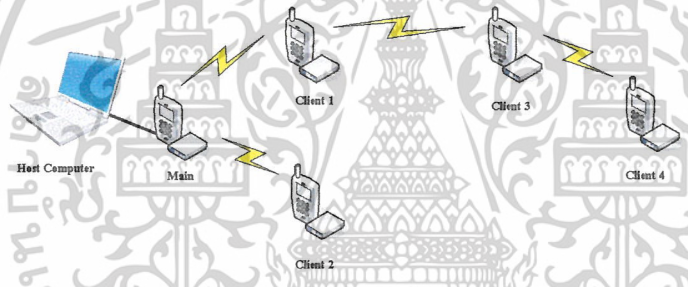
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Route 1, Route 2, ... ตั้งแต่ไบท์(Byte) นี้เป็นต้นไปจะเป็นค่าที่บอกถึงเส้นทางว่าจะต้องส่งผ่านตัวลูก(Client) ตัวใดบ้างเพื่อจะไปถึงตัวแม่(Main) โดยเส้นทางจะเป็นเส้นทางเดียวกันกับที่รับคำสั่งมา

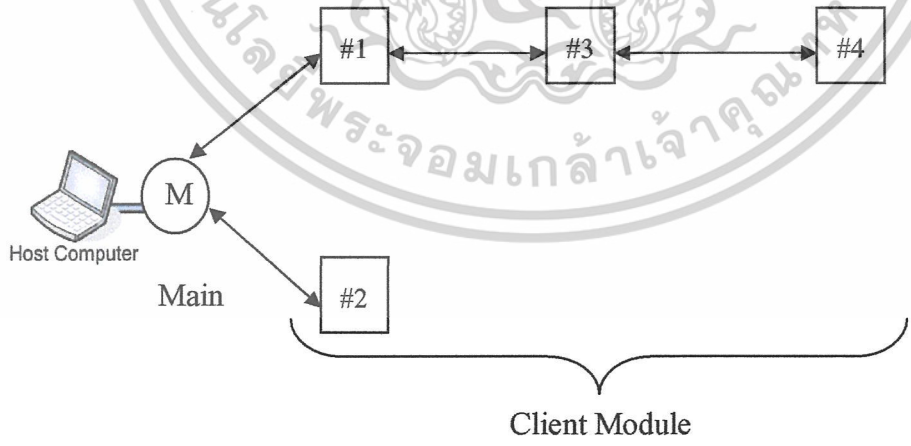
End Data Code เป็นไบท์(Byte) ที่ใส่ค่า 'E' แสดงถึงการจบของรหัสข้อมูล

3.4 ส่วนประกอบของฮาร์ดแวร์

สำหรับโครงงานชุดนี้ ในส่วนของตัวแม่ข่าย(Main)และลูกข่าย(Client) ที่ใช้ในการทดลองระบบ จะเป็นการจำลองขึ้นมา ซึ่งจะใช้วิธีการส่งข้อมูลผ่านสายสัญญาณแทนคลื่นวิทยุ โดยใช้หลักการที่ว่า ตัวลูกข่าย(Client) ที่อยู่ในย่านกำลังส่งของกันและกัน หรืออยู่ในย่านกำลังส่งของตัวแม่ข่าย(Main) ก็ตาม สามารถที่จะรับ-ส่งข้อมูลกันได้ ก็จะแทนด้วยสายสัญญาณเชื่อมต่อกัน ดังรูป



รูปที่ 3.7 การเชื่อมต่อกันโดยใช้คลื่นวิทยุในการรับส่งข้อมูล

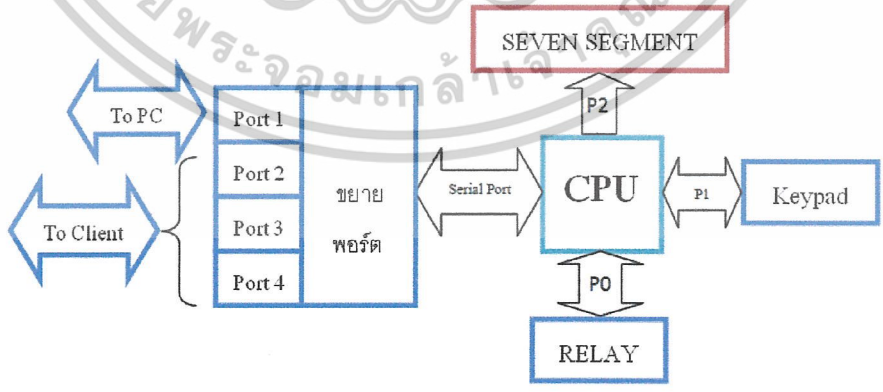


รูปที่ 3.8 การเชื่อมต่อกันโดยใช้สายสัญญาณแทนคลื่นวิทยุในการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างตัวแม่ข่าย(Main)และตัวลูกข่าย(Client)จำลองขึ้นมานั้น จะใช้ไมโครคอนโทรเลอร์(Microcontroler) เป็นหน่วยประมวลผล และใช้วิธีติดต่อสื่อสารกันผ่านพอร์ตอนุกรม(Serial Port) ของไมโครคอนโทรเลอร์(Microcontroler) โดยใช้สายสัญญาณแทนคลื่นวิทยุเป็นสื่อในการรับส่งข้อมูล แต่เนื่องจากพอร์ตอนุกรม(Serial Port) ของไมโครคอนโทรเลอร์(Microcontroler) มีเพียงพอร์ต(Port)เดียว รูปแบบการเชื่อมต่อกันจึงได้เพียงช่องทางเดียว ดังนั้นเพื่อให้มีรูปแบบการเชื่อมต่อที่ใกล้เคียงกับการใช้คลื่นวิทยุจริง จึงได้เพิ่มในส่วนของวงจรขยายพอร์ต(Port) เป็นการเพิ่มจำนวนพอร์ต(Port) ขึ้นมา และในส่วนของตัวลูกข่าย(Client) จะมีการจำลองตัวตรวจวัดอุณหภูมิ(Temperature Sensor) ขึ้นมา ใช้การกดตั้งค่าอุณหภูมิด้วยปุ่มกด(Keypad)และแสดงผลด้วย ตัวแสดงผลแบบเจ็ดส่วน(Seven Segment) และนอกจากนี้ยังมี รีเลย์(Relay) อีกสองตัว คือ Relay 1 และ Relay2 โดยมีโครงสร้างดังรูป

รูปที่ 3.9 ส่วนประกอบของฮาร์ดแวร์ชุดแม่ข่าย(Main)



รูปที่ 3.10 ส่วนประกอบของฮาร์ดแวร์ชุดลูกข่าย(Client)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 การทดลอง

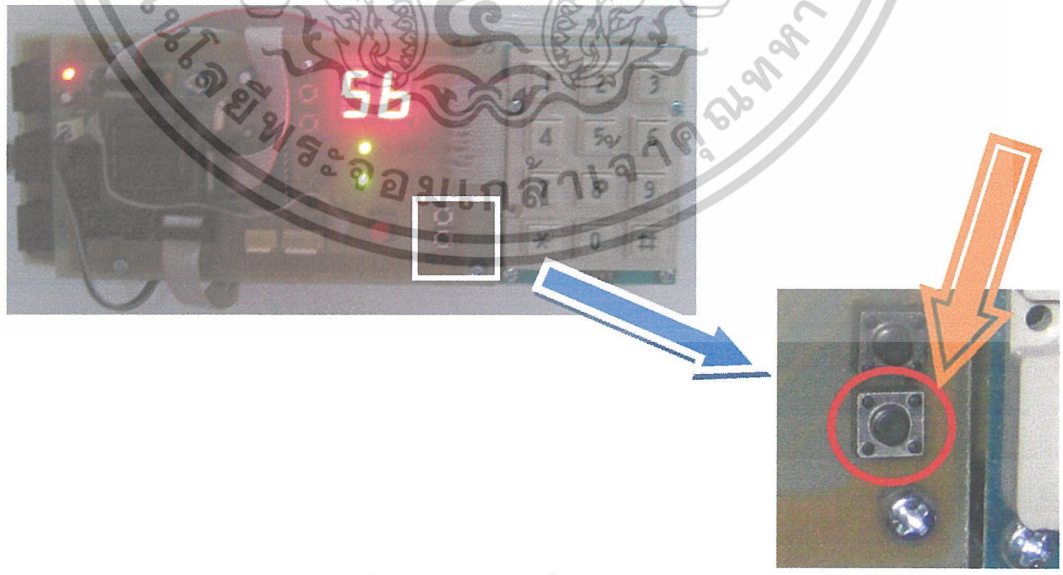
โปรแกรมชุดนี้จะเป็นตัวควบคุมการติดต่อ การสั่งงานและอ่านค่าจากเซ็นเซอร์โหนด โดยโครงานนี้เราใช้ไมโครคอนโทรลเลอร์เป็นตัวจำลองแทนเซ็นเซอร์โหนดที่เรียกว่า ตัวลูกข่าย(Client) โดยโปรแกรมจะควบคุมการติดต่อกันอัตโนมัติ ตั้งแต่เริ่มส่งคำสั่งจากตัวแม่ข่าย(Main) ไปสู่ตัวลูกข่าย (Client) ที่ต้องการติดต่อด้วยเป็นทอดๆ โดยใช้ตัวลูกข่าย (Client) ตัวที่ติดต่อกับตัวแม่ข่าย (Main) ได้ทำหน้าที่เป็นตัวกลางในการรับและส่งต่อข้อมูลไปยังตัวลูกข่าย (Client) เป้าหมาย

4.2 การตั้งค่าก่อนใช้งาน

เนื่องจากตัวลูกข่าย(Client) ที่ใช้ในการทดสอบระบบเป็นการจำลองขึ้นมาจากของจริง จึงต้องมีการกำหนดค่าเบื้องต้นก่อนเพื่อให้สอดคล้องกับรูปแบบการใช้งานจริง คือการตั้งค่าอุณหภูมิให้กับเครื่องลูกข่าย ซึ่งในที่นี้การตั้งค่าอุณหภูมิได้กำหนดตั้งได้เพียงสองหลักเท่านั้น

4.2.1 การตั้งค่าอุณหภูมิ

1.กดปุ่มตั้งค่าดังรูป หนึ่งครั้ง เพื่อให้สามารถกดแป้นกด(Keypad) ได้



รูปที่ 4.1 ปุ่มกดเพื่อตั้งค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กดค่าอุณหภูมิที่แป้นกด(Keypad) ตามต้องการ โดยสามารถกดได้เรื่อยๆ ในการกดครั้งแรกจะแสดงค่าที่กดใน Sevensegment ตำแหน่งหลักหน่วย และเมื่อกดครั้งต่อไปตัวเลขจะเลื่อนไปทางซ้ายเป็นหลักสิบ และตัวที่กดเข้าไปใหม่จะแสดงเป็นหลักหน่วยแทน และหากมีการกดอีกก็จะวนไปเรื่อยๆ จนกว่าจะกดยืนยัน

รูปที่ 4.2 แป้นกด(Keypad) ที่ใช้กดตั้งค่าอุณหภูมิ

3. หลังจากได้อุณหภูมิที่ต้องการแล้ว จะต้องกดยืนยันเพื่อบันทึกค่าอุณหภูมิเอาไว้ โดยการกดปุ่ม '*' หรือ ปุ่ม '#' ที่แป้นกด(Keypad) ปุ่มใดปุ่มหนึ่งก็ได้ จากนั้น Sevensegment จะกะพริบหนึ่งครั้งและแสดงค่าสุดท้ายที่เรากดเอาไว้

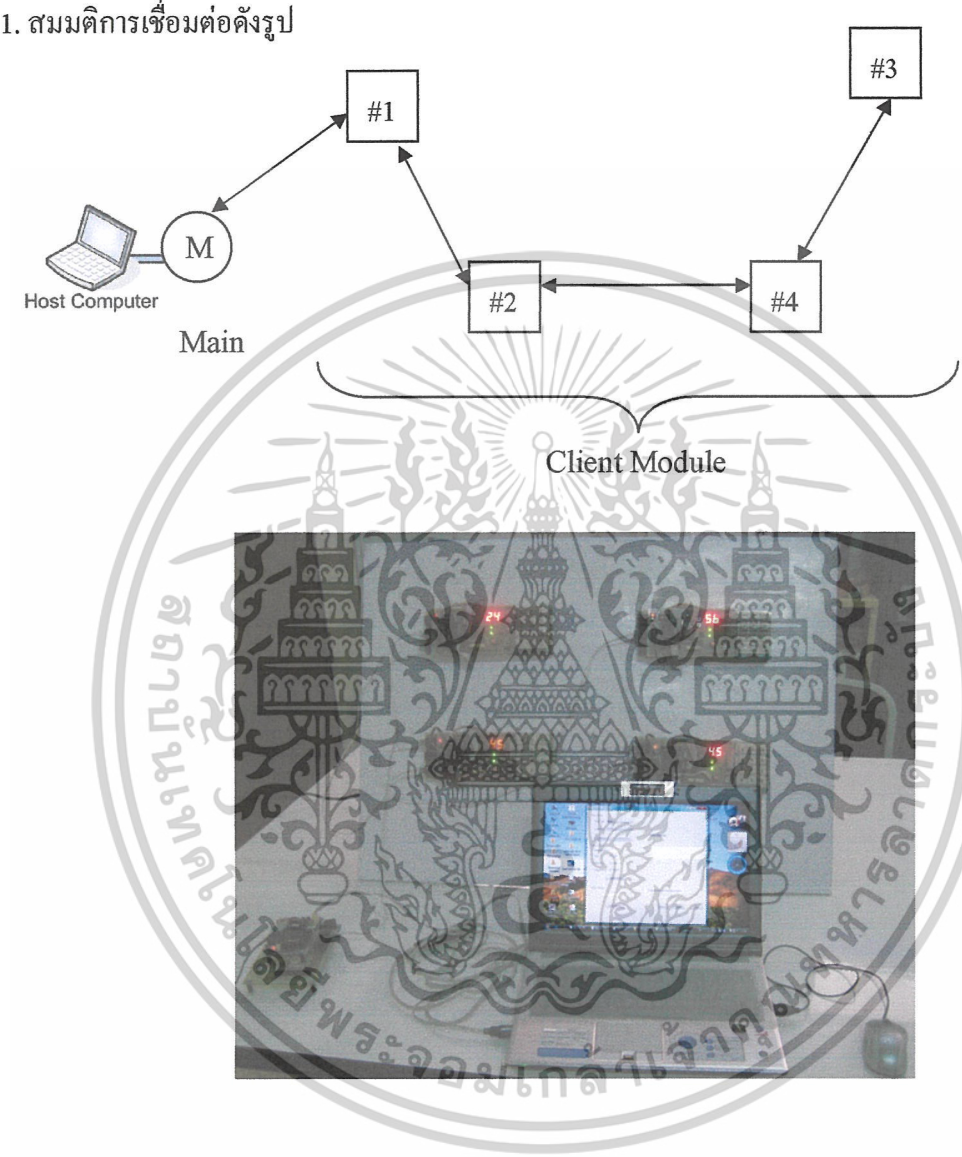
รูปที่ 4.3 ค่าอุณหภูมิที่ตั้งไว้

หากต้องการเปลี่ยนแปลงค่าอุณหภูมิใหม่ให้ทำตามขั้นตอนที่ 1 – 3 ในข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ขั้นตอนการทดลอง

1. สมมติการเชื่อมต่อดังรูป



รูปที่ 4.4 รูปแบบการเชื่อต่อ

จากรูปเป็นการสมมุติการเชื่อมต่อขึ้นมาโดยให้ตัวลูกข่าย(Client) มีการเชื่อมต่อกันเป็นทอดๆ โดยเริ่มจาก Main, Client#1, Client#2, Client#4 และ Client#3 เนื่องจากมีการเพิ่มจำนวนพอร์ต(Port) เพื่อให้สามารถเชื่อมต่อกันได้หลากหลายรูปแบบ ฉะนั้นจึงมีการกำหนดพอร์ต(Port) ในการติดต่อรับส่งข้อมูลกันให้ถูกต้อง ในการทดลองนี้มีการกำหนดพอร์ตดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Main สามารถเลือกพอร์ตอินพุท(Input Port)และเอาท์พุท(Output Port) ได้ทั้งสามพอร์ต คือ Port 2, Port3 และ Port 4

