สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบรักษาความปลอดภัยในหมู่บ้านโดยใช้คอมพิวเตอร์ผ่านระบบเครือข่าย RS-485

SECURITY SYSTEM FOR VILLAGE USING COMPUTER VIA RS-485

T050426

นายสุริยเทพ นาคฤทธิ์

นายอนุชา สาสงเคราะห์

ว.วท.
ร 84 ร
2545

เลขหมู่.....
เลขทะเบียน 50426
วัน,เดือน,ปี 13 พ.ค. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b1135807x

ปีการศึกษา 2545

SECURITY SYSTEM FOR VILLAGE USING COMPUTER VIA RS-485

SURIYATEP NAKKARIT

ANUCHAR SASONGKORK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2002

เอกสารนี้เป็นเอกสาร[illegible]ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือแสวงหาผลประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ระบบรักษาความปลอดภัยในหมู่บ้านโดยใช้คอมพิวเตอร์ผ่านระบบเครือข่าย RS-485

SECURITY SYSTEM FOR VILLAGE USING COMPUTER VIA RS-485

นักศึกษาผู้จัดทำ นายสุริยเทพ นาคฤทธิ์ **รหัสประจำตัว** 43015544

นายอนุชา สาสงเคราะห์ **รหัสประจำตัว** 43015546

ปริญญา วิศวกรรมศาสตร์บัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2545

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
ผศ. ทวีพล ซื่อสัตย์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 22 เมษายน พ.ศ. 2545

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบรักษาความปลอดภัยในหมู่บ้านโดยใช้คอมพิวเตอร์ผ่านระบบเครือข่าย RS-485 (SECURITY SYSTEM FOR VILLAGE USING COMPUTER VIA RS-485 NETWORK)
นักศึกษาผู้จัดทำ	นาย สุริยเทพ นาคฤทธิ์ นาย อนุชา สาสงเคราะห์
อาจารย์ที่ปรึกษา	ผ.ศ ทวีพล ซื่อสัตย์
ปีการศึกษา	2545

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการออกแบบระบบรักษาความปลอดภัยสำหรับศูนย์รักษาความปลอดภัยของหมู่บ้าน ผ่านระบบเครือข่ายอนุกรม RS -485 โดยได้ทำการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ประจำบ้านแต่ละหลังกับคอมพิวเตอร์ศูนย์กลาง ในบ้านแต่ละหลังจะประกอบด้วย เซนเซอร์แบบต่างๆ เช่น ลิมิตสวิทช์ สวิทช์ตรวจจับด้วยแสง สวิทช์ตรวจจับแก๊สและเปลวเพลิง ซึ่งทำหน้าที่ในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ ที่ได้เชื่อมต่อทำการกับคอมพิวเตอร์ศูนย์กลางแบบ RS-485 ที่คอมพิวเตอร์ขณะทำการตรวจสอบสภาวะของชุดควบคุมของบ้าน แต่ละหลังผ่านข้อตกลงในการติดต่อสื่อสาร เมื่อมีเหตุร้ายเกิดขึ้นจะมีการเตือนและสามารถตรวจสอบได้ว่าเกิดเหตุอะไรขึ้นที่บ้านหลังใด โดยสามารถใช้โปรแกรมที่พัฒนาขึ้นมาจาก Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Security system for Village using computer via RS-485 Network
Authors	Mr.Suriyatep Nakkarit
	Mr.Anuchar Sasongkork
Thesis Advisor	Asst.Prof Taweephol Suesut
Year	2002

ABSTRACT

This thesis present,the design of security system for the security center of the Village through RS-485 network system. Each house has the microcontroller (MCS51) It is I/O connect to the sensors such as limit switchs, photo switchs and fire alarm sensors in order to report any information. The microcontrolller of each houses are connected via twisted pair inform of RS-485 network to the computer center. The computer at the security center can monitor and alarm the alarm all house in the village on the screen by the computer software which developed using Visual Basic 6.0 and the custom communication protocol.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดีเพราะได้รับความกรุณาจาก อาจารย์ทวีพล ซื่อสัตย์ ที่ได้ให้คำแนะนำแก่ผู้จัดทำตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์ เครื่องมือต่างๆ ในการทำปริญญานิพนธ์นี้ ผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดทุกท่าน ที่ได้ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

ขอบคุณ คุณชุมพล แก้วประพันธิ์ และ นักศึกษาห้อง 3S/1 ที่ช่วยติชมปริญญานิพนธ์ฉบับนี้และช่วยเหลือเกี่ยวกับ เครื่องมือต่างๆ และเป็นกำลังใจซึ่งกันและกัน

และที่ลืมมิได้คือ ขอกราบขอบพระคุณคุณพ่อ คุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์ที่ได้จากปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญภาพ.....VII

บทที่ 1 บทนำ.....1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....2

1.3 ขอบเขตของปริญญานิพนธ์.....3

1.4 ขั้นตอนการศึกษา.....3

บทที่ 2 ส่วนประกอบที่สำคัญของระบบ.....4

2.1 ไมโครคอนโทรลเลอร์ MCS-51.....4

2.2 ลักษณะการสื่อสาร.....15

2.3 ลักษณะการสื่อสาร.....17

2.4 โครงสร้างของระบบควบคุม.....25

2.5 ตัวควบคุม.....28

2.6 ระบบโทรศัพท์แจ้งเหตุ.....30

2.7 Sensor.....32

บทที่ 3 การประยุกต์ใช้งานในระบบ.....33

3.1 การทำงานของระบบ.....33

3.2 การติดตั้งเครื่องตรวจจับในที่พัก(SENSOR).....36

3.3 อธิบายโปรแกรมการทำงาน.....38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 การใช้งานโปรแกรม.....42

4.1 เปิด Port ใช้งาน.....42

4.2 เปิดดูสถานะของบ้าน.....43

4.3 การ Set Config.....44

บทที่ 5 การทดลองใช้งานโปรแกรม.....45

5.1 การใช้ร่วมกับโปรแกรม Hyper Terminal.....45

5.2 การรันค่าที่โปรแกรม ไว้ใน MCS-51.....46

5.3 เปิดทดสอบโปรแกรมจริง.....47

5.4 เมื่อเกิดการทำงานของ Sensor 48

5.5 ตรวจดูการทำงาน.....49

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....50

6.1 บทสรุป.....50

6.2 ข้อเสนอแนะและแนวทางการพัฒนา.....50

บรรณานุกรม.....51

ภาคผนวก.....52

ภาคผนวก ก.....53

ภาคผนวก ข.....58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 การวางขาของไมโครคอนโทรลเลอร์....4
2.2 สัญญาณ ALE....7
2.3 สัญญาณ PSEN....8
2.4 สัญญาณ WR....9
2.5 สัญญาณ RD....10
2.6 PORT 0....11
2.7 PORT 1....12
2.8 PORT 2....13
2.9PORT 3....14
2.10 การสื่อสารแบบอนุกรม....15
2.11 การส่งข้อมูลแบบอนุกรม....15
2.12 แสดงลักษณะการสื่อสารแบบขนาน....16
2.13 แสดงการส่งข้อมูลแบบขนาน....16
2.14 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-232-C....18
2.15 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423....19
2.16 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423....20
2.17 แสดงวงจรการเชื่อมต่อแบบ Current loop....21
2.18 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-485....22
2.19 แสดงการเชื่อมต่อเครือข่ายตามมาตรฐาน RS-485....22
2.20 แสดงช่วงเวลาของกระแสและแรงดันในการทำงาน(Loading Curve of a FULL UL)....23
2.21 แสดงเครือข่ายที่สมมูลกันตามมาตรบาน RS-485....23
2.22 ความสัมพันธ์ของอัตราเร็วในการส่งข้อมูลและระยะทาง....24
2.23 แสดงให้เห็นถึงโครงสร้างของระบบควบคุม....25
2.24 แสดงการเชื่อมต่อเครือข่ายของระบบควบคุม....26
2.25 แสดงการเชื่อมต่อตัวควบคุมตัวลูกข่ายและอุปกรณ์ที่ต้องการควบคุม....26
2.26 ตัวอย่างการประยุกต์ใช้งาน....27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

2.27 โครงสร้างของตัวควบคุม.....28
2.28 แสดงส่วนประกอบของตัวควบคุม.....29
2.29 การทำงานของ Sensor.....32
3.1 รูปแบบการทำงานทั้งระบบ.....33
3.2 บล็อคไดอะแกรมการทำงานของระบบ.....34
3.3 ลักษณะภายนอกตัวบ้าน.....36
3.4 ลักษณะภายในตอนยังไม่ติดตั้งเครื่องตรวจจับ.....36
3.5 ลักษณะภายในตอนติดตั้งเครื่องตรวจจับแล้ว.....37
3.6 โฟรชาร์ทการทำงานของ MCS-51.....39
3.7 โฟร์ชาร์ทของโปรแกรม Computer.....40
3.8 หน้าตาโปรแกรมที่ใช้งาน.....41
4.1 แสดงหน้าจอเมื่อกด Open Port.....42
4.2 แสดงหน้าจอเมื่อกดหมายเลขบ้าน.....43
4.3 แสดงหน้าจอเมื่อกด Config.....44
5.1 การเปิดใช้ Hyper Terminal.....45
5.2 ลักษณะค่าที่แสดงออกมาใน Hyper terminal.....47
5.3 ลักษณะค่าที่แสดงออกมาในโปรแกรม.....47
5.4 ลักษณะหน้าจอจะแสดงเมื่อเกิด Alarm.....48
5.5 แสดงการทำงานSensor 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

ในปัจจุบันความก้าวหน้าของเทคโนโลยีได้มี การพัฒนาไปไกลมาก โดยสังเกตได้จากสภาพแวดล้อมโดยทั่วไปทางกายภาพ เช่น อาคารสถานที่ต่างๆ ตลอดจนวัฒนธรรมที่แปลกใหม่ที่เข้ามาในสังคมไทย แต่ภายในเบื้องหลังของการพัฒนาทางเทคโนโลยีที่มีที่ท่าว่าจะไม่หยุดยั้งนี้ ทำให้มนุษย์ มีจำนวนประชากรเพิ่มมากขึ้นก่อให้เกิดปัญหาในหลายๆด้านๆเช่น ปัญหารถติด ปัญหาการจราจรปัญหาความแออัดของชุมชน และปัญหาการว่างงาน ตลอดจนปัญหาอื่นๆในสังคม แต่ปัญหาที่สำคัญที่เกิดขึ้น และ เป็นอันตรายต่อความปลอดภัยในชีวิต และ ทรัพย์สินคือปัญหาการลักทรัพย์ ปัญหาที่เกิดจากอัคคีภัย ปัญหาที่เกิดจากมีผู้บุกรุกเข้ามาภายในบ้าน อาคาร สถานที่ต่างๆ จากปัญหาต่างๆดังที่ได้กล่าวมาข้างต้นทั้งหมดนี้

ในสภาพเศรษฐกิจที่รัดตัวผู้คนในปัจจุบันเกือบทั้งหมดต้องออกไปทำงานนอกบ้านทิ้งบ้านไว้โดยไม่มีใครดูแล ทรัพย์สินของมีค่าต่างๆ ที่วางอยู่ในบ้าน เครื่องใช้ไฟฟ้าต่างๆวางอยู่ในบ้าน เครื่องใช้ไฟฟ้าบางชนิดต้องเปิดไว้ตลอดเวลาบ้าง ดังนั้น ถ้าเกิดเหตุการณ์ที่ไม่คาดฝันขึ้น เช่นมีผู้บุกรุกเข้ามาภายในบริเวณบ้าน เกิดแก๊สรั่ว เกิดเหตุเพลิงไหม้

จากกระแสไฟฟ้าลัดวงจร เจ้าของบ้านนั้นไม่สามารถทราบได้เลยว่าเมื่อออกไปจากบ้านนั้นจะมีเหตุการณ์ต่างๆอะไรเกิดที่บ้านบ้างและไม่สามารถแก้ไขได้จนอาจทำให้สูญเสียทรัพย์สินไปทั้งหมดได้

จากปัญหาต่างๆเหล่านี้ ทางผู้จัดทำจึงได้เกิดความคิดที่จะสร้างอุปกรณ์อำนวยความสะดวกให้แก่ผู้ที่ต้องออกไปทำงานนอกบ้านโดยเครื่องจะทำการตรวจสอบแก๊สรั่วเพลิงไหม้ หรือมีผู้บุกรุกเข้ามาภายในบ้าน เมื่อเครื่องมือตรวจพบจะทำการโทรศัพท์ไปบอกเจ้าของบ้านให้ทราบ เพื่อให้เจ้าของบ้านได้รับรู้ว่าตอนนี้ได้มีเหตุการณ์ผิดปกติเกิดขึ้นที่บ้านของท่าน เพื่อให้เจ้าของบ้านหาทางป้องกันและทำการแก้ไขต่อไป

จากที่กล่าวมาทั้งหมดนี้ เป็นแนวคิดที่ได้จัดทำขึ้นในวิชาโปรเจค 1 ในปีการศึกษา 2545 ภาคเรียนที่ 1 แต่ในภาคเรียนที่ 2 นี้คณะผู้จัดทำได้ดำเนินการนำแนวความคิดจากภาคเรียนที่ 1 มาดัดแปลงทำการประยุกต์เพิ่มเติม โดยทางคณะผู้จัดทำได้ได้เล็งข้อผิดหลายอย่างที่เกิดขึ้น ไม่ว่าจะเป็นการป้องกันเหตุการณ์ต่างๆที่เกิดขึ้นไม่สามารถจะทำการป้องกันได้ทันเวลา เพราะแม้ว่าจะรู้ว่ามีเหตุการณ์ต่างๆ เกิดขึ้นกับบ้านของตัวเอง แต่เจ้าของบ้านไม่อาจจะกลับมาแก้ไข และ ป้องกันเหตุการณ์ต่างๆได้ทันท่วงทีเนื่องจากตัวเจ้าของบ้านอยู่ไกลหรือมีกิจการงานต่างๆที่ต้องทำเพื่อขจัดปัญหาต่างๆเหล่านี้ที่เกิดขึ้น คณะผู้จัดทำได้ทำการดัดแปลงเพิ่มเติมโดยสังเกตเห็นว่าใน หมู่บ้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละแห่งนั้น จะมีเจ้าหน้าที่รักษาความปลอดภัย(ยาม) ประจำอยู่หมู่บ้านละ 1 คนหรือ 2 คน ซึ่งทางคณะผู้จัดทำได้เล็งเห็นว่าจากปัญหาเรื่องการป้องกันละแก้ไขเหตุการณ์ของเจ้าของบ้าน ถ้าเราเปลี่ยนเป็นไปให้ เจ้าหน้าที่รักษาความปลอดภัยดูแล และ แก้ไขเหตุการณ์แทนเจ้าของบ้านสืบเนื่องจาก เจ้าหน้าที่รักษาความปลอดภัย อยู่ใกล้กับเหตุการณ์ และ เป็นหน้าที่โดยตรงของเจ้าหน้าที่รักษาความปลอดภัยอยู่แล้ว แต่ปัญหาที่เกิดขึ้นก็คือ เจ้าหน้าที่รักษาความปลอดภัยไม่สามารถไปตรวจไปตามบ้านแต่ละหลังได้ได้ตลอดเวลา ทางคณะผู้จัดทำจึงได้นำคอมพิวเตอร์มาช่วยแสดงผลเหตุการณ์ต่างๆที่เกิดขึ้นกับบ้านแต่ละหลังแต่ปัญหาที่พบต่อไปนั้นคือ การส่งข้อมูลของคอมพิวเตอร์โดยที่เราใช้มาตรฐานสัญญาณRS-232ทำให้ใช้ได้ในระยะทางสั้นๆ แต่ระยะทางที่ห่างกันมากๆระหว่าง ศูนย์รักษาความปลอดภัยของเจ้าหน้าที่รักษาความปลอดภัยกับบ้านแต่ละหลังนั้นๆบางหมู่บ้านก็ไกลกันมาก ทางคณะผู้จัดทำได้แก้ปัญหาที่เกิดขึ้น โดยนำมาตรฐาน RS 485 มาใช้แก้ปัญหาช่วยให้การส่งข้อมูลของคอมพิวเตอร์สามารถส่งได้ระยะทางที่ไกลขึ้น ส่วนอุปกรณ์ที่นำมาใช้ติดต่อกับคอมพิวเตอร์นั้นได้แก่ไมโครคอนโทรลเลอร์ MCS-51 โดยที่ตัวของ MCS-51 ไมโครคอนโทรลเลอร์ MCS-51แต่ละตัวจะแทนบ้านแต่ละหลังโดยเรานำค่าสภาวะของเซนเซอร์ที่ต่อเข้ากับพอร์ตอินพุตของ ไมโครคอนโทรลเลอร์ MCS-51ส่งไปยังคอมพิวเตอร์ เพื่อแสดงค่าสภาวะของเซนเซอร์

ที่หน้าจอคอมพิวเตอร์ โดยแสดงเป็นรูปภาพ โดยที่หน้าจอของคอมพิวเตอร์เราจะทำเป็นรูปภาพแทนบ้านแต่ละกลัง โดยข้างในรูปภาพนั้นจะแสดงเป็นแบบแปลนบ้านและก็เซนเซอร์ต่างๆแสดงสภาวะการทำงาน

จากที่กล่าวมาทั้งหมดนี้เป็นเพียงภาพรวมของระบบทั้งหมดที่กล่าวถึง สำหรับการที่จะนำไปประยุกต์ใช้งาน เราอาจจะใช้กล้องวงจรปิดตรวจจับและส่งภาพมายังคอมพิวเตอร์ทำให้เราสามารถเห็นเหตุการณ์ ต่างๆที่เกิดขึ้นจริงแม้ว่าจะไม่อยู่ในเหตุการณ์ สามารถนำไปประยุกต์ใช้งานแบบต่างๆแล้วแต่ชนิดและขอบเขตความต้องการของงานนั้นๆ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อต้องการออกแบบสร้างเครื่องมืออุปกรณ์ที่ทำหน้าที่ ตรวจสอบแก๊สรั่ว เพลิงไหม้ ผู้บุกรุกเข้ามาภายในที่อยู่อาศัย ให้ทำงาน โดยการอาศัยโทรศัพท์เป็นสื่อกลางระหว่างเครื่องกับเจ้าของบ้านและอาศัยคอมพิวเตอร์เป็นสื่อกลางระหว่างเครื่องกับผู้รักษาความปลอดภัย
2. เพื่อให้เจ้าของบ้านและผู้รักษาความปลอดภัย สามารถทราบเหตุการณ์ที่เกิดขึ้นภายในบ้านไม่ได้อยู่ภายในบ้าน
3. ต้องการตรวจเช็คเหตุการณ์จากจากบ้านหลายๆหลังพร้อมกัน(หมู่บ้าน)ว่ามีเหตุการณ์ผิดปกติเกิดขึ้นหรือไม่โดยใช้คอมพิวเตอร์ตรวจเช็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อศึกษาการทำงานของเซนเซอร์ ไมโครคอนโทรลเลอร์ สาย RS-485

1.3 ขอบเขตของปริญญานิพนธ์

1. เป็นเครื่องมือที่ใช้ตรวจสอบ ความผิดปกติภายในบ้าน เช่นแก๊สรั่ว เพลิงไหม้ และมีผู้บุกรุกเข้ามาในบ้าน แล้วแต่ว่าต้องการอะไรโดยตรวจสอบได้ 4 ชนิด ตามต้องการ เพียงแค่เปลี่ยนตัวตรวจจับเซนเซอร์เท่านั้น

2.ใช้ไมโครคอนโทรลเลอร์(MCS-51) เป็นตัวตรวจสอบสภาวะ การทำงานของเซนเซอร์ซึ่งไมโครคอนโทรลเลอร์แต่ละตัวจะควบคุมบ้านแต่ละหลัง

3.ใช้คอมพิวเตอร์ตรวจเช็คข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์แต่ละตัวแล้วนำข้อมูลที่ได้มาแสดงผลหน้าจอคอมพิวเตอร์เพื่อแสดงสภาวะต่างๆของเซนเซอร์

4. ใช้สาย RS-485 เป็นตัวส่งผ่านข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์โดยมีวงจรแปลง RS-485เป็น RS-232 เพื่อส่งข้อมูลเข้าคอมพิวเตอร์

1.4 ขั้นตอนการศึกษา

1. ศึกษาทฤษฎีเกี่ยวกับการส่งข้อมูลโดยใช้คอมพิวเตอร์ สายRS-485 การแปลงสัญญาณ RS-485เป็น RS-232

2. รวบรวมขั้นตอนการออกแบบเกี่ยวกับระบบเตือนภัยทั้งหมด เพื่อเขียนโปรแกรมให้ทำงานตามกำหนด

3. เขียนโปรแกรมโดยใช้ Delphi เพื่อทำการติดต่อกับไมโครคอนโทรลเลอร์ และทำการเขียนโปรแกรมไมโครคอนโทรลเลอร์

1.5 ประโยชน์ที่ได้รับจากการดำเนินงาน

1. ได้ทราบคุณสมบัติ สาย RS-485 สาย RS-232 ไมโครคอนโทรเลอร์ เซนเซอร์ชนิดต่างๆ การเขียนโปรแกรม

2. สามารถสร้างระบบเตือนภัยโดยใช้ ไมโครคอนโทรเลอร์เป็นตัวควบคุมและเชื่อมต่อกับคอมพิวเตอร์

3. นำโปรแกรมที่ออกแบบและระบบเตือนภัยไปใช้งานได้

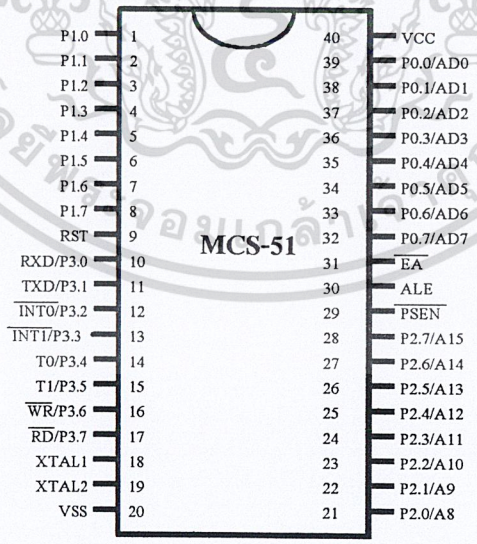
4. ทำให้ราคาถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ส่วนประกอบที่สำคัญของระบบ

2.1 ไมโครคอนโทรลเลอร์ MCS 51

จากความก้าวหน้าของเทคโนโลยี่ทางด้านไมโครอิเล็กทรอนิกส์ ทำให้เกิดการพัฒนาการทางด้านไมโครโปรเซสเซอร์ซึ่งได้เข้ามามีบทบาทเป็นอย่างมาก ทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์มีหลายบริษัทผลิต chip ออกมาแข่งขันกันมากขึ้นซึ่งแต่ละบริษัทก็จะมีออปชั่นการใช้งานให้เลือกที่ต่างๆกันไปทางด้านราคาก็จะแตกต่างกันไปตามความสามารถของแต่ละรุ่น ในขณะเดียวกันทางด้านซอฟแวร์หรือการเขียนโปรแกรมในการควบคุมไมโครคอนโทรนเลอร์ก็เช่นกันได้มีการพัฒนาให้ การเขียนโปรแกรมเป็นไปโดยง่ายขึ้น ยืดหยุ่น มีโครงสร้างที่เข้าใจง่ายขึ้นจากภาษาแอสแซมบลี ซึ่งเป็นภาษาเครื่องมาเป็น ภาษาซี ซึ่งเป็นภาษาที่คล้ายกับภาษามนุษย์มากขึ้นโดยที่โครงสร้างของภาษาสามารถเข้าใจได้ง่ายโดยเฉพาะโปรแกรมที่ใหญ่และมีความซับซ้อนจะสามารถเขียนโปรแกรมกันเป็นทีมได้เพราะโครงสร้างของภาษาซีนั้นง่ายต่อการตรวจสอบและสามารถที่จะทำความเข้าใจกว่าภาษาเครื่องนั้นเองในตอนแรกจะขอกล่าวถึง ไมโครคอนโทรลเลอร์กันก่อนว่าเป็นอย่างไรจากนั้นจะกล่าวถึงสถาปัตยกรรม โครงสร้าง ตำแหน่งขาการใช้งานว่าเป็นอย่างไรในที่นี้จะใช้คอนโทรลเลอร์ตระกูล 8051 โดยจะกล่าวถึง chip ที่มี 40 ขา



ภาพที่ 2.1 การวางขาของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- ต้องการแหล่งจ่ายไฟ + 5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจะสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไป มีถึง256 ไบต์
- มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
- มีไทม์เมอร์ เคาน์เตอร์ ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวคเตอร์ สำหรับเบอร์ 8052 ขึ้นไป มี 8 แหล่ง 6 เวคเตอร์
- มีพอร์ทรับส่งข้อมูลอนุกรม (UART) 2 พอร์ทแบบ Full Duplex เลือกรูปได้ 4 โหมด
- มีคำสั่งในการทำ AND , OR หรือ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

2.1.2 พอร์ทของ MCS-51

8051 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆดังนี้

- Vcc (ขา 40) ต่อกับ + 5V
- Vss (ขา 20) เป็นขา GND
- พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) มีโครงสร้างแบบ Open-Drain Bi-directional
- พอร์ท 0 (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.7-P0.0) ใช้งานได้ทั้ง 2 หน้าที่ คือ แอเดรสบัสและดาตาบัส เมื่อต้องการติดต่อกับหน่วยความจำภายนอกหรือเป็นที่ทำหน้าที่เป็นไอโอพอร์ทถ้าต้องการให้ทำงานเป็นอินพุทพอร์ทต้องส่งลอจิก "1" ไปยังพอร์ท
- พอร์ท 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้ายพอร์ท 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน InternAL Pull up Register
- พอร์ท 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.7-P2.0) มีโครงสร้างคล้าย พอร์ท0
- พอร์ท 3 (ขา 10-17) ญาณเข้าไปสัญญาณต่างๆมีดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม (UART)

P3.1/TXD (Serial Output Port) เป็นขาที่ใช่ส่งข้อมูลแบบอนุกรม (UART)

P3.2/INT0 (External Interrupt 0) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 0

P3.3/INT1 (External Interrupt 1) ใช้รับสัญญาณการขัดจังหวะจากภายนอกเบอร์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P3.4/T0 (Coumter 0 External Input) ขารับสัญญาณพัลซ์อินพุทเข้าไปยังวงจรCounter 0 (เป็นอินพุทโหมดเคาน์เตอร์)

P3.5/T1 (Coumter 1 External Input) ขาสัญญาณควบคุมการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมในการที่จะทำการเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

P3.7/Rd (External Data Memory Read Strobe) ขาสัญญาณควบคุมในการที่จะทำการอ่านข้อมูลจากหน่วยความจำภายนอก

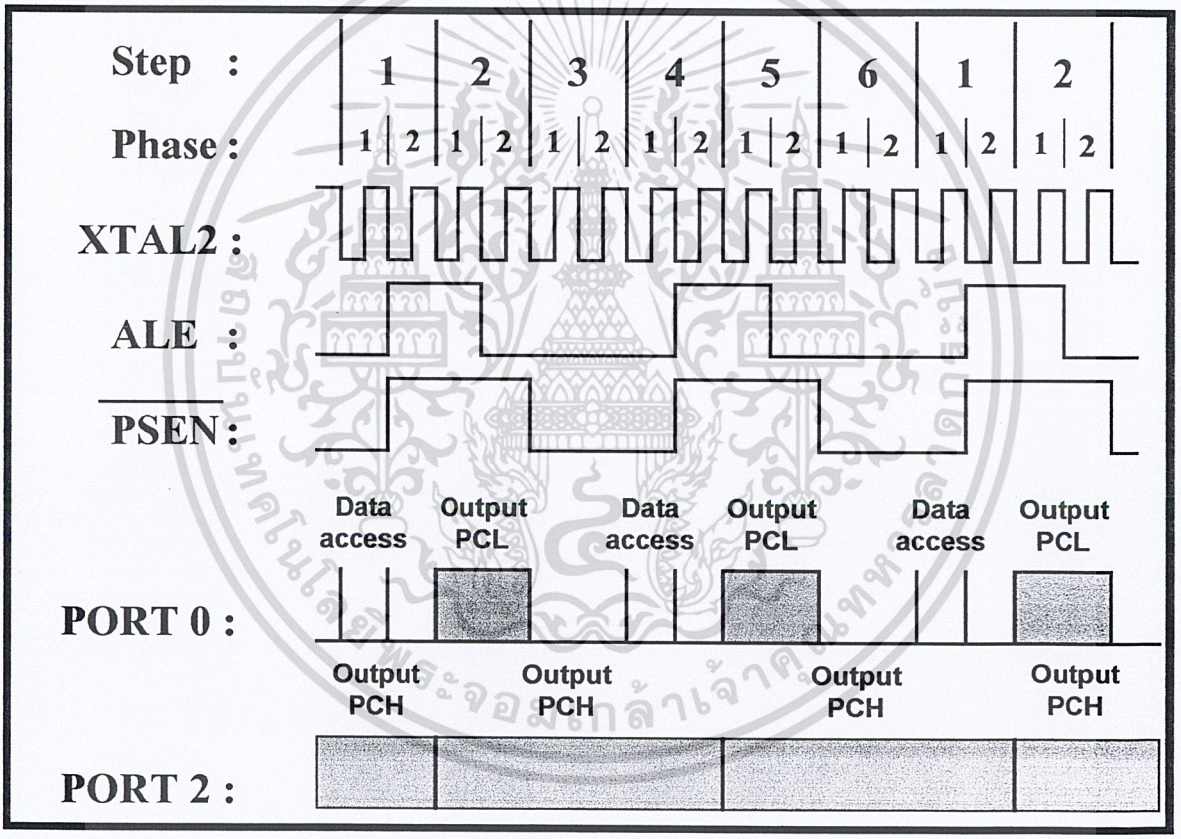
- ALE (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตซ์แอดเดรสไบต์ต่ำ (AT-AO) ที่ส่งออกมาจาก (พอร์ท0) สัญญาณนี้จะแอตทีฟทุกๆ 2 ครั้งใน 1 แมชชีนไซเคิล
- PSEN (ขา 29) เป็นขาสโตรบที่ใช้สำหรับอ่านข้อมูลจาก Program Memory ภายนอกสัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal - Program Memory จะไม่มีสัญญาณออกที่ขานี้
- EA (ขา 31) ใช้เลือกหน่วยความจำโปรแกรมภายนอก
  ป้อน “0” จะอ่านโปรแกรมจากภายนอกชิพ
  ป้อน “1” จะอ่านโปรแกรมจากภายในชิพ
- RST(ขา 9) ขารีเซ็ทจะรีเซ็ทได้ก็ต่อเมื่อป้อนลอจิก “1” เข้าที่ขานี้นานอย่างน้อย 2 แมชชีนไซเคิล
- XTAL 1 (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรออสซิสเลเตอร์ภายใน
- XTAL2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาท์พุทของวงจรออสซิสเลเตอร์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 สัญญาณต่างๆของไมโครคอนโทรลเลอร์

สัญญาณ ALE

Port 0 ส่งไบต์ต่ำเพื่ออ้างอิงตำแน่งของหน่วยความจำภายนอก นอกจากนี้ Port นี้ยังทำหน้าที่อ่านข้อมูล ดังนั้น การใช้ Port 0 นี้เพื่ออ้างอิงตำแหน่งหน่วยความจำ และ รับส่งข้อมูลจึงต้องทำตามลำดับกัน (จะทำพร้อมกันไม่ได้) ตำแหน่งอ้างอิงจึงต้องถูกบันทึกไว้ก่อนซึ่งเราเรียกว่า การแลตช์ข้อมูล จะกระทำที่ขอบขาขึ้นของสัญญาณ ALE (Address-Latch-Enable) นั่นคือตำแหน่งอ้างอิงหน่วยความจำโปรแกรมและข้อมูลทางไบต์ต่ำที่ถูกวางไว้แล้วบนแอดเดรสบัส โดยอุปกรณ์ (ชิป) Latch ส่วนตำแหน่งอ้างอิงไบต์สูงจะถูกส่งออกมาที่ Port 2 ตลอดเวลา หลังจากนั้นก็สามารถที่จะอ่านหรือเขียนข้อมูลลงบนหน่วยความจำที่อ่างอิงได้



ภาพที่ 2.2 สัญญาณ ALE

สัญญาณ ALE จะถูกสร้างขึ้น 2 ครั้งใน 1 วงรอบคำสั่ง มีกรณียกเว้น เมื่อทำคำสั่ง MOVX ซึ่งเป็นคำสั่งที่เกี่ยวข้องกับหน่วยความจำข้อมูลในกรณีนี้ ALE จะหายไป 1 ครั้ง ความถี่ของสัญญาณ ALE ที่ไม่ได้ทำงานกับหน่วยความจำข้อมูลจึงเท่ากับ

$$f_{ALE} = f_{OSC}/2$$

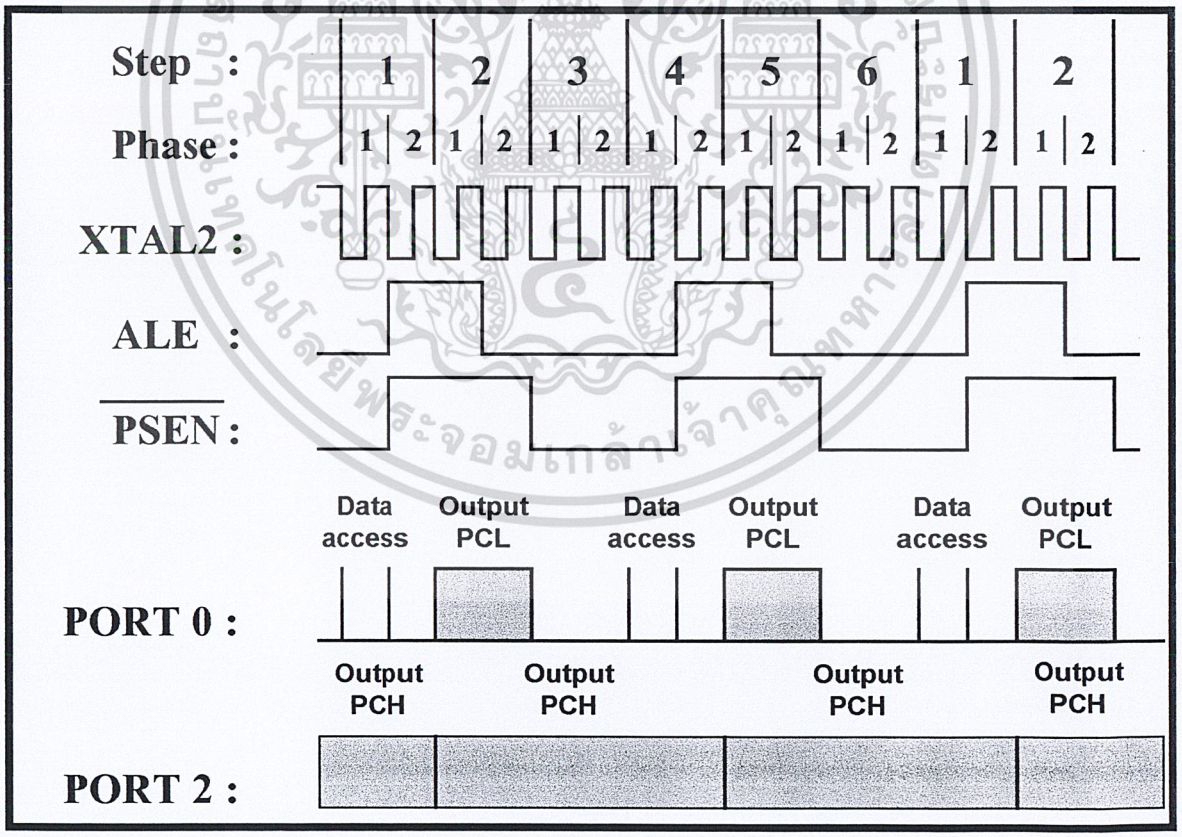
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ f_{OSC} หมายถึง ความถี่สัญญาณนาฬิกา สัญญาณความถี่ของ f_{ALE} นี้เรายังสามารถนำมาใช้ประโยชน์อย่างอื่นได้อีก

สัญญาณ PSEN

สัญญาณ PSEN(Program-Store-Enable) จะแอคทีฟที่ขอบขาลงในระหว่างที่ติดต่อกับหน่วยความจำโปรแกรม ถ้าดูจากผังเวลาแล้วจะเห็นได้ว่า สัญญาณ PSEN จะแอคทีฟหลังจากที่ได้แลตช์ตำแหน่งอ้างอิงหน่วยความจำสิ้นสุดลงแล้ว PSEN จะเชื่อมต่ออยู่กับขา OE (Output-Enable) ในระหว่างที่ PSEN (Low) แอ็กทีฟ หน่วยความจำโปรแกรมจะว่างข้อมูลที่ถูก อ้างอิงถึงลงบนบัสข้อมูล ส่วนควบคุมจะรับข้อมูลผ่านมาทาง Port 0

สัญญาณ PSEN จะมีพฤติกรรมคล้ายกับสัญญาณ ALE คือ จะหายไประหว่างที่โค้ดคำสั่งเป็น MOVX ในกรณีที่ติดต่อกับหน่วยความจำโปรแกรมภายในของ 8051 สัญญาณ PSEN นี้จะไม่ถูกสร้างขึ้นมา

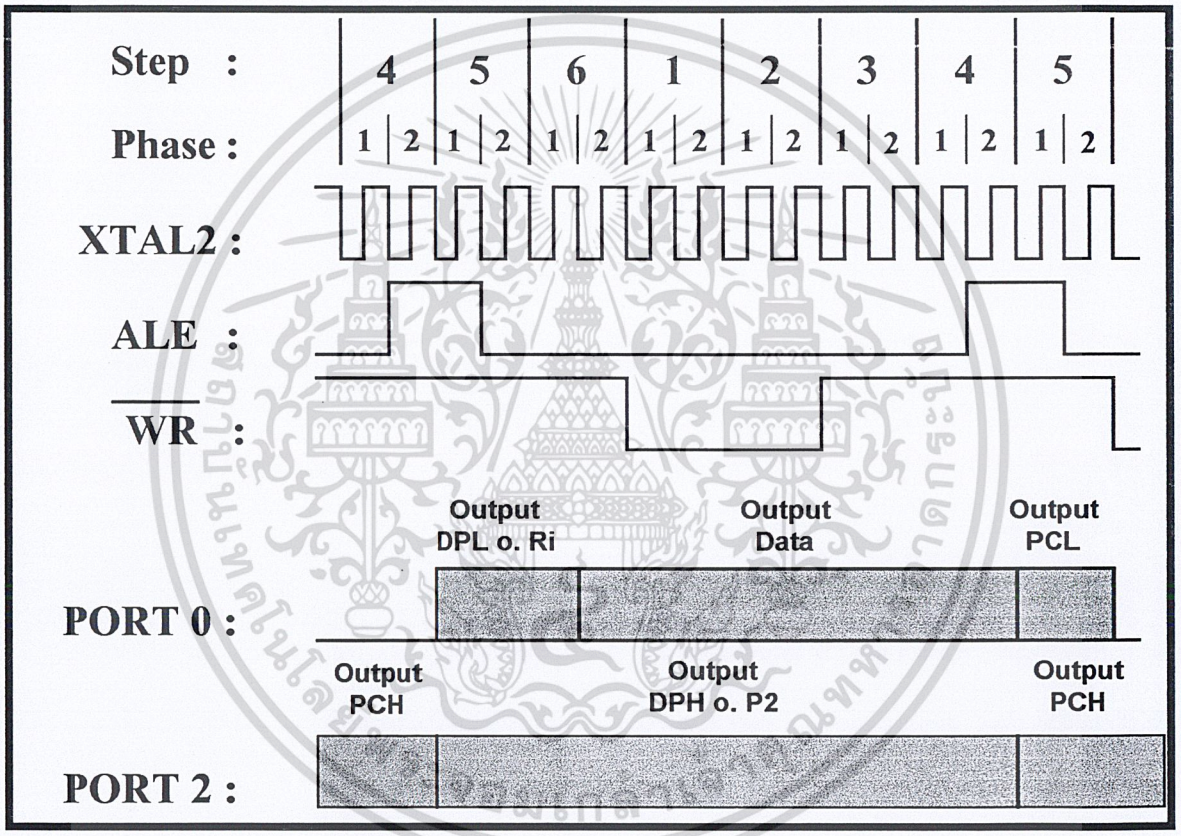


ภาพที่ 2.3 **สัญญาณ** PSEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ WR

ส่วนควบคุมจะส่งสัญญาณ WR (Write-Enable) ในระหว่างที่มีการเขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอกตำแหน่งอ้างอิงของหน่วยความจำข้อมูลจะเหมือน กับ การอ้างอิงหน่วยความจำโปรแกรมบนแอดเดรสบัส ในระหว่างที่ของขาลงของสัญญาณ ALE ตำแหน่งอ้างอิงจากไบต์ต่ำที่ Port 0 จะถูกเก็บไว้ในชิปแลตช์ในไซเกิลถัดไปส่วนควบคุมจะให้สัญญาณ WR สู่สภาวะต่ำ (Low) ซึ่งทำให้ขา WE (Write Enable) ของหน่วยความจำข้อมูลภายนอกแอ็กทีฟ และจะเขียนข้อมูลเข้าไปในตำแหน่งที่ถูกอ้างอิง



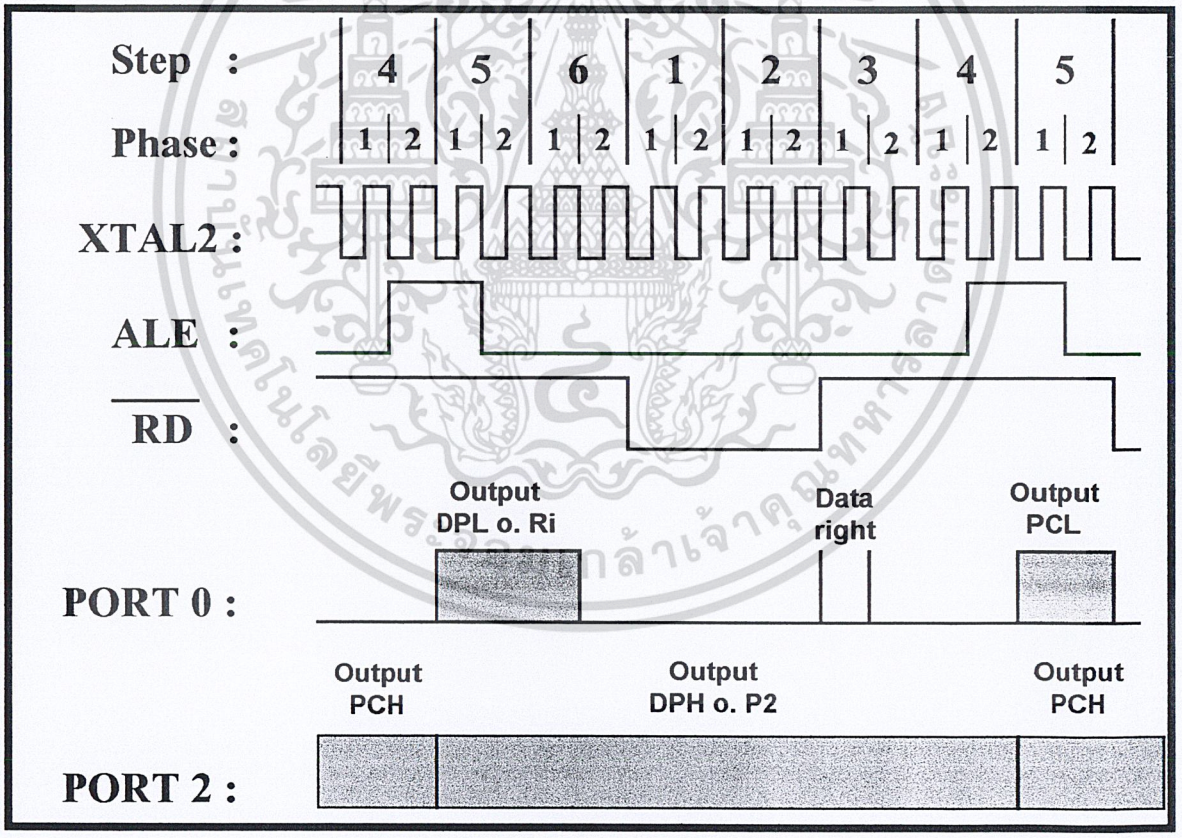
ภาพที่ 2.4 สัญญาณ WR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ RD

โปรเซสเซอร์จะสร้างสัญญาณ RD(Read Enable) ในขณะที่อ่านข้อมูลออกจากหน่วยความจำข้อมูลภายนอก หลังจากที่ไบต์ต่ำอ้างอิงตำแหน่งได้ถูกแลตช์แล้ว โปรเซสเซอร์จะทำให้ RD สู่สภาวะต่ำ (Low) ดังนั้นที่ขา OE (Output - Enable) จึงแอ็กทีฟ หน่วยความจำข้อมูลก็จะวางข้อมูลในตำแหน่งที่ถูกอ้างอิงบนบัสข้อมูล และ ส่วนควบคุมก็จะอ่านข้อมูลจากบัสข้อมูลซึ่งผ่านมาทาง Port 0

จากบล็อกไดอะแกรมเวลาจะเห็นว่าสัญญาณ ALE แอ็กทีฟเพียงครั้งเดียวระหว่างทำคำสั่ง MOVX ตรงที่ S2P1 ที่ซึ่งให้ขอบขาสัญญาณลงของ ALE สัญญาณ RD หรือ WR จะแอ็กทีฟตรงจุดเวลานี้องที่จะต้องไม่มีการเปลี่ยนแปลง ของตำแหน่งอ้างอิงแอดเดรสที่หน่วยความจำข้อมูล สัญญาณ PSEN ที่เวลานี้ต้องหายไป เพื่อที่ขณะที่คำสั่ง MOVX จะมีเพียงหน่วยความจำข้อมูลเท่านั้นที่วางข้อมูลลงบนบัสข้อมูลไม่ใช่ EPROM



ภาพที่ 2.5 สัญญาณ RD

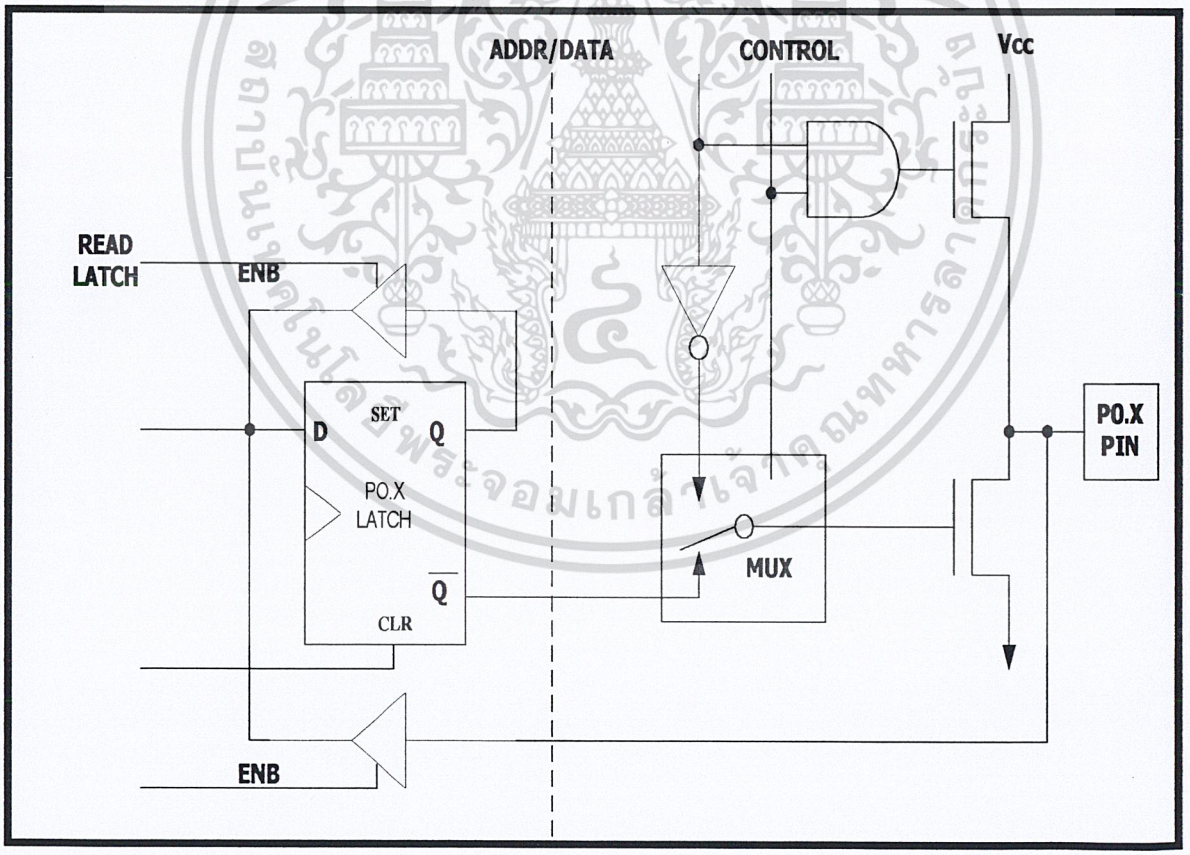
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 Port สัญญาณของไมโครคอนโทรลเลอร์

Port 0 เป็นพอร์ตอินพุตเอาท์พุต 2 ทางแบบ Open drain ขนาด 8 บิต P0.1 - P0.7 เมื่อใช้เป็นเอาต์พุตสามารถต่อกับไอซี TTL ตระกูล LS ได้ 8 ตัว เมื่อต้องการใช้งานเป็น อินพุต ต้อง ส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้ลอยซึ่งจเป็นอิมพีแดนซ์สูง

พอร์ต P0 จะทำงานอีกหน้าที่หนึ่งคือ เป็นมัลติเพล็กซ์ของสัญญาณตำแหน่งค้านต่ำ และสัญญาณข้อมูลในกาติดต่อกับหน่วยความจำภายนอก ในการทำงานในลักษณะนี้จะใช้การพูลอัพ (Pull up) จากภายในที่สามารถจ่ายกระแสให้กับอินพุตของ TTL ได้ 8 ตัว นอกจาก2 หน้าที่ดังกล่าวแล้ว พอร์ต P0 ยังใช้เป็นตัวรับข้อมูลในช่วงการโปรแกรม EPROM และเป็นตัวส่งข้อมูลออกมาในช่วงการตรวจสอบโปรแกรมภายใน ROM หรือ EPROM ซึ่งจะต้องใช้พูลอัพจากภายนอกในขณะทำการตรวจสอบโปรแกรม

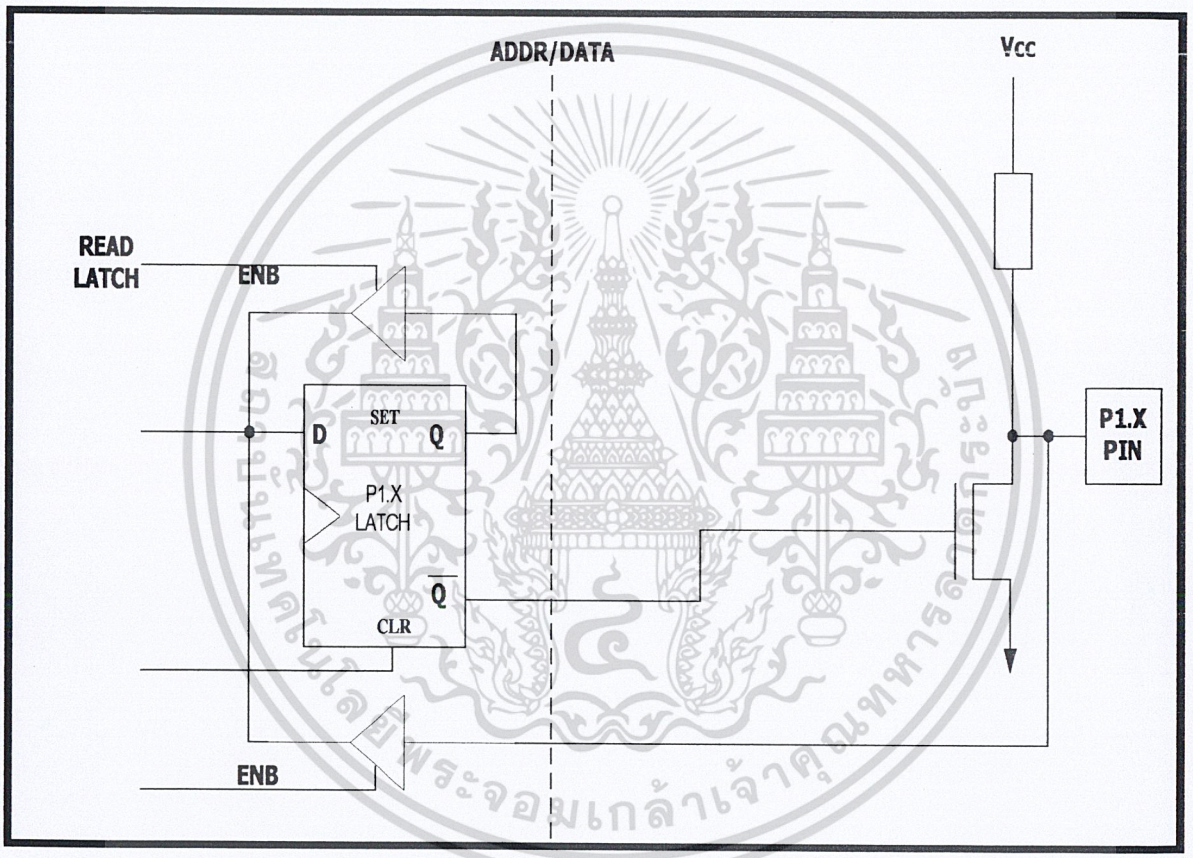
อุปกรณ์ที่นำมาแลตช์ตำแหน่ง A0-A7 ที่ออกมาจากพอร์ต P0 คือ ไอซี 74LS373 ซึ่งเป็น 8 บิต แลทช์เราสามารถนำมาต่อใช้งานได้ดังแสดงในรูป



ภาพที่ 2.6 Port 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

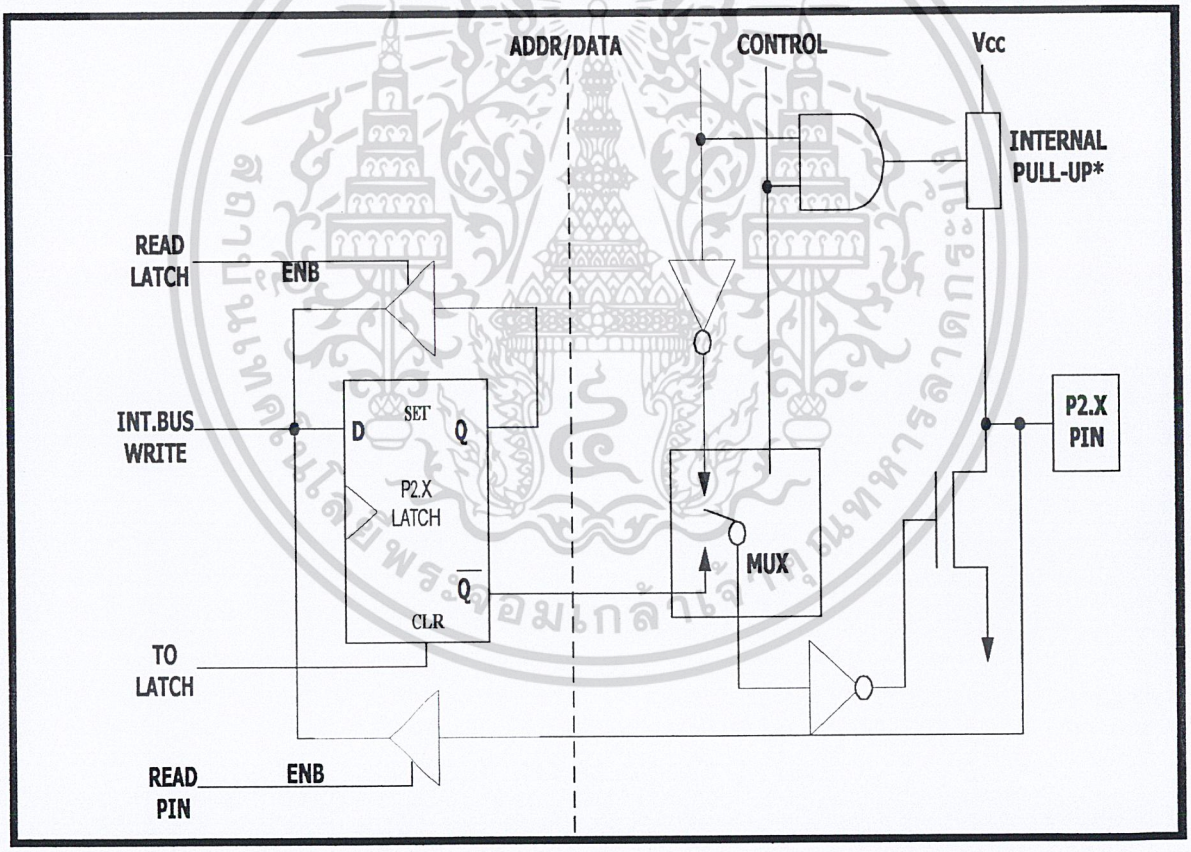
Port 1 เป็นพอร์ตอินพุต / เอาต์พุทแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตจะต่อกับอินพุตของ TTK ตระกูล LS ได้ 4 ตัว เมื่อต้องการใช้เป็นอินพุตต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปที่พอร์ตนี้ก่อน เพื่อทำให้เกิดพูลอัพภายใน เมื่อมีสัญญาณอินพุตเป็น 0 เข้ามาจะทำให้ พอร์ตจ่ายกระแสออกเนื่องจากมีพูลอัพอยู่ภายใน นอกจากนี้พอร์ต P1 ยังทำหน้าที่รับตำแหน่งด้านต่ำ ในช่วงของการโปรแกรม EPROM และช่วงการตรวจสอบโปรแกรมใน ROM หรือ EPROM อีกด้วย สำหรับเบอร์ 8032 AH และ 8052 AH ขา P1.0 และ P1.1 จะทำหน้าที่เป็น T2 และ T2EX อีกหนึ่งหน้าที่



ภาพที่ 2.7 Port 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

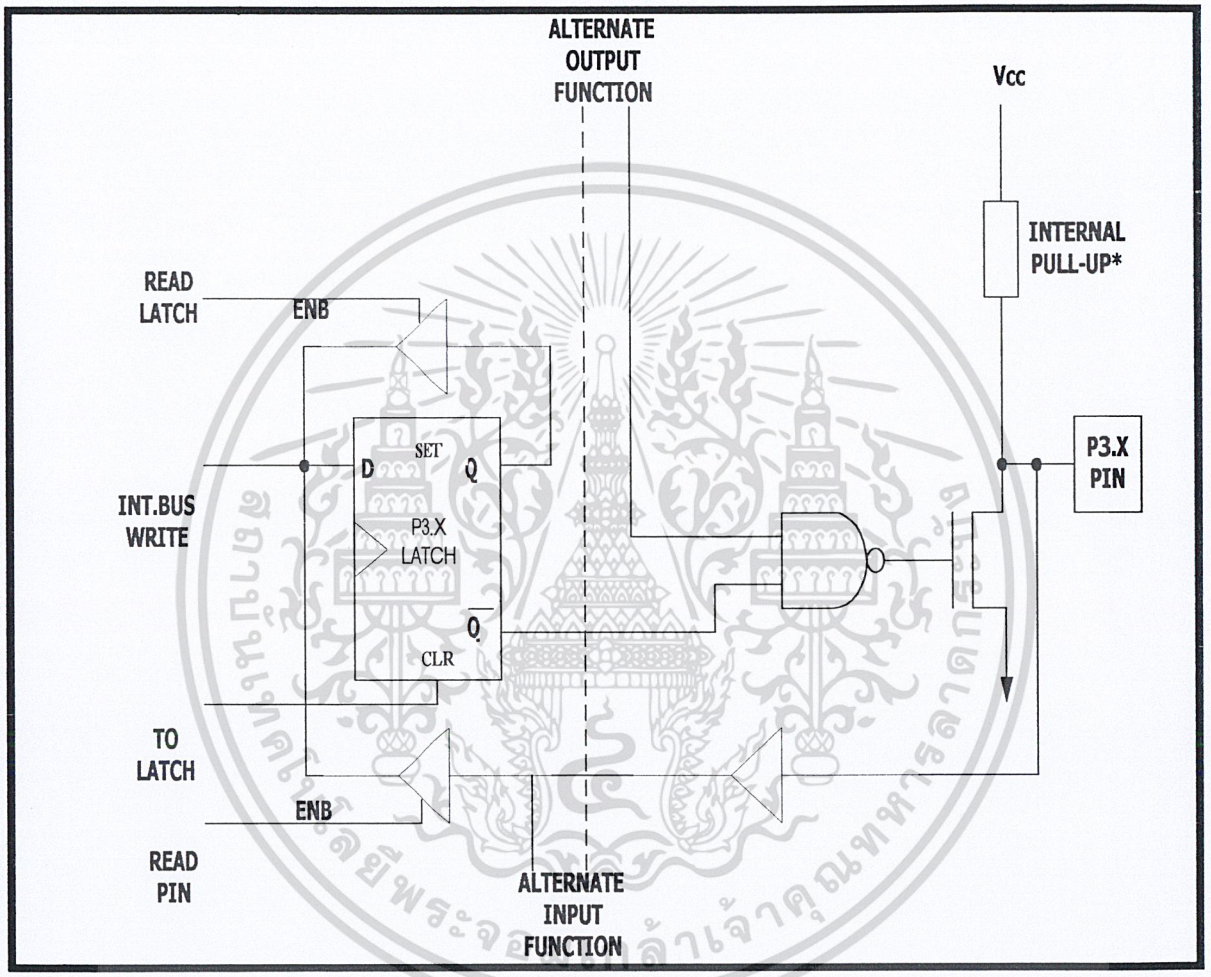
Port 2 เป็นพอร์ตอินพุต / เอาต์พุตแบบ 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตจะต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุต ต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเข้ามาเป็น 0 จะทำให้พอร์ต P2 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน ในระหว่างการติดต่อกับโปรแกรมภายนอกหรือการติดต่อกับหน่วยความจำข้อมูลภายนอกที่มีการอ้างอิงตำแหน่งแบบ 16 บิต ในที่นี้พอร์ต P2 จะส่งตำแหน่งไบต์สูงออกไป ซึ่งการทำงานในลักษณะนี้จะมีการพูลอัพภายในอยู่ ในช่วงการติดต่อ กับ หน่วยความจำข้อมูลภายนอก ที่ใช้การอ้างอิงตำแหน่งแบบ 8 บิต (คำสั่ง MOVX @Ri) สัญญาณที่ขาของพอร์ต 2 จะมีค่าเท่ากับรีจิสเตอร์ P2 ที่อยู่ใน SFR นอกจากนี้พอร์ต P2 ยังทำหน้าที่รับตำแหน่งไบต์สูง ในช่วงของการโปรแกรม EPROM และการตรวจสอบโปรแกรมใน ROM และ EPROM อีกด้วย