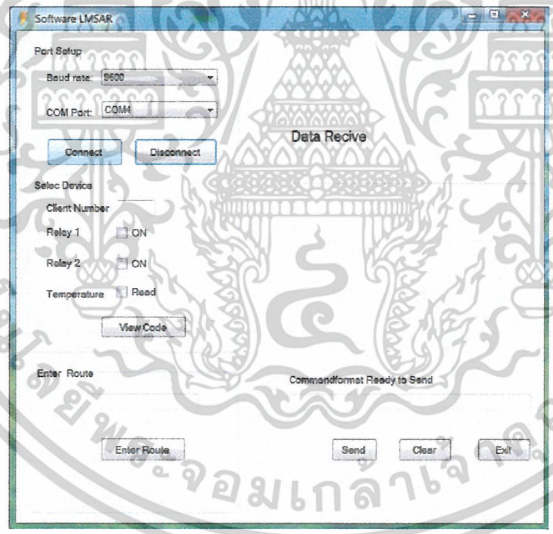
Client#1 กำหนดให้ พอร์ตอินพุท(Input Port) คือ Port 2 และ พอร์ตเอาท์พุท(Output Port) คือ Port 3

Client#2 กำหนดให้ พอร์ตอินพุท(Input Port) คือ Port 3 และ พอร์ตเอาท์พุท(Output Port) คือ Port 4

Client#4 กำหนดให้ พอร์ตอินพุท(Input Port) คือ Port 4 และ พอร์ตเอาท์พุท(Output Port) คือ Port 2

Client#3 กำหนดให้ พอร์ตอินพุท(Input Port) และ พอร์ตเอาท์พุท(Output Port) คือ Port 2

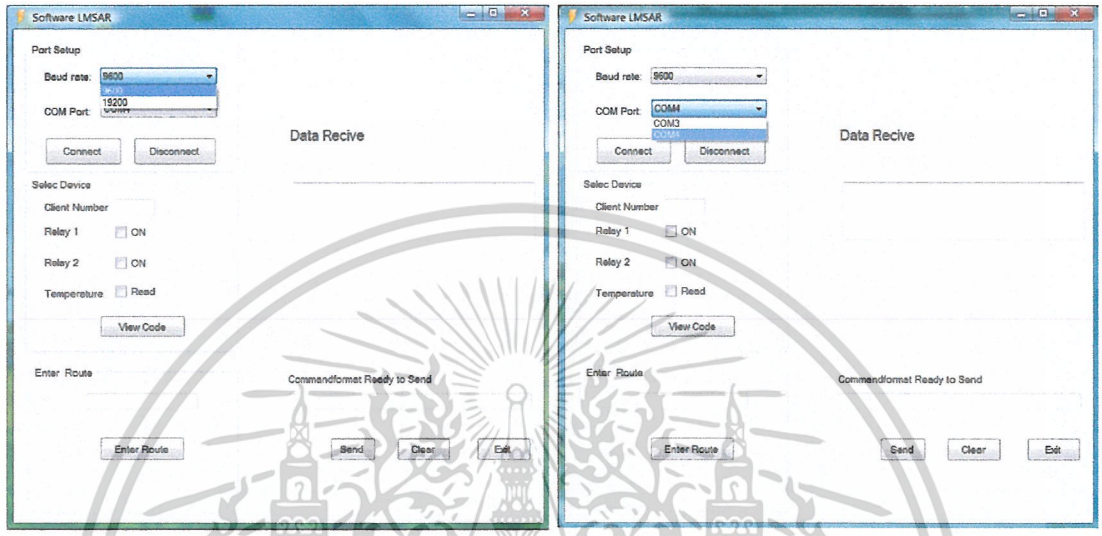
2. เปิดโปรแกรมควบคุมระบบตรวจสอบระยะไกลผ่านวิทยุสื่อสาร



รูปที่ 4.5 หน้าจอของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตั้งค่าการรับส่งข้อมูลโดยเลือก Baud rate ที่ 9600 จากนั้นเลือกหมายเลข COM Port



รูปที่ 4.6 การตั้งค่าในการรับส่งข้อมูล

จากนั้นกดปุ่ม Connect เพื่อติดต่อกับตัวแม่ข่าย(Main)

4. ทำการกำหนดค่าคำสั่ง โดยจะประกอบด้วย

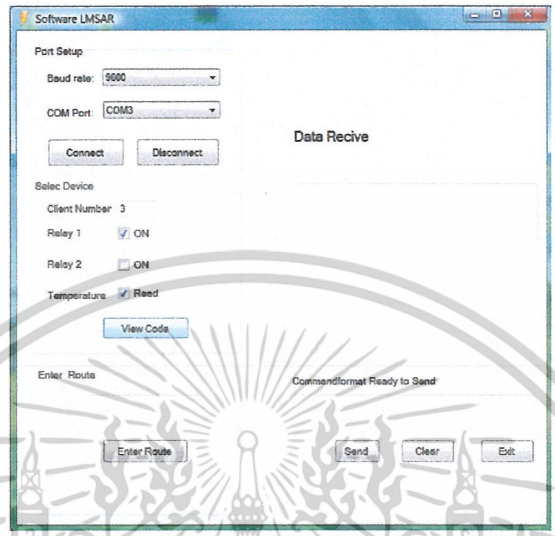
4.1 ใส่หมายเลขอุปกรณ์ตัวลูกข่าย(Client)ที่ต้องการสั่งงานหรืออ่านค่า ในช่อง Client Number

4.2 เลือกสั่งงานรีเลย์(Relay) ที่ต้องการ โดยถ้าเลือกในช่อง ON จะเป็นการสั่งให้รีเลย์(Relay) อยู่ในสถานะเปิด แต่ถ้าไม่ได้เลือกจะเป็นสถานะปิดอัตโนมัติ

4.3 กำหนดการอ่านค่าอุณหภูมิ โดยถ้าเลือกในช่อง Read จะเป็นการสั่งให้อ่านค่า อุณหภูมิ แต่ถ้าไม่ได้เลือกไว้ก็ไม่ต้องอ่านค่าอุณหภูมิกลับมา

ในการทดลองนี้ กำหนดให้อ่านค่าอุณหภูมิจาก Client # 3 ให้ Relay 1 ON และอ่านค่าอุณหภูมิกลับมา โดยกำหนดค่าดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 กำหนดคำสั่งที่จะส่ง

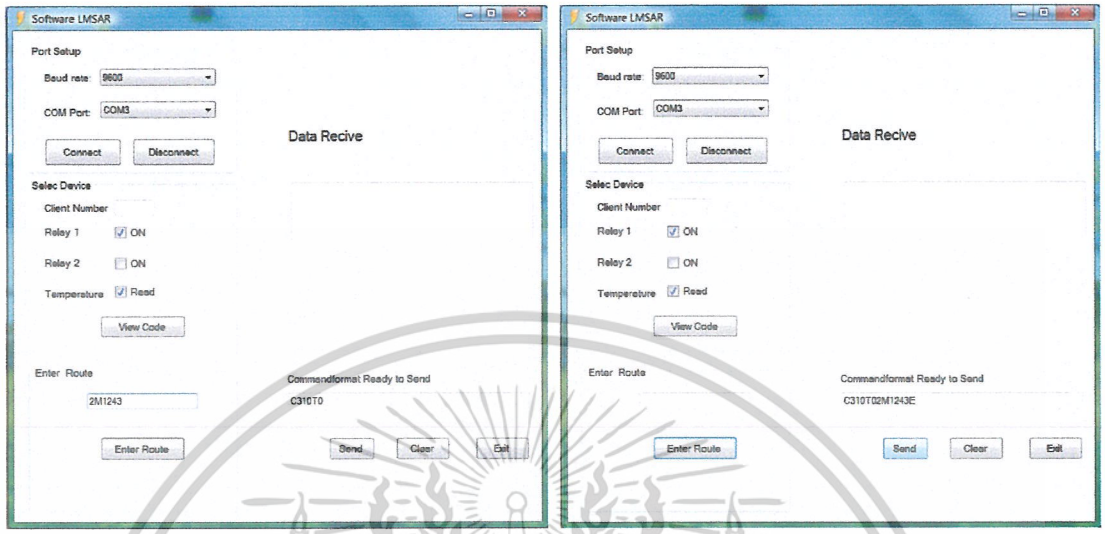
5. กดปุ่ม View Code โปแกรมจะแสดงคำสั่งในช่อง Command format จากนั้นกำหนดเส้นทางในการส่งต่อข้อมูลโดยตัวอย่างนี้สามารถกำหนดดังนี้ คือ 2M1243 โดยมีรายละเอียดคือ

2 หมายถึง พอร์ตของตัว Main ที่จะส่งข้อมูล

M หมายถึง ตัว Main

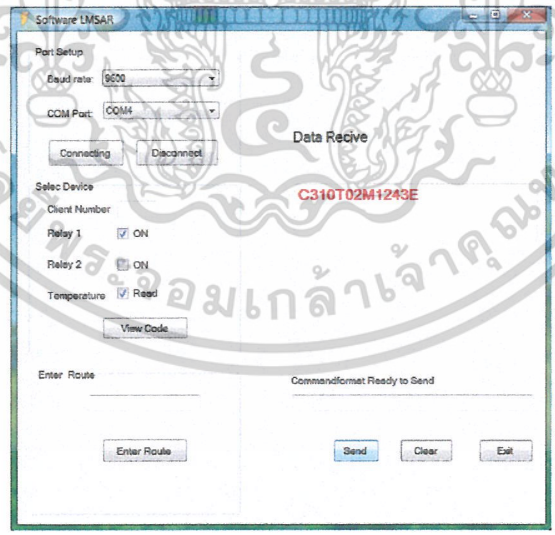
1243 หมายถึง เส้นทางที่เรากำหนดไว้ว่าผ่านตัว Client ตัวไหนบ้างเพื่อส่งไปยัง Client #3 การกำหนดเป็นหมายเลขของตัวลูกข่าย(Client) ลงไปในช่อง Enter Route ต่อจากนั้นกดปุ่ม Enter Route จะได้รหัสคำสั่งที่จะส่งออกมา และเมื่อกดปุ่ม Send ก็จะส่งคำสั่งไปยังตัวแม่ข่าย (Main)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 กำหนดเส้นทางที่จะส่ง

6. จากนั้นตัวโปรแกรมจะทำส่งคำสั่งไปยังตัวแม่ข่าย(Main) และทำการส่งต่อไปยังตัวลูกข่าย(Client) ตามเส้นทางที่ระบุไว้ในคำสั่ง



รูปที่ 4.9 ชุดคำสั่งที่ถูกส่งออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดลอง

4.4.1 ผลการอ่านค่าอุณหภูมิ

รูปที่ 4.10 ค่าอุณหภูมิที่อ่านค่าและตอบกลับมา

ค่าอุณหภูมิที่อ่านได้จาก Client# 3 มีค่าเท่ากับ 56 องศา ซึ่งตรงกับค่าที่ตั้งเอาไว้ แสดงผลที่หน้าจอคอมพิวเตอร์ของผู้ใช้งาน

4.4.2 ผลการสั่งงานรีเลย์ (Relay)

รูปที่ 4.11 Relay 1 ON

Relay 1 เปลี่ยนสถานะจาก OFF เป็น ON ตามคำสั่งที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 ผลการทดลอง สั่งงาน และ อ่านค่า Client ตัวอื่นๆ

	ค่าที่ตั้งเอาไว้			ผลการสั่งงานและอ่านค่าได้		
	Client#1	**Client#2**	**Client#4**	**Client#1**	**Client#2**	**Client#4**
Relay 1	OFF	OFF	OFF	ON	ON	OFF
Relay 2	OFF	OFF	OFF	OFF	ON	ON
Temperature	24	45	45	24	45	45

ตารางที่ 4.1 ผลการทดลอง สั่งงาน และ อ่านค่า Client ตัวอื่นๆ

หมายเหตุ Relay สถานะ ON จะเป็นไฟสีแดง

Relay สถานะ OFF จะเป็นไฟสีเขียว

4.5 สรุปผลการทดลอง

1. โปรแกรมสามารถติดต่อสื่อกันระหว่าง Computer กับ Main และ Client ได้
2. สามรถรับและส่งข้อมูลกันเป็นทอดๆ ได้
3. สามารถสั่งงาน และ อ่านค่า จาก Client ที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 บทสรุป

เพื่อให้ระบบการตรวจสอบข้อมูลระยะไกลผ่านวิทยุสื่อสารมีประสิทธิภาพมากขึ้น และสามารถที่จะเชื่อมต่อกันได้เป็นจำนวนมากๆเป็น โครงข่าย จึงได้จัดทำโครงงานชุดนี้ขึ้นมา ซึ่งเป็นโปรแกรมที่ใช้ในการควบคุมการติดต่อสื่อสารกันระหว่างเครื่องแม่ข่าย(Main) กับเครื่องลูกข่าย(Client) โดยจะใช้วิธีการส่งข้อมูลกันเป็นทอดๆ ช่วยให้สามารถติดตั้งเครื่องลูกข่ายในระยะทางที่ไกลๆได้ โดยที่ไม่ต้องเพิ่มกำลังส่งของเครื่องแม่ข่าย ในโครงงานนี้ได้จำลองเครื่องแม่ข่าย(Main)และครื่องลูกข่าย (Client) ขึ้นจากไมโครคอนโทรลเลอร์ เพื่อใช้ทดสอบระบบ ผลจากการดำเนินการ

ส่วนโปรแกรมด้านผู้ใช้งาน

ได้ออกแบบหน้าจอการใช้งานด้วยโปรแกรม Visual C# 2005 ให้สามารถรับคำสั่งจากผู้ใช้งาน และจัดส่งคำสั่งไปยังเครื่องลูกข่ายต่างที่ต้องการได้ ตลอดจนแสดงผลลัพธ์ที่ได้มายังผู้ใช้ได้ถูกต้อง

ส่วนโปรแกรมควบคุมการติดต่อสื่อสาร

พัฒนาโดยใช้ภาษาซีสำหรับไมโครคอนโทรลเลอร์ ด้วยโปรแกรม Keil C51 โปแกรมสามารถที่จะจัดการการับส่งข้อมูลกันในลักษณะที่เป็นทอดๆได้

ส่วนของฮาร์ดแวร์จำลองเป็นเซ็นเซอร์โหนด

สามารถป้อนค่าจากโหนดจำลอง และรับคำสั่งจากผู้ใช้งานแล้วส่งค่าที่ป้อนส่งกลับไปยังเส้นทางที่กำหนดไปแสดงผลหน้าจอให้ผู้ใช้งานได้ทราบค่าได้อย่างถูกต้อง

5.2 ปัญหาและอุปสรรคในการทำงาน

1. อุปกรณ์บางอย่างไม่สามารถหาได้ต้องใช้อย่างอื่นทดแทน เช่น Criystal ความถี่ 29.4912 หาไม่ได้ต้องใช้ Criystal ความถี่ 18.432 แทน แต่ก็สามารถใช้งานได้

2. ในขั้นตอนการรับและส่งข้อมูลต่อกันเป็นทอดๆ มักจะมี Code Command ที่เกินออกมาจาก Command จริงๆ ที่ใช้รับส่งข้อมูล ทำให้การทำงานผิดพลาดไป จึงแก้ไขโดยการเขียนโปรแกรมตัดส่วนที่เกินนี้ออกไป จึงสามารถทำงานได้อย่างถูกต้อง

3. อุปกรณ์บางอย่างแม้มีราคาถูกแต่ก็มีข้อเสียในเรื่องของเสียรภาพของตัวอุปกรณ์นั้น ทำให้ต้องคอยปรับแต่งค่าอยู่เสมอเช่น แป้นกดที่ใช้ตั้งค่าอุณหภูมิ มักจะมีค่าผิพลาดออกมาเสมอเวลากด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ชุด Power Supply ไม่เสถียร เมื่อเปิดนานไฟจะพอกับชุดทดลองตัวลูก Client จึงแก้ไขโดยการเปลี่ยน ไอซีเบอร์ 7805 ตัวใหม่เข้าไปแทน

5.3 ข้อเสนอแนะ

ปริญญานิพนธ์ซอฟร์แวร์สำหรับระบบตรวจสอบข้อมูลระยะไกลผ่านวิทยุสื่อสาร นี้สามารถใช้งานได้ถูกต้องตามขอบเขตแล้วก็ตาม แต่ยังมีบางส่วนที่สามารถเพิ่มเติมได้ ดังต่อนี้

1. ปรับปรุงกระบวนการให้สามารถตรวจสอบเส้นทางใหม่เมื่อติดต่อเส้นทางหลักไม่ได้ให้ติดต่อเส้นทางใหม่เพื่อส่งข้อมูลต่อไปได้

2. มีกระบวนการแจ้งความผิดปกติที่เกิดขึ้นกับตัว Client แต่ละตัว

3. จัดทำฐานข้อมูลเพื่อจัดเก็บค่าที่ได้รับเพื่อสามารถนำไปใช้ประโยชน์ได้หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

สัจจะ จรัสรุ่งระวีวร. **คู่มือ Visual C# 2005 ฉบับสมบูรณ์**. พิมพ์ครั้งที่1 นนทบุรี: ไอดีซีฯ, 2550

ขจร อนุดิตย์. **การเขียนโปรแกรมควบคุมไมโครคอลโทรลเลอร์ MCS-51 ด้วยภาษา C**. นนทบุรี : เอ-บุ๊ค ดิสทริบิวชั่น, 2550

ชาญกิจ อังสโวทัย, ภิญโญ พงษ์โพธิ์. **ไมโครคอนโทรเลอร์ตระกูล 8051**. [ออนไลน์] . เข้าถึงได้จาก : http://www.cpe.ku.ac.th/~yuen/204471/micro/mcs51/ (วันที่ค้นข้อมูล: 8 สิงหาคม 2551).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

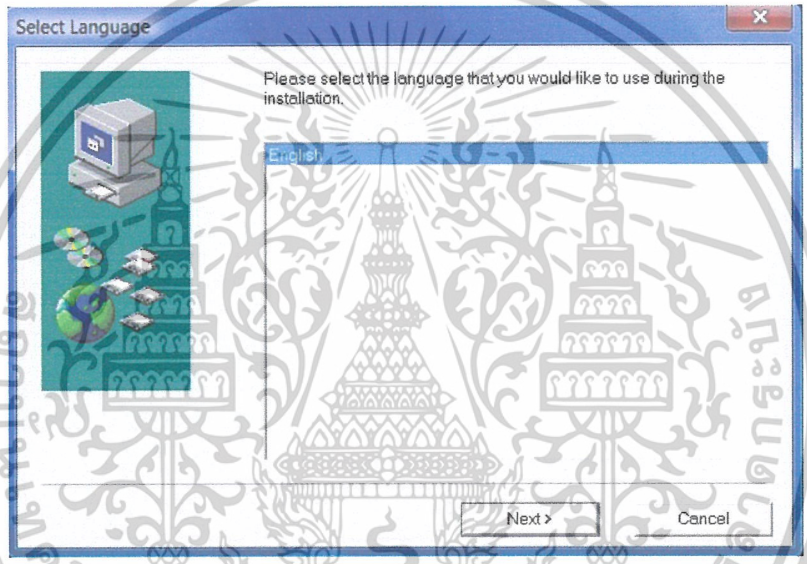
การติดตั้งโปรแกรมใช้งาน LMSAR

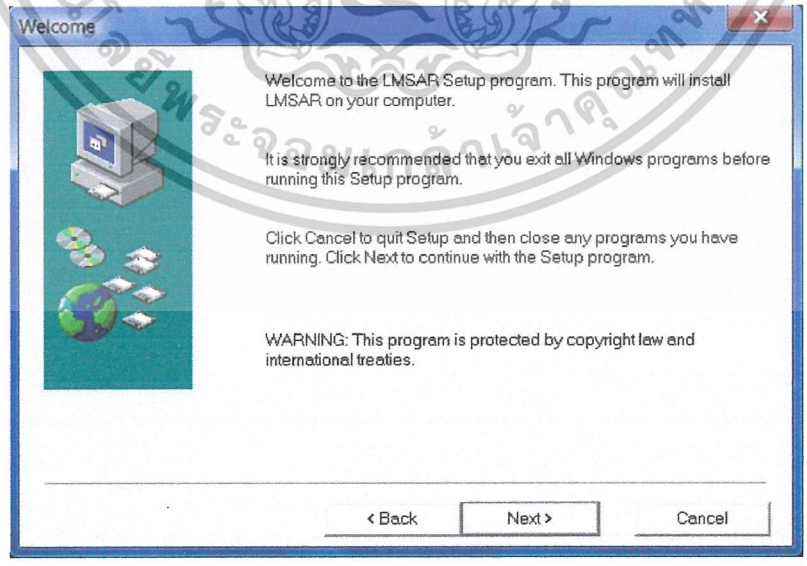
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการติดตั้งโปรแกรมใช้งาน LMSAR

1.ดับเบิ้ลคลิกไฟล์ Setup

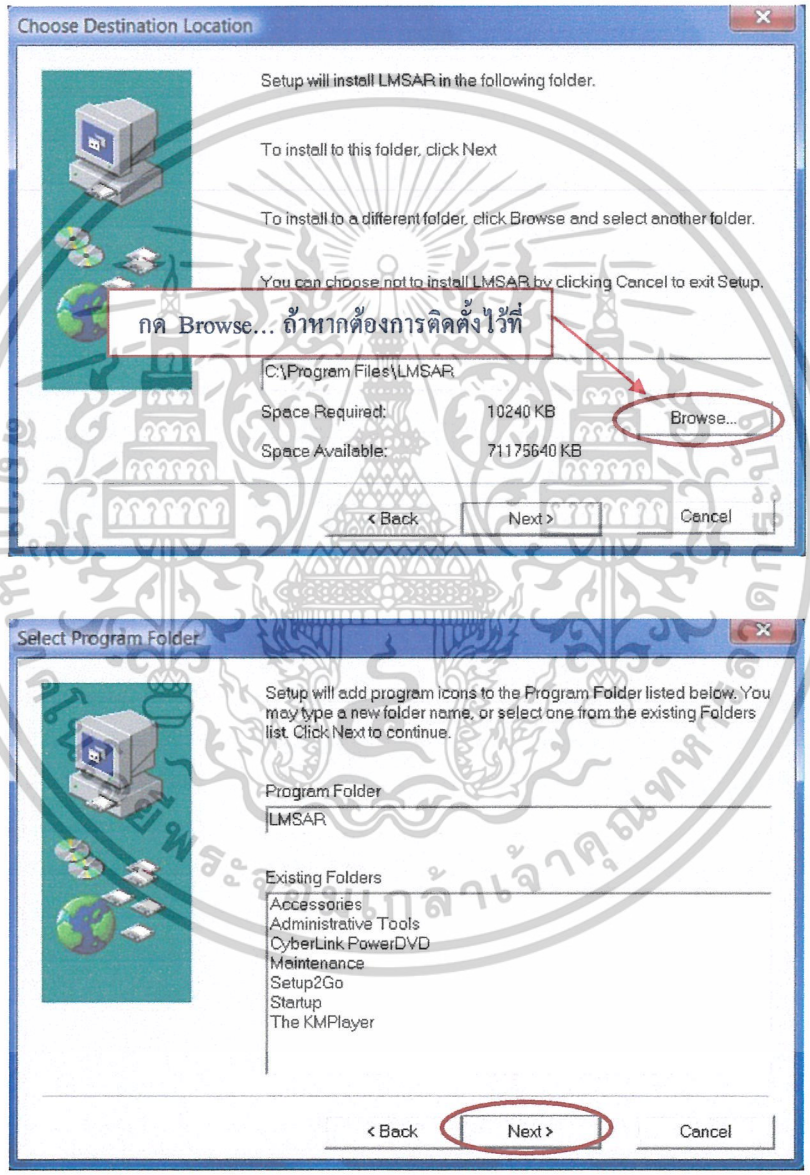
2.กด Next





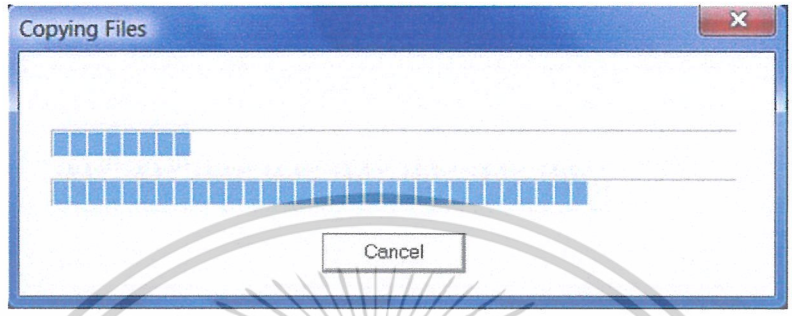
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.กด Brown เพื่อเลือกไดร์ที่ต้องการติดตั้งจากนั้นกด Next



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.รอการติดตั้ง



5.ติดตั้งโปรแกรมเสร็จแล้วกด Finish เป็นการเสร็จสิ้นขบวนการ Setup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมติดต่อผู้ใช้งาน (User Inter face Program)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Collections;

namespace rs232LMSAR
{
  public partial class Form1 : Form
  {
    // ตัวแปร rs232
    private SerialPort rs232 = new SerialPort("COM1", 9600, Parity.None, 8, StopBits.One);
    static string ss;
    static string rxTemp;
    public Form1()
    {
      InitializeComponent();
      // เพิ่มการดักเหตุการณ์รับข้อมูลจากพอร์ต RS232 โดยให้เรียกฟังก์ชัน rs232_DataRecieved
      rs232.DataReceived += new SerialDataReceivedEventHandler(rs232_DataReceived);
    }
    private void rs232_DataReceived(object sender, SerialDataReceivedEventArgs e)
    {
      ss += rs232.ReadExisting();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
   if (rs232.IsOpen) rs232.Close();
}
private void btSend_Click(object sender, EventArgs e)
{
   //rs232.Write(textBox2.Text+"\r\n");
   rs232.Write(txtSend.Text + "\r");
   txtSend.Clear();
}
private void btDisconnect_Click(object sender, EventArgs e)
{
   if (rs232.IsOpen) rs232.Close();
   btConnect.Text = "Connect";
   txtRecive.Clear();
   txtSend.Clear();
}
private void btConnect_Click(object sender, EventArgs e)
{
   btConnect.Text = "Connecting";
   // จะทำงานถ้ามีรายการพอร์ตติดตั้งในเครื่อง
   if (cmbPort.Items.Count > 0)
   {
      if (rs232.IsOpen)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            rs232.Close(); // ปิดพอร์ต
            timer1.Stop();
            txtRecive.Text = "";
            ss = "";
        }
        timer1.Start();
        rs232.BaudRate = int.Parse(cmbBaud.Text);
        rs232.DataBits = 8;
        rs232.Parity = (Parity)Enum.Parse(typeof(Parity), "None");
        rs232.StopBits = (StopBits)Enum.Parse(typeof(StopBits), "One");
        rs232.PortName = cmbPort.Text;
        rs232.Open();
    }
}
private void btClear_Click(object sender, EventArgs e)
{
    txtRecive.Clear();
    txtSend.Clear();
}
private void timer1_Tick(object sender, EventArgs e)
{
    txtRecive.Text = ss;
    rxTemp = ss;
    ss = "";
}
private void Form1_Load(object sender, EventArgs e)
{
    // ให้ตัวแปร s เก็บรายการชื่อพอร์ตอนุกรม
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
string[] s = SerialPort.GetPortNames();
int i = 0;
timer1.Stop();
// คอมโบสำหรับเลือกความเร็วในการสื่อสาร
cmbBaud.Items.Clear();
cmbBaud.Items.Add("9600");
cmbBaud.Items.Add("19200");
cmbBaud.SelectedIndex = 0;
// คอมโบสำหรับเลือกหมายเลขพอร์ต
cmbPort.Items.Clear();
foreach (string port in s) // เพิ่มรายชื่อพอร์ตใน comboBox2
{
   cmbPort.Items.Add(s[i]);
   i++;
}
if (i > 0)
{
   cmbPort.SelectedIndex = 0; // เลือกพอร์ตแรกที่พบเป็นค่าปกติ
}
else
{
   cmbPort.Enabled = false; // ไม่ให้เลือก เนื่องจากไม่พบรายการพอร์ตอนุกรมในเครื่อง
   btConnect.Enabled = false; // ไม่ให้คลิกปุ่มเชื่อมต่อ
}
}
private void bntComman_Click(object sender, EventArgs e)
{
   txtSend.Text = 'C' + txtClient.Text;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (chON.Checked)
{
   txtSend.Text += "1";
}
else
{
   txtSend.Text += "0";
}
if (chON2.Checked)
{
   txtSend.Text += "1";
}
else
{
   txtSend.Text += "0";
}
if (chRead.Checked)
{
   txtSend.Text += "T" + "0";
}
else
{
   txtSend.Text += "0" + "0";
}
txtClient.Clear();
}
private void btPart_Click(object sender, EventArgs e)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            txtSend.Text += txtPart.Text + "E";
            txtPart.Clear();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

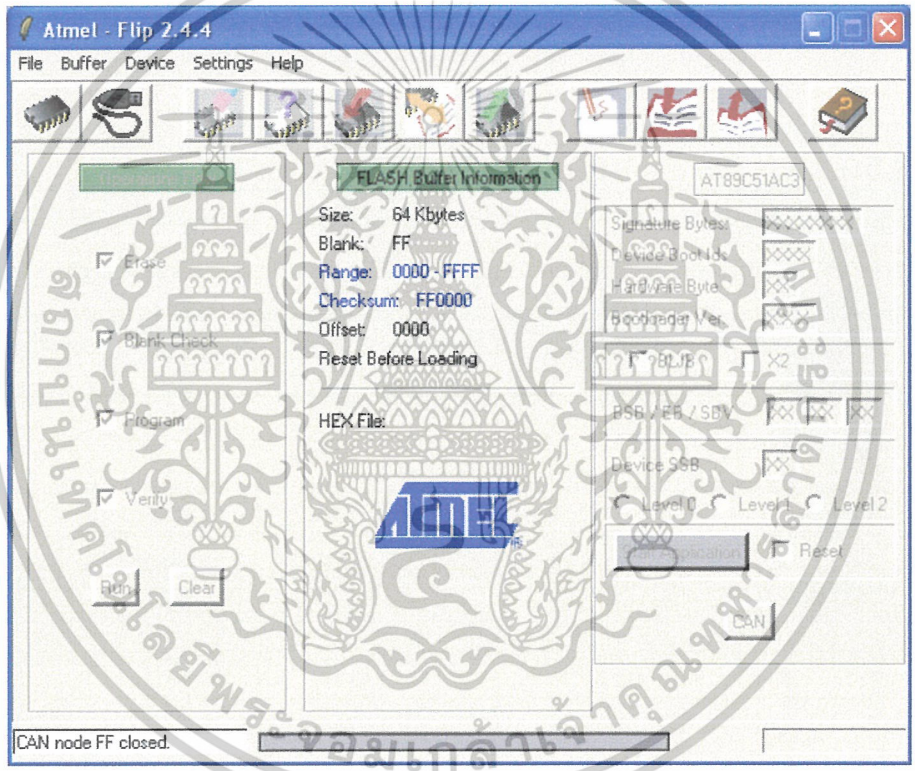
ภาคผนวก ข.