ภาพที่ 2.8 Port 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 3 เป็นพอร์ตอินพุต / เอาต์พุต 2 ทาง ขนาด 8 บิต ที่มีพูลอัพอยู่ภายใน ในกรณีเอาต์พุตสามารถต่อกับอินพุตของ TTL ตระกูล LS ได้ 4 ตัว เมื่อต้องการทำเป็นอินพุต ต้องเริ่มต้นด้วยการส่งค่าลอจิก 1 ออกไปที่พอร์ตก่อนเพื่อทำให้เกิดการพูลอัพภายใน เมื่อสัญญาณอินพุตเข้ามาเป็น 0 จะทำให้พอร์ต P3 จ่ายกระแสออกเนื่องจากการพูลอัพภายใน นอกจากนี้พอร์ต P3 ยังทำหน้าที่เป็นสัญญาณอื่นๆอีกดังนี้



ภาพที่ 2.9 Port 3

เมื่อต้องการใช้งานพอร์ต P3 ให้ทำหน้าที่เป็นสัญญาณต่างๆ จะต้องเริ่มต้นด้วยการส่งค่าลอกจิก1 ออกไปแลทช์ที่พอร์ต P3 ก่อนเพื่อให้เกิดการพูลอัพขึ้นภายใน หากเรากำหนดให้มีค่าลอจิก 0 จะทำให้สัญญาณที่ขาต่างๆ มีค่าเป็น 0 ตลอดเวลา

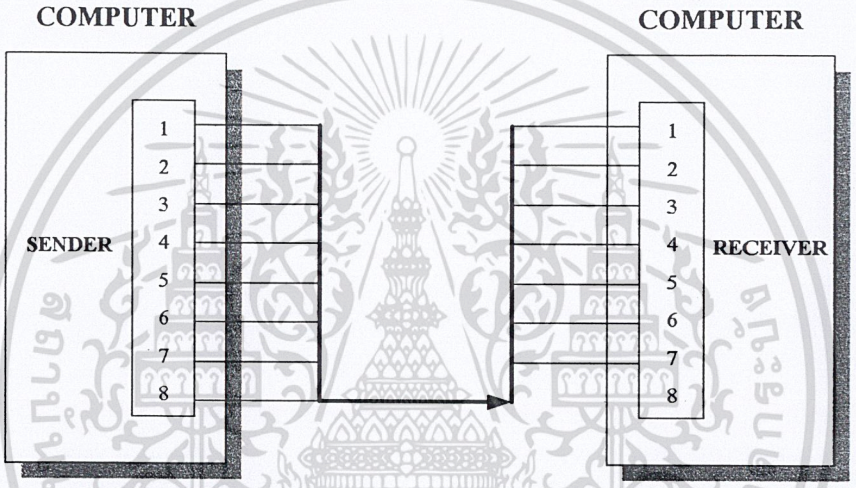
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ลักษณะของการสื่อสาร

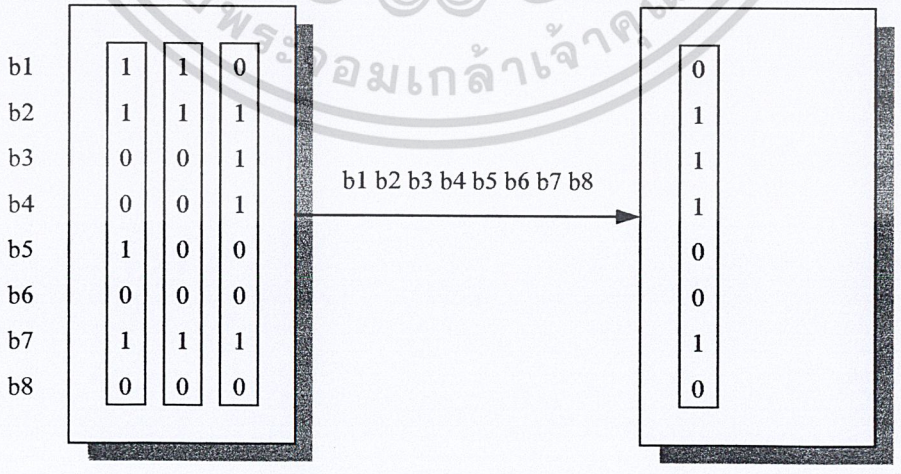
ในการสื่อสารหรือการส่งข้อมูลโดยใช้คอมพิวเตอร์ส่วนบุคคลนั้น มีรูปแบบในการสื่อสารที่สำคัญอยู่ 2 รูปแบบ คือ

2.2.1 การสื่อสารแบบอนุกรม(Serial Communication)

เป็นการสื่อสารโดยการส่งข้อมูลทีละบิท(bit) ผ่านสายสัญญาณเส้นเดียวจนครบทั้ง 8 บิท(bit) โดยจะส่งบิทต่ำ(LSB)ออกไปก่อน ซึ่งการสื่อสารและการส่งข้อมูลแบบอนุกรมจะแสดงให้เห็นได้ดังภาพที่ 2.10 และ 2.11



ภาพที่ 2.10 การสื่อสารแบบอนุกรม

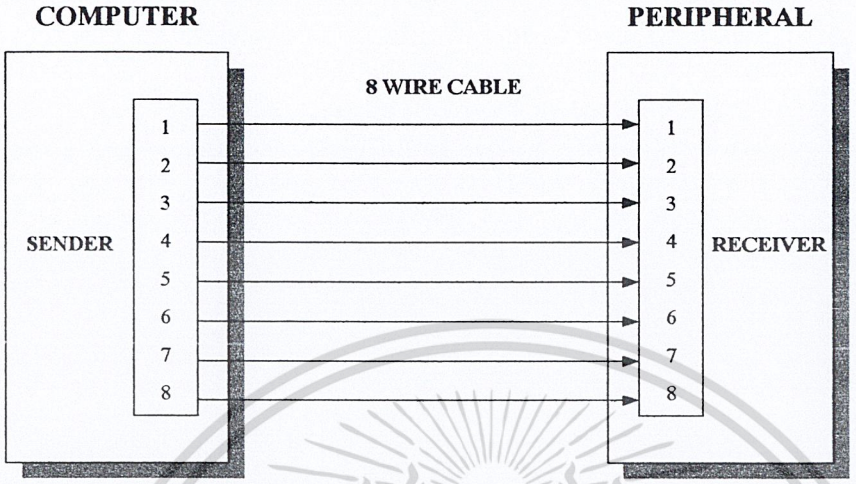


ภาพที่ 2.11 การส่งข้อมูลแบบอนุกรม

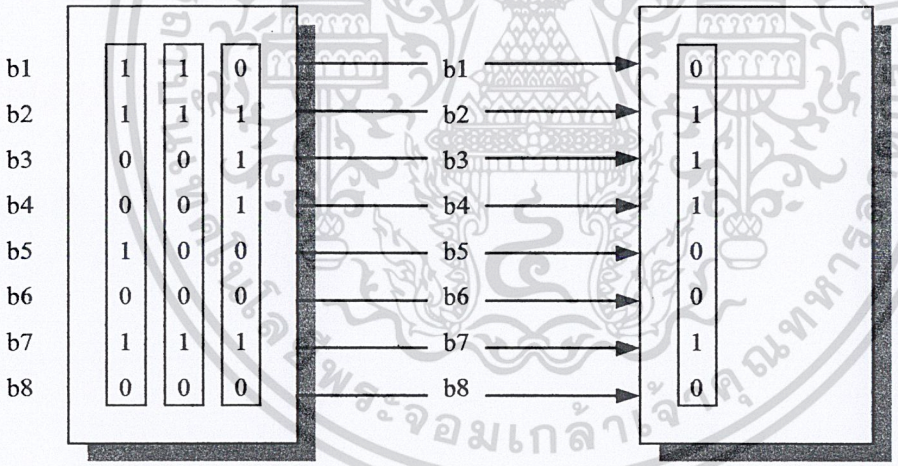
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 **การสื่อสารแบบขนาน**(Parallel Communication)

เป็นการสื่อสารและการส่งข้อมูลแบบขนานจะแสดงให้เห็นได้ดังภาพที่ 2.12 และ 2.13



ภาพที่ 2.12 แสดงลักษณะการสื่อสารแบบขนาน



ภาพที่ 2.13 แสดงการส่งข้อมูลแบบขนาน

จะเห็นได้ว่าการสื่อสารแบบขนาน มีข้อดีคือทำให้สามารถส่งข้อมูลได้ทีละมากๆ และ มีความรวดเร็วกว่าการส่งข้อมูลแบบอนุกรม แต่การสื่อสารแบบขนานก็มีข้อจำกัด คือ ไม่สามารถส่งข้อมูลได้ระยะทางไกลๆได้และยังต้องใช้สายสัญญาณหลายเส้น ในการส่งข้อมูลทำให้สิ้นเปลืองกว่าการสื่อสารแบบอนุกรม รวมทั้งทำให้ไม่สะดวกในการใช้งาน

ตัวอย่างของการสื่อสารแบบอนุกรม เช่นการสื่อสารด้วย MODEM และ เมาส์

ตัวอย่างของการสื่อสารแบบขนาน เช่น เครื่องพิมพ์ (Printer) และ การสื่อสารทางพอร์ทขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3 ลักษณะการสื่อสารตามมาตรฐาน

ในการติดต่อสื่อสารและการรับส่งข้อมูลนั้นจำเป็นตองมีมาตรฐาน ในการสื่อสารเพื่อให้สามารถติดต่อสื่อสารกันได้โดยไม่มีปัญหาในการสื่อสรเกิดขึ้น โดยส่วนมากมาตรฐานในการสื่อสารจะกล่าวถึงคุณสมบัติสำคัญๆที่ใช้ในการสื่อสาร เช่น ลักษณะทางกล ลักษณะของสัญญาณไฟฟ้า ลักษณะการทำงานที่ใช้เชื่อมต่อ

ในปัจจุบันสัญญาณทีใช้ในการสื่อสารส่วนใหญ่จะใช้สัญญาณดิจิตอล(Digital Signal) เป็นสัญญาณหลักและสัญญาณที่ใช้ในการติดต่อสื่อสารส่วนมากจะเป็น สัญญาณดิจิตอลที่มีระดับสัญญาณแบบ TTL ซึ่งทำการติดต่อสื่อสารในลักษณะของระดับสัญญาณแบบ TTL แล้วจะสามารถทำการติดต่อสื่อสารได้โดยตรง แต่ในหลายกรณีที่มีความจำเป็นที่จะต้องสื่อสารด้วยระดับสัญญาณในระดับอื่นที่ไม่ใช่ระดับสัญญาณแบบ TTL หรือไม่ได้เป็นสัญญาณดิจิตอล ซึ่งจะทำให้เกิดปัญหาในการสื่อสาร แต่ปัญหาดังกล่าวสามารถแก้ไขได้โดยอาศัยการแปลงสัญญาณที่ต้องการสื่อสารให้เป็นสัญญารตามมาตรฐาน โดยที่แต่ละมาตรฐานจะมีข้อกำหนดที่แตกต่างกันออกไป ในที่นี้จะกล่าวถึงมาตรฐานสำคัญๆที่ใช้ในการสื่อสารดังนี้

2.3.1 ลักษณะการเชื่อมต่อตามมาตรฐาน RS-232-C

เป็นมาตรฐานที่ได้รับการพัฒนาและถูกใช้งานอย่างแพร่หลาย รวมทั้งยังใช้เป็นมาตรฐานในการเชื่อมต่อ DTE(Data Terminal Equipment) เข้ากับ DCE (Data Communication Equipment) เช่น การเชื่อมต่อคอมพิวเตอร์ส่วนบุคคลเข้ากับโมเด็ม

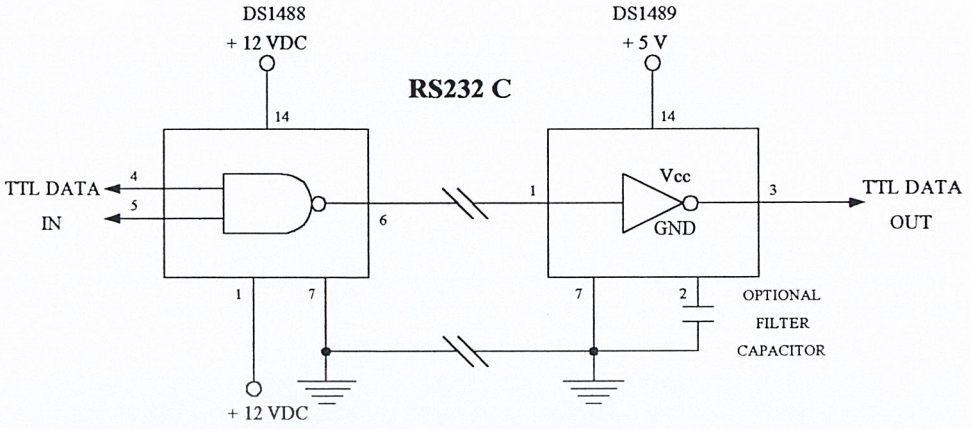
ลักษณะการเชื่อมต่อตามมาตรฐาน RS-232-C นั้นจะใช้สายสัญญาณเพียงคู่เดียวในการส่งสัญญาณ โดยสามารถส่งสัญญาณไปได้ในทิศทางเดียว อัตราเร็วสูงสุดในการส่งข้อมูลมีค่าเท่ากับ 20 kbps(กิโลบิทต่อวินาที) และระยะทางที่ใช้ในการส่งข้อมูลไม่ควรเกิน 50 ฟุต สำหรับการส่งข้อมูลจะใช้ระดับแรงดัน แทนค่าทางตรรก(Logic) ของข้อมูล โดยระดับแรงดันที่มีค่าอยู่ระหว่าง 5 โวลท์ ถึง –15 โวลท์ จะแทนค่าสถานะ 0 และระดับแรงดันที่มีค่าอยู่ระหว่าง+5 โวลท์ ถึง + 15 โวลท์ จะแทนค่าสถานะ 1 ส่วนในช่วงระดับแรงดัน ที่มีค่าอยู่ระหว่าง-5 โวลท์ ถึง + 15 โวลท์ นั้นจะใช้แทนการแบ่งแยกระดับสถานะของสัญญาณระหว่างสถานะ 0 และสถานะ 1

ตัวอย่างของอุปกรณ์ที่ใช้ในการเชื่อมต่อตามาตรฐานRS-232-Cได้แก่เทอร์มินัล(Terminal) พล็อตเตอร์(Plotter) ลอจิกอนาไลเซอร์(Logic analyzer) และเครื่องพิมพ์

ตัวอย่างของวงจรที่ใช้ในการเชื่อมต่อตามาตรฐาน RS-232-C นั้นจะแสดงให้เห็นได้ดังในภาพที่ 2.14 โดยจะมีการแปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณตามที่กำหนดไว้ในมาตรฐาน RS-232-C แล้วส่งไปตามสายสัญญาณและแปลงกลับจากระดับแรงดันสัญญาณในมาตรฐาน RS-232-C ไปเป็นระดับสัญญาณแบบ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50426



ภาพที่ 2.14 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-232-C

2.3.2 **ลักษณะการเชื่อมต่อตามมาตรฐาน** RS-423

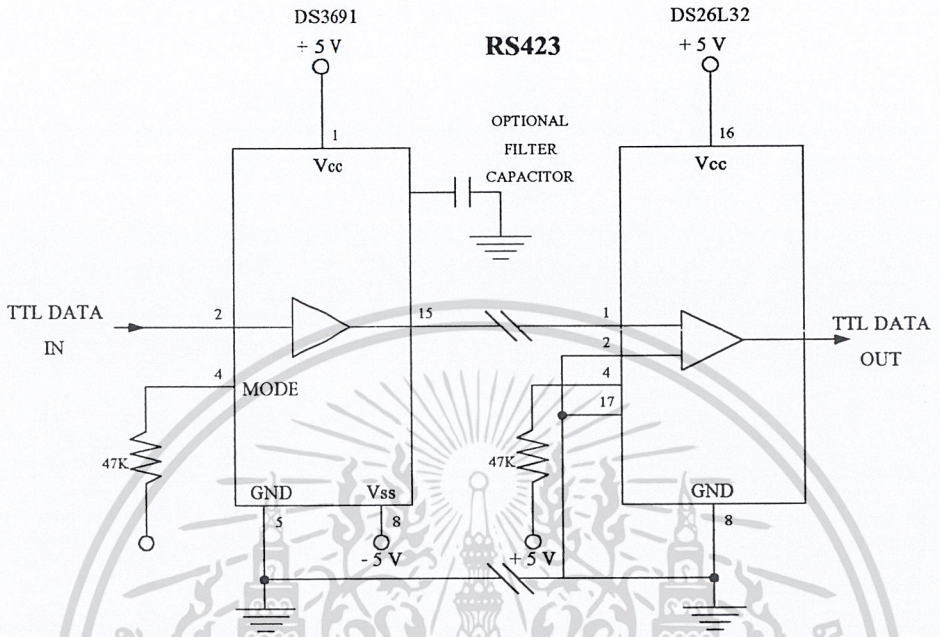
มาตรฐาน RS-423 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-232-C เพื่อทำให้มี ประสิทธิภาพในการสื่อสารมากขึ้น โดยมีการพัฒนาให้ตัวรับข้อมูลมีความสามารถในการรับข้อมูลสูงมากขึ้นและอัตราเร็วในการส่งข้อมูลได้ถูกพัฒนาให้มีอัตราเร็วสูงกว่า อัตราการส่งข้อมูลตามมาตรฐาน RS-232-C ซึ่งอุปกรณ์ที่ผลิตขึ้นมาใหม่ๆ มักจะใช้การเชื่อมต่อแบบนี้ โดยเฉพาะอุปกรณ์ที่ต้องการให้อัตราเร็วในการส่งข้อมูลมีค่าสูงขึ้นกว่าเดิม

ลักษณะการเชื่อมต่อตามมาตรฐาน RS-423 นั้นจะใช้สายสัญญาณเพียงคู่เดียวช่วยในการส่งสัญญาณ โดยสามารถส่งสัญญาณไปได้ทิศทางเดียว อัตราเร็วสูงสุดในการส่งข้อมูลมีค่าเท่ากับ 100 kbps ที่ระยะห่าง 40 ฟุต ตัวรับข้อมูลเป็นแบบ Balance 1ดังนั้นตัวรับข้อมูล (reciver) จึงทำการรับ ข้อมูลขยายความแตกต่างของสัญญาณระหว่างสายกราวด์กับตัวขับสัญญาณ (Driver) การทำเช่นนี้ช่วยแก้ปัญหาในกรณีที่เกิดความแตกต่างระหว่างแรงดันที่กราวด์ของ ตัวรับข้อมูลกับตัวขับสัญญาณ สำหรับในการส่งข้อมูลจะใช้ระดับแรงดันแทนค่าทางตรรกของข้อมูลโดยระดับแรงดันที่มีค่าอยู่ระหว่าง –4 โวลท์ถึง + 6 โวลท์ จะแทนค่าสถานะ 0 และระดับแรงดันที่มีค่าอยู่ระหว่าง+4 โวลท์ ถึง + 6โวลท์ จะแทนค่าสถานะ 1 ส่วนในช่วงระดับแรงดัน ที่มีค่าอยู่ระหว่าง -4 โวลท์ ถึง + 4 โวลท์ นั้นจะใช้แทนการแบ่งแยกระดับสถานะของสัญญาณระหว่างสถานะ 0 และสถานะ 1

ตัวอย่างของอุปกรณ์ที่ใช้ในการเชื่อมต่อตามาตรฐาน RS-423 จะทำการแสดงให้เห็นได้ดังภาพที่ 2.15 โดยจะมีการแปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณตามที่กำหนดไว้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน RS-423แล้วส่งไปตามสายสัญญาณ และ แปลงกลับจากระดับแรงดันสัญญาณใน มาตรฐาน RS-423 ไปเป็นระดับสัญญาณแบบ TTL



ภาพที่ 2.15 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423

2.3.3 ลักษณะการเชื่อมต่อตามมาตรฐาน RS-422

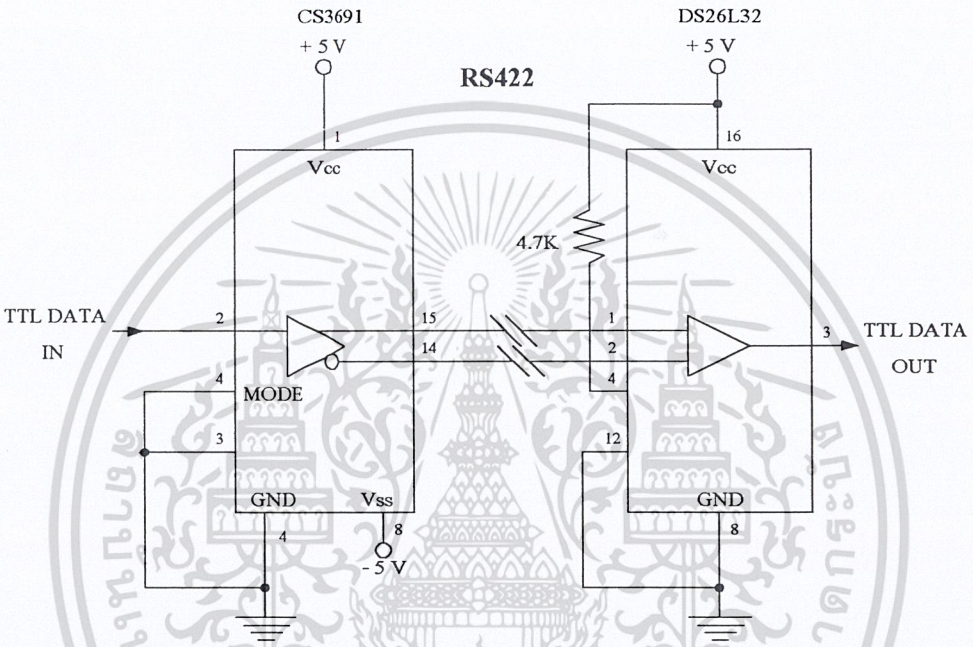
มาตรฐาน RS-422 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-423 ช่วยทำให้มีประสิทธิภาพในการสื่อสารมากขึ้น โดยมีการพัฒนาให้อัตราเร็วในการส่งข้อมูลมูลมีค่าสูงกว่าอัตราการส่งข้อมูลตามมาตรฐาน RS-423 และระยะทางที่ใช้ในการส่งข้อมูลระหว่างตัวส่ง และตัวรับก็มากกว่าในมาตรฐาน RS-423 นอกจากนี้ยังทำให้มีความไวต่อสัญญาณมากกว่าในมาตรฐาน RS-423 อีกด้วย

ลักษณะการเชื่อมต่อตามมาตรฐานRS-422 นั้นจะใช้สายสัญญาณเพียงคู่เดียวในการส่งสัญญาณโดยสามารถส่งสัญญาณไปได้ใน ทิศทางเดียวอัตราเร็วสูงสุดใน การส่งข้อมูลมีค่า เท่ากับ 10Mbps (กิโลบิทต่อวินาที) โดยระยะทางที่ใช้ในการส่งข้อมูลสามารถขยาย ได้ถึง 4000 ฟุต สำหรับการส่งข้อมูลจะใช้ระดับแรงดันแทนค่าทางตรรกของข้อมูล โดยระดับแรงดันที่มีค่า อยู่ระหว่าง –2 โวลท์ถึง - 6 โวลท์ จะแทนค่าสถานะ 0 และระดับแรงดันที่มีค่าอยู่ ระหว่าง+2โวลท์ ถึง + 6 โวลท์ จะแทนค่าสถานะ 1 ส่วนในช่วงระดับแรงดัน ที่มีค่าอยู่ระหว่าง-2โวลท์ ถึง + 2 โวลท์ นั้นจะใช้แทนการแบ่งแยกระดับสถานะของสัญญาณระหว่างสถานะ 0 และสถานะ 1 นอกจากนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวรับสัญญาณยังสามารถจับสัญญาณที่มีระดับแรงดันต่ำกว่า 200 mV ได้อีกด้วยทำให้มีความไว ต่อสัญญาณเพิ่มขึ้นด้วย

ตัวอย่างของอุปกรณ์ที่ใช้ในการเชื่อมต่อตามาตรฐาน RS-422 จะทำการแสดงให้เห็นได้ดังภาพที่ 2.16 โดยจะมีการแปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณตามที่กำหนดไว้ในมาตรฐานRS-422แล้วส่งไปตามสายสัญญาณและแปลงกลับจากระดับแรงดันสัญญาณในมาตรฐาน RS-422 ไปเป็นระดับสัญญาณแบบ TTL

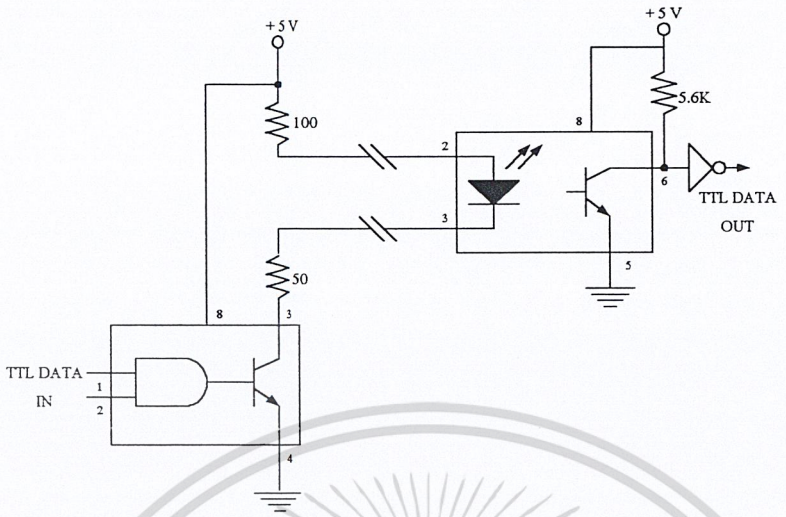


ภาพที่ 2.16 แสดงวงจรขับและรับสัญญาณที่ใช้กับมาตรฐาน RS-423

2.3.4 **การเชื่อมต่อแบบ** Current Loop

ลักษณะการเชื่อมต่อแบบ Current Loop นั้นจะทำการใช้สายสัญญาณเพียงคู่เดียวในการส่งสัญญาณ โดยมีหลักการส่งสัญญาณคือ เมื่อลูปถูกปิดวงจร (close loop) ระดับแรงดันนั้นจะถูกเปลี่ยนเป็นกระแสตามสมการ V=IR แต่เมื่อลูปถูกเปิดวงจร (open loop) จะไม่มีกระแสไหลในลูป ทำให้สามารถส่งข้อมูลได้โดยอาศัยการไหลของกระแสในวงจร เนื่องจากวงจรของการเชื่อมต่อแบบนี้มีอิมพีแดนซ์ต่ำดังนั้นจึงทนต่อสัญญาณรบกวนได้ดี เรามักใช้การเชื่อมต่อแบบนี้ในกรณีที่ต้องเดินสายผ่านบริเวณที่มีสัญญาณรบกวนมากๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



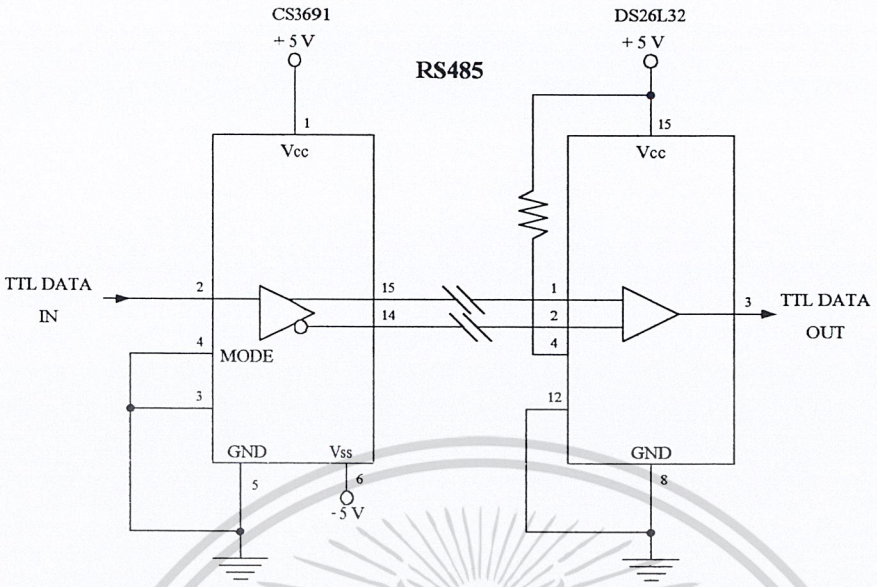
ภาพที่ 2.17 แสดงวงจรเชื่อมต่อแบบ Current loop

จากภาพที่ 2.17 แสดงให้เห็นถึงการเชื่อมต่อแบบ Current loop ซึ่งใช้แยกกราวด์ของระบบ 2 ระบบ ออกจากกัน ตามในรูปเป็นการแยก Current loop ของวงจรผลิตสัญญาณ (Transmitter) ออกจากวงจรรับสัญญาณ (Receiver) นอกจากนี้ยังสามารถใช้ส่งข้อมูลได้ในอัตราเร็ว 50 Kbps โดยระยะทางที่ใช้ในการส่งสัญญาณไม่เกิน 3000 ฟุต ซึ่งตัวจำกัดระยะทางที่ใช้ในการส่งสัญญาณคือ ความต้านทานของสายที่ประกอบกันอยู่เป็น Current Loop นี้มีส่วนดีก็ คือสามารถส่งข้อมูลได้ในระยะทางไกลๆและค่าใช้จ่ายในการเชื่อมต่อแบบนี้ราคาไม่สูงนัก

2.3.5 ลักษณะการเชื่อมต่อตามมาตรฐาน RS-485

มาตรฐาน RS-485 เป็นมาตรฐานที่ได้รับการพัฒนามาจากมาตรฐาน RS-422 ช่วยทำให้มีประสิทธิภาพในการสื่อสารเพิ่มขึ้นเป็นอย่างมาก โดยมีการพัฒนาให้วงจรของตัวขับสัญญาณเป็น 3 สถานะ(Tri State) ทำให้สามารถส่งข้อมูลได้สองทิศทางบนสายคู่เดียวและสามารถต่อเครือข่ายแบบ Multidrop ซึ่งอุปกรณ์หลายๆตัว ทำให้สามารถรับและส่งข้อมูลแบบ (Half- Duplex) บนสายสัญญาณคู่เดียวได้

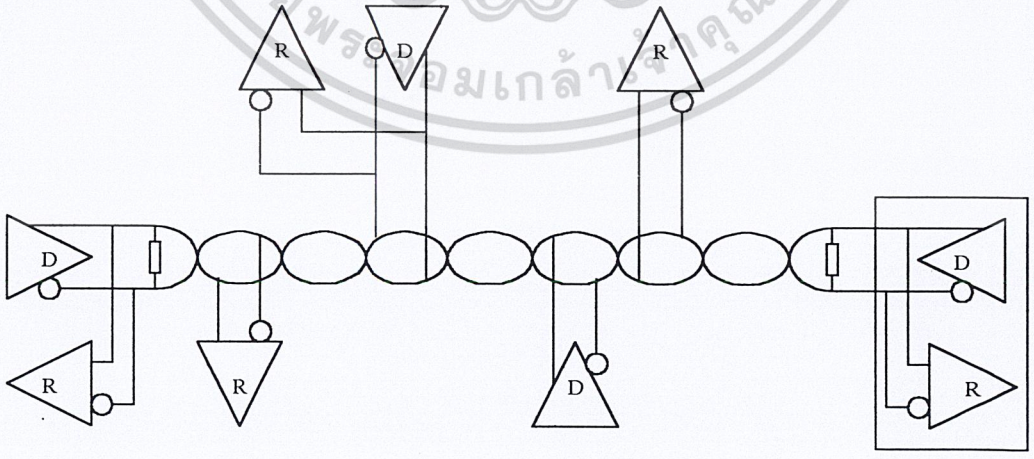
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.18 แสดงวงจรขับและรับสัญญาณที่ใช้มาตรฐาน RS-485

2.3.6 **เหตุผลในการเลือกใช้มาตรฐาน** RS-485

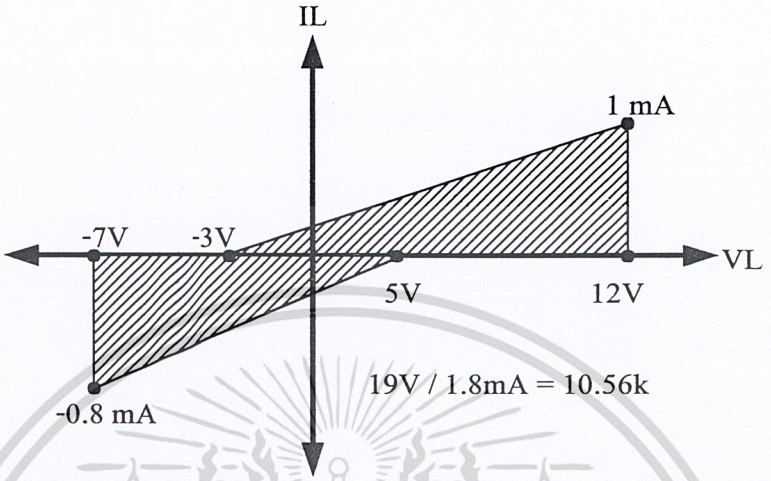
คุณสมบัติในการสื่อสารแบบสองทิศทางในคนละเวลา(Half- duplex)เนื่องจากตัวรับและตัวส่งตามมาตรฐาน RS-485 ถูกออกแบบให้เป็นแบบ 3 สถานะ (Tri-State) จึงสามารถช่วยทำการสื่อสารได้สองทิศทางบนสายสัญญาณเพียงคู่เดียว (bidirection) ซึ่งมีข้อดีก็คือทำให้สะดวกและประหยัดต่อการใช้งาน



ภาพที่ 2.19 แสดงการเชื่อมต่อเครือข่ายตามมาตรฐาน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

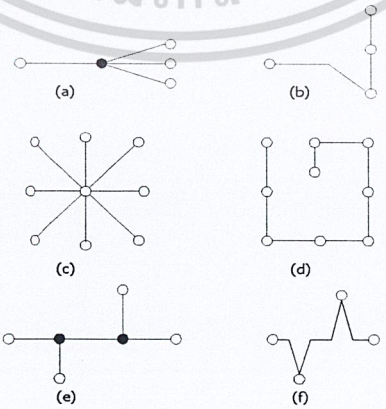
นอกจากนี้ตามมาตรฐาน RS-485ยังสามารถที่จะเชื่อมต่อเป็นเครือข่ายได้มากถึง 32 จุดจะแสดงให้เห็นได้ดังภาพที่ 2.20



ภาพที่ 2.20 แสดงช่วงเวลาของกระแสและแรงดันในการทำงาน(Loading Curve of a FULL UL)

2.3.7 **คุณสมบัติการเชื่อมต่อเป็นเครือข่าย**

ตามมาตรฐาน RS-485 นั้นเครือข่ายสามารถเชื่อมต่อได้หลายจุดบนสายสัญญาณเพียงคู่เดียว (Multidrop Transceivers) และการเชื่อมต่อตามาตรฐาน RS-485 นั้นจะมีพื้นฐานอยู่บนการเชื่อมต่อเครือข่ายแบบบัส (Bus-Type Network) ซึ่งสามารถเชื่อมต่อได้หลายแบบโดยอาศัยการแปลงให้เป็นครือข่ายเสมือนเพื่อให้เกิดประสิทธิภาพสูงสุดในการทำงาน แสดงดังภาพที่ 2.21 เครือข่ายทางขวามือจะเป็นเครือข่ายเสมือนของเครือข่ายเสมือนของเครือข่ายทางซ้ายมือ



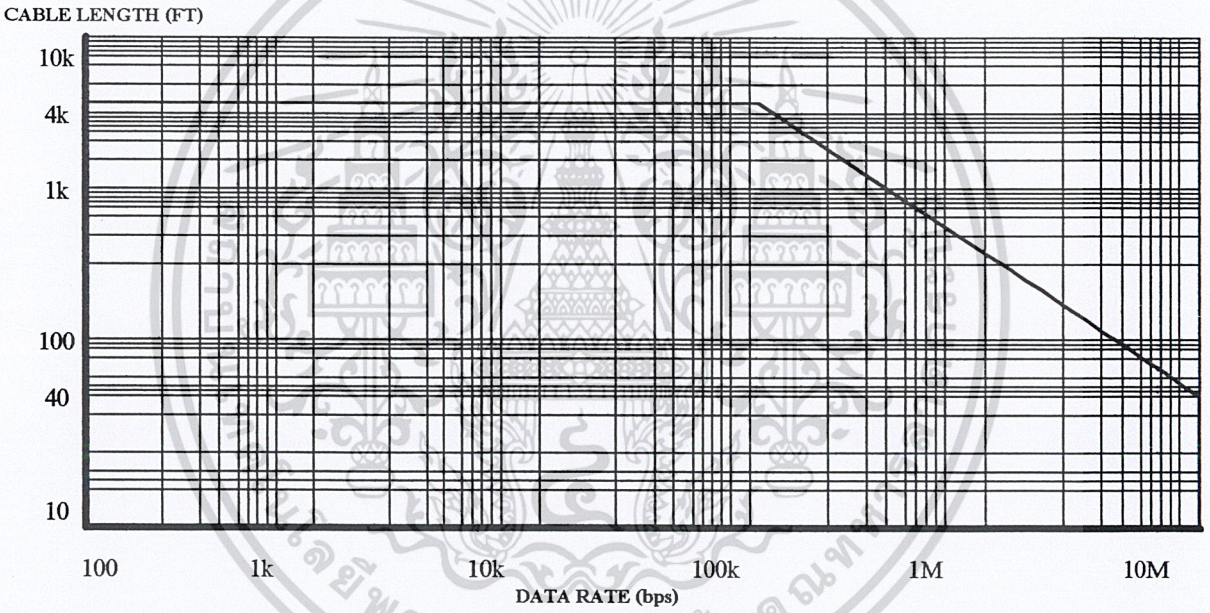
ภาพที่ 2.21 แสดงเครือข่ายที่สมมูลกันตามมาตรฐาน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.8 คุณสมบัติของสายสัญญาณที่ใช้

สายสัญญาณที่ใช้ตามมาตรฐาน RS-485 นั้นสามารถที่จะใช้สายเกลียวคู่ (Twist Pair) ซึ่งเป็น สายสัญญาณโทรศัพท์ ที่ใช้งานโดยทั่วๆไปได้โดยไม่มีผลกระทบ ต่อการสื่อสาร แต่ถ้าต้องการคุณภาพของ การสื่อสารที่สูงข้นอาจใช้ สายสัญญาณที่มีคุณภาพดีกว่าสายเกลียวคู่ เช่น สายโคแอกเชียล(Coaxial) เป็นสายสัญญาณที่ใช้สื่อสารแทนกันได้

คุณสมบัติด้านอัตราเร็ว และ ระยะทางในการสื่อสาร แทนกันได้ในการส่งข้อมูลตามมาตรฐาน RS-485 นั้นสามารถที่จะส่งข้อมูลได้สูงที่สุดถึง 10 Mbps และส่งข้อมูลได้ไกลสุดถึง 4000 ฟุต(1200เมตร) โดยความสัมพันธ์ของอัตราเร็วในการส่งข้อมูลและระยะทางในการส่งข้อมูลแสดงให้เห็นได้ดังภาพที่ 2.22



ภาพที่ 2.22 ความสัมพันธ์ของอัตราเร็วในการส่งข้อมูลและระยะทาง

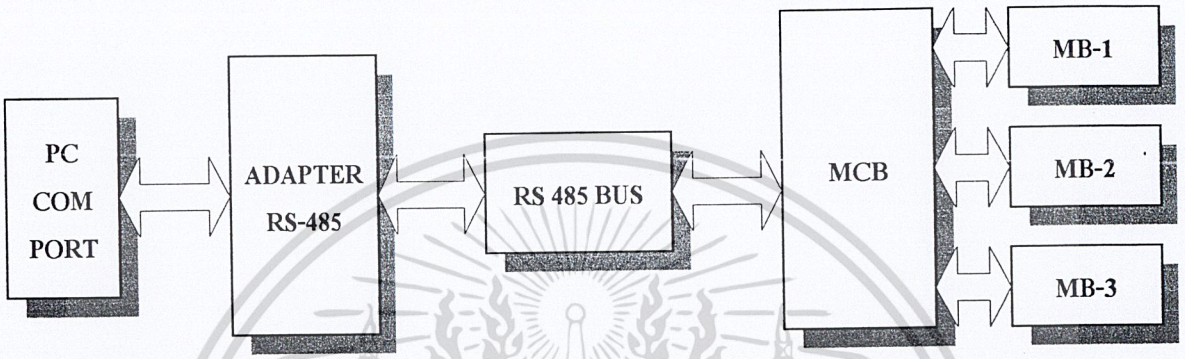
2.3.9 คุณสมบัติทางด้านสัญญาณรบกวน

สัญญาณรบกวนจะมีผลต่อการสื่อสารตามมาตรฐานRS-485 น้อยมากถ้าเลือกอัตราเร็วในและระยะทางในการส่งข้อมูลให้เหมาะสมนั้น เนื่องจากตามมาตรฐานRS-485นั้นการสื่อสารจะเป็นแบบ Current Loop และใช้ความต่างศักย์ของคู่สายสัญญาณในการส่งข้อมูล ทำให้สามารถทนต่อสัญญาณรบกวนได้ดีโดยเฉพาะสัญญาณรบกวนในลักษณะของ Common- mode noise

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างของระบบควบคุม

การทำงานของระบบควบคุมในหมู่บ้านทั้งหมดจะถูกควบคุมโดย คอมพิวเตอร์ส่วนบุคคล (PERSONAL computer PC)โดยจะเป็นการควบคุมแบรวมศูนย์(CENTRALIZE CONTROL) ซึ่งคอมพิวเตอร์ส่วนบุคคลจะทำหน้าที่เป็นตัวหลักในการติดต่อกับผู้ใช้งาน (User Interfzce) และระบบควบคุมโดยการจ่ายงานให้กับตัวควบคุมที่เชื่อมต่อกันเป็นเครือข่าย



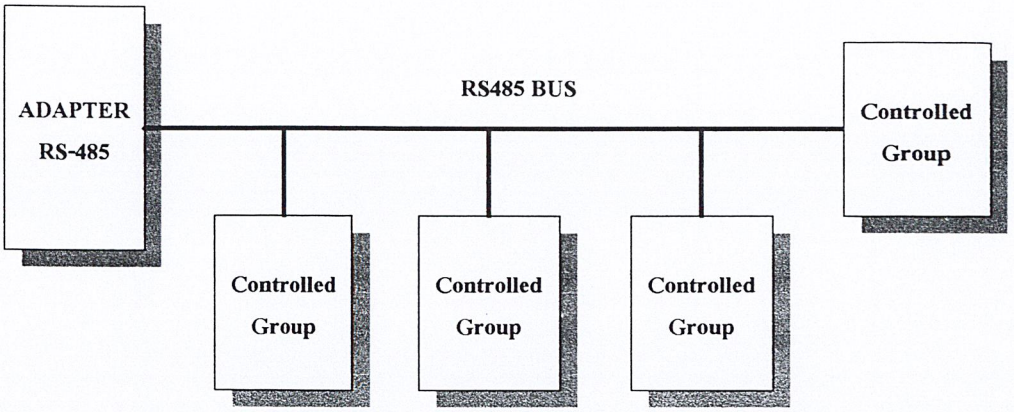
ภาพที่ 2.23 แสดงให้เห็นถึงโครงสร้างของการเชื่อมต่อกับระบบควบคุม

โดยการใช้การเชื่อมต่อ(Interface)กับคอมพิวเตอร์ส่วนบุคคล ซึ่งในที่นี้เราจะทำการใช้พอร์ท DB-25 (Com Port2) ในการเชื่อมต่อ โดยการสื่อสารของระบบควบคุม จะเป็นดังนี้คือ สัญญาณที่ออกจากพอร์ทสื่อสารจะแปลงจากสัญญาณตามาตรฐาน RS-232c ให้เป็นสัญญาณตามมาตรฐาน RS-485 โดยอแดปเตอร์(Adapter) จากนั้นสัญญาณจะถูกส่งผ่านสายสัญญาณ(RS-485) ไปยังตัวควบคุม และตัวลูกข่ายตามลำดับ

ส่วนการเชื่อมต่อตัวควบคุมเข้าเป็นเครือข่าย (Network) นั้น จะเป็นการเชื่อมต่อเครือข่ายแบบบัส(Bus type Network) โดยที่ไม่ต้องมีการปิดปลายสายสัญญาณ (Terminate) เนื่องจากอัตราการส่งข้อมูลที่ใช้มีค่าต่ำ สัญญาณรบกวนจึงมีผลต่อการส่งสัญญาณในระดับต่ำซึ่งจะไม่ทำให้เกิดความเสียหายในการสื่อสาร

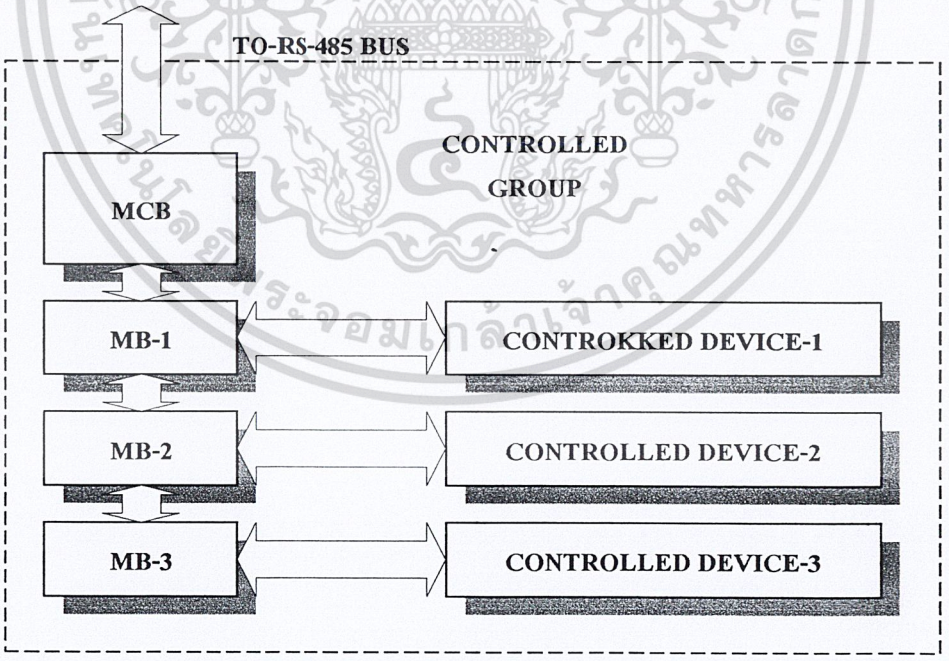
การเชื่อมต่อเป็นเครือข่ายของระบบควบคุม จะแสดงให้เห็นดังภาพที่ 2.24 โดยที่จะจัดตัวควบคุมกับตัวลูกข่ายที่เชื่อมต่อเข้าด้วยกันเป็นกลุ่มเรียกว่ากลุ่มตัวควบคุม(MCB area) และในการเชื่อมต่อกลุ่มตัวควบคุมเข้าเป็นเครือข่ายจะใช้สายสัญญาณเพียงคู่เดียวในการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.24 แสดงการเชื่อมต่อเครือข่ายของระบบควบคุม

ในกลุ่มของตัวควบคุมประกอบไปด้วย ตัวควบคุม ตัวลูกข่าย และอุปกรณ์ที่ต้องการควบคุม โดยตัวลูกข่ายแต่ละตัวจะเป็นอิสระต่อกันในการทำงาน กล่าวคือสามารถสลับที่หรือต่อตัวลูกข่ายแต่ละตัวจะเป็นอิสระต่อกันในการทำงาน กลบ่าวคือสามารถสลับที่หรือต่อตัวลูกข่ายเพิ่มเข้ามาในกลุ่มของตัวควบคุมได้โดยไม่มีผลกระทบต่อระบบควบคุม แต่มีข้อจำกัดหรือตัวลูกข่ายเพิ่มเข้ามาในกลุ่มของตัวควบคุมได้แบบหนึ่งต่อหนึ่ง เนื่องจากตัวลูกข่ายหนึ่งตัวสามารถควบคุมอุปกรณ์ได้เพียงหนึ่งอุปกรณ์เท่านั้น โดยจะแสดงการเชื่อมต่อกลุ่มตัวควบคุมดังภาพที่ 2.25



ภาพที่ 2.25 แสดงการเชื่อมต่อตัวควบคุมตัวลูกข่ายและอุปกรณ์ที่ต้องการควบคุม

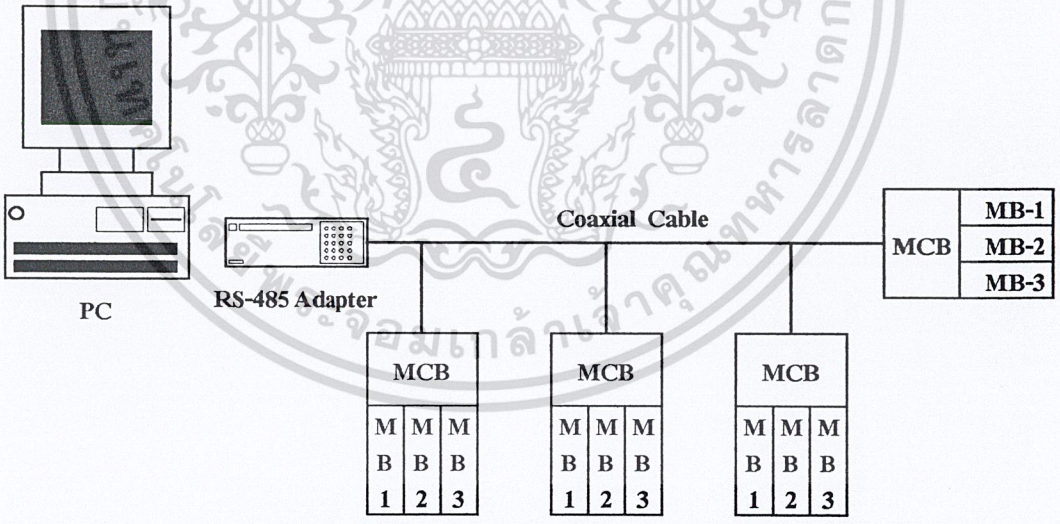
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การนำระบบควบคุมไปประยุกต์การใช้งาน

ระบบควบคุมนี้นอกจากจะนำไปประยุกต์ใช้งานในการควบคุมอุปกรณ์ภายใน ตัวอาคาร แล้วยังสามารถนำระบบควบคุมนี้ไปประยุกต์การใช้ในการควบคุม ระบบงานประเภทอื่นได้อีกด้วย โดยอาศัยการปรับเปลี่ยนส่วนประกอบที่ใช้ในการควบคุมอุปกรณ์ให้มีความเหมาะสมกับงาน ที่จะใช้ในการควบคุมให้มีความเหมาะสมกับงานที่จะใช้ ซึ่งในส่วนประกอบสำคัญที่เรา จะต้องมีการปรับเปลี่ยนประกอบด้วย 2 ส่วนใหญ่ๆคือ

1.ส่วนของตัวลูกข่ายที่ใช้ควบคุมอุปกรณ์ จะต้องทำให้มีการปรับเปลี่ยนให้ตัวลูกข่ายมีความสามารถในการควบคุมอุปกรณ์ได้ดี มีการทำงานที่เหมาะสมและเข้ากันได้กับอุปกรณ์ที่เราต้องการควบคุม โดยการปรับเปลี่ยนตัวลูกข่ายนี้จะขึ้นอยู่กับคุณสมบัติของอุปกรณ์ที่ต้องการควบคุมว่าเป็นอุปกรณ์ชนิดใด มีการทำงานแบบใด และอาศัยอะไรในการควบคุมอุปกรณ์ เนื่องจากปัจจัยเหล่านี้มีผลต่อประสิทธิภาพในการทำงานของระบบควบคุม

2. ส่วนของโปรแกรมที่ใช้ในการควบคุมการทำงานของอุปกรณ์ จะต้องมีการปรับเปลี่ยนให้โปรแกรมมีความสามารถในการควบคุมตัวลูกข่ายได้อย่างมีประสิทธิภาพ โดยมีความเข้ากันได้กับตัวลูกข่ายและมีการทำงานที่สอดคล้องกับตัวลูกข่ายที่ใช้ในการควบคุมอุปกรณ์ ตัวอย่างการประยุกต์ใช้งานแสดงดังภาพที่ 2.26



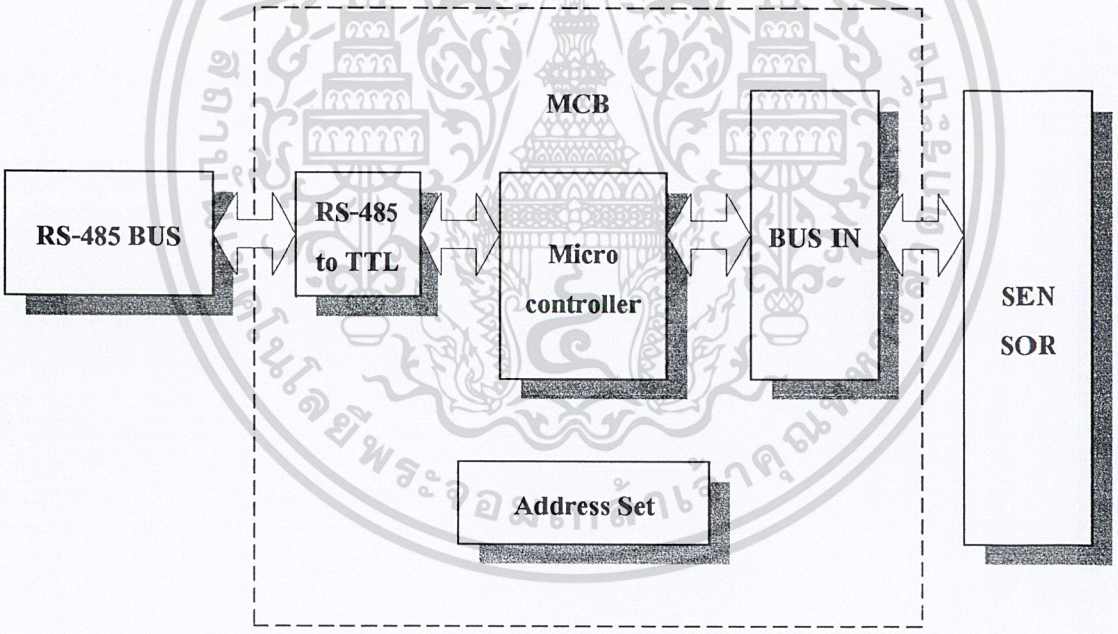
ภาพที่ 2.26 ตัวอย่างการประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ตัวควบคุม

2.5.1โครงสร้างของตัวควบคุม

ตัวควบคุมจะต้องคอยทำหน้าที่เป็นตัวเชื่อมต่อระหว่างสายสัญญาณ RS-485 ตัวลูกข่าย (เซนเซอร์) โดยจะรับข้อมูลจากสายสัญญาณแล้วทำการแปลงจากสัญญาณมาตรฐาน RS-485 ให้เป็นระดับสัญญาณแบบ TTL เพื่อนำไปประมวลผลลำการติดต่อกับตัวลูกข่ายต่อไป หรือให้รับค่าสภาวะของเซนเซอร์ที่ส่งเข้ามายัง ไมโครคอนโทรลเลอร์จากนั้นก็จะทำการ ประมวลผลแล้วทำ การส่งค่าผ่านออกทางพอร์ตอนุกรมของ ไมโครคอนโทรลเลอร์ แล้วส่งผ่านไปยัง คอมพิวเตอร์ โครงสร้างของตัวควบคุมสามารถแสดงให้เห็นได้ดังรูปโดยภายในตัวควบคุมนอกจาก จะมีส่วนที่ทำหน้าที่ปรับระดับสัญญาณแล้วยังมีส่วนที่ทำการตั้งค่าแอดเดรสสำหรับตัวควบคุม เพื่อกำหนดค่าแอดเดรสให้กับตัวควบคุมทำให้สามารถเลือกใช้งานตัวควบคุมได้ โดยการอ้างอิงถึงแอดเดรสของตัวควบคุมแต่ละตัว

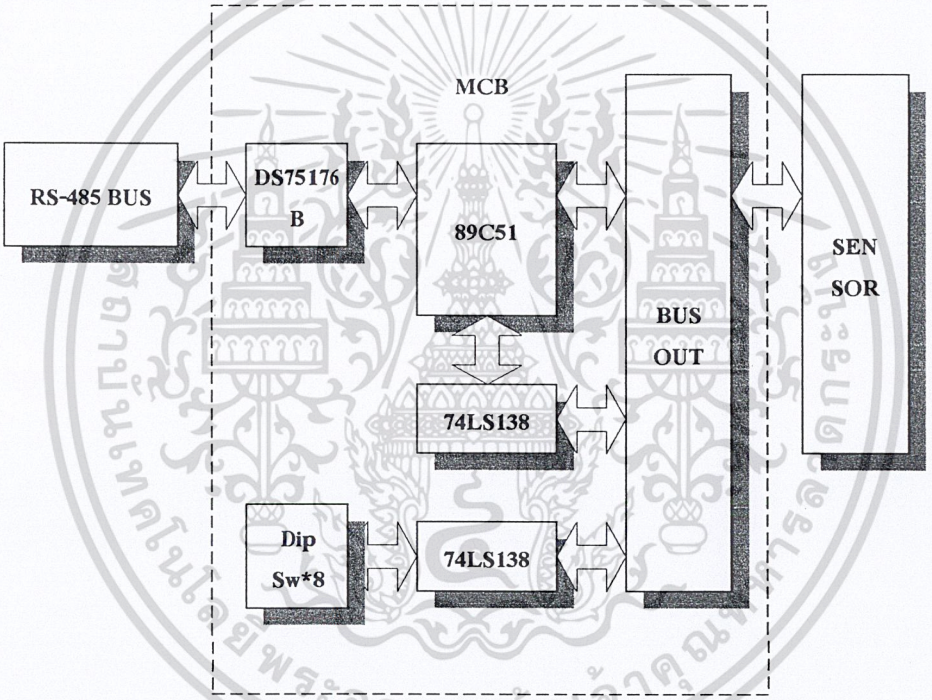


ภาพที่ 2.27 โครงสร้างของตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 การออกแบบตัวควบคุม

โดยอาศัยข้อมูลที่ได้จากโครงสร้างของตัวควบคุมทำให้สามารถ ออกแบบตัวควบคุมให้มีลักษณะการทำงานและหน้าที่ตามที่ต้องการได้โดยส่วนประกอบของตัวควบคุมสามารถ แสดงให้เห็นได้ดังรูปโดยในส่วนของตัวควบคุมนั้นจะมีไมโครคอนโทรลเลอร์คอยทำหน้าที่ ในการจัดการกับข้อมูล การจ่ายงาน การตัดสินใจขั้นพื้นฐาน ซึ่งเป็นลักษณะของอุปกรณ์อัตโนมัติ เพื่อลดภาระส่วนบุคคลของคอมพิวเตอร์ส่วนบุคคล และ ยังช่วยเพิ่มประสิทธิภาพ ของระบบควบคุมอีกด้วย นอกจากการทำงานข้างต้น แล้วไมโครคอนโทรลเลอร์ยังทำหน้าที่ในการเลือกปฏิบัติตามคำสั่งโดยดูจากค่าของแอดเดรสที่มากับข้อมูลว่าตรงกับแอดเดรสของตนหรือไม่ ถ้าตรงก็จะนำไปปฏิบัติคำสั่งต่อไป