คู่มือการใช้งานโปรแกรม Keil C-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

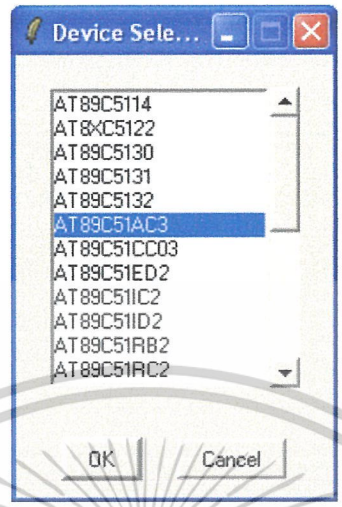
ลำดับขั้นตอนการ Download HEX File ด้วยโปรแกรม FLIP 2.4.4 แบบ Auto Download

1. ต่อสายสัญญาณ RS232 จาก Com Port ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว ET-DOWNLOAD แบบ 5 Pin ของบอร์ด
2. จ่ายไฟเลี้ยงวงจรให้บอร์ด ซึ่งจะสังเกตเห็น LED แสดงสถานะของ PWR สีแดงติดสว่างอยู่
3. สั่ง Run โปรแกรม FLIP V2.4.4 ซึ่งจะได้ผลดังรูป



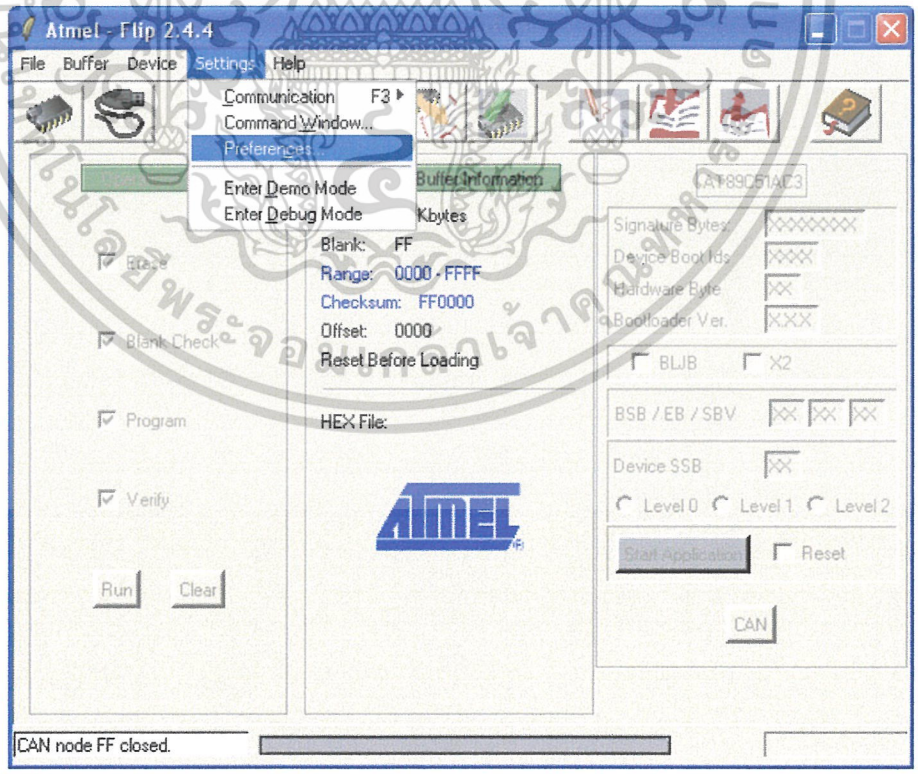
4. สั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ด โดยเลือก Device ⟶ Select ซึ่งต้องเลือกกำหนดให้ตรงกับที่ทำการติดตั้งไว้จริงๆในบอร์ดด้วย ดังตัวอย่าง (AT89C51AC3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

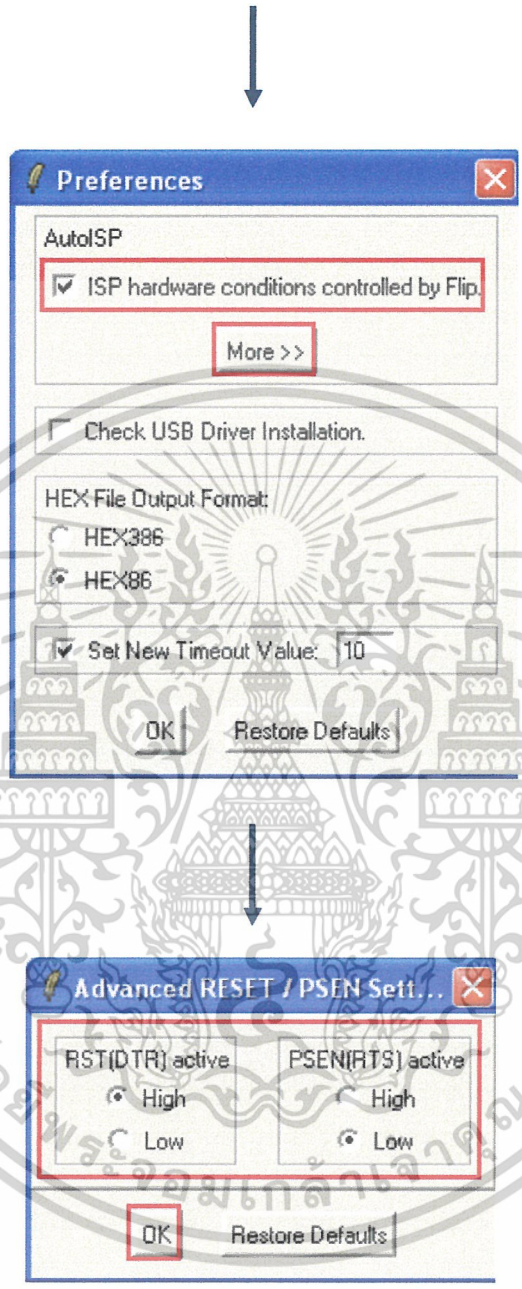


รูปแสดง การเลือกกำหนดเบอร์ CPU ของ ET-BASE51 AC3 (AT89C51AC3)

5. ทำการกำหนดค่า Option ของการสื่อสาร RS232 สำหรับใช้ Download แบบอัตโนมัติ โดยให้เลือกคลิกเมาส์ที่ Setting ⟶ Preferences... แล้วเลือกกำหนดค่า ดังรูป

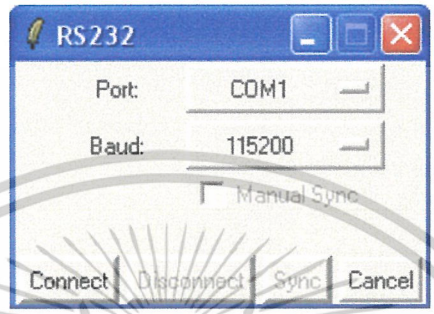


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

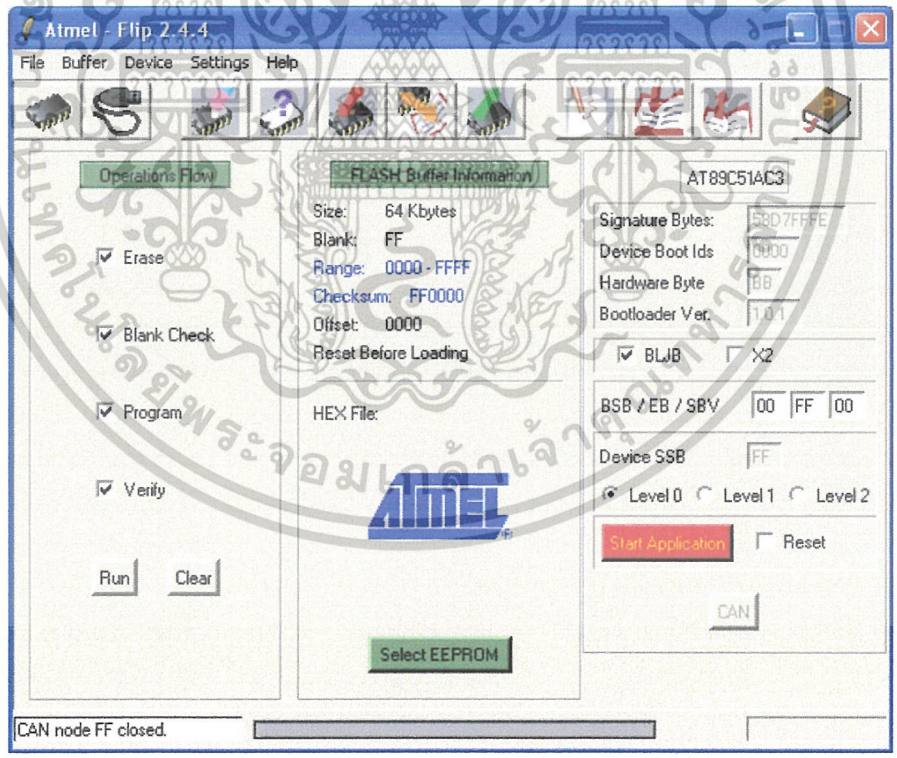


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. คลิกเมาส์ที่คำสั่ง Setting ⟶ Communication ⟶ RS232 จากนั้นเลือกกำหนด Comport ให้ตรงกับที่ต่อสายไว้จริง ดังรูป (ในตัวอย่างใช้ Com1)

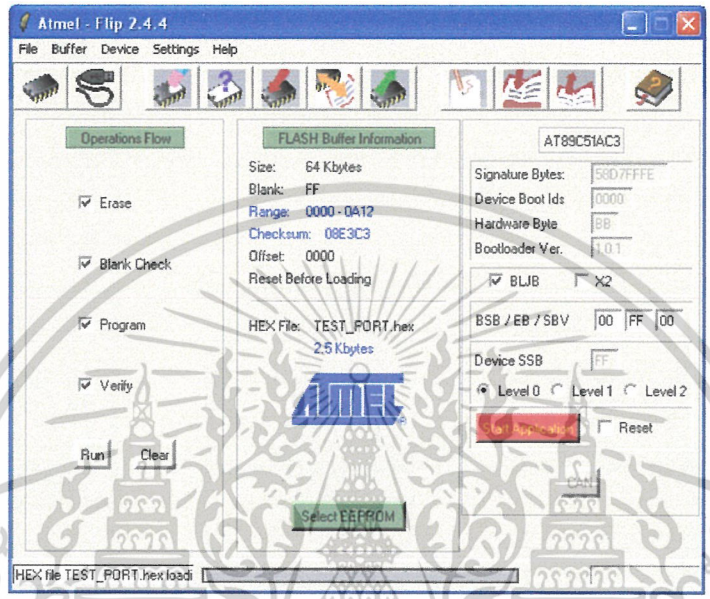


7. คลิกเมาส์ที่ปุ่ม Connect เพื่อทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode ซึ่งจะได้ผล ดังรูป

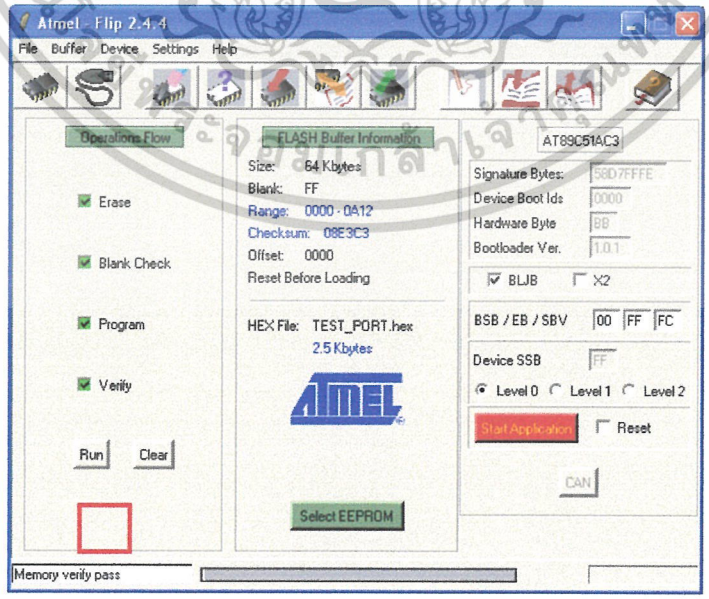


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. สั่งเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU มารอไว้ใน Buffer ของโปรแกรม FLIP โดยใช้คำสั่ง “File ⟶ Load Hex File…”

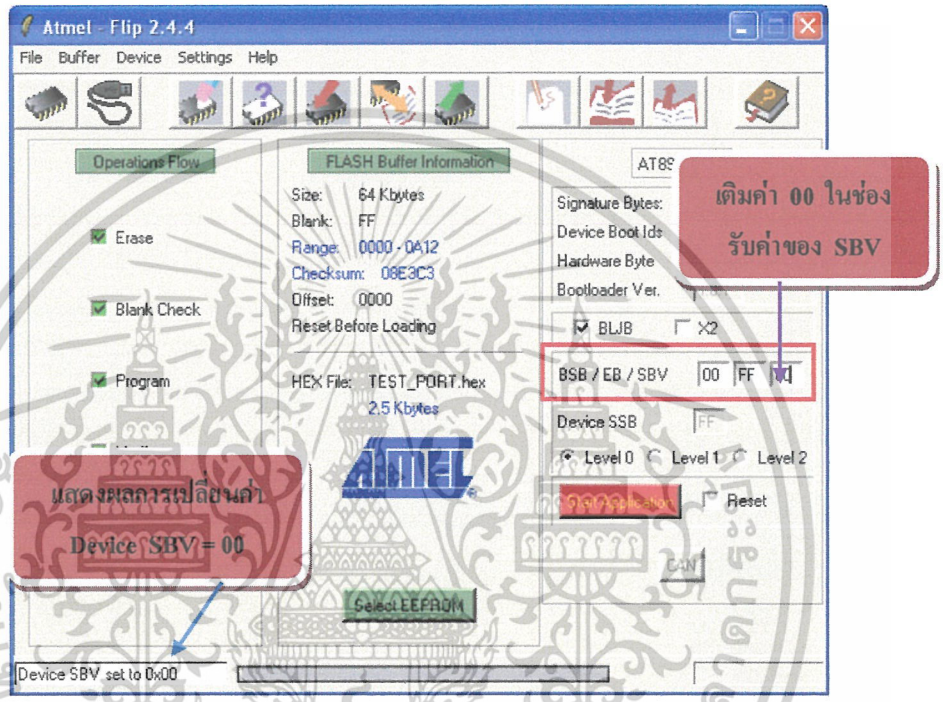


9. คลิกเมาส์ที่หน้าตัวเลือกคำสั่งใน Tab ของ Operation Flow ให้ครบทุกคำสั่ง ซึ่งได้แก่ Erase, Bank Check, Program, Verify จากนั้นคลิกเมาส์ที่ปุ่มคำสั่ง Run และรอจนการทำงานของโปรแกรมเสร็จเรียบร้อยดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ตรวจสอบค่า Device BSB และ SBV ว่ามีค่าเป็น 00 ทั้งหมดแล้วหรือยัง ซึ่งถ้ายังไม่เป็น 00 ให้ทำการแก้ไขค่าให้เป็น 00 โดยคลิกเมาส์ในช่องตัวเลขแล้วพิมพ์ค่า 00 แทนที่ลงไปทั้ง 2 ช่องดังรูป



11. ทำการคลิกเมาส์ที่ “Start Application” หรือกดสวิตช์ Reset ให้กับบอร์ดเพื่อให้บอร์ดเริ่มต้นทำงานตามโปรแกรมที่ได้ทำการ Download ไปให้ ซึ่งถ้าไม่เกิดความผิดพลาดใดๆ จะเห็น MCU เริ่มต้นทำงานทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโค้ดโปรแกรม Keil C-51 เครื่อง Main