ภาพที่ 2.28 แสดงส่วนประกอบของตัวควบคุม

ในส่วนของการแปลงสัญญาณจากสัญญาณตามมาตรฐาน RS-485 ให้เป็นระดับสัญญาณแบบ TTL จะใช้สำหรับการติดต่อระหว่างตัวควบคุมกับตัวลูกข่ายเพื่อสั่งงาน และ การส่งข้อมูลนั้นจะใช้พอร์ตของไมโครคอนโทรลเลอร์ในการติดต่อวงจรของตัวควบคุมสามารถแสดง ให้เห็นได้ดังภาพ โดยมีการเชื่อมต่อตามที่ได้ออกแบบไว้และมีการใช้ไฟเลี้ยงวงจรจากวงจรภายนอก ในที่นี้จะใช้ไอซี 7805 ทำหน้าที่ในการจ่ายไฟเลี้ยงให้แก่วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรจะทำงานโดยเริ่มจากรับสัญญาณตามาตรฐาน RS-485 จากสายสัญญาณแล้วทำการแปลงให้เป็นระดับแรงสัญญาณแบบ TTL โดยใช้ไอซี DS 75176 B ส่งข้อมูลที่ได้ออกไปยังที่ตัวของไมโครคอนโทรเลอร์ AT89C51 เพื่อตรวจดูแอดเดรสในคำสั่งว่าเป็นแอดเดรสของตนหรือไม่ ถ้าไม่ใช่ก็จะ ไม่นำไปปฏิบัติคำสั่ง แต่ถ้าเป็นแอดเดรสของตนก็จะนำไปปฏิบัติตามคำสั่งต่อไป โดยส่งค่าไปทำ การตรวจสภาวะของพอร์ทอินพุทว่าขณะนี้มีเซนเซอร์อันไหนทำงานก็จะทำการส่งค่าสภาวะมาหรือยัง ถ้าเซนเซอร์ทำงานส่งค่าสภาวะมาที่พอร์ทอินพุทไมโครคอนโทรลเลอร์ก็จะทำการเปลี่ยนค่าเป็นเลขฐาน 16 แล้วทำการส่งค่าออกไปยัง พอร์ทอนุกรม เพื่อส่งค่าต่อไปยังคอมพิวเตอร์

2.6 ระบบโทรศัพท์แจ้งเหตุ

ทำการในระบบ สองสายไฟเดินคู่สายมาจาก ชุมสายควบคุมมายัง โทรศัพท์แต่ละบ้าน ในคู่สายนี้จะทำหน้าที่ส่งสัญญาณเสียงพูดคุย เสียงกริ่งและสัญญาณหมุนหมายเลขกับทางชุมสาย ปลายทางชุมสายจะรับทราบว่า มีผู้ต้องการใช้สายโทรศัพท์(ต้องการโทรออก) เมื่อท่านยกหูทางชุมสายจะครบวงรอบกับตัวโทรศัพท์ขณะนั้น เพื่อรับทราบหมายเลขจากท่าน แล้วทำการต่อคู่สาย ปลายทางให้คู่สายโทรศัพท์ขณะนั้นจะมีอิมพีแดนซ์ 600 โอห์ม สำหรับการส่งเสียงสนทนา

ภายในวงจรยามโทรศัพท์แบ่งเป็น 2 ส่วน คือ ส่วนที่เชื่อมต่อโยงกับคู่สายโทรศัพท์ และส่วนวงจรการเชื่อมโยงกระทำผ่าน ออพโตไอโซเลเตอร์ วงจรหลักจะรับไฟจากแบตเตอรี่แม้ว่าสามารถใช้ไฟบ้านได้ก็ตามส่วนอุปกรณ์ภาคเชื่อมโยงสายโทรศัพท์จะทำการรับไฟเลี้ยงจาก คู่สายโทรศัพท์เอง

เมื่อพบว่ามีผู้บุกรุกหรือต้องการแจ้งภัย ส่วนวงจรหลักจะสร้างรหัสพัลส์ตามเบอร์โทรศัพท์ที่ตั้งไว้ จากนั้นจึงส่งสัญญาณเสียงแจ้งภัยให้ทราบไดโอดบริดจ์(D1-D4) ทำหน้าที่รักษาขั้วแรงดันไฟตรงเลี้ยงวงจรให้คงที่ตลอดเวลา ซึ่งในขณะนี้ RLA 1 ทำหน้าที่ต่อวงจรเข้ากับคู่สายโทรศัพท์ แรงดัน 50 V จากคู่สายโทรศัพท์จะปรากฏคร่อม ไดโอดบริดจ์ทำให้ FET ทั้งคู่ทำงานปิดวงจรจึงเกิดลูปครบวงจรขึ้น

2.6.1 การใช้งาน

ใช้ง่ายมาก โดยโปรแกรมเลขหมายปลายทางแก่ระบบนี้ก่อน ด้วยการกดปุ่ม Reset ชั่วครู่ตามด้วยการกดปุ่มเลขหมายปลายทาง(ใส่เลขหมายได้มากถึง 10 หลัก) หากกดปุ่ม Test หรือใช้วงจรสั่งงานด้านอินพุทเกิดปิดวงจร เครื่องนี้จะหมุนหมายเลขปลายทางและส่งเสียงเตือนให้ผู้ใช้ทราบเป็นเวลา 1 นาทีก่อนรีเซ็ทตัวเองโดยอัตโนมัติเพื่อพร้อมทำงานครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. กดหมายเลขที่ต้องการไม่เกิน 10 ตัว

2. กดปุ่ม MODE ค้างไว้

3. กดปุ่ม * แล้วปล่อยโดยปุ่ม MODE ยังกดค้างไว้อยู่

4. กดปุ่ม * กดหมายเลขช่องที่ต้องการบันทึก (1-3)

5. ปล่อยปุ่ม MODE

ตัวอย่าง ต้องการกดหมายเลข 3758058 ไว้ที่ ช่อง 1

หมายเลข 7310015 ไว้ที่ช่อง 2

หมายเลข 7310013 ไว้ที่ช่อง 3

1. กดหมายเลข 3758058

2. กดปุ่ม MODE ค้างไว้

3. กดปุ่ม* (เครื่องหมายดอกจันทร์) ปุ่ม MODE ยังค้างไว้อยู่

4. ปล่อยปุ่ม * กดหมายเลข 1

5. ปล่อยปุ่ม MODE

6. ทำตามข้อ 1 ,2 ,3, 4, 5 โดยข้อ 1 จะต้องเปลี่ยนหมายเลขเป็น 7310015 และข้อ 4 เปลี่ยนเป็นกดหมายเลข 2

7. ทำตามข้อ 1 2 3 4 5 โดยข้อ 1 จะต้องเปลี่ยนหมายเลขเป็น 7310013 และข้อ 4 เปลี่ยนเป็นหมายเลข 3

2.6.3 **การทำงานของเครื่อง**

เมื่อเราทำการ ON สวิตช์ตอนนี้ LED1 จะติดวงจรจะทำการหน่วงเวลา 30 วินาทีซึ่งในตอนนี้วงจรจะไม่ทำการตรวจจับสวิตช์เซนเซอร์ เมื่อผ่านช่วงเวลานี้ไปแล้ววงจรก็พร้อมจะทำงาน เมื่อใดก็ตามสวิตช์ เซนเซอร์ ถูกเปิดออกแล้วปิดเข้าหรือเปิดค้างไว้ก็ตาม LED 3 จะติดแสดงว่าวงจรทำการตรวจจับแล้ววงจรจะทำการหมุนหมายเลขโปรแกรม ที่ 1 หลังจากนั้น LED 2 จะติดแสดงว่าวงจรได้ทำการหมุนหมายเลขโปรแกรมที่ 1 เรียบร้อยแล้ว นานประมาณ 1 นาที LED 2 จะดับเป็นเวลาอีกประมาณ 5 วินาที LED 2 จะติดอีกเป็นครั้งที่ 2 แสดงว่าโปรแกรมจะทำการหมุนหมายเลขโปรแกรมที่ 2 เป็นเวลา 1 นาที LED 2 จะดับเป็นเวลา 5 วินาที จะติดอีกเป็นครั้งที่ 3 แสดงว่าวงจรทำการหมุนหมายเลขโปรแกรมที่ 3 เป็นเวลา 1 นาที LED 2 ก็จะดับ แสดงว่าตอนนี้เครื่องได้ทำการหมุนหมายเลขครบ 1 รอบ

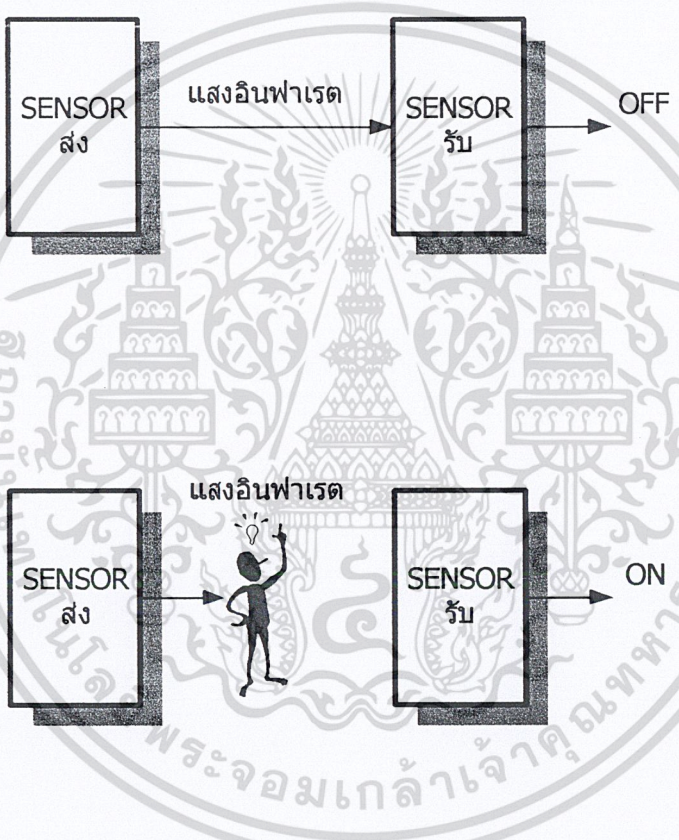
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 SENSOR

จะใช้ Sensor ที่เป็นตัวหลักจากโปรเจคที่แล้วในการนำมาประกอบ ประยุกต์ใช้ในครั้งนี้ โดยจะอธิบายพอคร่าวๆ คือ

2.7.1 Sensor อินฟาเรด

จะทำงานเมื่อมีสิ่งผิดปกติมาขวางกั้น การยิงของลำแสงอินฟาเรด ซึ่งจะทำการประกอบด้วย ตัวส่งและตัวรับ เมื่อมีการ On (เมื่อมีการผิดปกติ) ตัวรับก็จะส่งสัญญาณออกไป อาจจะทำการส่งสัญญาณเสียง Buffer หรือ ส่งไป On ใช้ร่วมกับวงจรตัวอื่นเพื่อใช้งานต่อไป



ภาพที่ 2.29 การทำงานของ Sensor

***<u>หมายเหตุ</u> วงจร Sensor อินฟาเรดจะเซ็ทภาวะการทำงานที่ NO

2.7.2 ตัวตรวจจับเปลวไฟ

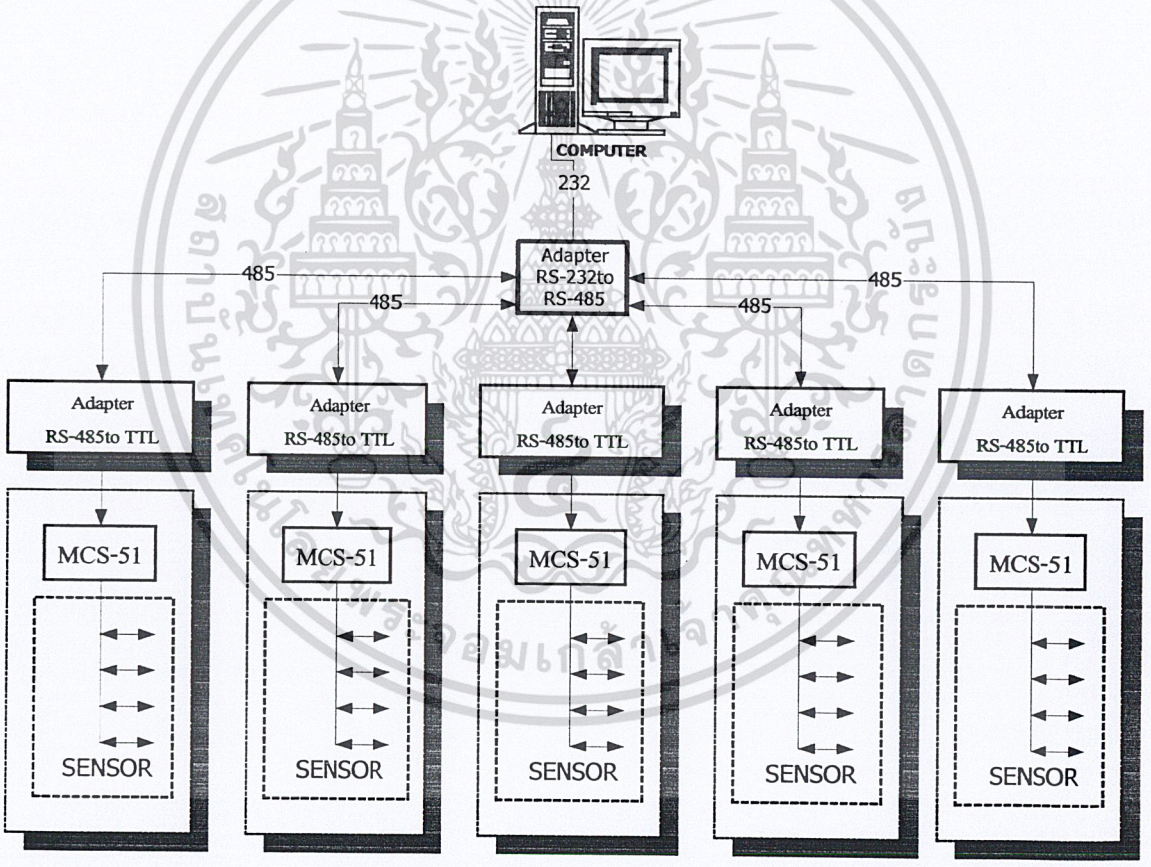
เมื่อไม่มีเปลวไฟมาสัมผัสที่บริเวณขั้วเส้นลวดวงจรก็จะทำงานปกติคือไม่มีการส่งสัญญาณ On ออกไป แต่เมื่อมีเปลวไฟมาสัมผัสทำให้หน้าสัมผัสเส้นลวดร้อนถึงระดับหนึ่งแล้ว วงจรก็จะทำการ ส่งสัญญาณ On ออกไป เพื่อขับ Buffer หรือส่งสัญญาณ เพื่อใช้ร่วมกับวงจรอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
การประยุกต์ใช้งานในระบบ

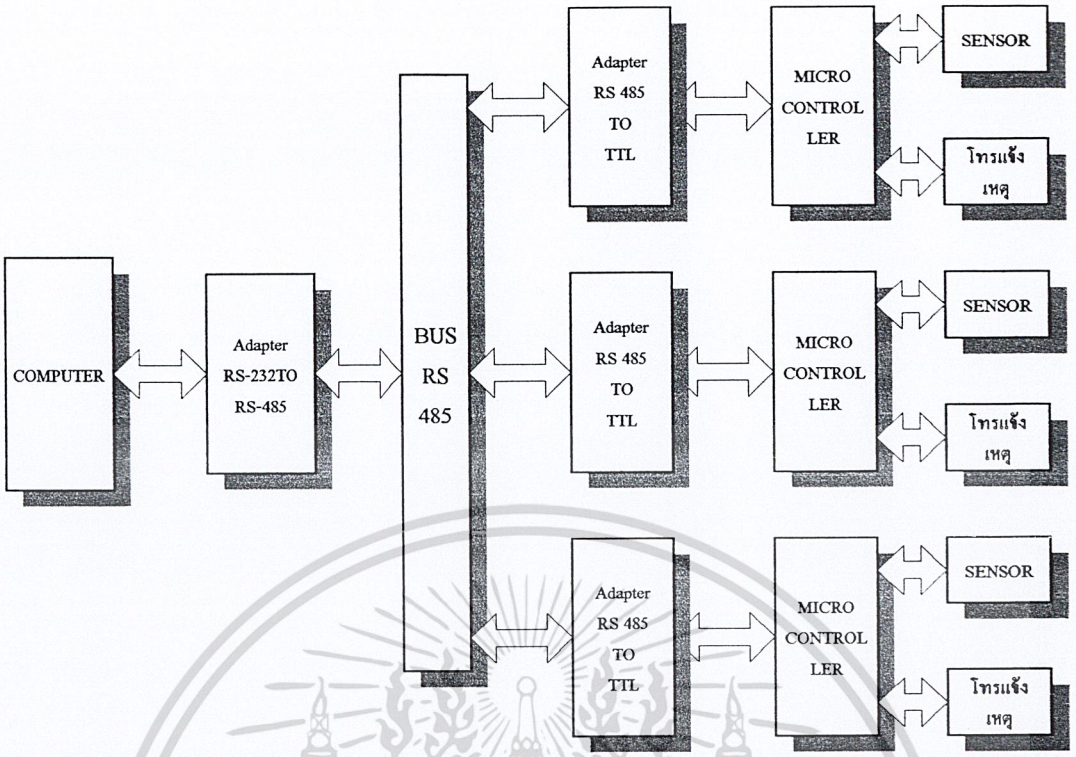
3.1 การทำงานของระบบ

ระบบโดยรวม เราจะทำการดูแลทั้งหมู่บ้านโดยการติดตั้งคอมพิวเตอร์อยู่ที่ศูนย์รักษาความปลอดภัยของหมู่บ้าน โดยเราจะให้ผู้รักษาความปลอดภัยเป็นผู้เป็นผู้ตรวจดูสภาวะต่างๆที่เกิดขึ้นกับคอมพิวเตอร์โดยเราจะให้ ไมโครคอนโทรลเลอร์แต่ละตัวควบคุมดูแลบ้านแต่ละหลัง โดยการที่เราจะใช้เซนเซอร์ทำการติดตั้ง ภายในบ้านเพื่อตรวจจับสิ่งผิดปกติที่เกิดขึ้นภายในบ้าน จากนั้นจะทำส่งค่าสภาวะของเซนเซอร์ไปที่ ไมโครคอนโทรลเลอร์ เพื่อส่งต่อไปยังคอมพิวเตอร์เพื่อแสดงผล



ภาพที่ 3.1 รูปแบบการทำงานทั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 บล็อคไดอะแกรมการทำงานของระบบ

3.1.1 **การทำงานของระบบรักษาความปลอดภัย**

ระบบรักษาความปลอดภัยที่เราได้จัดทำขึ้นนี้การทำงานของระบบนั้นเราได้ กำหนดให้คอมพิวเตอร์ เป็นตัวแสดงผลแสดงค่าสถานะของเซนเซอร์ตามจุดๆที่เราได้ทำการติดตั้งไว้ภายในบ้านแต่ละหลัง โดยเรากำหนดให้คอมพิวเตอร์เป็นตัวหลัก Masterในการตรวจเช็คการทำงานของไมโครคอนโทรเลอร์และเซนเซอร์ และเรากำหนดไมโครคอนโทรลเลอร์เป็นตัวส่งข้อมูลSLAVE โดยเราจะทำการกำหนดค่าแอดเดรสให้กับไมโครคอนโทรลเลอร์ แต่ละตัวมีดังนี้

ไมโครคอนโทรลเลอร์MCS 51	ADDRESS
1	#00H
2	#01H
3	#02H

โดยคอมพิวเตอร์จะทำการส่งค่าแอดเดรสที่มีค่าตรงกับค่าของ ไมโครคอนโทรลเลอร์ที่ต้องการจะทำการตรวจสอบผ่านไปยังบัส RS-485 จากนั้นไมโครคอนโทรลเลอร์จะรับค่าแอดเดรสที่ส่งผ่านมาเข้าไปทำการตรวจสอบโดยตอนที่ ไมโครคอนโทรลเลอร์ทำการรับค่าสภาวะเข้าไปทำการตรวจสอบนั้นไมโครคอนโทรลเลอร์จะรับค่าแอดเดรสที่ส่งมาเข้าไปพร้อมกันทั้ง3 ตัว จากนั้นจะทำการเปรียบเทียบแอดเดรสของคอมพิวเตอร์ที่ส่งเข้ามากับแอดเดรสของตัวเองว่าตรงกันหรือ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ถ้าแอดเดรสไม่ตรงกัน ไมโครคอนโทรเลอร์ก็จะไปรอรับค่าจาก Searial Port ถ้าค่าแอดเดรสของตนเองตรงกับค่าแอดเดรสของคอมพิวเตอร์ ไมโครคอนโทรลเลอร์จะเข้าไปที่โปรแกรมของตัวเองเพื่อเช็คค่าสภาวะของเซนเซอร์ที่ต่ออยู่กับพอร์ทของไมโครคอนโทรลเลอร์ ว่ามีเหตุการณ์ผิดปกติเกิดขึ้นทำให้เซนเซอร์ทำงาน

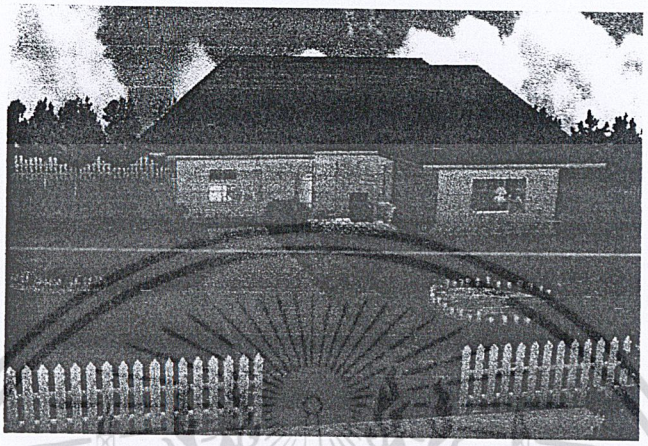
เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้นภายในบ้านนั้นเซนเซอร์จะเปลี่ยนสภาวะส่งค่าญญาณ มาที่พอร์ตของไมโครคอนโทรลเลอร์หลังจากนั้นไมโครคอนโทรลเลอร์จะทำการประมวลผลค่าสภาวะที่ได้เข้ามาที่พอร์ทของไมโครคอนโทรลเลอร์ว่ามีค่าเป็นเลขฐาน 16 จากนั้นไมโครคอนโทรลเลอร์จะทำการส่งค่าสภาวะของเซนเซอร์ออกไปยัง SearialPort ที่อยู่ภายในตัวของไมโครคอนโทรลเลอร์เพื่อทำการส่งไปแสดงค่าสภาวะของเซนเซอร์ยังคอมพิวเตอร์โดยที่คอมพิวเตอร์นั้นจะมีการเขียนโปรแกรมแสดงการรับค่าเข้ามาของเซนเซอร์ โดยที่จะทำการแสดงเป็นรูปบ้านและอุปกรณ์เซนเซอร์แสดงอยู่ที่หน้าจอคอมพิวเตอร์ โดยเมื่อเกิดเหตุการณ์ ผิดปกติขึ้นรูปเซนเซอร์และรูปบ้านจะทำการเปลี่ยนสภาวะเพื่อแสดงค่าให้เห็น

การส่งค่าแอดเดรสไปตรวจเช็คไมโครคอนโทรลเลอร์ของคอมพิวเตอร์จะทำการตรวจเช็คไปเรื่อยๆ ไล่ไปตังแต่ไมโครคอนโทรลเลอร์ ตัวที่1 2 3 วนลูปไปเรื่อยๆ จากนั้นก็จะกลับมาทำการตรวจเช็คตัวที่ 1 ใหม่ อย่างนี้ต่อไปเรื่อยๆ

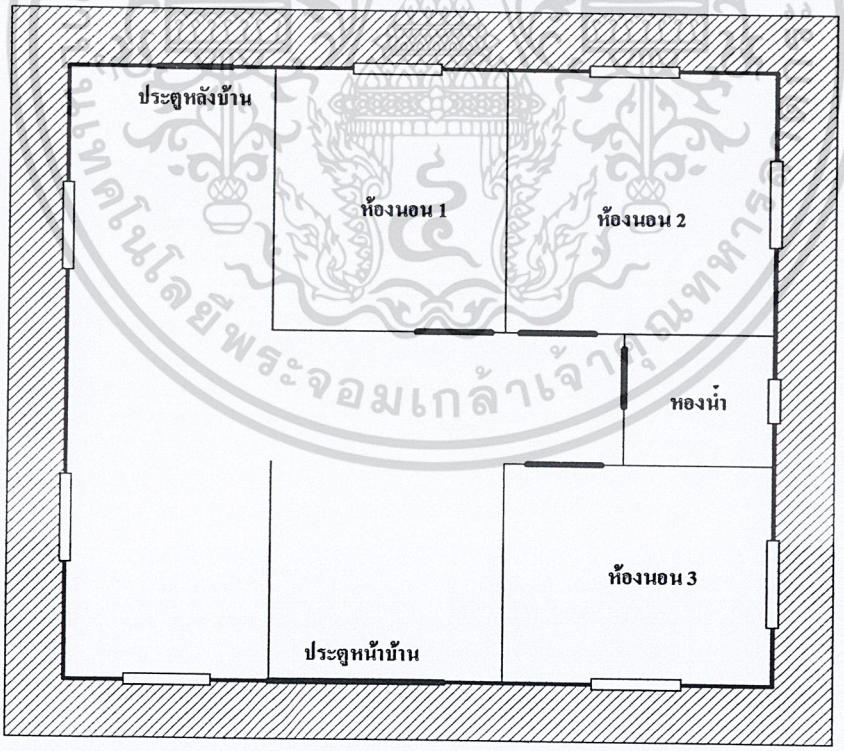
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การติดตั้งเครื่องตรวจจับในที่พักอาศัย(SENSOR)

เราจะทำการติดตั้งเซนเซอร์ไว้ตามจุดต่างๆ ของบ้าน เช่น ที่ประตู หน้าต่าง ผนัง เมื่อมีเหตุการณ์ผิดปกติเกิดขึ้นเซนเซอร์ก็จะทำการส่งค่าไปที่ไมโครคอนโทรลเลอร์เพื่อส่งเข้าไปที่คอม

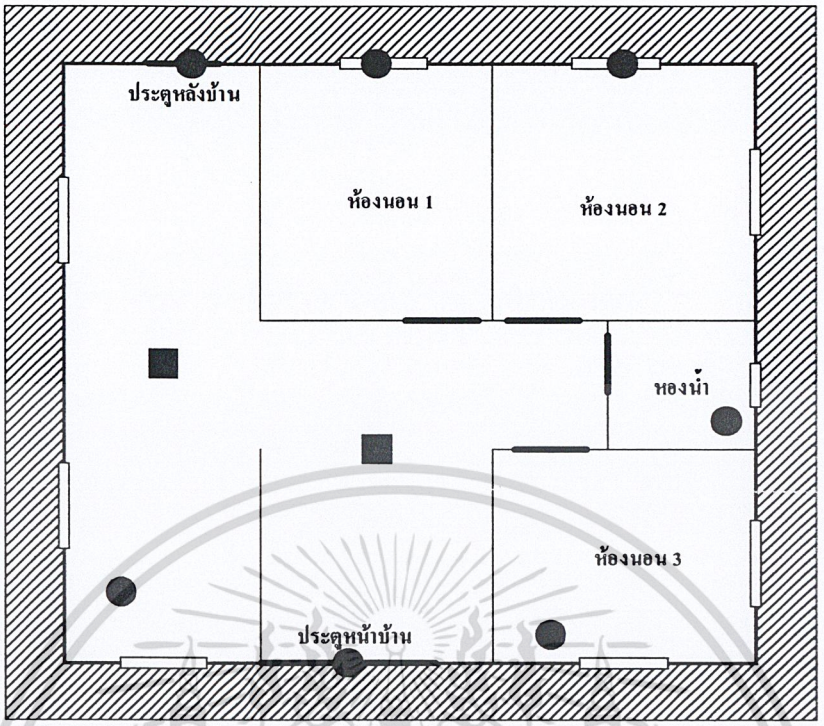


ภาพที่ 3.3 ลักษณะภายนอกตัวบ้าน



ภาพที่ 3.4 ลักษณะภายในตอนยังไม่ติดตั้งเครื่องตรวจจับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 ลักษณะภายในตอนยังติดตั้งเครื่องตรวจจับแล้ว

รูปแบบการติดตั้งโดยรวมภายในบ้านทั้งหมด โดย MCS ในแต่ละชุดแต่ละหลังสามารถรองรับ Sensor ได้ถึง 8 ตัว จากภาพที่ 3.5 ในลักษณะการจัดวาง

วงกลม คือ Sensor ทั่วไปอาจจะเป็นแบบตรวจจับโดยใช้ อินฟาเรด หรือตรวจจับการปิด-เปิด ประตูหน้าต่าง

สี่เหลี่ยม คือ Sensor ตรวจจับเปลวไฟ การติดตั้งจะติดไว้ กลางบ้าน หรือ ในครัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 อธิบายโปรแกรมการทำงาน

โดยจะประกอบด้วยโปรแกรม หลักๆอยู่ 2 โปรแกรมด้วยกัน คือ

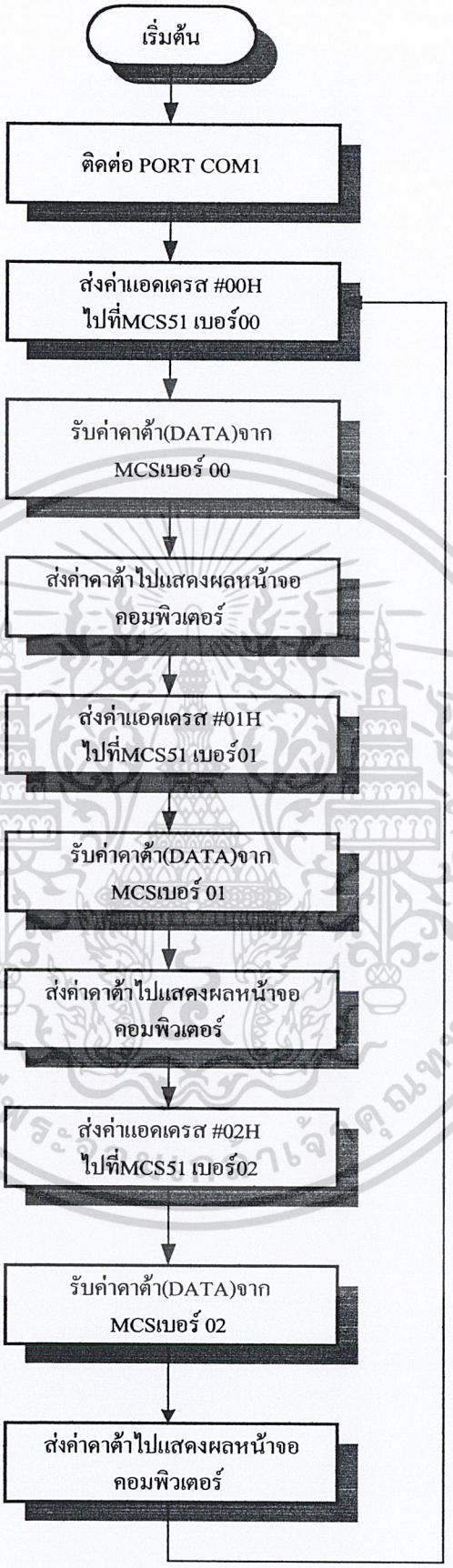
3.3.1 อธิบายการทำงานของโฟร์ชาร์ทไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะรอรับค่า ADDRESS ที่ส่งมาตรวจเช็คจากคอมพิวเตอร์ แล้วนำค่าที่ได้มาเปรียบเทียบกับค่า ADDRESS ของตัวเอง หากมีค่าตรงกัน ไมโครคอนโทรลเลอร์จะทำการตรวจสอบ พอร์ทอินพุทที่ทำการต่อกับเซนเซอร์ จากนั้นจะส่งค่าDATA ซึ่งแทนค่าสภาวะของเซนเซอร์ไปยังคอมพิวเตอร์เพื่อแสดงผล หากไม่ใช่ก็จะกลับไปรอรับค่า ADDRESS ที่ส่งมาจากคอมพิวเตอร์ใหม่