Main Program

```
#include<reg51.h>
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>

int portIO;
sbit  Line_1 = P0^0;
sbit  Line_2 = P0^1;
sbit  Line_3 = P0^2;
//=========================Function=========================
void Line_En(int Ln_Num){        //Line Enable
     int x;
          x=Ln_Num;
          if(x==1){//==============Line No.1 ==================
                    Line_1 = 0;
                    Line_2 = 0;
                    Line_3 = 0;
                    }
               else if(x==2){//========Line No.2 ====================
                         Line_1 = 1;
                         Line_2 = 0;
                         Line_3 = 0;
                        }
               else if(x==3){//========Line No.3 ==================
                         Line_1 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Line_2 = 1;
                    Line_3 = 0;
                    }
          else if(x==4){//=====Line No.4====================
                     Line_1 = 1;
                     Line_2 = 1;
                     Line_3 = 0;
                    }
        }
void serial_init () //================ serial port setup ========================
        {
         SCON=0x50;
         TMOD=0x20;
         TH1=0xFB;//29.4912 use 0xF8 or 18.432 use 0xFB
         TR1=1;
         TI=1;
          RI=0;
        }
void delay(unsigned int z)          //======= Function Delay ======================
       {
         unsigned int count,y;
         for(count=0;count<z;count++)
         {
           for(y=0;y<100;y++);
         }
       }
 //=====MAINProgram=============================================
 void main ()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        char rxCOM[15];
        char sendREQ = 'r',rdyACK = 'Y';
        int i,j,k,SeqData=0,TF=1,com_l,Line=0,d,estop;


        serial_init( );   //Init serial port ========================
            START:
                        Line_En(1);
                        printf("\Enter Command ");
                        for(j=0;j<15;j++){
                                        rxCOM[j] = '\0';
                                        }
                        delay(100);
                        Line_En(1);
                    gets(rxCOM,sizeof(rxCOM)+1);       //Recive Command From PC ====
com_l=toint(rxCOM[0]);
                   delay(200);
//============== Fomat New Command ================================
if(rxCOM[7]=='M')
{
   portIO = toint(rxCOM[6]);
  for(d=6;d<15;d++)
  {
      if(rxCOM[d]=='E')
      {
        estop = d;
        for(k=estop;k<15;k++){
                        rxCOM[k] = '\0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                }
                        break;
        }
    else{
        rxCOM[d] = rxCOM[d+1];
        }
}
//====================== Send ============================
delay(100);
Line_En(portIO);//Port Send Out put ============
delay(100);
for(j=0;j<15;j++){
                while(!TI);
                TI=0;
                SBUF=rxCOM[j];
                delay(80);
                }
 delay(50);
}
else{
      goto START;
    }
 //=============== RECIVE ========================
Line_En(1);
delay(80);
Line_En(portIO);          //Port Send In put ===============
gets(rxCOM,sizeof(rxCOM)+1);   //Recive Command From Client ==================
com_l=toint(rxCOM[0]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(100);
//============ Fillter Data ========================
if(rxCOM[0]!='D'){
                    for(i=0;i<14;i++){
                                        rxCOM[i] = rxCOM[i+1];
                                      }
                   }
else;

  Line_En(1);
  delay(300);
  Line_En(1);
  printf("\n%c%c\n",rxCOM[4],rxCOM[5]);//========== Send To PC ==========
  delay(100);
  delay(100);
}
```

Keypad Program

```
#define SEGMENT P2
#define KEYPAD  P1
//==============Function==================
void Switch_lnit(void)//keypad init
{
        KEYPAD=0xFF;
}
unsigned char Keypad_scan(void)          // keypad scan...
{
        int in,out,row,col,pressed,temp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        while(1)
        {
          pressed=0;
          out=0x01;
          for(row=0;row<=3;row++)
          {
             KEYPAD=~out;       //P1 = ~out...
             in=~KEYPAD>>4;
             if(in !=0){
                        pressed=1;
                        break;
                        }
                        out<<=1;
          }       //end for...
    if(pressed){
                  col=0;
                  while(in>>(col+1)!=0){
                                        col++;
                                        }
                   temp =  row*3 + col+1;
                  if(temp == 11){
                                  return 0;
                                 }
                   else
                        return temp;
                 }
          }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void delay_k(unsigned int z)        // delay...
    {
      unsigned int count ,t;
      for(count=0;count<z;count++)
       {
          for(t=0;t<200;t++);
       }
    }
// ============ Main ====================
char KeyPad(void)
{
        unsigned  char Key,temp,TempValue;
        temp = 0x00;
        SEGMENT = 0x00;//start 00 ...
        Switch_Init();
        delay_k(100);
        while(1)
        {
         Key=Keypad_scan();
         switch(Key){
         case 10:
              TempValue = SEGMENT;
                          SEGMENT = 0x00;
                          delay_k(50);
                          SEGMENT = TempValue;
                          return  TempValue;
         break;
         case 12:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                TempValue = SEGMENT;
                        SEGMENT = 0x00;
                        delay_k(50);
                        SEGMENT = TempValue;
                        return  TempValue;
            break;
            default:
                SEGMENT = temp+Key;
                delay_k(200);
                temp = Key<<4;
            break;
            }
    }
}
```

ส่วนของโค้ดโปรแกรม Keil C-51 เครื่อง Client

Client Program

```
#include<reg51.h>
#include<stdio.h>
#include<stdlib.h>
#include<ctype.h>
#include<string.h>
#include<keypad.c>
//======================== Init ======================
char T_ValueDEC;
char T_ValueUNIT;
unsigned char TEMP;
unsigned char buff[15];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char rxFrom;
int RL = 2;  //========== Input port ============
sbit  Line_1 = P0^0;
sbit  Line_2 = P0^1;
sbit  Line_3 = P0^2;
sbit  Relay1 = P0^6;
sbit  Relay2 = P0^7;
//=======================Function=======================
void Line_En(int Ln_Num){        //Line Enable
    int x;
        x=Ln_Num;
        if(x==1){
                Line_1 = 0;
                Line_2 = 0;
                Line_3 = 0;
                }
        else if(x==2){
                        Line_1 = 1;
                        Line_2 = 0;
                        Line_3 = 0;
                        }
          else if(x==3){
                          Line_1 = 0;
                          Line_2 = 1;
                          Line_3 = 0;
                         }
        else if(x==4){
                        Line_1 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Line_2 = 1;
                    Line_3 = 0;
                   }
        }
void serial()
        {
         SCON=0x50;
         TMOD=0x20;
         TH1 = 0xFB; //====29.4912 use 0xF8 or 18.432 use 0xFB ===
         EA=1;
         TR1=1;
         TI=1;
         RI=0;
        }
void delay(unsigned int z)          //========== delay ====================
        {
         unsigned int count,y;
         for(count=0;count<z;count++)
         {
          for(y=0;y<100;y++);
         }
        }
char ckCH(unsigned char C){
                        char buffCH;
                        switch(C){
                                case 0:
                                        buffCH = '0';
                                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 1:
        buffCH = '1';
break;
case 2:
        buffCH = '2';
break;
case 3:
        buffCH = '3';
break;
case 4:
        buffCH = '4';
break;
case 5:
        buffCH = '5';
break;
case 6:
        buffCH = '6';
break;
case 7:
        buffCH = '7';
break;
case 8:
        buffCH = '8';
break;
case 9:
        buffCH = '9';
break;
default:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
                }
            return buffCH;
        }
void int0_init(void){                   //Interrupt Ex_0 ==================
                    PX0 = 1;
                    IT0 = 1;
                    EX0 = 1;
                    EA = 1;
                    }
void ex0_isr(void) interrupt 0      // Interrupt Keypad to setup Temperature==========
{
  unsigned char numDEC,numUN;
  char bufChar1,bufChar2;
  TEMP = KeyPad();
  numDEC = TEMP>>4;
  bufChar1 =  ckCH(numDEC);
  delay(50);
  numUN = TEMP<<4;
  numUN = numUN>>4;
  bufChar2 = ckCH(numUN);
  T_ValueDEC = bufChar1;
  T_ValueUNIT = bufChar2;
}
//========== SEND ========================
int SendFunct(int L){
                    int j;
                    Line_En(L);  // Is connect ================
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                for(j=0;j<15;j++){
                                while(!TI);
                                TI=0;
                                SBUF=buff[j];
                                delay(80);
                                }
                                return 1;
                }
//===========MAIN====================
void main()
{
                char CN = '1'; //============Client Number=============
                char stR1,stR2;
                int Line=0,sendOK;
                int com_l=0;
                int i=0,j,TF=0,d=0,estop,k;
                serial();
                int0_init();
//============== START ============================
        START:
                RL = 2;
                com_l=0;
                for(j=0;j<15;j++){
                                buff[j] = '\0';
                                }
                delay(100);
                Line_En(1);
                printf("\nClient#1");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                delay(100);
//============= Recive ========================
                Line_En(RL);
                delay(100);
                gets(buff,sizeof(buff)+1);    //Recive Command From PC =====
                com_l=toint(buff[1]);
                delay(100);
                 while(com_l>13);
                 delay(100);
//============= SEND ========================
                 do{
                    if(buff[0]!='C'){
                                    for(i=0;i<14;i++){
                                                    buff[i] = buff[i+1];
                                                    }
                                    }
                      else;
                    }while(buff[0] != 'C');
        Line_En(1);
        delay(800);
        printf("\ndata in = ");
        for(j=0;j<15;j++){
                            printf("\n buff %d : %c",j,buff[j]);
                          }
        printf("\nCheck Command");
        delay(100);
  Line_En(RL);
  if(buff[7]=='1')    //== check send to me? ===
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  rxFrom = buff[6];
  switch(buff[0])              //== Yes check Header is Command or Data ==========
  {
    case 'C':                  //== If is command ===========
    if(buff[1]=='1')           //taget is i ============
    {
    switch(buff[2])
    {
     case '1':
              Relay1 = 1;
              stR1 = '1';
    break;
    case '0':
              Relay1 = 0;
              stR1 = '0';
    break;
    default :
    break;
   }
  switch(buff[3])
  {
     case '1':
              Relay2 = 1;
             stR2 = '1';
     break;
   case '0':
              Relay2 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            stR2 = '0';
    break;
   default :
   break;
  }
switch(buff[4])
{
 case 'T':
            buff[4]= T_ValueDEC;
           buff[5]= T_ValueUNIT;
 break;
 default :
           buff[4]='0';
           buff[5]='0';
 break;
}
for(d=9;d<15;d++)
{
   buff[d] = '\0';
}
delay(100);
buff[0]='D';
buff[1]='M';
buff[2]= stR1;
buff[3]= stR2;
buff[6]= rxFrom; //To ====
buff[7]= '1';       //From ===
buff[8]='E';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(2000);
sendOK = SendFunct(2);
delay(100);
goto START;
}
else{     //Taget is not i ==========
for(d=6;d<15;d++){
                    if(buff[d]=='E'){
                                estop = d;
                                for(k=estop;k<15;k++){
                                                buff[k] = '\0';
                                }
                                break;
                                }
                    else{
                          buff[d] = buff[d+1];
                          }
                }
RL = 3;
sendOK = SendFunct(RL);
delay(100);
goto RX;
}
break;
default :
break;
}
goto START;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
else{
      delay(100);
      for(j=0;j<15;j++){
                        buff[j] = '\0';
                        }
      delay(100);
      goto START;
    }
//============ RECIVE ================
RX:
       delay(80);
       Line_En(3);          //Port Send In put
       gets(buff,sizeof(buff)+1); //Recive Command From Client ======
       com_1=toint(buff[0]);
       delay(100);
//============= Fillter Data ====================
       do{
          if(buff[0]!='D'){
                           for(i=0;i<14;i++){
                                             buff[i] = buff[i+1];
                                            }
                           }
            else;
          }while(buff[0] != 'D');
 //================ Format New Data ============
 if(buff[6]=='1'){
                  for(d=8;d<15;d++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

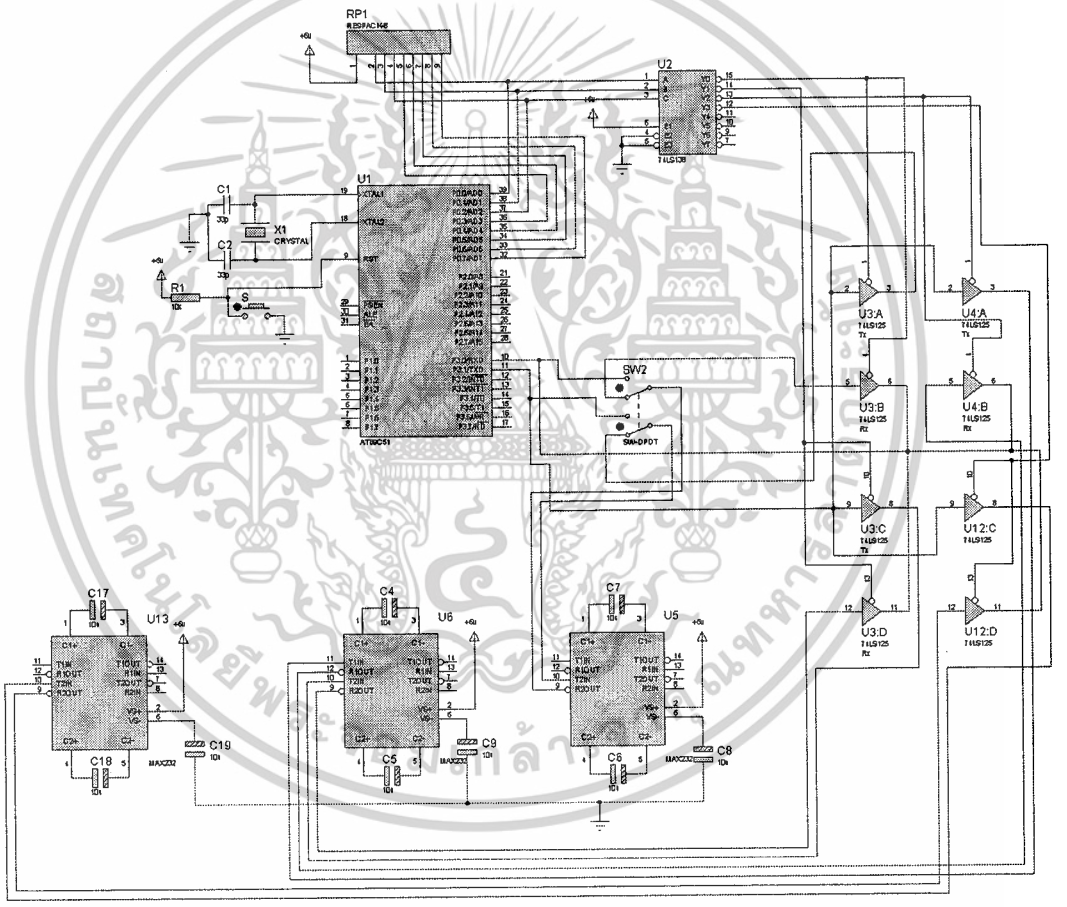
```
                    if(buff[d]== 'E'){
                                delay(80);
                                estop = d;
                                for(i=estop;i>=6;i--){
                                        buff[i+1] = buff[i];
                                        }
                                break;
                                }
else;
}
buff[6]= rxFrom;
buff[7]= CN;
//=== Ans Send ==========================================
Line_En(1);
printf("\n\n data recive from %C is %s\n",buff[7],buff);
delay(200);
RL = 2;
delay(80);
sendOK = SendFunct(RL);
delay(100);
}
else{
      goto START;
     }
      delay(80);
}