3.3.2 อธิบายการทำงานของโฟร์ชาร์ทของคอมพิวเตอร์

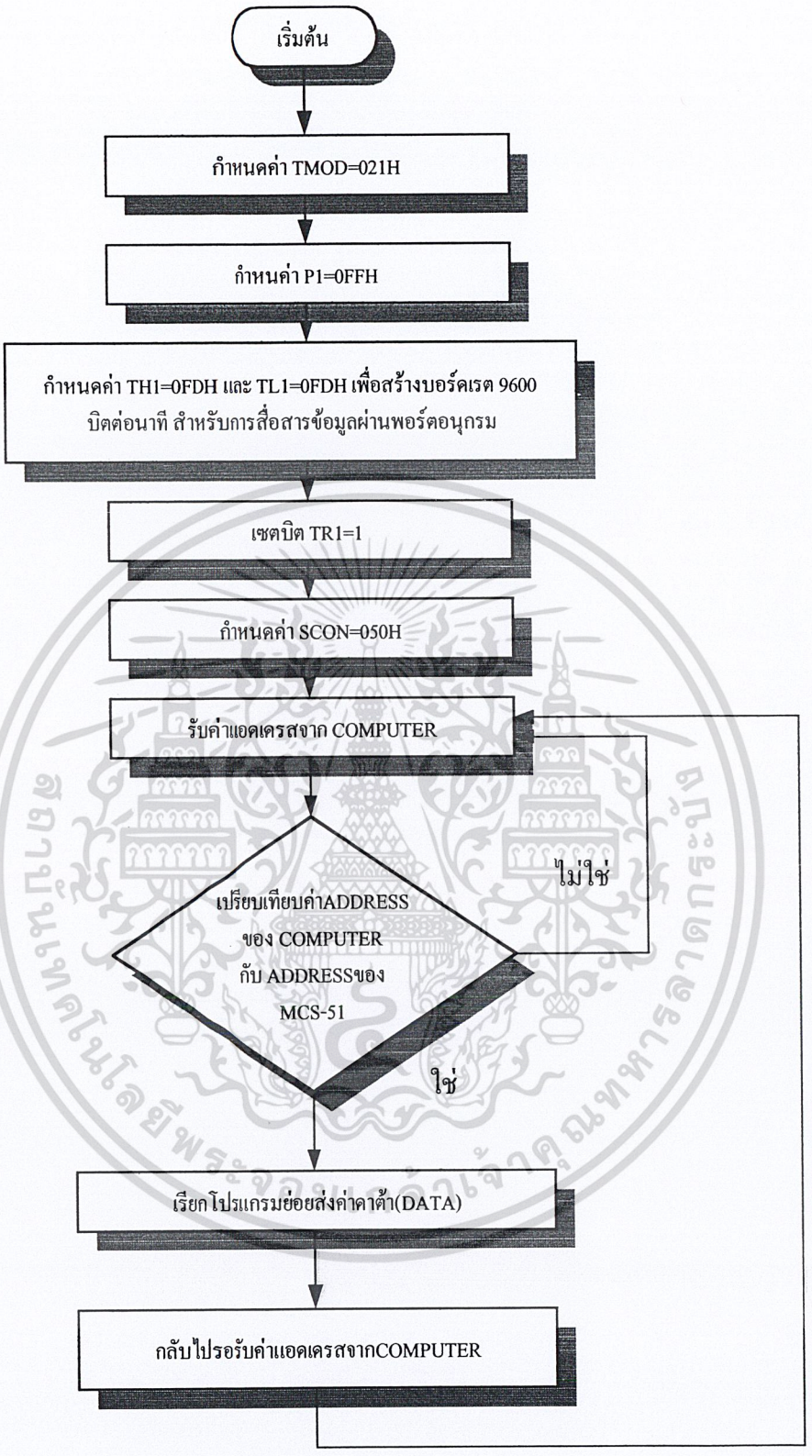
คอมพิวเตอร์จะส่งค่า ADDRESS ของไมโครคอนโทรลเลอร์ออกไป เพื่อไปตรวจเช็คการทำงานของ ไมโครคอนโทรลเลอร์จากนั้นจะรอรับค่า DATA จากไมโครคอนโทรลเลอร์เพื่อนำไปแสดงผลที่หน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



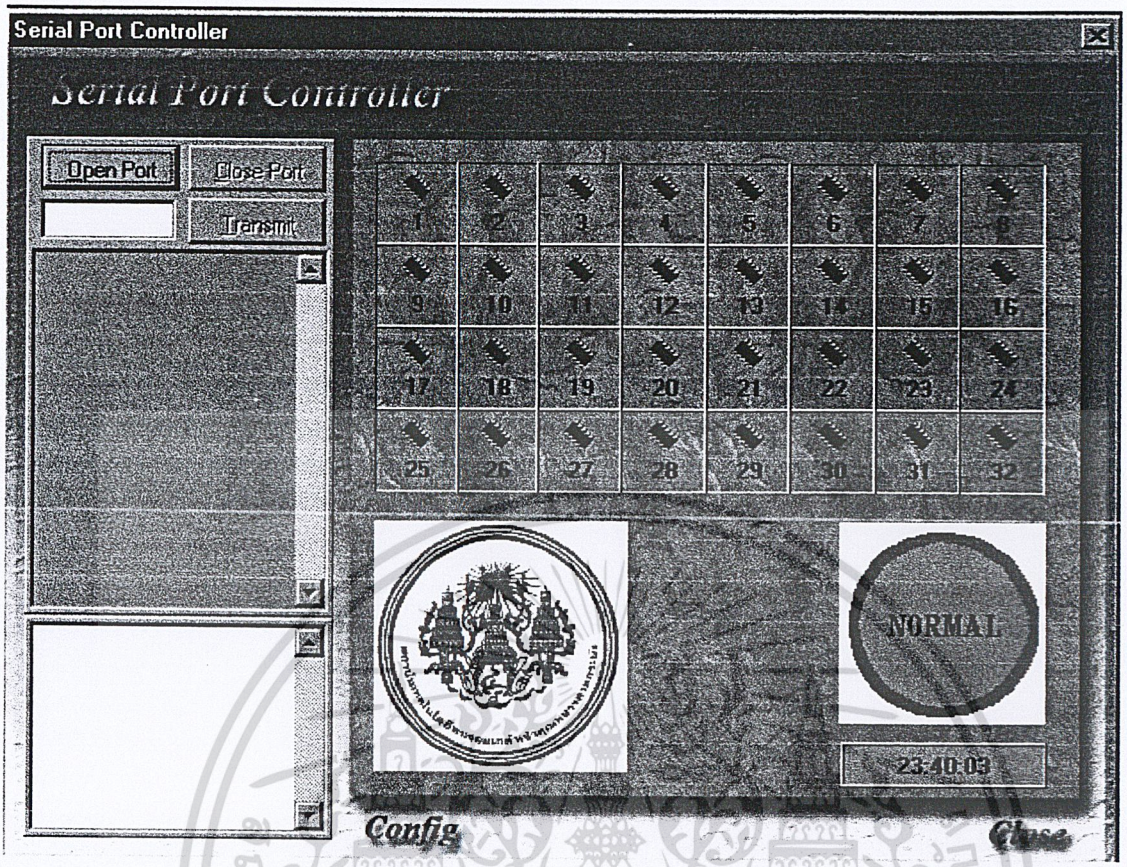
ภาพที่ 3.6 โฟรชาร์ตของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.7 โฟรชาร์ตของ โปรแกรม Computer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.8 หน้าตาโปรแกรมที่ใช้งาน

จาก ภาพที่ 3.8โปรแกรมที่ใช้งานนี้ ตรงส่วน

OpenPort แสดงการเปิดพอร์ตหรือการติดต่อกับ Searial Port ใช้งาน

ClosePort แสดงการปิดพอร์ตหรือการติดต่อกับ Searial Port ใช้งาน

Transmitter แสดงการทดสอบค่าที่ไว้ใช้งาน

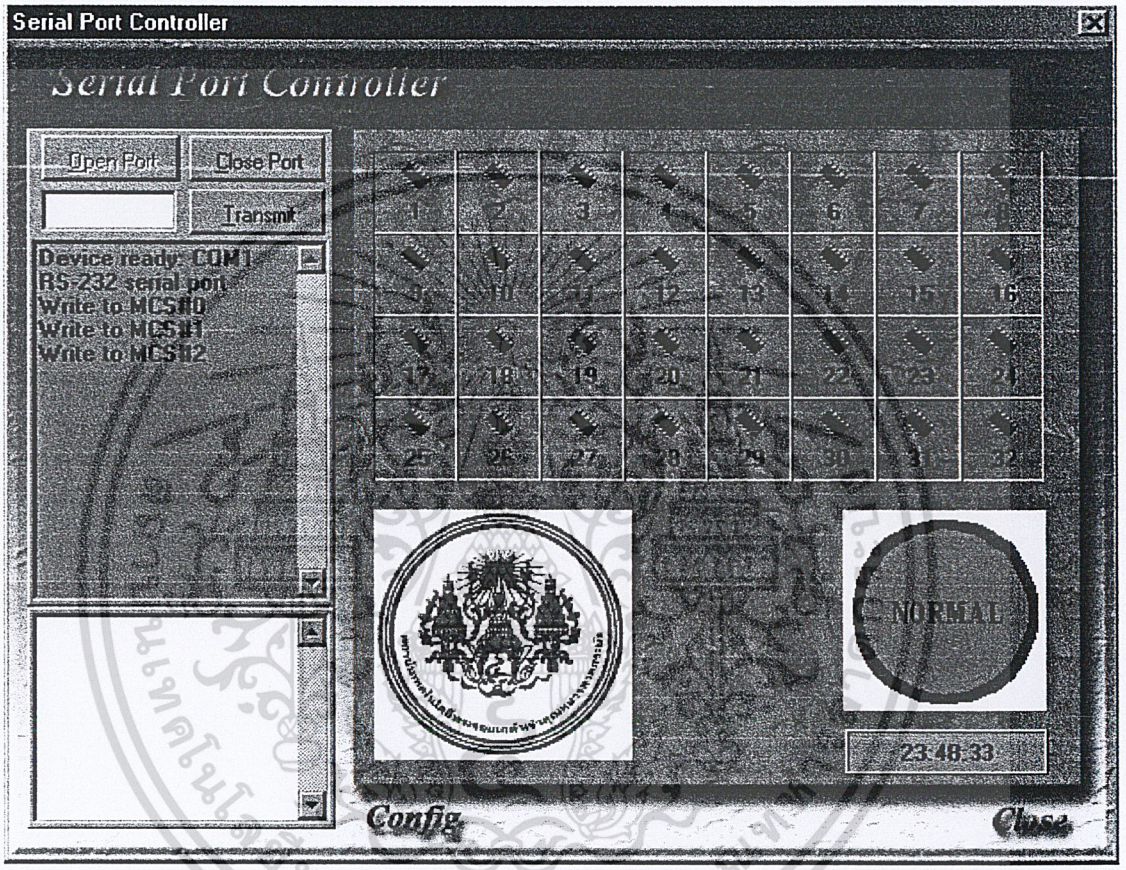
ตรงรูป ไอซี เอาไว้แสดงการติดต่อกับ MCS51 โดยหมายเลข1 หมายถึง MCS51 ตัวที่ 1

ตรง Alarm เอาไว้แสดงค่าและเสียงเตือนเวลาที่เซนเซอร์ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
หลักการใช้งานโปรแกรม

4.1 เปิด Port ใช้งาน

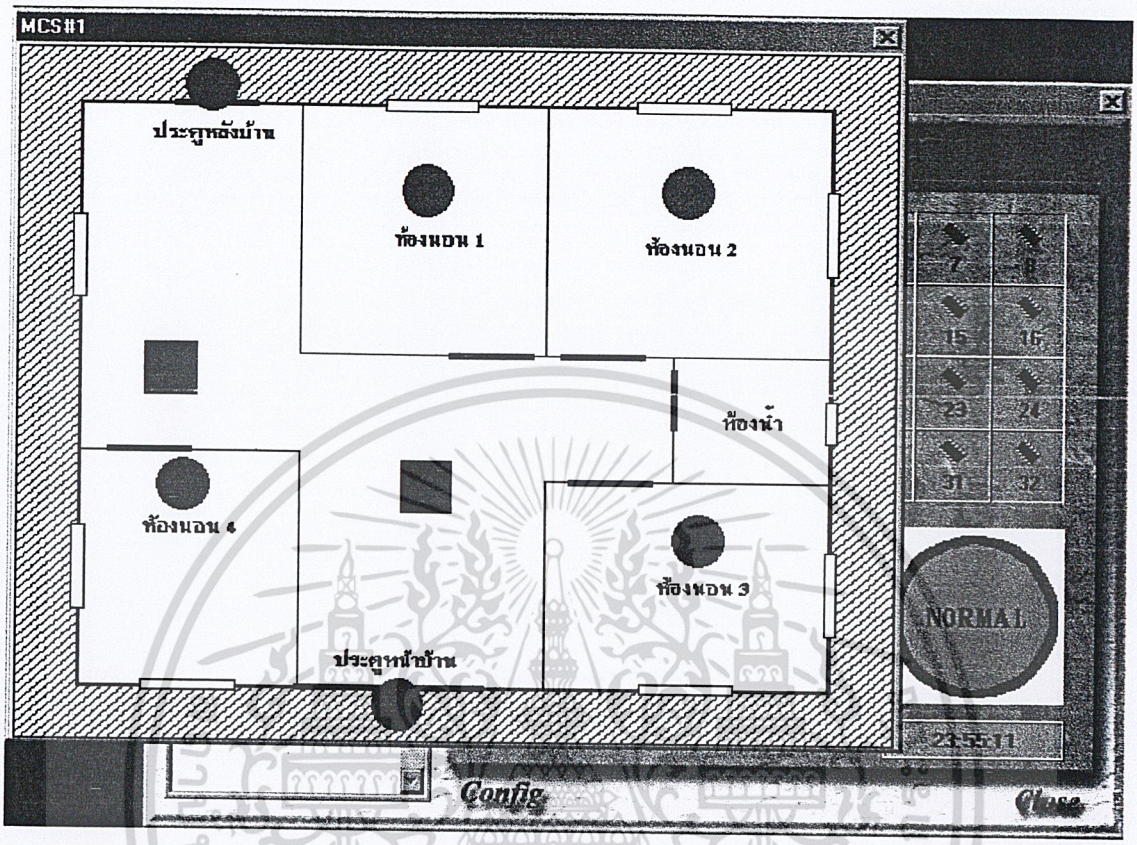


ภาพที่ 4.1 แสดงหน้าจอเมื่อกด Open Port

จาก ภาพที่4.1 จะเห็นว่าเมื่อเราใช้คำสั่งเปิดพอร์ต ที่หน้าจอจะขึ้นคำว่า Device Ready: COM1 หมายความว่า ขณะนี้ คอมพิวเตอร์ได้ทำการติดต่อกับ Searial Port แล้ว พร้อมจะทำการติดต่อแล้ว จากนั้นคอมพิวเตอร์จะส่งค่า แอดเดรสไปที่ Searial Port เพื่อทำการเช็ค MCS51 โดยหน้าจอจะขึ้น WRITE TO #MCS0, WRITE TO #MCS1 WRITE TO #MCS2 ขณะนี้คอมพิวเตอร์กำลังทำการส่งแอดเดรสอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 เปิดดูสถานะของบ้าน

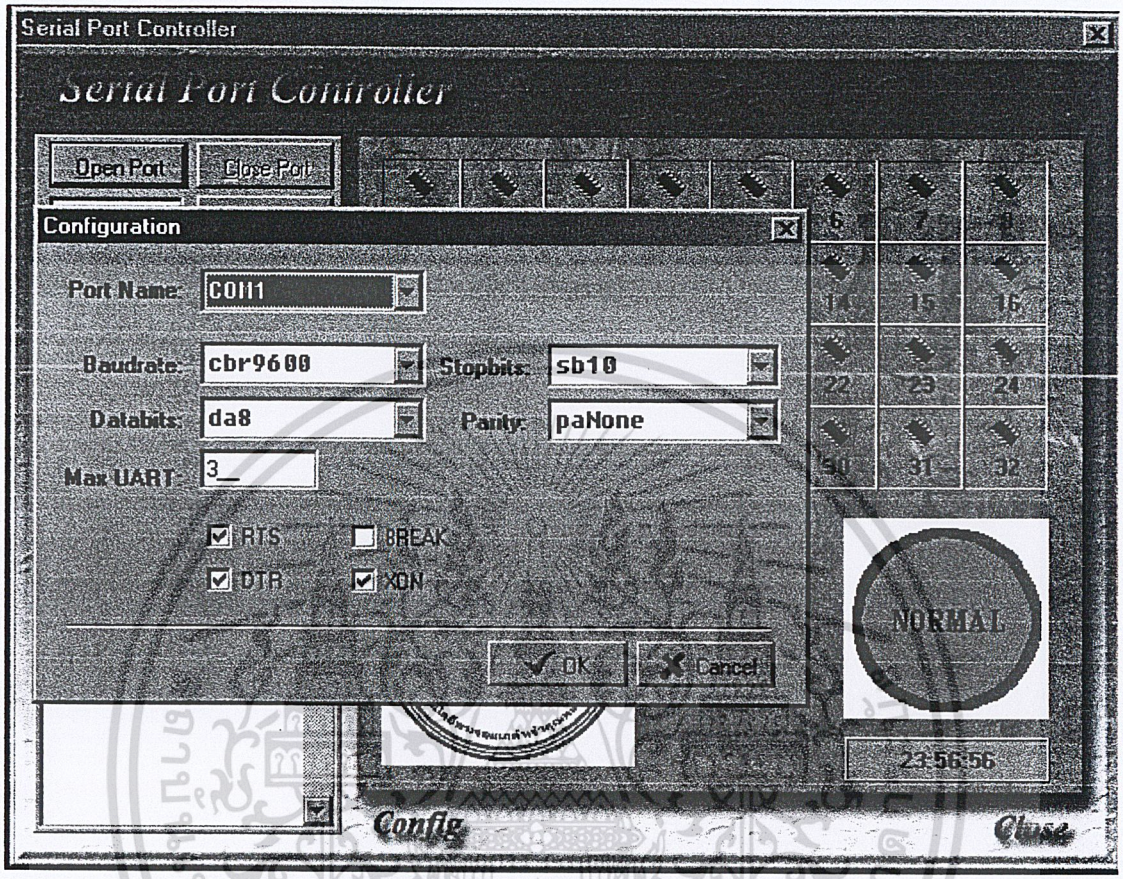


ภาพที่ 4.2 แสดงหน้าจอเมื่อกด หมายเลขบ้าน

โดยภาพ ไอซีแต่ละอันจะแทน MCS แต่ละตัว เมื่อเราคลิ๊กเข้าไปที่รูป ไอซีจะปรากฏแบบแปลนบ้านขึ้น ซึ่งจะมีการติดตั้งเซนเซอร์ตามจุดต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การ Set Config



ภาพที่ 4.2 แสดงหน้าจอเมื่อกด Config

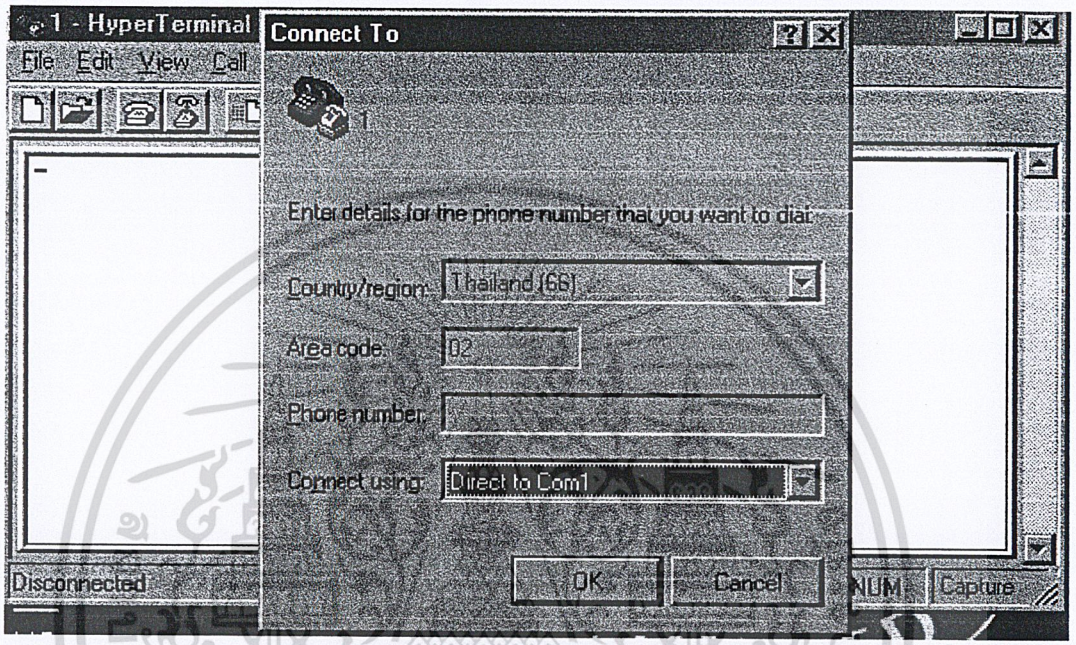
จากภาพที่ 4.2 เมื่อเรากด Config จะปรากฏค่าต่างๆซึ่งมีความหมายดังนี้

Port Name	คือ พอร์ตที่เราใช้งานติดต่อกับ อุปกรณ์ภายนอกในที่นี้เราต่อกับ PORT COM1. ใช้งาน
Baud Rate	คือ อัตราการส่งข้อมูล ความเร็ว เราใช้ 9600 บิตต่อวินาที
Databit	คือ จำนวนบิตที่ส่งออกไปกี่บิตใน 1 ครั้ง เราใช้ 8 บิต
Max UART	แทนจำนวนตัวของ MCS ในการวนลูปส่งใน 1 ครั้ง เราใช้ 3 ตัว
STOPBIT	บิตที่ใช้ส่งให้รู้ว่าข้อมูลสิ้นสุดแล้วในการส่ง
RTS	ใช้ควบคุมการทำงาน ขา RTS
DTR	ใช้ควบคุมการทำงาน ขา DTR
Parity	ใช้การส่งว่าจะกำหนดเป็นคี่หรือเป็นคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองใช้งานโปรแกรม

5.1 การใช้ร่วมกับโปรแกรม Hyper Terminal

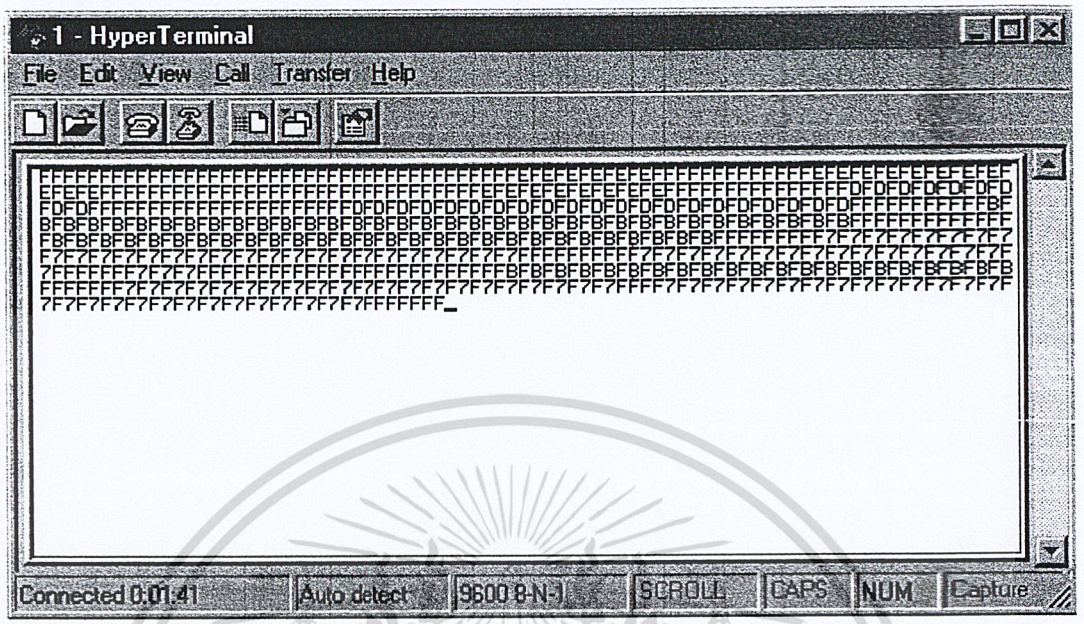


ภาพที่ 5.1 การเปิดใช้ Hyper Terminal

การเลือกใช้โปรแกรมเทอร์มินัล เพื่อใช้ในการทดสอบข้อมูลในไมโครคอนโทรลเลอร์ MCS-51ที่ได้ทำการโปรแกรมเอาไว้แล้ว เมื่อเราเข้าโปรแกรมเทอมินัลจะปรากฏหน้าจอดังภาพ 5.1 ให้ทำการ เซต DIREC TO COM 1 จากนั้นก็คลิ๊ก OK แล้วเข้าไป set ความเร็ว bit ที่จะใช้งาน คลิ๊ก OK. จะปรากฏหน้าจอให้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 การรันค่าที่โปรแกรมไว้ใน MCS-51

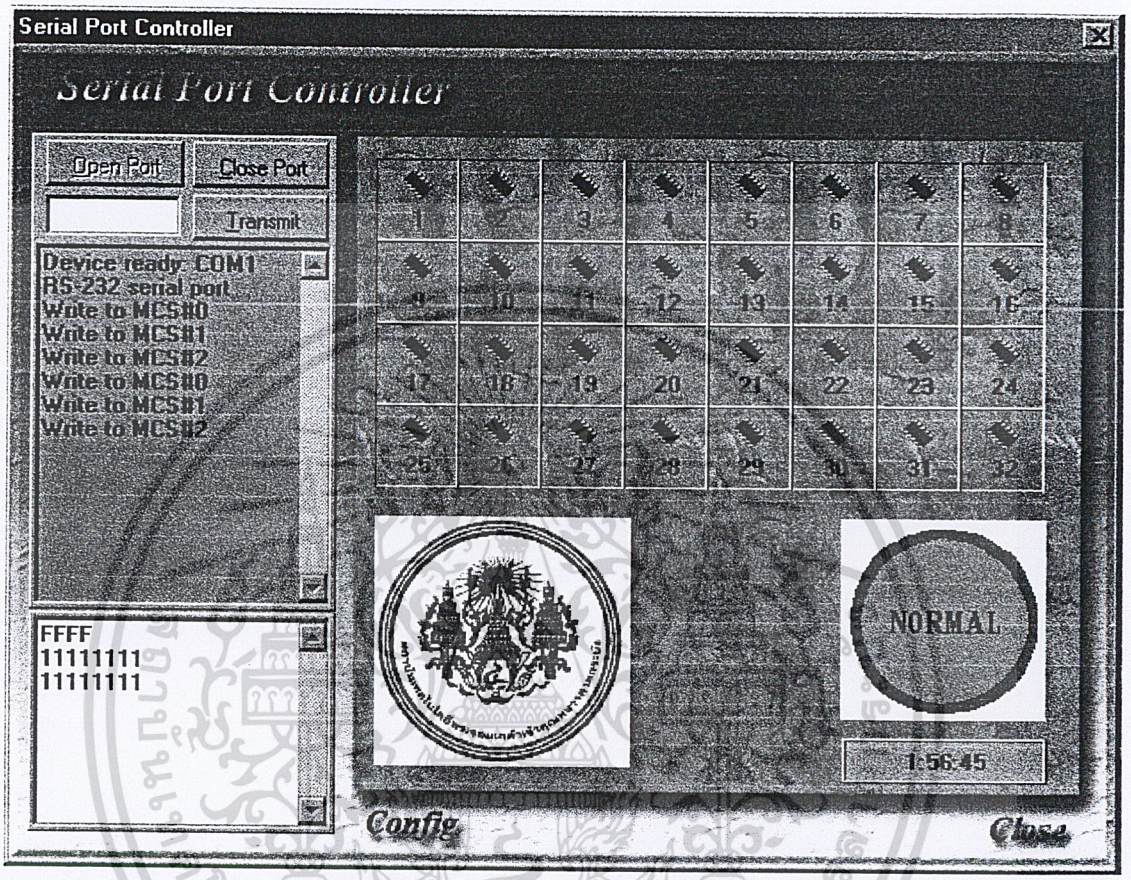


ภาพที่ 5.2 ลักษณะค่าที่ออกมา

จาก ภาพที่ 5.2 เมื่อเราต่อไมโครคอนโทรลเลอร์ MCS 51 เข้ากับโปรแกรมเทอร์มินัลจากนั้นลองส่งค่าเข้ามาที่โปรแกรมจะปรากฏเป็นเลขฐาน 16 แสดงค่าขึ้นมา FF คือ เซนเซอร์ไม่ทำงาน ถ้าเป็นลักษณะอื่น เช่น 7F BF จะแสดงว่ามีการกดของ สวิทช์ Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 เปิดทดสอบโปรแกรมจริง

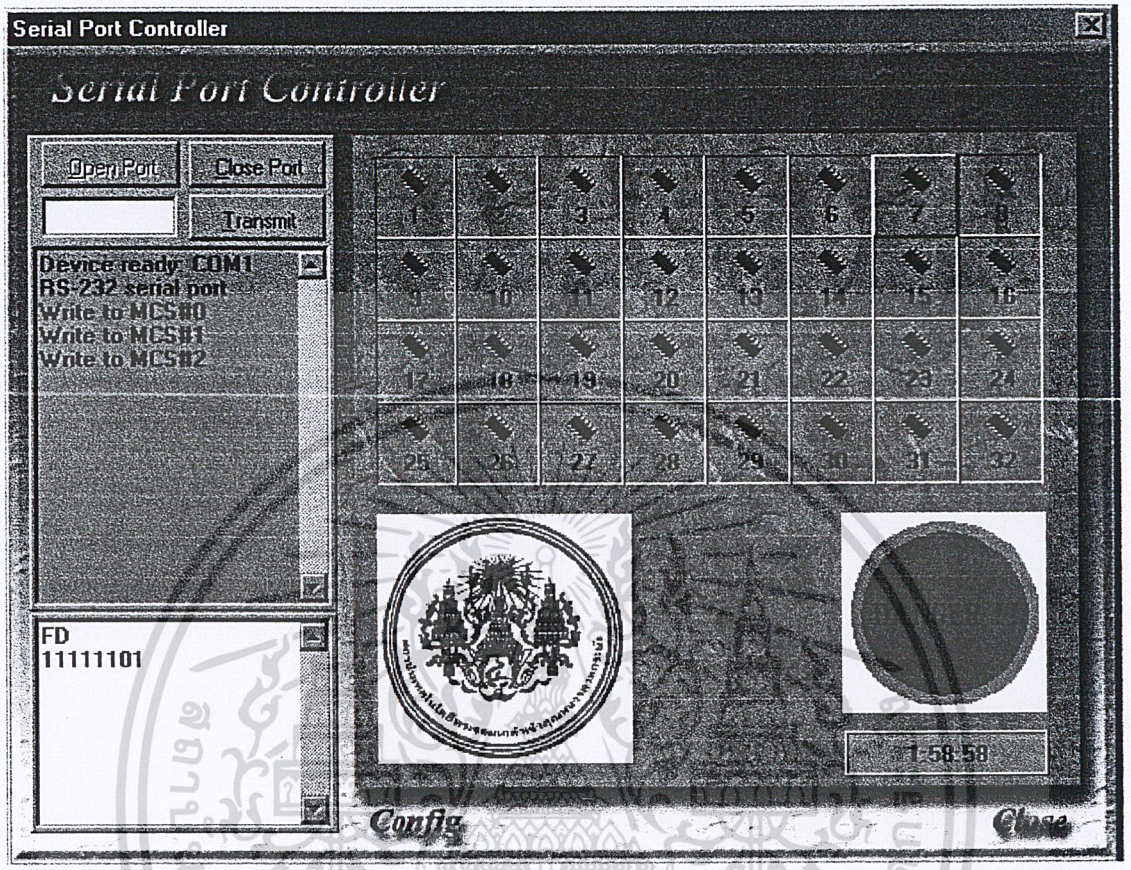


ภาพที่ 5.3 ลักษณะค่าที่รันออกมา

จากภาพที่ 5.3 จะเห็นว่า เมื่อคอมส่งค่าแอดเดรสไปที่ไมโครคอนโทรลเลอร์ MCS- 51 เพื่อจะให้ส่งค่าดาต้ากลับมาที่คอมพิวเตอร์จะเห็นว่าค่าดาต้าที่ส่งออกมาเป็น 11111111 หรือ FF ซึ่งแสดงว่าตอนนี้ เซนเซอร์ยังไม่ทำงานเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 เมื่อเกิดการทำงานของ Sensor

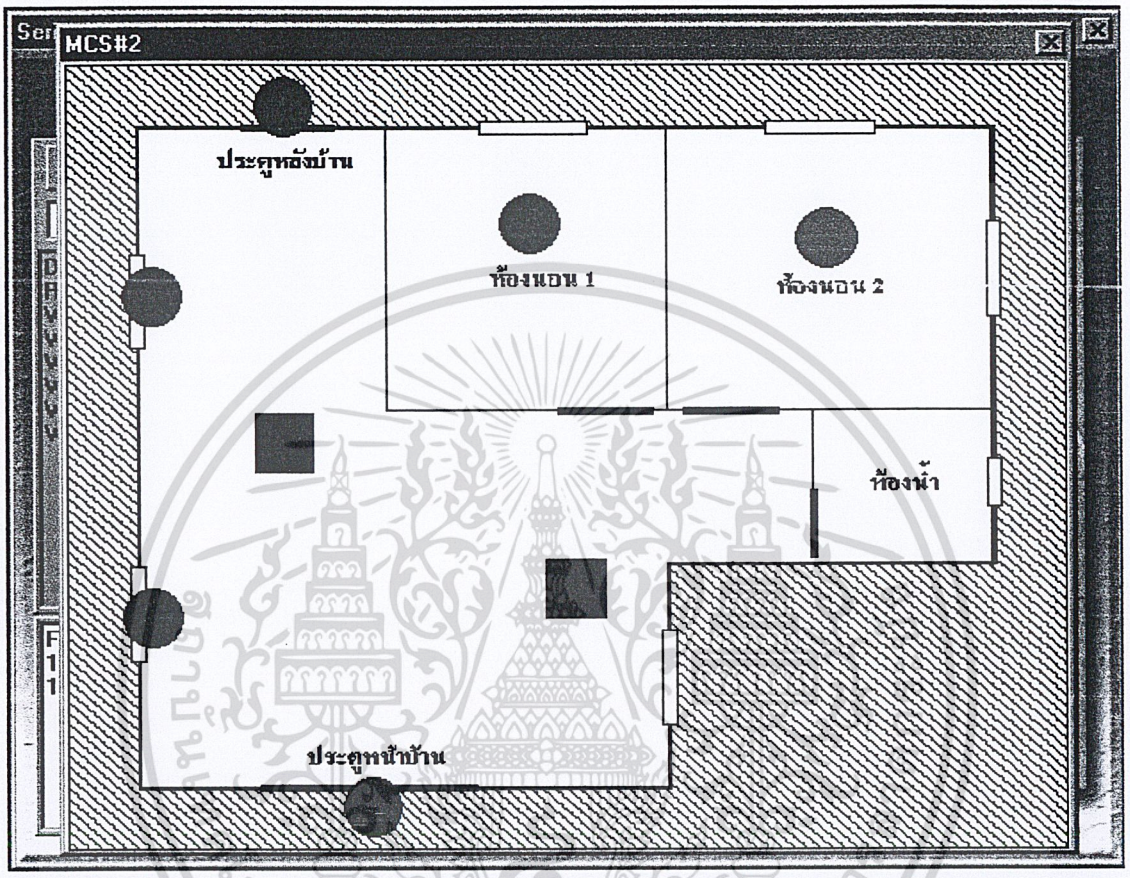


ภาพที่ 5.4 ลักษณะหน้าจอจะฟ้องเมื่อเกิด Alarm

จาก ภาพที่ 5.4 เมื่อเกิด Alam ขึ้นสัญลักษณ์ที่แทนไซเรนจะเปลี่ยนสีจากสีเขียวไปเป็นสีแดงและจะส่งเสียงร้องออกมาที่ลำโพงที่ต่อกับคอมเนื่องจากเซนเซอร์ตัวใดตัวหนึ่งได้ทำงาน สังเกตุจากค่าที่ปรากฏหน้าจอไม่เป็น FF แต่ เป็น FD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 ตรวจดูการทำงาน



ภาพที่ 5.5 แสดง Sensor ว่าตัวไหนทำงาน

จากภาพที่ 5.5 ตรวจดูการทำงานเราใช้เม้าคลิ๊กเข้าไปที่รูปไอซีที่เกิดเหตุการณ์ผิดปกติเกิดขึ้นจะเปลี่ยนสภาวะจากสีน้ำเงินไปเป็นสภาวะสีแดงจากนั้นดูสัญลักษณ์ซึ่งแทนเซนเซอร์แต่ละจุดที่เรา ได้ทำการติดตั้งไว้ภายในบริเวณบ้านตามจุดต่างๆถ้าเกิดเซนเซอร์ตัวใดทำงานก็จะเปลี่ยนสภาวะจากสีเขียวไปเป็นสีแดงแสดงให้เห็นว่า ขณะนี้ได้มีเหตุการณ์ ผิดปกติเกิดขึ้นตรงบริเวณนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 บทสรุป

การทำปริญญานิพนธ์นี้เป็นต้นแบบในการพัฒนาผลงานทำให้สามารถนำไปประยุกต์ใช้ในชีวิตประจำวัน โดยสามารถทดแทนทรัพยากรมนุษย์และต้องสามารถทำงานได้อย่างถูกต้อง แม่นยำ ไม่มีข้อผิดพลาด ซึ่งในสิ่งเหล่านี้มนุษย์อาจมีความผิดพลาดเกิดขึ้นได้ และในปัจจุบันนี้สิ่งอำนวยความสะดวกต่างๆ ก็เป็นปัจจัยหนึ่งที่มนุษย์จะขาดเสียมิได้

ปริญญานิพนธ์ฉบับนี้กล่าวถึงการทำงานของระบบ 2 ส่วนหลักคือ ส่วนทางด้านของไมโครคอนโทรลเลอร์ ส่วนทางด้านคอมพิวเตอร์ ในการทำปริญญานิพนธ์ฉบับนี้มีปัญหาต่างๆ มากมายหลายประการ ปัญหาบางอย่างก็แก้ไขได้อย่างง่าย แต่ปัญหาบางอย่างต้องใช้เวลาในการแก้ไข ไม่สามารถที่จะแก้ไขได้ทันที จึงถือว่าการทำปริญญานิพนธ์ฉบับนี้ให้หลายสิ่งหลายอย่างในการดำเนินชีวิตในยุคโลกไร้พรมแดนได้อย่างดี เนื่องจากโครงงานนี้ ต้องใช้ความคิดตั้งแต่ต้นในการวิเคราะห์ปัญหาที่อาจจะเกิดขึ้น แล้วนำปัญหาที่ได้ไปประเมินว่าควรหรือไม่ควรทำโครงงานนี้ จากนั้นก็นำไปสู่การ ออกแบบชิ้นงานให้มีความสอดคล้องและสัมพันธ์กัน จากนั้นก็มีการทดสอบการทำงานว่าสามารถทำงานได้ตามเป้าหมายที่กำหนดไว้ จากนั้นจึงสรุปผลที่ได้ออกมา หาสาเหตุของปัญหาที่เกิดขึ้นระหว่างกระบวนการต่างๆ พร้อมเสนอวิธีการแก้ไขปัญหาที่เกิดขึ้น จึงถือได้ว่าเป็นกระบวนการ ทางวิทยาศาสตร์ที่สมบูรณ์

การทำงานของตัวเครื่องเริ่มจากที่คอมพิวเตอร์ส่งค่าแอดเดรสมาที่ไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ส่งค่าดาต้า ซึ่งแทนค่าสภาวะของเซนเซอร์ทั้งหมด 8 ตัว ไปให้คอมพิวเตอร์เพื่อทำการนำค่าดาต้าเหล่านั้นไปแสดงผลหน้าจอคอมพิวเตอร์

6.2 ข้อเสนอแนะและแนวทางการพัฒนา

การทำปริญญานิพนธ์ครั้งนี้ยังมีปัญหาอีกมากมายในการทำงานคือ ไซเรนเตือนภัยไม่ดังตลอดเวลาเมื่อเซนเซอร์ทำงาน เมื่อใช้ 2 อันขึ้นไป และ กดสวิทช์ 2 อันพร้อมกันขึ้นไป ไม่สามารถแสดงผลที่หน้าจอได้ทั้งหมด

ทั้งนี้ได้ทำการเขียนโปรแกรมแบบวนลูปส่ง จึงทำให้เกิดปัญหาดังที่ได้กล่าวมาข้างต้น เนื่องจากแบบวนลูปส่งนี้ต้องมีการ StartBit และ StopBitmทำให้ไม่สามารถส่งแบบซ้ำๆกันได้

และสุดท้ายนี้คิดว่าควรที่จะมีผู้สนใจพัฒนาโครงงานนี้เพื่อที่จะได้สามารถนำไปใช้งานจริงในชีวิตประจำวันได้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม (PC-Serial port Interfacing text-Lab manual) โดย อรรถพล บุญยะโภคา,วรพจน์ กรแก้ววัฒนกุล , ชัยวัฒน์ ลิ้มพรจิตรวิไล

2.เรียนรู้และปฏิบัติไมโครคอนโทรลเลอร์ MCS-51Flash Microcontroller ฉบับ AT89C5X ของ Atmel

3.DELPHI 4 ผู้แต่ง กนก กุสุมาลย์นุกูล และ ไกรวุฒิ มั่นเสถียรสิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code MCS-51

```
; Program          : Send Data port to Pc
; Description      : Serial UART Exchange Data with PC
;                  : using Serial interrupt
; Filename         : Controller0.asm
; Assembler        : RAD51
;*****************************************************************************
RX_OK              BIT       F0              ; Define bit to keep received
already
;-----------------------------------------------------------------
; Define User Register
;-----------------------------------------------------------------
TX_DATA            EQU       02FH            ; For keep TX Data
DATASEND           EQU       030H
HEX_IN             EQU       031H
ASC_FO             EQU       032H
ASC_OF             EQU       033H
;-----------------------------------------------------------------
; Main Program.
;-----------------------------------------------------------------
                   ORG       0000H           ; Reset Vector
                   AJMP      MAIN            ; Jump to Main

                   ORG       0023H           ; TI+RI Vector
                   AJMP      SER_INT         ; Goto Serial interrupt
subroutine

MAIN:              MOV       TMOD,#021H      ; T1 8Bit Auto, T0 16Bit
                   MOV       TH1,#0FDH       ; 9600 bps Timer1 Default
                   MOV       TL1,#0FDH       ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     IE,#10010000B       ; En. EA, ES
                SETB    TR1                 ; Start Timer1
                MOV     SCON,#050H          ; Mode1 RX Enable
                CLR     RX_OK               ; Clear RX_OK flag
LOOP:           JNB     RX_OK,$             ; Wait message received
                ;CLR    TI
                ;MOV    TX_DATA,#040H
                ;SETB   TI
                MOV     DATASEND,#000H      ;SEND address 0
                ACALL   ASC_SEND

                MOV     DATASEND,P1         ;Send data port 1
                ACALL   ASC_SEND

                ;CLR    TI
                ;MOV    TX_DATA,#00AH
                ;SETB   TI
                ;CLR    TI
                ;MOV    TX_DATA,#00DH
                ;SETB   TI
                CLR RX_OK
                AJMP    LOOP                ; Back to received next
message
;---------------------------------------------------------------------
; Serial interrupt subroutine
;---------------------------------------------------------------------
SER_INT:        PUSH    ACC                 ; Push ACC. to stack

                JBC     RI,SER_RX           ; RX or TX? If RX then
received and clear RI
SER_TX:         CLR     TI                  ; Clear TI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV         SBUF,TX_DATA
SER_TX_WAIT:        JBC         TI,SER_EXIT         ; Wait until TX already
(TI=1) then clear TI
                    AJMP        SER_TX_WAIT         ; Wait for TX
SER_RX:             CLR         RX_OK               ; Clear RI
                    MOV         A,SBUF                      ; Get data to ACC.
                    ;MOV        @R0,A               ; Move to Buffer @R0
                    ;INC        R0                  ; Increase R0
                    CJNE        A,#030H,SER_EXIT    ; Check address controlier
                    SETB        RX_OK               ; Set RX_OK received
message ok.
SER_EXIT:           POP         ACC         ; Pop Acc. to Main
                    RETI                    ; Return interrupt
;-------------------------------------------------------------
;Send ASC Code to serial
;-------------------------------------------------------------
ASC_SEND:           MOV         HEX_IN,DATASEND
                    LCALL HEX_ASC
                    CLR         TI
                    MOV         TX_DATA,ASC_FO
                    SETB        TI
                    CLR         TI
                    MOV         TX_DATA,ASC_OF
                    SETB        TI
                    RET
;-------------------------------------------------------------------
;Convers Hex to ASC Code data in = HEX_IN data out = ASC_OF,ASC_FO
;-------------------------------------------------------------------
HEX_ASC:            MOV         A,HEX_IN
                    ANL         A,#00FH
                    MOV         B,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

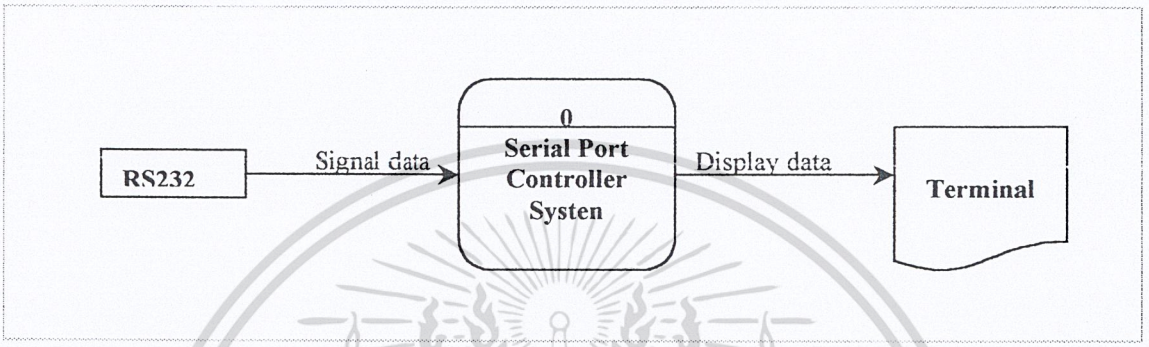
```
                CLR   C
                SUBB  A,#00AH
                MOV       A,B
                      JC          HEX_ASC1
                ADD       A,#007H
HEX_ASC1:             ADD         A,#030H
                MOV       ASC_OF,A
                MOV       A,HEX_IN
                ANL       A,#0F0H
                SWAP      A
                MOV       B,A
                CLR       C
                SUBB      A,#00AH
                MOV       A,B
                JC        HEX_ASC2
                ADD       A,#007H
HEX_ASC2:             ADD         A,#030H
                MOV       ASC_FO,A
                RET

;-------------------------------------------------------------
; Dummy Delay time 100m
;-------------------------------------------------------------
DELAY_100ms:    MOV       R7,#100         ; Do 100 times
DELAY_100ms_1:  MOV       R6,#0E6H        ; Each loop = 1 ms
DELAY_100ms_2:  NOP
                NOP
                DJNZ      R6,DELAY_100ms_2
                DJNZ      R7,DELAY_100ms_1
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System design

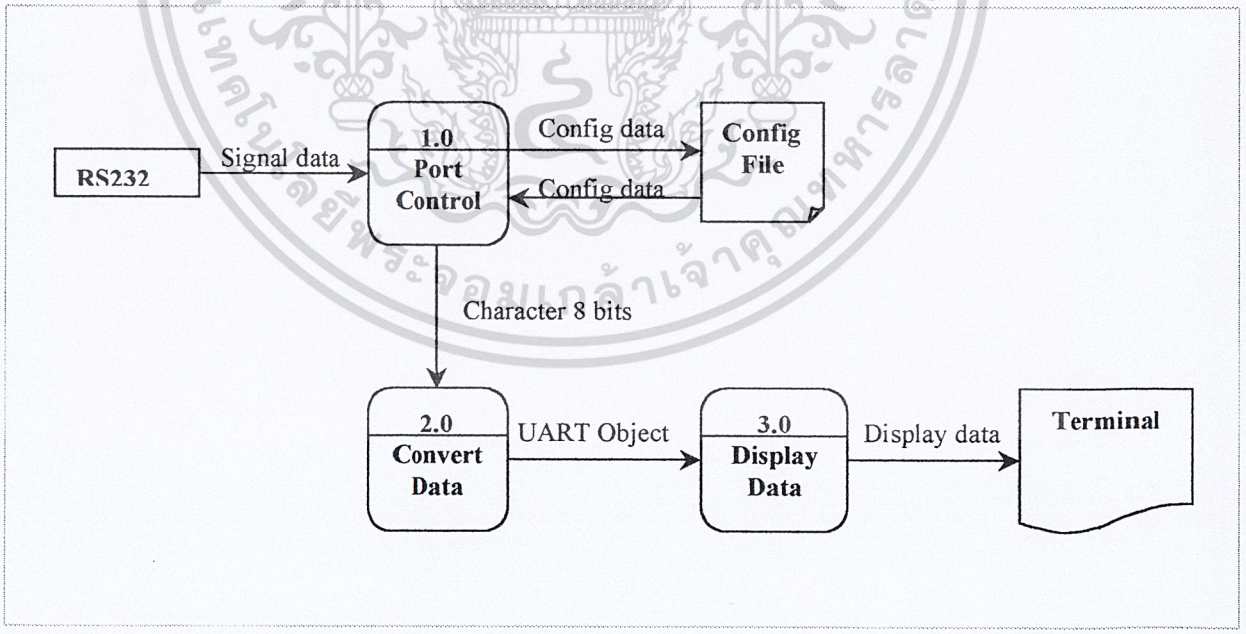
Context Diagram
Level 0 Data Flow Diagram

Context diagram is the overview of all environment scope in the system. It's show communication between external variable and the system, that's mean RS232 and terminal computer



Data Flow Diagram
Level 1 Serial Port Controller System

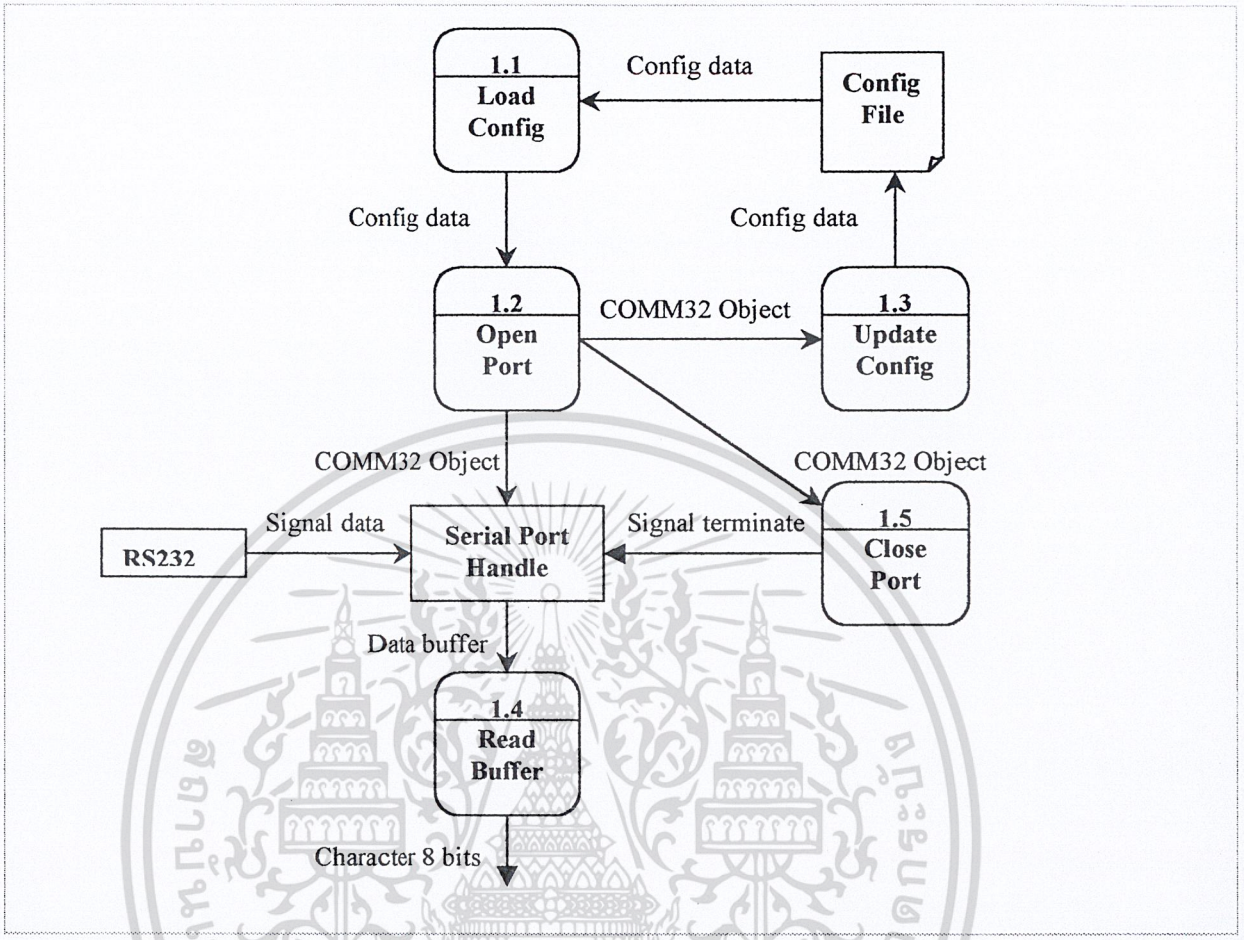
This figure show data flow diagram in all main details module



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Level 2 Data Flow Diagram

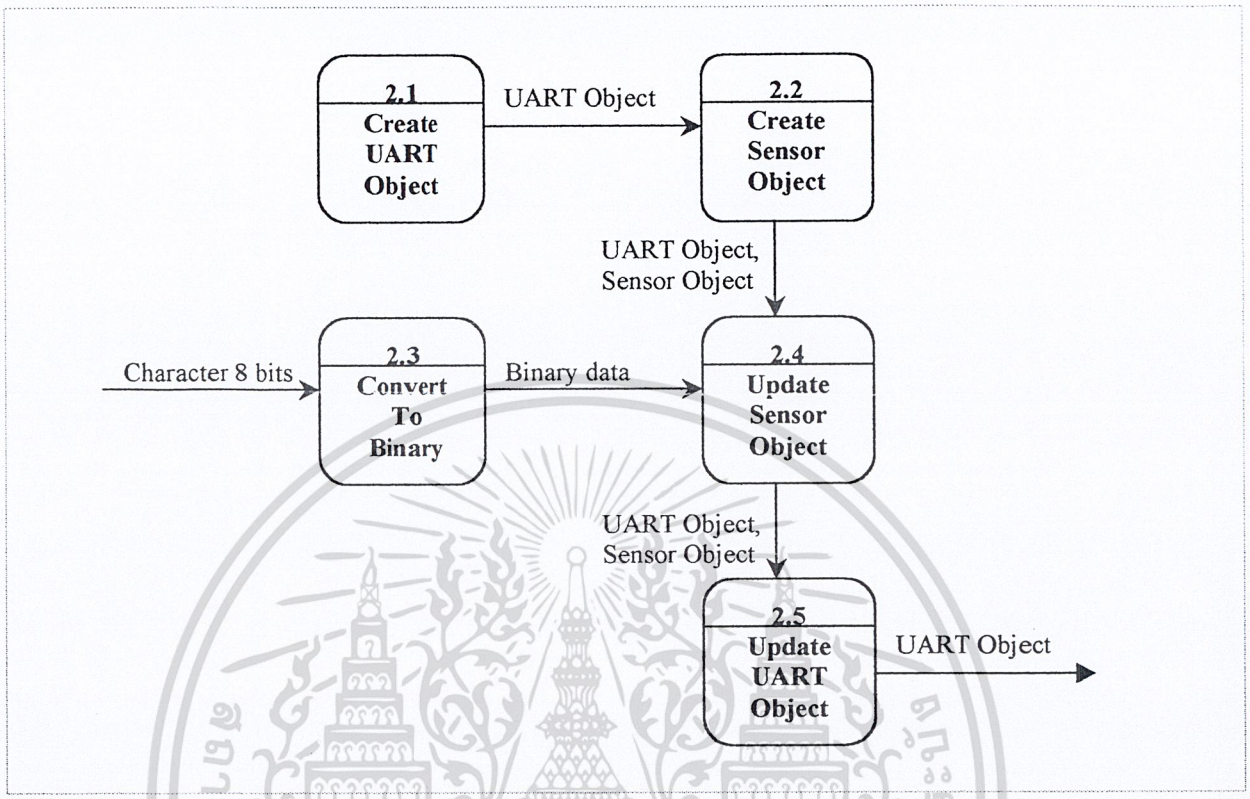
1.0 Port Control



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

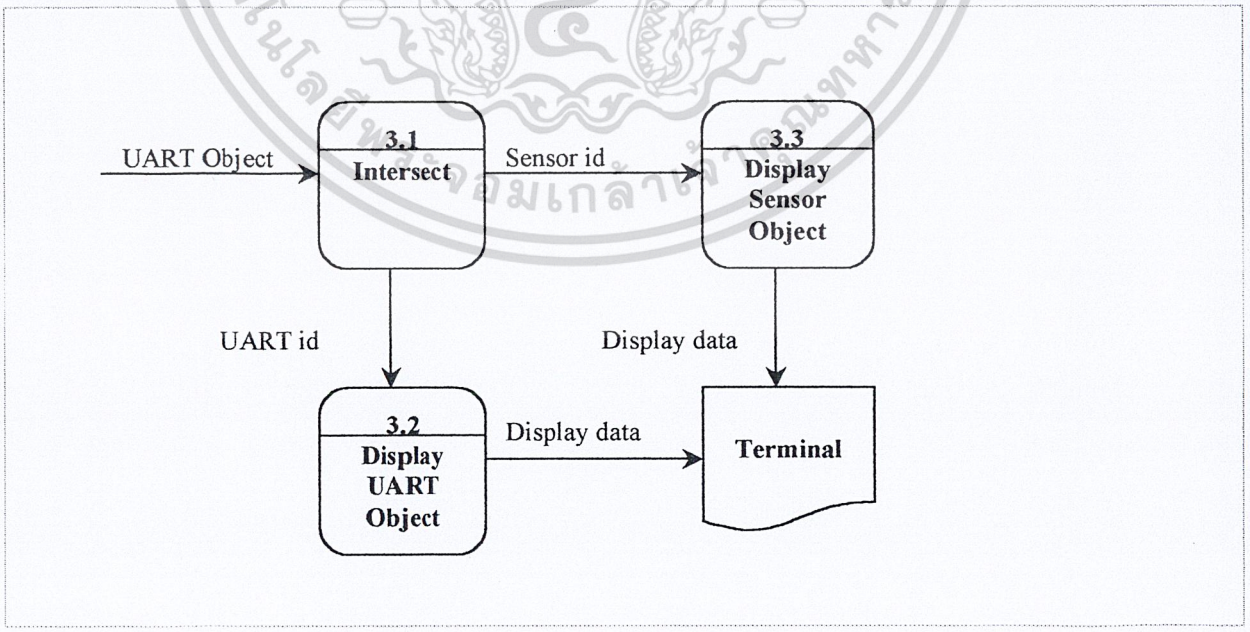
Level 2 Data Flow Diagram

2.0 Convert Data



Level 2 Data Flow Diagram

3.0 Display Data



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program design

- Program Module Specification

Designed by Data Flow Diagram

- 1.Port Control
  - 1.1 Load Config
  - 1.2 Open Port
  - 1.3 Update Config
  - 1.4 Read Buffers
  - 1.5 Close Port

- 2.Convert Data
  - 2.1 Create UART Objects
  - 2.2 Create Sensor Objects
  - 2.3 Convert to Binary
  - 2.4 Update Sensor Objects
  - 2.5 Update UART Objects

- 3.Display Data
  - 3.1 Intersection
  - 3.2 Display UART Objects
  - 3.3 Display Sensor Objects

For program coding,this devide to 3 parts of main module

1. Processing for serial port receiving communication by create a class of COMM object.
2. After received data from buffers of serial port then calculation for data display.
3. Display graphic user interface.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code