```

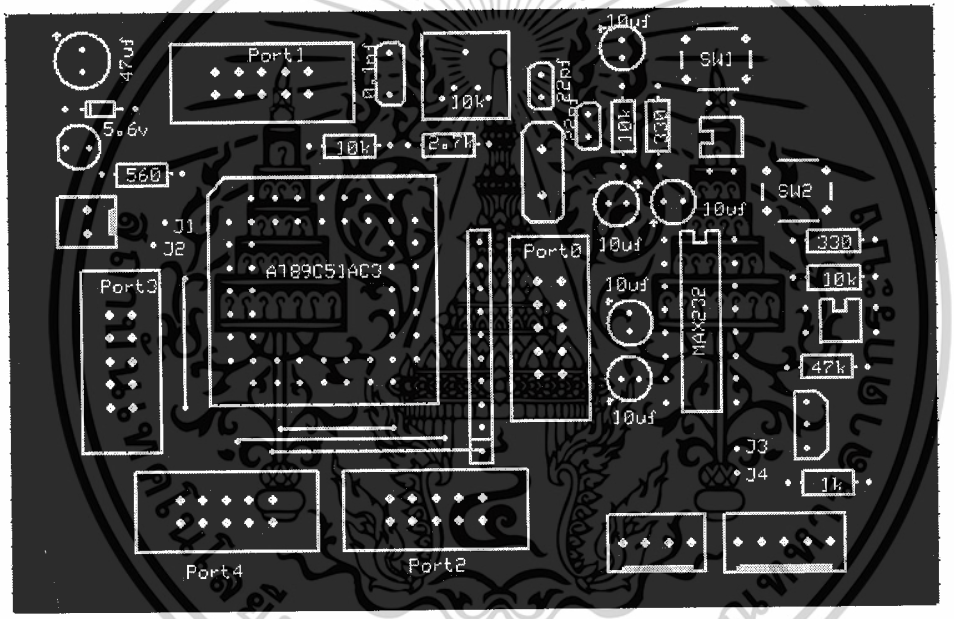
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
วงจรและลายวงจรทั้งหมด

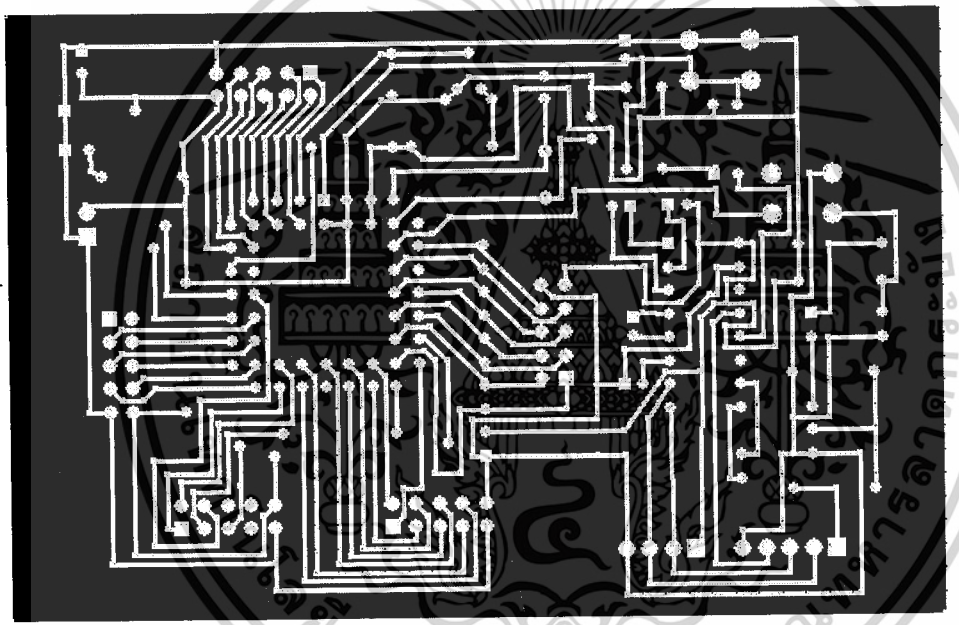
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรของ Main

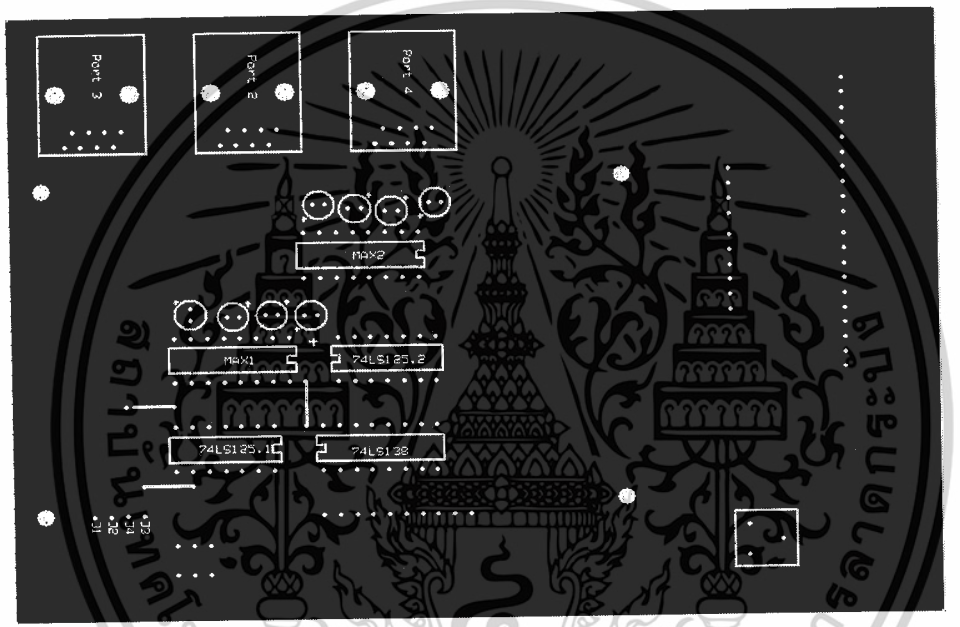




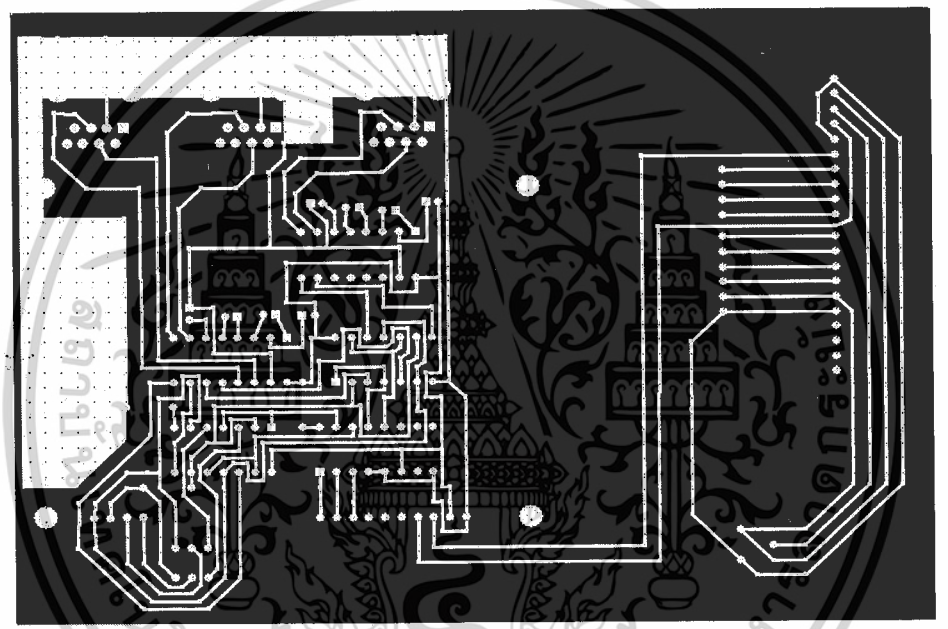
การวางอุปกรณ์ของ Main



ลายวงจรของ Main

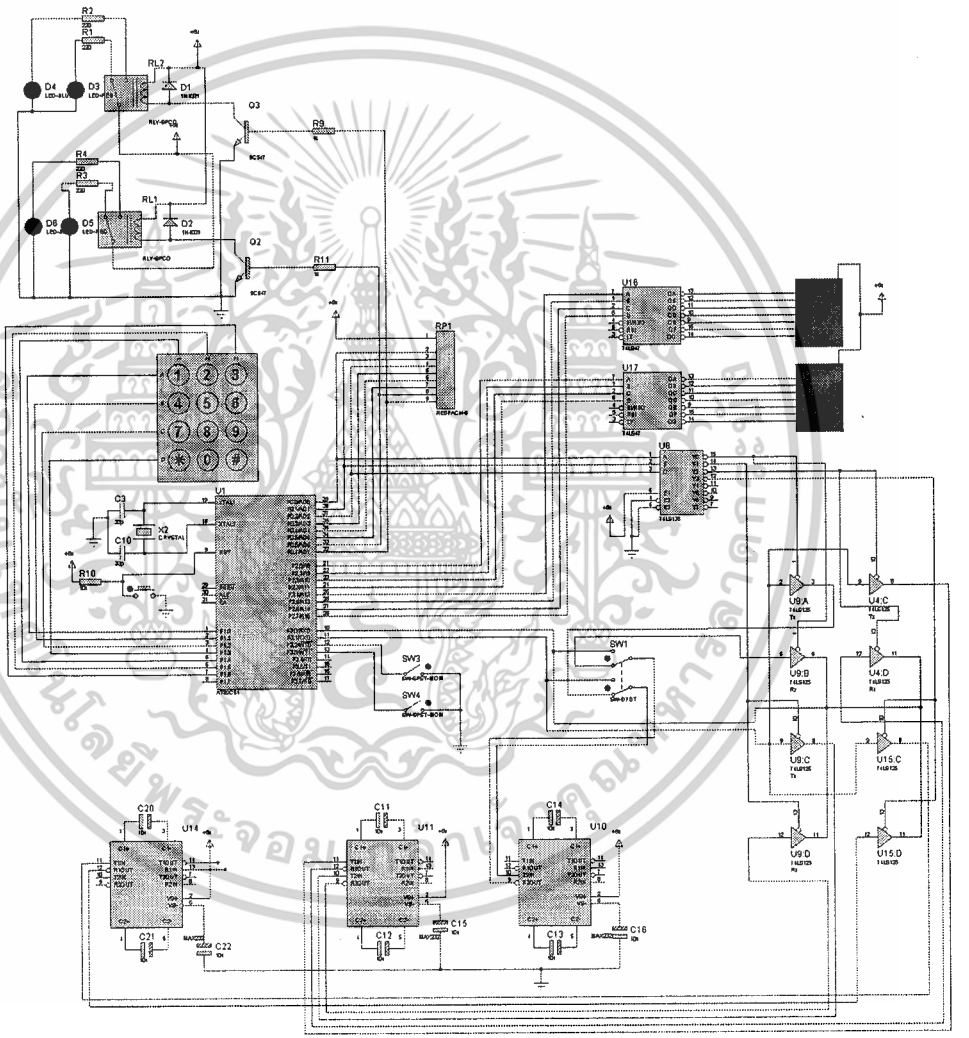


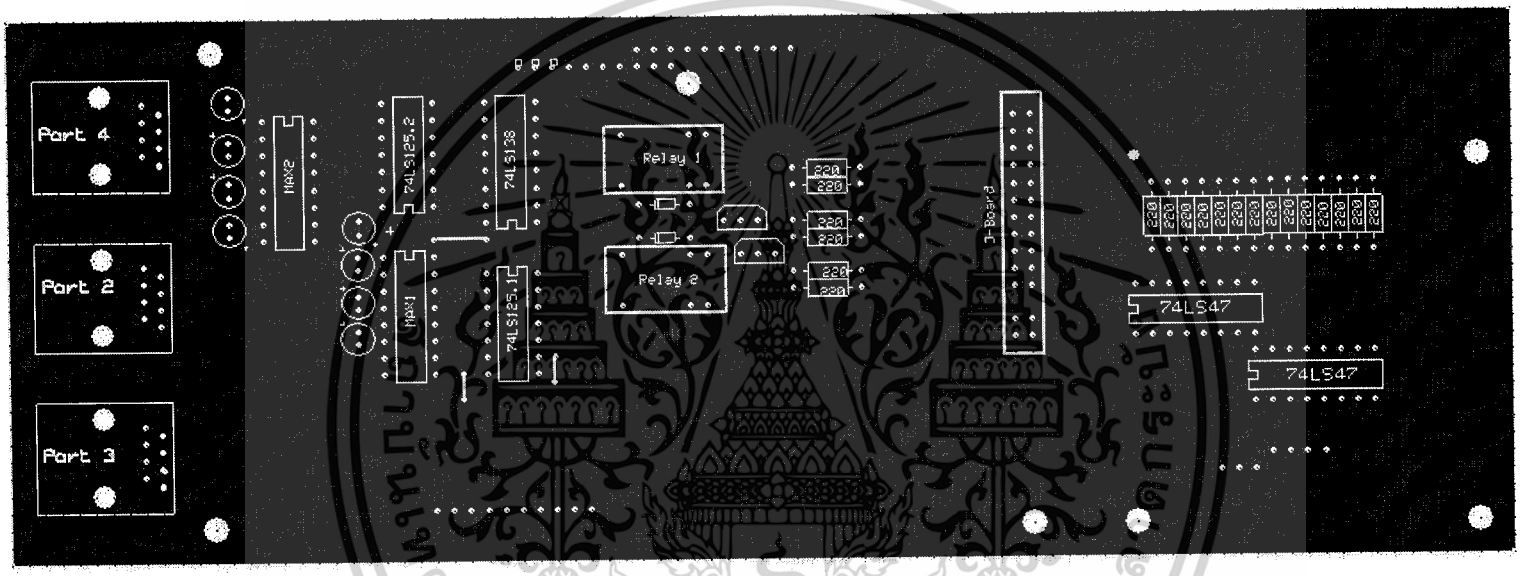
การวางอุปกรณ์ของชุดขยายพอร์ต ของ Main



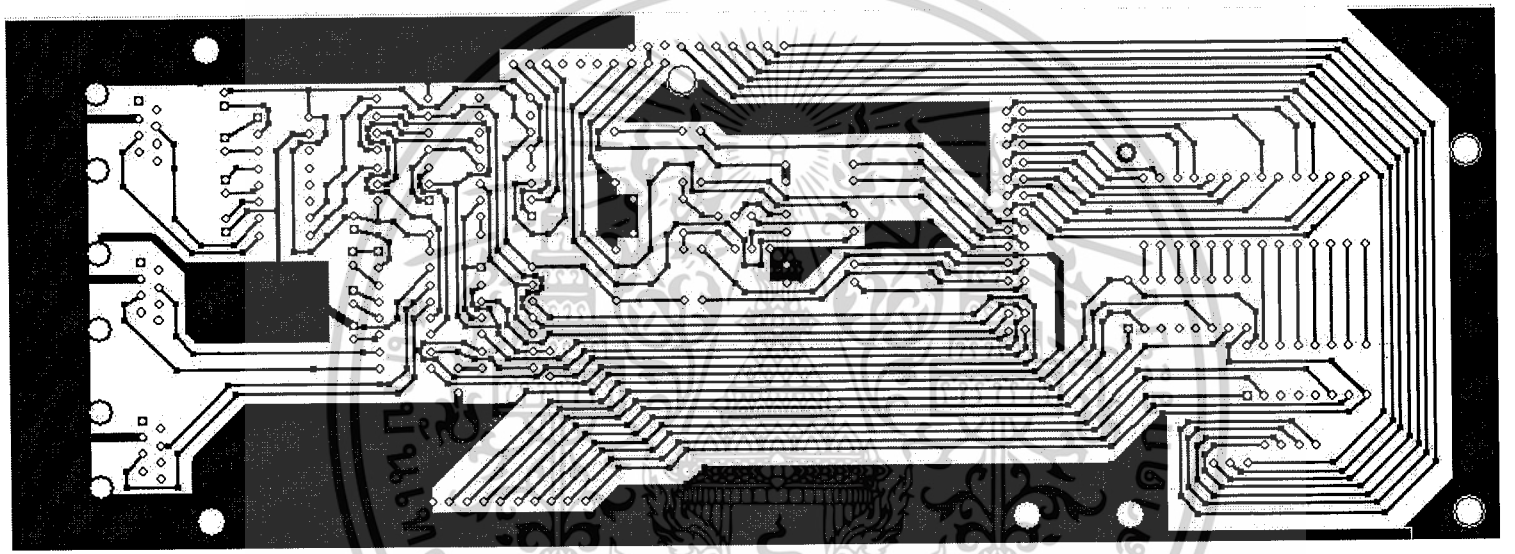
ลายวงจรของชุดขยายพอร์ต ของ Main

วงจรของ Client

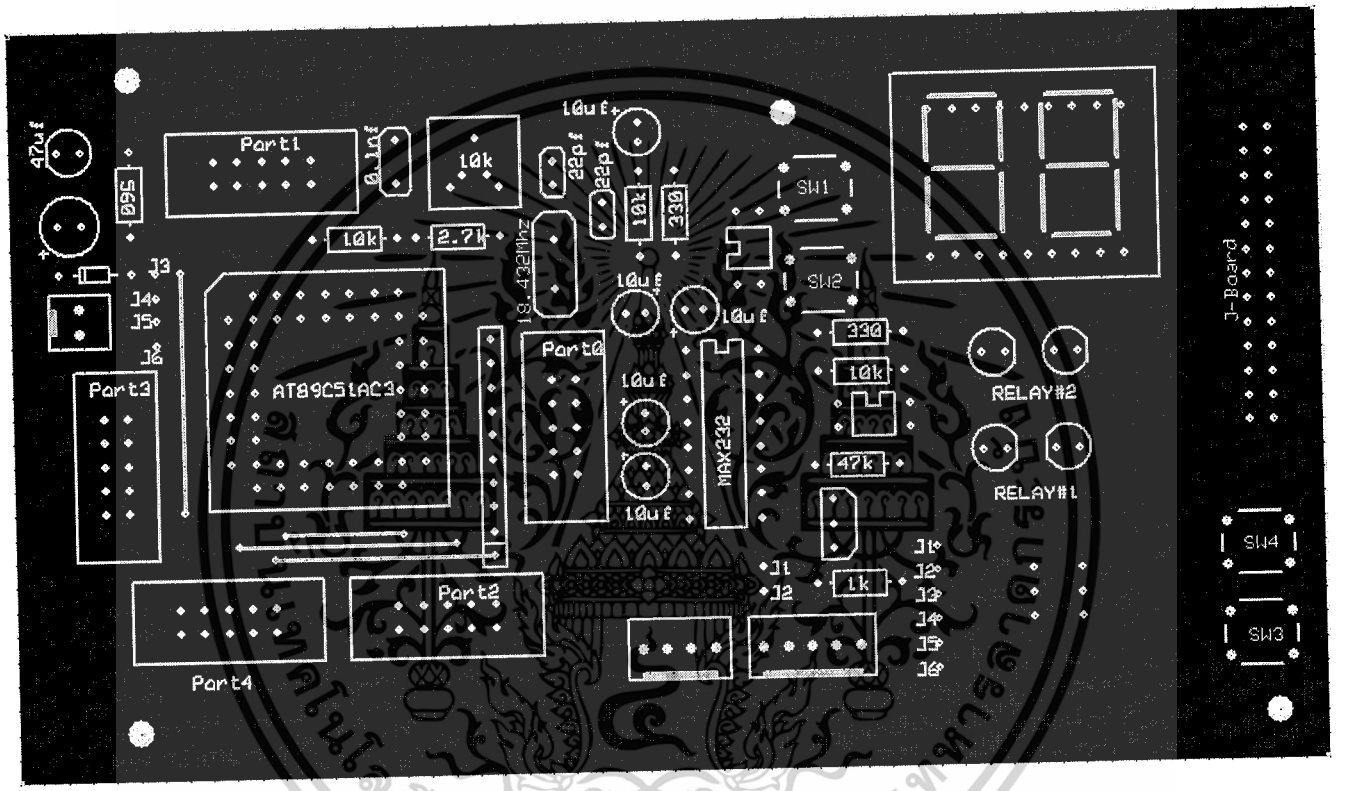




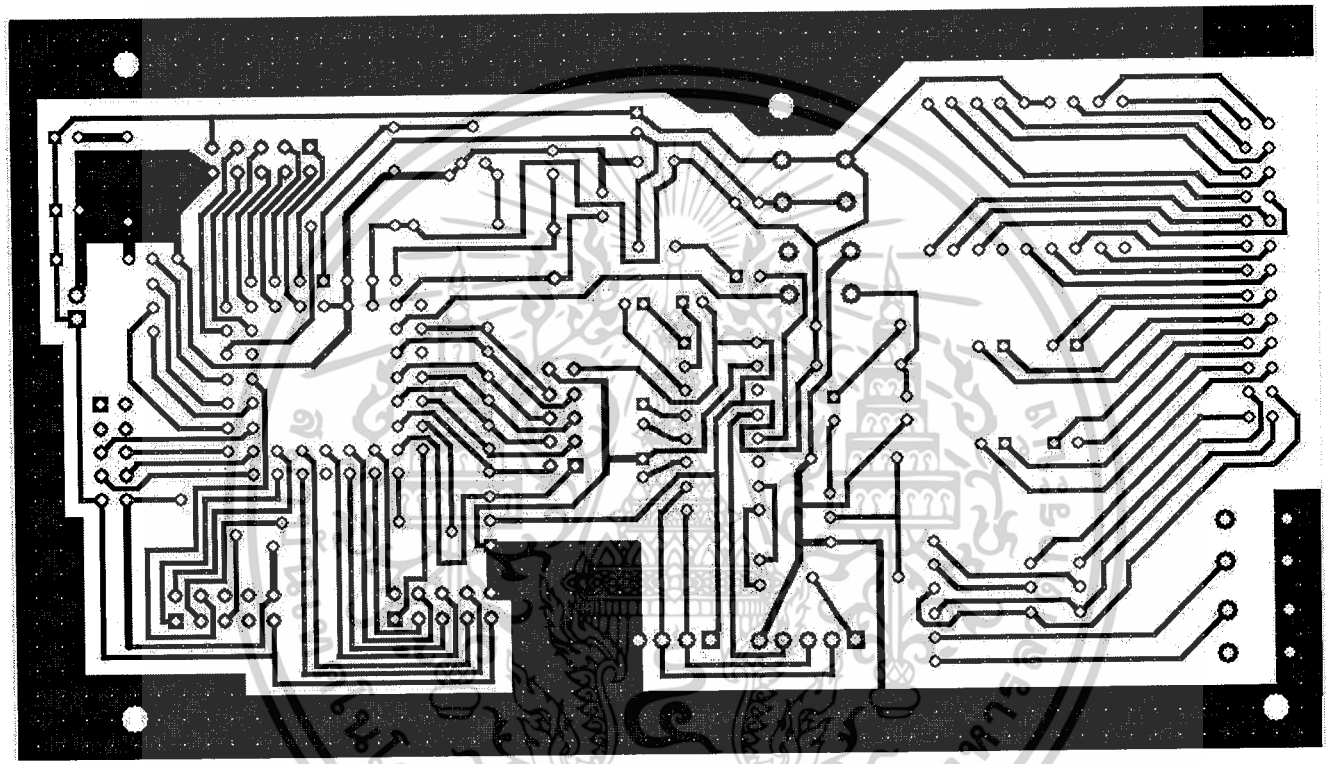
การวางอุปกรณ์ของชุดขยายพอร์ต ของ Client



ลายวงจรของชุดขยายพอร์ต ของ Client



การวางอุปกรณ์ของ Client



ลายวงจรของ Client

ภาคผนวก ง.

DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- 80C51 Core Architecture
- 256 Bytes of On-chip RAM
- 2048 Bytes of On-chip ERAM
- 64K Bytes of On-chip Flash Memory
  - Data Retention: 10 Years at 85°C
  - Read/Write Cycle: 100K
- Boot Code Section with Independent Lock Bits
- 2K Bytes of On-chip Flash for Bootloader
- In-System Programming by On-Chip UART Boot Program and IAP Capability
- 2K Bytes of On-chip EEPROM
  Read/Write Cycle: 100K
- Integrated Power Monitor (POR: PFD) To Supervise Internal Power Supply
- 14-sources 4-level Interrupts
- Three 16-bit Timers/Counters
- Full Duplex UART Compatible 80C51
- High-speed Architecture
  - In Standard Mode:
    40 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
    60 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
  - In X2 mode (6 Clocks/machine cycle)
    20 MHz (Vcc 3V to 5.5V, both Internal and external code execution)
    30 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)
- Five Ports: 32 + 4 Digital I/O Lines
- Five-channel 16-bit PCA with
  - PWM (8-bit)
  - High-speed Output
  - Timer and Edge Capture
- Double Data Pointer
- 21-bit WatchDog Timer (7 Programmable Bits)
- A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs
- SPI Interface (PLCC52 and VPFP64 packages only)
- On-chip Emulation Logic (Enhanced Hook System)
- Power Saving Modes
  - Idle Mode
  - Power-down Mode
- Power Supply: 3 volts to 5.5 volts
- Temperature Range: Industrial (-40° to +85°C)
- Packages: VQFP44, PLCC44, VQFP64, PLCC52

Description

The AT89C51AC3 is a high performance Flash version of the 80C51 single chip 8-bit microcontrollers.

In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time.

Besides the AT89C51AC3 provides 64K Bytes of Flash memory including In-System Programming (ISP) and IAP, 2K Bytes Boot Flash Memory, 2K Bytes EEPROM and 2048 byte ERAM.

Primary attention is paid to the reduction of the electro-magnetic emission of AT89C51AC3.

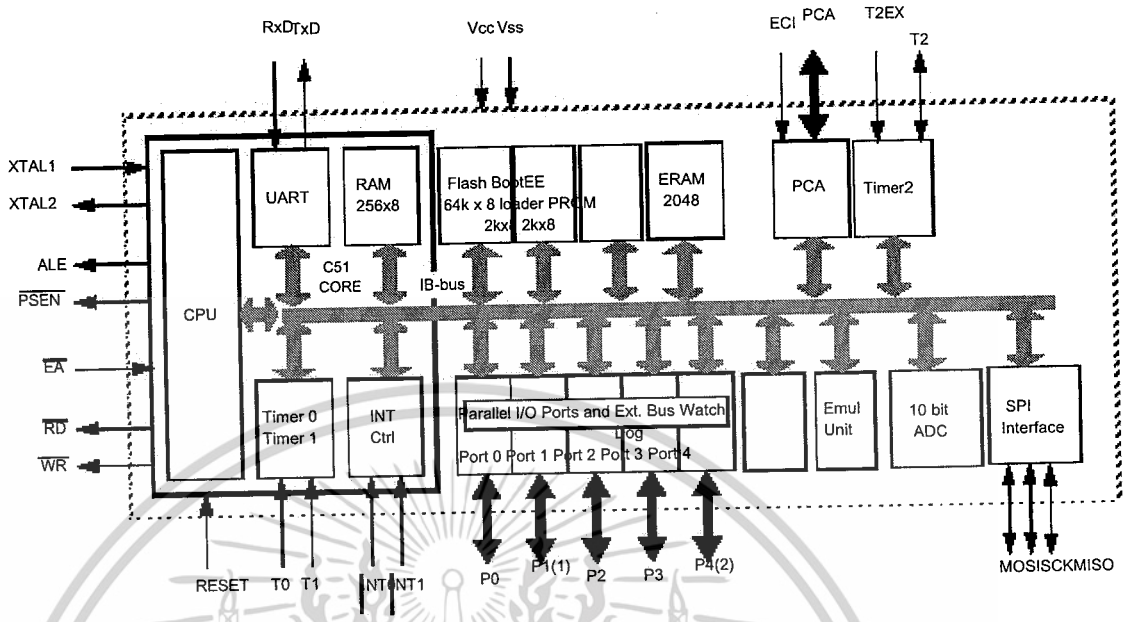
Enhanced 8-bit Microcontroller with 64KB Flash Memory

AT89C51AC3

Rev. 4383B–8051–01/05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



Notes: 1. 8 analog Inputs/8 Digital I/O
2. 5-Bit I/O Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Name	Type	Description
VSS	GND	Circuit ground
TESTI	I	Must be connected to VSS
VCC		Supply Voltage
VAREF		Reference Voltage for ADC
VAGND		Reference Ground for ADC
P0.0:7	I/O	Port 0: Is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in this state can be used as high-impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pull-ups when emitting 1's. Port 0 also outputs the code Bytes during program validation. External pull-ups are required during program verification.
P1.0:7	I/O	Port 1: Is an 8-bit bi-directional I/O port with internal pull-ups. Port 1 pins can be used for digital input/output or as analog inputs for the Analog Digital Converter (ADC). Port 1 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 1 pins that are being pulled low externally will be the source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 1 pins are assigned to be used as analog inputs via the ADCCF register (in this case the internal pull-ups are disconnected). As a secondary digital function, port 1 contains the Timer 2 external trigger and clock input; the PCA external clock input and the PCA module I/O. P1.0/AN0/T2 Analog input channel 0, External clock input for Timer/counter2. P1.1/AN1/T2EX Analog input channel 1, Trigger input for Timer/counter2. P1.2/AN2/ECI Analog input channel 2, PCA external clock input. P1.3/AN3/CEX0 Analog input channel 3, PCA module 0 Entry of input/PWM output. P1.4/AN4/CEX1 Analog input channel 4, PCA module 1 Entry of input/PWM output. P1.5/AN5/CEX2 Analog input channel 5, PCA module 2 Entry of input/PWM output. P1.6/AN6/CEX3 Analog input channel 6, PCA module 3 Entry of input/PWM output. P1.7/AN7/CEX4 Analog input channel 7, PCA module 4 Entry ot input/PWM output. Port 1 receives the low-order address byte during EPROM programming and program verification. It can drive CMOS inputs without external pull-ups.
P2.0:7	I/O	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Name	Type	Description
P3.0:7	I/O	Port 3: Is an 8-bit bi-directional I/O port with internal pull-ups. Port 3 pins that have 1's written to them are pulled high by the internal pull-up transistors and can be used as inputs in this state. As inputs, Port 3 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. The output latch corresponding to a secondary function must be programmed to one for that function to operate (except for TxD and $\overline{WR}$). The secondary functions are assigned to the pins of port 3 as follows: P3.0/RxD: Receiver data input (asynchronous) or data input/output (synchronous) of the serial interface P3.1/TxD: Transmitter data output (asynchronous) or clock output (synchronous) of the serial interface P3.2/$\overline{INT0}$: External interrupt 0 input/timer 0 gate control input P3.3/$\overline{INT1}$: External interrupt 1 input/timer 1 gate control input P3.4/T0: Timer 0 counter input P3.5/T1/$\overline{SS}$: Timer 1 counter input SPI Slave Select P3.6/$\overline{WR}$: External Data Memory write strobe; latches the data byte from port 0 into the external data memory P3.7/$\overline{RD}$: External Data Memory read strobe; Enables the external data memory. It can drive CMOS inputs without external pull-ups.
P4.0:4	I/O	Port 4: Is an 2-bit bi-directional I/O port with internal pull-ups. Port 4 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 4 pins that are being pulled low externally will be a source of current (IIL, on the datasheet) because of the internal pull-up transistor. The secondary functions are assigned to the 5 pins of port 4 as follows: P4.0: Regular Port I/O P4.1: Regular Port I/O P4.2/MISO: Master Input Slave Output of SPI controller P4.3/SCK: Serial Clock of SPI controller P4.4/MOSI: Master Ouput Slave Input of SPI controller It can drive CMOS inputs without external pull-ups.

Port 2:
Is an 8-bit bi-directional I/O port with internal pull-ups. Port 2 pins that have 1's written to them are pulled high by the internal pull-ups and can be used as inputs in this state. As inputs, Port 2 pins that are being pulled low externally will be a source of current (I_{IL}, see section "Electrical Characteristic") because of the internal pull-ups. Port 2 emits the high-order address byte during accesses to the external Program Memory and during accesses to external Data Memory that uses 16-bit addresses (MOVX @DPTR). In this application, it uses strong internal pull-ups when emitting 1's. During accesses to external Data Memory that use 8 bit addresses (MOVX @Ri), Port 2 transmits the contents of the P2 special function register.
It also receives high-order addresses and control signals during program validation.
It can drive CMOS inputs without external pull-ups.

4383B–8051–01/05
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Name	Type	Description
RESET	I/O	Reset: A high level on this pin during two machine cycles while the oscillator is running resets the device. An internal pull-down resistor to VSS permits power-on reset using only an external capacitor to VCC.
ALE	O	ALE: An Address Latch Enable output for latching the low byte of the address during accesses to the external memory. The ALE is activated every 1/6 oscillator periods (1/3 in X2 mode) except during an external data memory access. When instructions are executed from an internal Flash (EA = 1), ALE generation can be disabled by the software.
PSEN	O	$\overline{\text{PSEN}}$: The Program Store Enable output is a control signal that enables the external program memory of the bus during external fetch operations. It is activated twice each machine cycle during fetches from the external program memory. However, when executing from of the external program memory two activations of PSEN are skipped during each access to the external Data memory. The PSEN is not activated for internal fetches.