This application deverlope by Delphi language

Class object of COMM32 : File ASYNC32.PAS

Main class : File Main.pas

Sensor class : File Sensor.pas

Config class : File Config.pas

Common function : File KConfig.pas

```
unit KConfig;

interface
uses SysUtils,Math;
    function KPower(base : Integer;x : Integer):Integer;
    function ConvertHexToBinary(x : string):String;
    function ConvertBinaryToDec(x : string):Integer;
implementation
//---------------------------------------------------------------------------
function KPower(base : Integer;x : Integer):Integer;
var ret,i : integer;
begin
   if x = 0 then
      result := 1
   else
   begin
      ret := 0;
      For i := 1 to x do
      begin
         ret := ret + base * 2;
      end;
      result := ret;
   end;
end;
//---------------------------------------------------------------------------
function ConvertHexToBinary(x : string):String;
var
   A :Array[1..8] of char;
   i,remain,dev,res,Temp1,Temp2 : Integer;
   ch1,ch2 : char;
begin
   Temp1 := 0;
   Temp2 := 0;
   for i := 1 to 8 do
      A[i] := '0';
   //---------- Decode first character----------
   ch1 := x[1];
```

เอกสารนี้จัดทำขึ้นเพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ch2 := x[2];

IF(ch1 >= '0') AND (ch1 <= '9') Then
    Temp1 := StrToInt(ch1)
ELSE
Begin
    case ch1 of
      'A': Temp1 := 10;
      'B': Temp1 := 11;
      'C': Temp1 := 12;
      'D': Temp1 := 13;
      'E': Temp1 := 14;
      'F': Temp1 := 15;
    end;

End;
i := 4;
dev := Temp1;res := Temp1;
while dev > 1 do
begin
    remain := dev mod 2;
    if remain = 1 then
        A[i] := '1'
    else A[i] := '0';
    res := dev div 2;
    dev := res;
    dec(i);
end;
if res = 1 then
    A[i] := '1'
else A[i] := '0';
//---------- Decode second character ----------
IF(ch2 >= '0') AND (ch2 <= '9') Then
    Temp2 := StrToInt(ch2)
ELSE
Begin
    case ch2 of
      'A': Temp2 := 10;
      'B': Temp2 := 11;
      'C': Temp2 := 12;
      'D': Temp2 := 13;
      'E': Temp2 := 14;
      'F': Temp2 := 15;
    end;

End;
i := 8;
dev := Temp2;res := Temp2;
while dev > 1 do
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        remain := dev mod 2;
        if remain = 1 then
            A[i] := '1'
        else A[i] := '0';
        res := dev div 2;
        dev := res;
        dec(i);
    end;
    if res = 1 then
        A[i] := '1'
    else A[i] := '0';
    //---------------------------------

    result := A;
end;
//-------------------------------------------------------------------------
function ConvertBinaryToDec(x : string):Integer;
var i,ret,base : integer;
begin
    ret := 0;
    For i := 1 to Length(x) do
    begin
        IF x[i] = '1' Then
        begin
            base := KPower(2,i - 1);
            ret := ret + base;
        end;
    end;

    result := ret;
end;
//-------------------------------------------------------------------------
end.
```

==========

File Main.pas

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, ExtCtrls, ToolWin, Menus, StdCtrls, Async32, Buttons,
  IniFiles,QControls, OleCtnrs, MPlayer;
type
  TFMain = class(TForm)
    Panel1: TPanel;
    Comm1: TComm;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Panel2: TPanel;

Memo2: TMemo;

Panel3: TPanel;

BOpen: TBitBtn;

BClose: TBitBtn;

BTransmit: TBitBtn;

ETransmit: TEdit;

Memo1: TMemo;

Splitter1: TSplitter;

Image1: TImage;

BExit: TImage;

BConfig: TImage;

Timer1: TTimer;

PTime: TPanel;

B1: TSpeedButton;

B2: TSpeedButton;

B3: TSpeedButton;

B4: TSpeedButton;

B5: TSpeedButton;

B6: TSpeedButton;

B7: TSpeedButton;

B8: TSpeedButton;

B9: TSpeedButton;

B10: TSpeedButton;

B11: TSpeedButton;

B12: TSpeedButton;

B13: TSpeedButton;

B14: TSpeedButton;

B15: TSpeedButton;

B16: TSpeedButton;

B17: TSpeedButton;

B18: TSpeedButton;

B19: TSpeedButton;

B20: TSpeedButton;

B21: TSpeedButton;

B22: TSpeedButton;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B23: TSpeedButton;

B24: TSpeedButton;

B25: TSpeedButton;

B26: TSpeedButton;

B27: TSpeedButton;

B28: TSpeedButton;

B29: TSpeedButton;

B30: TSpeedButton;

B31: TSpeedButton;

B32: TSpeedButton;

Bevel1: TBevel;

Bevel2: TBevel;

Bevel3: TBevel;

Bevel4: TBevel;

Bevel5: TBevel;

Bevel6: TBevel;

Bevel7: TBevel;

Bevel8: TBevel;

Bevel9: TBevel;

Bevel10: TBevel;

Bevel11: TBevel;

Bevel12: TBevel;

Bevel13: TBevel;

Bevel14: TBevel;

Bevel15: TBevel;

Bevel16: TBevel;

Bevel17: TBevel;

Bevel18: TBevel;

Bevel19: TBevel;

Bevel20: TBevel;

Bevel21: TBevel;

Bevel22: TBevel;

Bevel23: TBevel;

Bevel24: TBevel;

Bevel25: TBevel;

Bevel26: TBevel;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Bevel27: TBevel;
Bevel28: TBevel;
Bevel29: TBevel;
Bevel30: TBevel;
Bevel31: TBevel;
Bevel32: TBevel;
Image2: TImage;
ImageWarning: TImage;
Timer2: TTimer;
procedure FormCreate(Sender: TObject);
procedure BOpenClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure BCloseClick(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure BTransmitClick(Sender: TObject);
procedure Comm1DSR(Sender: TObject);
procedure Comm1Error(Sender: TObject; Errors: Integer);
procedure Comm1RxChar(Sender: TObject; Count: Integer);
procedure Comm1TxEmpty(Sender: TObject);
procedure BExitClick(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure BConfigClick(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure B1Click(Sender: TObject);
procedure B2Click(Sender: TObject);
procedure B3Click(Sender: TObject);
procedure B4Click(Sender: TObject);
procedure B5Click(Sender: TObject);
procedure B6Click(Sender: TObject);
procedure B7Click(Sender: TObject);
procedure B8Click(Sender: TObject);
procedure B9Click(Sender: TObject);
procedure B10Click(Sender: TObject);
procedure B11Click(Sender: TObject);
procedure B12Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure B13Click(Sender: TObject);
    procedure B14Click(Sender: TObject);
    procedure B15Click(Sender: TObject);
    procedure B16Click(Sender: TObject);
    procedure B17Click(Sender: TObject);
    procedure B19Click(Sender: TObject);
    procedure B21Click(Sender: TObject);
    procedure B23Click(Sender: TObject);
    procedure B25Click(Sender: TObject);
    procedure B18Click(Sender: TObject);
    procedure B27Click(Sender: TObject);
    procedure B29Click(Sender: TObject);
    procedure B31Click(Sender: TObject);
    procedure B20Click(Sender: TObject);
    procedure B22Click(Sender: TObject);
    procedure B24Click(Sender: TObject);
    procedure B26Click(Sender: TObject);
    procedure B28Click(Sender: TObject);
    procedure B30Click(Sender: TObject);
    procedure B32Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
  private
    { Private declarations }
    FCurrentLine: Integer;
    procedure UpdateControls;
    //Catch global exceptions
    procedure HandleException(Sender: TObject; E: Exception);
    procedure UARTClick(Sender: integer);
    procedure LoadConfig;
    procedure DecodeData(Data : String);
  public
    { Public declarations }
    //UART : array[1..32] of TSpeedButton;
    SENSOR  : array[1..32,1..16] of Integer;
    MAXUART : Integer;
    MAXMCS  : Integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MCS     : Integer;
    SendFlag : Boolean;
    MCSWarning : boolean;
    ImageFlag : Integer;
  end;

var
  FMain: TFMain;

implementation

uses Config, Sensor, KConfig, Sensor1, Sensor2, Sensor3;

const
  OnOff: array[0..1] of string = ('Off', 'On');

{$R *.dfm}
//---------------------------------------------------------------------------
procedure TFMain.FormCreate(Sender: TObject);
begin
    SendFlag := False;
    MAXMCS := 1;
    MCS := 0;
    Memo1.Clear;
    Memo2.Clear;
    Application.OnException := HandleException;
    FCurrentLine := 0;
    LoadConfig;
    UpdateControls;
    MCSWarning := False;
end;
//---------------------------------------------------------------------------
procedure TFMain.FormDestroy(Sender: TObject);
begin
    Application.OnException := nil;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//---------------------------------------------------------------------------
procedure TFMain.LoadConfig;
var
    ConfigFile : TIniFile;
    i,j : Integer;
begin
    For i := 1 to 32 do
        For j := 1 to 16 do
            SENSOR[i,j] := 1;
    ConfigFile  := TIniFile.Create('KConfig.sys');
    try
        Comm1.DeviceName := 'COM' + IntToStr(ConfigFile.ReadInteger('COMM32','Port',0) + 1);
        Comm1.BaudRate := TBaudrate(ConfigFile.ReadInteger('COMM32','BaudRate',0));
        Comm1.Databits := TDataBits(ConfigFile.ReadInteger('COMM32','DataBits',0));
        Comm1.Parity   := TParity(ConfigFile.ReadInteger('COMM32','Parity',0));
        Comm1.StopBits := TStopBits(ConfigFile.ReadInteger('COMM32','StopBits',0));
        MAXMCS := ConfigFile.ReadInteger('COMM32','MAXUART',0)
    except
        on E: ECommError do
            Memo1.Lines.add(E.Message);
    end;
    ConfigFile.Free;
end;
//---------------------------------------------------------------------------
procedure TFMain.UpdateControls;
begin
    BOpen.Enabled :=not Comm1.Enabled;
    BClose.Enabled := Comm1.Enabled;
    BTransmit.Enabled := Comm1.Enabled and Comm1.DSR;
end;
//---------------------------------------------------------------------------
procedure TFMain.UARTClick(Sender: integer);
begin
    case Sender of
    1:
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        IF(FSensor1 = nil)then
        begin
            Application.CreateForm(TFSensor1, FSensor1);
            FSensor1.ShowParam(Sender);
        end;
      end;
      2:
      begin
        IF(FSensor2 = nil)then
        begin
            Application.CreateForm(TFSensor2, FSensor2);
            FSensor2.ShowParam(Sender);
        end;
      end;
      3:
      begin
        IF(FSensor3 = nil)then
        begin
            Application.CreateForm(TFSensor3, FSensor3);
            FSensor3.ShowParam(Sender);
        end;
      end;
      else
      begin
        IF(FSensor = nil)then
        begin
            Application.CreateForm(TFSensor, FSensor);
            FSensor.ShowParam(Sender);
        end;
      end;
    end;
end;
//---------------------------------------------------------------------------
procedure TFMain.DecodeData(Data : String);
var i,j,temp,count,flag : Integer;
    str : String;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
    MCSWarning := False;
    For i := 1 to 16 do
        SENSOR[MCS,i] := 1;
    count := 0;
    For i := 1 to Length(Data) do
    begin
        count := count + 1;
        str := str + Data[i];
        IF count = 2 then
        begin
            //--------- Decode to decimal ----------
            str := ConvertHexToBinary(str);
            Memo2.Lines.Add(str);
            temp := 0;
            For j := Length(str) downto 1 do
            begin
                temp := temp + 1;
                IF str[j] = '0' Then
                begin
                    flag := temp;
                    break;
                end;
            end;
            IF (flag >= 1) AND (flag <= 16) Then
            begin
                SENSOR[MCS,flag] := 0;

                MCSWarning := True;
                ImageFlag := 1;

                case MCS of
                  1: B1.Font.Color := clRed;
                  2: B2.Font.Color := clRed;
                  3: B3.Font.Color := clRed;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        4: B4.Font.Color := clRed;
        5: B5.Font.Color := clRed;
        6: B6.Font.Color := clRed;
        7: B7.Font.Color := clRed;
        8: B8.Font.Color := clRed;
        9: B9.Font.Color := clRed;
        10: B10.Font.Color := clRed;
        11: B11.Font.Color := clRed;
        12: B12.Font.Color := clRed;
        13: B13.Font.Color := clRed;
        14: B14.Font.Color := clRed;
        15: B15.Font.Color := clRed;
        16: B16.Font.Color := clRed;
        17: B17.Font.Color := clRed;
        18: B18.Font.Color := clRed;
        19: B19.Font.Color := clRed;
        20: B20.Font.Color := clRed;
        21: B21.Font.Color := clRed;
        22: B22.Font.Color := clRed;
        23: B23.Font.Color := clRed;
        24: B24.Font.Color := clRed;
        25: B25.Font.Color := clRed;
        26: B26.Font.Color := clRed;
        27: B27.Font.Color := clRed;
        28: B28.Font.Color := clRed;
        29: B29.Font.Color := clRed;
        30: B30.Font.Color := clRed;
        31: B31.Font.Color := clRed;
        32: B32.Font.Color := clRed;
      end;

end
Else
begin
    case MCS of
        1: B1.Font.Color := clBlue;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        2: B2.Font.Color := clBlue;
        3: B3.Font.Color := clBlue;
        4: B4.Font.Color := clBlue;
        5: B5.Font.Color := clBlue;
        6: B6.Font.Color := clBlue;
        7: B7.Font.Color := clBlue;
        8: B8.Font.Color := clBlue;
        9: B9.Font.Color := clBlue;
        10: B10.Font.Color := clBlue;
        11: B11.Font.Color := clBlue;
        12: B12.Font.Color := clBlue;
        13: B13.Font.Color := clBlue;
        14: B14.Font.Color := clBlue;
        15: B15.Font.Color := clBlue;
        16: B16.Font.Color := clBlue;
        17: B17.Font.Color := clBlue;
        18: B18.Font.Color := clBlue;
        19: B19.Font.Color := clBlue;
        20: B20.Font.Color := clBlue;
        21: B21.Font.Color := clBlue;
        22: B22.Font.Color := clBlue;
        23: B23.Font.Color := clBlue;
        24: B24.Font.Color := clBlue;
        25: B25.Font.Color := clBlue;
        26: B26.Font.Color := clBlue;
        27: B27.Font.Color := clBlue;
        28: B28.Font.Color := clBlue;
        29: B29.Font.Color := clBlue;
        30: B30.Font.Color := clBlue;
        31: B31.Font.Color := clBlue;
        32: B32.Font.Color := clBlue;
      end;
  end;
  //---------- Reset variable ----------
  count := 0;
  str := '';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;

        case MCS of
          1:
            IF FSensor1 <> nil Then
                FSensor1.UpdateControl(MCS);
          2:
            IF FSensor2 <> nil Then
                FSensor2.UpdateControl(MCS);
          3:
            IF FSensor3 <> nil Then
                FSensor3.UpdateControl(MCS);
          else
            IF FSensor <> nil Then
                FSensor.UpdateControl(MCS);
        end;
    end;
end;
//---------------------------------------------------------------------
procedure TFMain.HandleException(Sender: TObject; E: Exception);
begin
    if E is ECommError then
        with E as ECommError do
        ShowMessage('Async32 error: ' + Message);
end;
//---------------------------------------------------------------------
procedure TFMain.BOpenClick(Sender: TObject);
begin
    LoadConfig;
    MCS := 0;
    ImageFlag := 0;
    try
        Comm1.Open;
        Memo1.Lines.add('Device ready: ' + Comm1.DeviceName);
        Memo1.Lines.add(GetProviderSubtypeName(Comm1.ProviderSubtype));
        SendFlag := True;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    except
        on E: ECommError do
            Memo1.Lines.add(E.Message);
    end;
    UpdateControls;
end;
//---------------------------------------------------------------------------

procedure TFMain.BCloseClick(Sender: TObject);
begin
    ImageFlag := 0;
    SendFlag := False;
    Comm1.Close;
    Memo1.Lines.Add('Device closed: ' + Comm1.DeviceName);
    UpdateControls;
    ImageWarning.Picture.LoadFromFile('Normal.bmp');
end;
//---------------------------------------------------------------------------
procedure TFMain.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
    if(Comm1 = nil)then
        Comm1.Close;
    CanClose := true;
end;
//---------------------------------------------------------------------------
procedure TFMain.BTransmitClick(Sender: TObject);
var
  S: string;
  Count: Integer;
begin
  S := ETransmit.Text;
  if Length(S) = 0 then exit;
  Count := Comm1.Write(S[1], Length(S));
  if Count = -1 then
    Memo1.Lines.add('Error writing to: ' + Comm1.DeviceName)
  else Memo1.Lines.add('Transmitting ' + IntToStr(Count) + ' characters');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
//---------------------------------------------------------------------------
procedure TFMain.Comm1DSR(Sender: TObject);
begin
    Memo1.Lines.add('DSR: ' + OnOff[ord(Comm1.DSR)]);
    UpdateControls;
end;
//---------------------------------------------------------------------------
procedure TFMain.Comm1Error(Sender: TObject; Errors: Integer);
begin
 if (Errors and CE_BREAK > 0) then
   Memo1.Lines.add('The hardware detected a break condition.');
 if (Errors and CE_DNS > 0) then
   Memo1.Lines.add('Windows 95 only: A parallel device is not selected.');
 if (Errors and CE_FRAME > 0) then
   Memo1.Lines.add('The hardware detected a framing error.');
 if (Errors and CE_IOE > 0) then
   Memo1.Lines.add('An I/O error occurred during communications with the device.');
 if (Errors and CE_MODE > 0) then
 begin
   Memo1.Lines.add('The requested mode is not supported, or the hFile parameter');
   Memo1.Lines.add('is invalid. If this value is specified, it is the only valid error.');
 end;
 if (Errors and CE_OOP > 0) then
   Memo1.Lines.add('Windows 95 only: A parallel device signaled that it is out of paper.');
 if (Errors and CE_OVERRUN > 0) then
   Memo1.Lines.add('A character-buffer overrun has occurred. The next character is lost.');
 if (Errors and CE_PTO > 0) then
   Memo1.Lines.add('Windows 95 only: A time-out occurred on a parallel device.');
 if (Errors and CE_RXOVER > 0) then
 begin
   Memo1.Lines.add('An input buffer overflow has occurred. There is either no');
   Memo1.Lines.add('room in the input buffer, or a character was received after');
   Memo1.Lines.add('the end-of-file (EOF) character.');
 end;
 if (Errors and CE_RXPARITY > 0) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Memo1.Lines.add('The hardware detected a parity error.');
  if (Errors and CE_TXFULL > 0) then
  begin
    Memo1.Lines.add('The application tried to transmit a character, but the output');
    Memo1.Lines.add('buffer was full.');
  end;
end;
//---------------------------------------------------------------------------
procedure TFMain.Comm1RxChar(Sender: TObject; Count: Integer);
var
  Buffer: array[0..1024] of Char;
  P,Bytes : Integer;
  Data : String;
begin
  Fillchar(Buffer, Sizeof(Buffer), 0);
  Bytes := Comm1.Read(Buffer, Count);
  if Bytes = -1 then
    Memo1.Lines.add('Error reading data...')
  else
  begin
    //Memo1.Lines.add('Reading ' + IntToStr(Bytes) + ' characters');
    Data := '';
    for P := 0 to Bytes do
      case Buffer[P] of
        #10: begin
               Memo2.Lines.add('');
               Inc(FCurrentLine);
             end;
        #13:;
        else
          begin
            Data := Data + Buffer[P];
            Memo2.Lines[FCurrentLine] := Memo2.Lines[FCurrentLine] + Buffer[P];
          end;
    end; // For
    DecodeData(Data);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  end;
  SendFlag := True;
 end;
 //---------------------------------------------------------------------------
 procedure TFMain.Comm1TxEmpty(Sender: TObject);
 begin
 //    Memo1.Lines.add('TxEmpty signal detected...');
 end;
 //---------------------------------------------------------------------------
 procedure TFMain.BExitClick(Sender: TObject);
 begin
     Close;
 end;
 //---------------------------------------------------------------------------
 procedure TFMain.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
 begin
    if(Key = 27)then Close;
 end;
 //---------------------------------------------------------------------------
 procedure TFMain.BConfigClick(Sender: TObject);
 begin
    if FConfig.ShowModal = mrOK then
    begin
       UpdateControls;
    end;
 end;
 //---------------------------------------------------------------------------
 procedure TFMain.Timer1Timer(Sender: TObject);
 var Buffer : String;
 begin
    PTime.Caption := TimeToStr(Time);


    if (SendFlag = True)Then
    Begin
       if(MCS < MAXMCS)then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Begin
            //SendFlag := False;
            IF(MCS = 0)THEN
                Buffer := '0'
            ELSE Buffer := IntToStr(MCS);
            Buffer := Buffer;// + #13#10;
            Comm1.Write(Buffer[1], Length(Buffer));
            Memo1.Lines.Add('Write to MCS#'+ Buffer);
            MCS := MCS + 1;
        End
        Else MCS := 0;
    End
end;
//---------------------------------------------------------------------
procedure TFMain.B1Click(Sender: TObject);
begin
    UARTClick(1);
end;
//---------------------------------------------------------------------
procedure TFMain.B2Click(Sender: TObject);
begin
    UARTClick(2);
end;
//---------------------------------------------------------------------
procedure TFMain.B3Click(Sender: TObject);
begin
    UARTClick(3);
end;
//---------------------------------------------------------------------
procedure TFMain.B4Click(Sender: TObject);
begin
    UARTClick(4);
end;
//---------------------------------------------------------------------
procedure TFMain.B5Click(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    UARTClick(5);
end;
//---------------------------------------------------------------------
procedure TFMain.B6Click(Sender: TObject);
begin
    UARTClick(6);
end;
//---------------------------------------------------------------------
procedure TFMain.B7Click(Sender: TObject);
begin
    UARTClick(7);
end;
//---------------------------------------------------------------------
procedure TFMain.B8Click(Sender: TObject);
begin
    UARTClick(8);
end;
//---------------------------------------------------------------------
procedure TFMain.B9Click(Sender: TObject);
begin
    UARTClick(9);
end;
//---------------------------------------------------------------------
procedure TFMain.B10Click(Sender: TObject);
begin
    UARTClick(10);
end;
//---------------------------------------------------------------------
procedure TFMain.B11Click(Sender: TObject);
begin
    UARTClick(11);
end;
//---------------------------------------------------------------------
procedure TFMain.B12Click(Sender: TObject);
begin
    UARTClick(12);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
//---------------------------------------------------------------------------
procedure TFMain.B13Click(Sender: TObject);
begin
    UARTClick(13);
end;
//---------------------------------------------------------------------------
procedure TFMain.B14Click(Sender: TObject);
begin
    UARTClick(14);
end;
//---------------------------------------------------------------------------
procedure TFMain.B15Click(Sender: TObject);
begin
    UARTClick(15);
end;
//---------------------------------------------------------------------------
procedure TFMain.B16Click(Sender: TObject);
begin
    UARTClick(16);
end;
//---------------------------------------------------------------------------
procedure TFMain.B17Click(Sender: TObject);
begin
    UARTClick(17);
end;
//---------------------------------------------------------------------------
procedure TFMain.B19Click(Sender: TObject);
begin
    UARTClick(18);
end;
//---------------------------------------------------------------------------
procedure TFMain.B21Click(Sender: TObject);
begin
    UARTClick(19);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//---------------------------------------------------------------------------
procedure TFMain.B23Click(Sender: TObject);
begin
    UARTClick(20);
end;
//---------------------------------------------------------------------------
procedure TFMain.B25Click(Sender: TObject);
begin
    UARTClick(21);
end;
//---------------------------------------------------------------------------
procedure TFMain.B18Click(Sender: TObject);
begin
    UARTClick(25);
end;
//---------------------------------------------------------------------------
procedure TFMain.B27Click(Sender: TObject);
begin
    UARTClick(22);
end;
//---------------------------------------------------------------------------
procedure TFMain.B29Click(Sender: TObject);
begin
    UARTClick(23);
end;
//---------------------------------------------------------------------------
procedure TFMain.B31Click(Sender: TObject);
begin
    UARTClick(24);
end;
//---------------------------------------------------------------------------
procedure TFMain.B20Click(Sender: TObject);
begin
    UARTClick(26);
end;
//---------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TFMain.B22Click(Sender: TObject);
begin
    UARTClick(27);
end;
//---------------------------------------------------------------------------
procedure TFMain.B24Click(Sender: TObject);
begin
    UARTClick(28);
end;
//---------------------------------------------------------------------------
procedure TFMain.B26Click(Sender: TObject);
begin
    UARTClick(29);
end;
//---------------------------------------------------------------------------
procedure TFMain.B28Click(Sender: TObject);
begin
    UARTClick(30);
end;
//---------------------------------------------------------------------------
procedure TFMain.B30Click(Sender: TObject);
begin
    UARTClick(31);
end;
//---------------------------------------------------------------------------
procedure TFMain.B32Click(Sender: TObject);
begin
    UARTClick(32);
end;
//---------------------------------------------------------------------------
procedure TFMain.Timer2Timer(Sender: TObject);
begin
    if (SendFlag = False)Then
        Exit;
    IF MCSWarning = True Then
    begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Beep;
        IF ImageFlag = 1 Then
        begin
            ImageWarning.Picture.LoadFromFile('Danger1.bmp');
            ImageFlag := 2;
        end
        Else
        begin
            ImageWarning.Picture.LoadFromFile('Danger2.bmp');
            ImageFlag := 1;
        end;
    end
    Else If MCSWarning = False then
        ImageWarning.Picture.LoadFromFile('Normal.bmp');
end;
//---------------------------------------------------------------------
end.
```

==========

File Sensor.pas

```
unit Sensor;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls;

type
  TFSensor = class(TForm)
    Shape1: TShape;
    Shape2: TShape;
    Shape3: TShape;
    Shape4: TShape;
    Shape5: TShape;
    Shape6: TShape;
    Shape7: TShape;
    Shape8: TShape;
    Image1: TImage;
    procedure BCloseClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  private
    { Private declarations }
  public
    { Public declarations }
    procedure ShowParam(Param :integer);
    procedure UpdateControl(Param :integer);
  end;

var
  FSensor: TFSensor;
  SHIPID : Integer;
implementation

uses KConfig, Main;

{$R *.dfm}
//---------------------------------------------------------------------------
procedure TFSensor.UpdateControl(Param :integer);
begin
    IF SHIPID <> Param Then Exit;
    IF FMain.SENSOR[Param,1] = 0 Then
        Shape1.Brush.Color := clRed
    Else Shape1.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,2] = 0 Then
        Shape2.Brush.Color := clRed
    Else Shape2.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,3] = 0 Then
        Shape3.Brush.Color := clRed
    Else Shape3.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,4] = 0 Then
        Shape4.Brush.Color := clRed
    Else Shape4.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,5] = 0 Then
        Shape5.Brush.Color := clRed
    Else Shape5.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,6] = 0 Then
        Shape6.Brush.Color := clRed
    Else Shape6.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,7] = 0 Then
        Shape7.Brush.Color := clRed
    Else Shape7.Brush.Color := clGreen;
    IF FMain.SENSOR[Param,8] = 0 Then
        Shape8.Brush.Color := clRed
    Else Shape8.Brush.Color := clGreen;

end;
//---------------------------------------------------------------------------
procedure TFSensor.ShowParam(Param :integer);
begin
    Caption := 'MCS#' + IntToStr(Param);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SHIPID := Param;
    UpdateControl(Param);
    Show;
end;
//---------------------------------------------------------------------------
procedure TFSensor.BCloseClick(Sender: TObject);
begin

end;
//---------------------------------------------------------------------------
procedure TFSensor.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Action := caFree;
    FSensor := nil;
end;
//---------------------------------------------------------------------------
end.
```

File Config.pas

```
unit Config;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls,IniFiles, Mask;

type
  TFConfig = class(TForm)
    BOK: TBitBtn;
    BClose: TBitBtn;
    LabelBaudrate: TLabel;
    LabelDataBits: TLabel;
    LabelStopbits: TLabel;
    LabelParity: TLabel;
    ComboParity: TComboBox;
    ComboStopbits: TComboBox;
    ComboDatabits: TComboBox;
    ComboBaudrate: TComboBox;
    Bevel1: TBevel;
    CheckBoxRTS: TCheckBox;
    CheckBoxDTR: TCheckBox;
    CheckBoxBREAK: TCheckBox;
    CheckBoxXON: TCheckBox;
    Label1: TLabel;
    ComboPort: TComboBox;
    Label2: TLabel;
    EUART: TMaskEdit;
    procedure BOKClick(Sender: TObject);
    procedure BCloseClick(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
    procedure SetConfig(Mode : Integer);
  public
    { Public declarations }
  end;

var
  FConfig: TFConfig;

implementation

uses KConfig;

{$R *.dfm}
//---------------------------------------------------------------------------
procedure TFConfig.SetConfig(Mode : Integer);
var
    ConfigFile : TIniFile;
begin
    ConfigFile := TIniFile.Create('KConfig.sys');
    if Mode = 1 then // Save to file
    begin
        ConfigFile.WriteInteger('COMM32','Port'    ,ComboPort.ItemIndex);
        ConfigFile.WriteInteger('COMM32','BaudRate',ComboBaudrate.ItemIndex);
        ConfigFile.WriteInteger('COMM32','DataBits',ComboDataBits.ItemIndex);
        ConfigFile.WriteInteger('COMM32','StopBits',ComboStopbits.ItemIndex);
        ConfigFile.WriteInteger('COMM32','Parity'  ,ComboParity.ItemIndex);
        ConfigFile.WriteInteger('COMM32','MAXUART' ,StrToInt(EUART.Text));
        ConfigFile.UpdateFile;
    end
    else if Mode = 0 then // Load from file
    begin
        ComboPort.ItemIndex     := ConfigFile.ReadInteger('COMM32','Port',0);
        ComboBaudrate.ItemIndex :=
ConfigFile.ReadInteger('COMM32','BaudRate',0);
        ComboDataBits.ItemIndex := ConfigFile.ReadInteger('COMM32','DataBits',0);
        ComboStopbits.ItemIndex := ConfigFile.ReadInteger('COMM32','StopBits',0);
        ComboParity.ItemIndex   := ConfigFile.ReadInteger('COMM32','Parity',0);
        EUART.Text := ConfigFile.ReadString('COMM32','MAXUART','100');
    end;
    ConfigFile.Free;
end;
//---------------------------------------------------------------------------
procedure TFConfig.FormShow(Sender: TObject);
begin
    SetConfig(0);
end;
//---------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TFConfig.BOKClick(Sender: TObject);
begin
    SetConfig(1);
    ModalResult := mrOK;
end;
//---------------------------------------------------------------------------
procedure TFConfig.BCloseClick(Sender: TObject);
begin
    ModalResult := mrCancel;
end;
//---------------------------------------------------------------------------


end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้