EA	I	$\overline{\text{EA}}$: When External Access is held at the high level, instructions are fetched from the internal Flash. When held at the low level, AT89C51AC3 fetches all instructions from the external program memory.
XTAL1	I	XTAL1: Input of the inverting oscillator amplifier and input of the internal clock generator circuits. To drive the device from an external clock source, XTAL1 should be driven, while XTAL2 is left unconnected. To operate above a frequency of 16 MHz, a duty cycle of 50% should be maintained.
XTAL2	O	XTAL2: Output from the inverting oscillator amplifier.

I/O Configurations

Each Port SFR operates via type-D latches, as illustrated in Figure 1 for Ports 3 and 4. A CPU "write to latch" signal initiates transfer of internal bus data into the type-D latch. A CPU "read latch" signal transfers the latched Q output onto the internal bus. Similarly, a "read pin" signal transfers the logical level of the Port pin. Some Port data instructions activate the "read latch" signal while others activate the "read pin" signal. Latch instructions are referred to as Read-Modify-Write instructions. Each I/O line may be independently programmed as input or output.

Port 1, Port 3 and Port 4

Figure 1 shows the structure of Ports 1 and 3, which have internal pull-ups. An external source can pull the pin low. Each Port pin can be configured either for general-purpose I/O or for its alternate input output function.

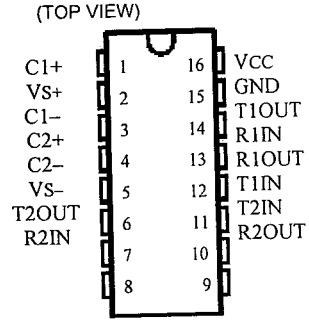
To use a pin for general-purpose output, set or clear the corresponding bit in the Px register (x = 1,3 or 4). To use a pin for general-purpose input, set the bit in the Px register. This turns off the output FET drive.

To configure a pin for its alternate function, set the bit in the Px register. When the latch is set, the "alternate output function" signal controls the output level (see Figure 1). The operation of Ports 1, 3 and 4 is discussed further in the "quasi-Bidirectional Port Operation" section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ±30–ç Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30–ç inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC□ library.

ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
	SOIC (D)	Tube	MAX232D	MAX232
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
–40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
	SOIC (D)	Tube	MAX232ID	MAX232I
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

D0h
C8h
C0h
B8h
B0h
A8h
A0h
98h
90h
88h
80h

EFh
E7h
DFh
D7h
CFh
C7h
BFh
B7h
AFh
A7h
9Fh
97h
8Fh
87h

Reserved

Note: 1. Do not read or write Reserved Registers

2. These registers are bit–addressable.
Sixteen addresses in the SFR space are both byte–addressable and bit–addressable. The bit–addressable SFR's are those whose address ends in 0 and 8. The bit addresses, in this area, are 0x80 through to 0xFF.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

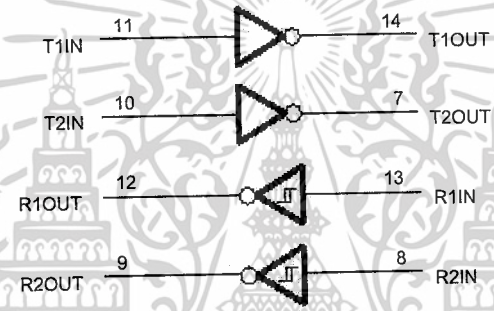
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, VCC (see Note 1) . . . . . –0.3 V to 6 V
Positive output supply voltage range, VS+ . . . . . VCC – 0.3 V to 15 V
Negative output supply voltage range, VS– . . . . . –0.3 V to –15 V
Input voltage range, VI: Driver . . . . . –0.3 V to VCC + 0.3 V
Receiver . . . . . ±30 V
Output voltage range, VO: T1OUT, T2OUT . . . . . VS– – 0.3 V to VS+ + 0.3 V
R1OUT, R2OUT . . . . . –0.3 V to VCC + 0.3 V
Short-circuit duration: T1OUT, T2OUT . . . . . Unlimited
Package thermal impedance, θJA (see Note 2): D package . . . . . 73°C/W
DW package . . . . . 57°C/W
N package . . . . . 67°C/W
NS package . . . . . 64°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds . . . . . 260°C
Storage temperature range, Tstg . . . . . –65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.
2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
VCC	Supply voltage		4.5	5	5.5	V
VIH	High-level input voltage (T1IN,T2IN)		2			V
VIL	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
TA	Operating free air temperaturefree-air	MAX232	0		70	°X
		MAX232I	–40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER		TEST CONDITIONS		MIN	TYP‡	MAX	UNIT
ICC	Supply current	VCC = 5.5 V, TA = 25°C	All outputs open,		8	10	mA

‡ All typical values are at VCC = 5 V and TA = 25°C.
NOTE 3: Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
VOH	High-level output voltage	T1OUT, T2OUT	RL = 3 kΩ to GND		5	7		V
VOL	Low-level output voltage‡	T1OUT, T2OUT	RL = 3 kΩ to GND			–7	–5	V
ro	Output resistance	T1OUT, T2OUT	VS+ = VS– = 0, VCC = 5.5 V,	VO = ±2 V VO = 0	300			&
IOS§	Short-circuit output current	T1OUT, T2OUT	±10 mA					
IIS	Short-circuit input current	T1IN, T2IN	VI = 0 to 200 μA					

† All typical values are at VCC = 5 V, TA = 25°C.
‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
§ Not more than one output should be shorted at a time.
NOTE 3: Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V.

switching characteristics, VCC = 5 V, TA = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	RL = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kbit/s

NOTE 3: Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
VOH	High-level output voltage	R1OUT, R2OUT	IOH = –1 mA		3.5			V
VOL	Low-level output voltage‡	R1OUT, R2OUT	IOL = 3.2 mA				0.4	V
VIT+	Receiver positive-going input threshold voltage	R1IN, R2IN	VCC = 5 V,	TA = 25°C		1.7	2.4	V
VIT–	Receiver negative-going input threshold voltage	R1IN, R2IN	VCC = 5 V,	TA = 25°C	0.8	1.2		V
Vhys	Input hysteresis voltage	R1IN, R2IN	VCC = 5 V			0.5	1	V
ri	Receiver input resistance	R1IN, R2IN	VCC = 5, TA = 25°C		3	5	7	kΩ

† All typical values are at VCC = 5 V, TA = 25°C.
‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 3: Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V.

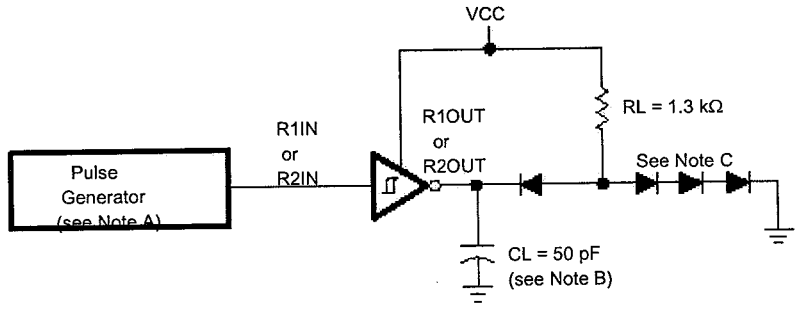
switching characteristics, VCC = 5 V, TA = 25°C (see Note 3 and Figure 1)

PARAMETER		TYP	UNIT
tPLH(R)	Receiver propagation delay time, low- to high-level output	500	ns
tPHL(R)	Receiver propagation delay time, high- to low-level output	500	ns

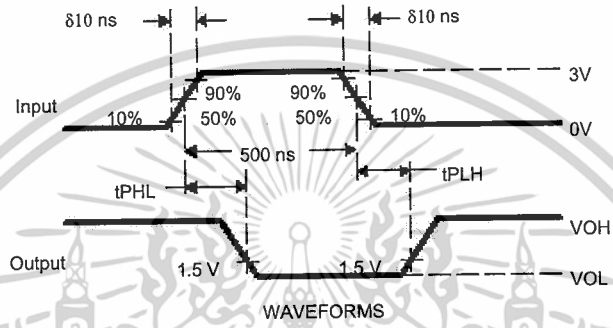
NOTE 3: Test conditions are C1–C4 = 1 μF at VCC = 5 V ± 0.5 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



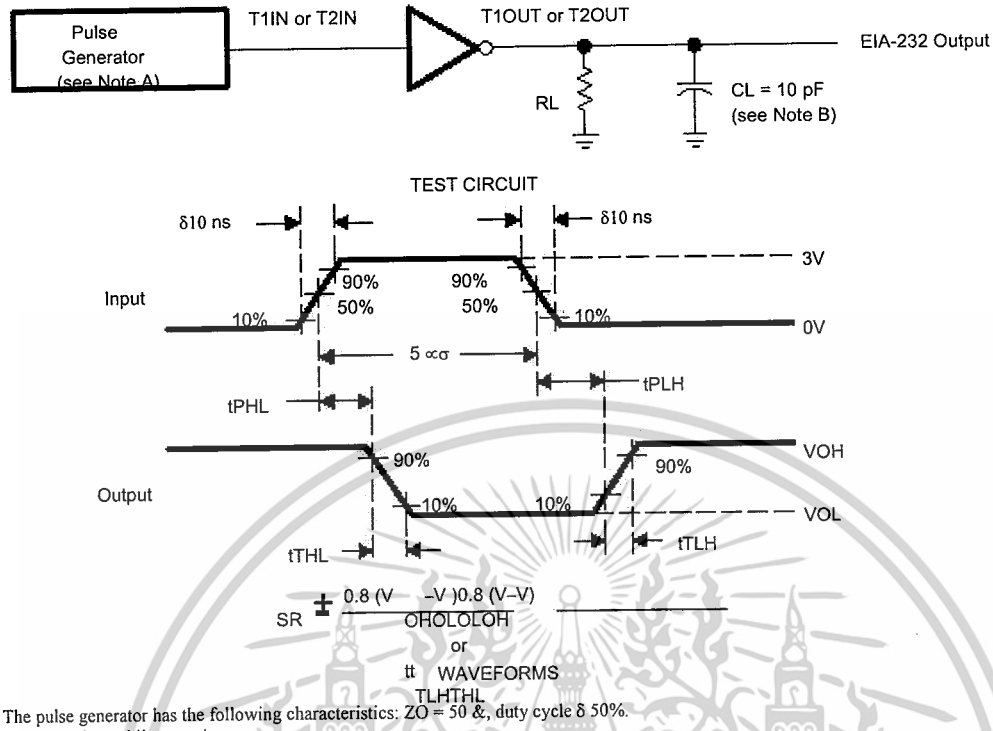
WAVEFORMS

NOTES: A. The pulse generator has the following characteristics: ZO = 50 &, duty cycle δ 50%.
B. CL includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

Figure 1. Receiver Test Circuit and Waveforms for tPHL and tPLH Measurements

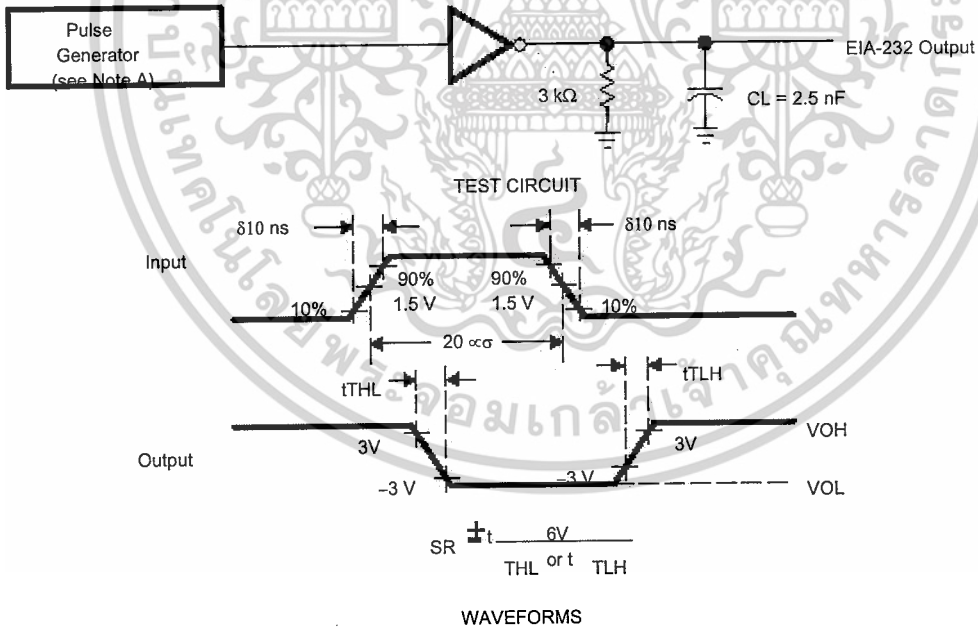
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: ZO = 50 &, duty cycle δ 50%.
B. CL includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for tPHL and tPLH Measurements (5-µs Input)

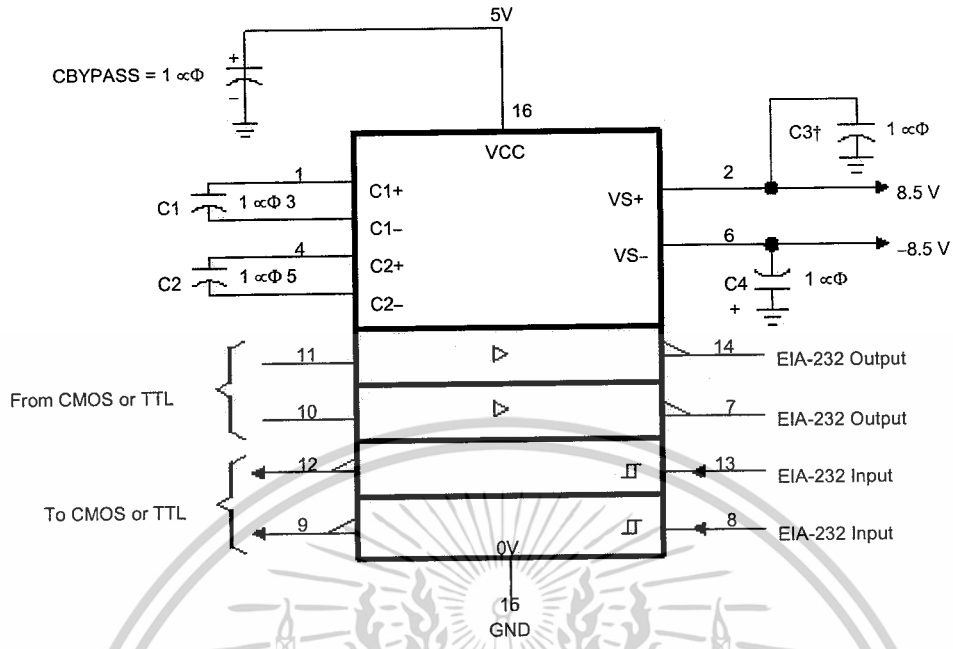


NOTE A: The pulse generator has the following characteristics: ZO = 50 &, duty cycle δ 50%.

Figure 3. Test Circuit and Waveforms for tTHL and tTLH Measurements (20-µs Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION



† C3 can be connected to VCC or GND.

Figure 4. Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third–party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265

Copyright © 2002, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS125A
Quad 3-STATE Buffer

General Description

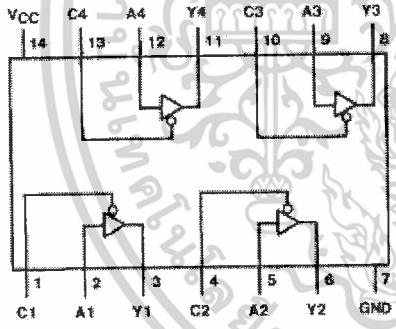
This device contains four independent gates each of which performs a non-inverting buffer function. The outputs have the 3-STATE feature. When enabled, the outputs exhibit the low impedance characteristics of a standard LS output with additional drive capability to permit the driving of bus lines without external resistors. When disabled, both the output transistors are turned off presenting a high-impedance state to the bus line. Thus the output will act neither as a significant load nor as a driver. To minimize the possibility that two outputs will attempt to take a common bus to opposite logic levels, the disable time is shorter than the enable time of the outputs.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS125AM	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS125ASJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS125AN	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

Y=A

Inputs		Output
A	C	Y
L	L	L
H	L	H
X	H	Hi-Z

H = HIGH Logic Level
L = LOW Logic Level
X = Either LOW or HIGH Logic Level
Hi-Z = 3-STATE (Outputs are disabled)

© 2000 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้