สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องตรวจวัดอุณหภูมิระยะไกลผ่านระบบวิทยุสื่อสาร

REMOTE TEMPERATURE MEASURING MACHINE VIA RADIO

T119534

นางสาว ธัญญลักษณ์ เขียวแท้

MISS THANYALAK KHEAWTAE

นางสาว พจนา ไกรเสือ

MISS POTJANA KRAISUA

นางสาว วริษฐา รื่นมล

MISS VARITTHA RUENMOL

เลขหมู่.....
เลขทะเบียน 119534
วัน,เดือน,ปี - 8 S.A. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REMOTE TEMPERATURE MEASURING MACHINE VIA RADIO

MISS THANYALAK KHEAWTAE

MISS POTJANA KRAISUA

MISS VARITTHA RUENMOL

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องตรวจวัดอุณหภูมิระยะไกลผ่านระบบวิทยุสื่อสาร	
รายชื่อนักศึกษา	นางสาวธัญญลักษณ์ เขียวแท้	รหัสนักศึกษา 50010675
	นางสาวพจนา ไกรเสือ	รหัสนักศึกษา 50011024
	นางสาววริษฐา รื่นมล	รหัสนักศึกษา 50011399
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2553	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.อรรถสิทธิ์ หล่าสกุล	

ปริญญานิพนธ์ฉบับนี้ ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

.....

(รศ.ดร.อรรถสิทธิ์ หล่าสกุล)

อาจารย์ผู้ควบคุมปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	เครื่องตรวจวัดอุณหภูมิระยะไกลผ่านระบบวิทยุสื่อสาร	
รายชื่อนักศึกษา	นางสาวธัญญลักษณ์ เขียวแท้	รหัสนักศึกษา 50010675
	นางสาวพจนา ไกรเสือ	รหัสนักศึกษา 50011024
	นางสาววริษฐา รื่นมล	รหัสนักศึกษา 50011399
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมสารสนเทศ	
พ.ศ.	2553	
อาจารย์ที่ปรึกษาปริญญานิพนธ์	รศ.ดร.อรรถสิทธิ์ หล่าสกุล	

บทคัดย่อ

โครงงานนี้นำเสนอการออกแบบสร้างเครื่องตรวจสอบวัดค่าอุณหภูมิระยะไกลแบบใหม่ ที่อาศัยการรับส่งข้อมูลแบบอนุกรม (RS-232C) ผ่านระบบ FSK ทำให้สามารถรับส่งด้วยระบบวิทยุได้ ซึ่งจะมีการตั้งเครื่องวัดของตัวลูก (Client) ได้ครอบคลุมพื้นที่มากขึ้น โดยไม่ต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุ โดยหลักการคือ ให้ตัวลูก (Client) ซึ่งมีอยู่หลายจุดนี้ มีการเชื่อมต่อกันคล้ายเครื่อข่าย (Network) โปรแกรมจะควบคุมการติดต่อกันอัตโนมัติ ตั้งแต่เริ่มส่งคำสั่งจากตัวแม่ (Main) ไปสู่ตัวลูก (Client) ที่ต้องการติดต่อกันเป็นทอดๆ โดยใช้ตัวลูก (Client) ที่อยู่ในพื้นที่กำลังส่งของตัวแม่ ทำหน้าที่เป็นตัวกลางในการรับและส่งข้อมูล และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถจะใช้ซอฟต์แวร์ที่ได้ออกแบบเป็นแบบ GUI ไว้บนคอมพิวเตอร์ทั่วไปเพื่อเข้าไปดูข้อมูลได้ทุกจุด ทุกเวลาตามความต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Remote Temperature Measuring Machine Via Radio	
Student	Miss Thanyalak Kheawtae	Student ID. 50010675
	Miss Potjana Kraisua	Student ID. 50011024
	Miss Varittha Ruenmol	Student ID. 50011399
Degree	Bachelor of Engineering	
Program	Information Engineering	
Year	2010	
Thesis Advisor	Assoc. Prof. Dr. Attasit Lasakul	

ABSTRACT

This project presents a new idea of designing the machine for long distance temperature measurement. The machine uses serial data conveyance by Frequency Shift Keying (FSK) which can convey data by radio communication system. The client can communicate to other client , this system can cover more and more area. There is no need to enhance the power propagation of the radio. A designed program will control the connection automatic since sending the command code from the main to desire client that wants to contact some other client located between them (main and desire client). This system will work like a network system. When the user wants to know the temperature from any client , the user can use GUI software installed on general computer (PC) to enter seeing the data every node or client at all times.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ ด้วยความกรุณาของรองศาสตราจารย์ ดร.อรรถสิทธิ์ หล่าสกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งได้คอยให้คำปรึกษา ความรู้ ข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่างตลอดการศึกษาที่ผ่านมาจนกระทั่งสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำรู้สึกซาบซึ้งต่อความกรุณาดังกล่าวเป็นอย่างมาก จึงขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

ขอบคุณ คุณชัยวัฒน์ ที่ได้ให้คำปรึกษา และคำแนะนำในเรื่องของฮาร์ดแวร์ และวงจร รวมถึงอุปกรณ์เครื่องมือเครื่องใช้ต่างๆ

สุดท้ายนี้ กราบขอบพระคุณ คุณพ่อ คุณแม่ บุพการีผู้ให้ทุกสิ่งทุกอย่างกับผู้จัดทำ และความดีอันเกิดจากการศึกษาค้นคว้าครั้งนี้ คณะผู้จัดทำขอมอบแด่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่าน คณะผู้จัดทำมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมา และขอกราบขอบพระคุณมา ณ โอกาสนี้

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 บทนำ	1
1.2 แนวคิดและที่มาของโครงงาน	1
1.3 วัตถุประสงค์ของโครงงาน	2
1.4 ขอบเขตของโครงงาน	2
1.5 ขั้นตอนการดำเนินงาน	3
1.6 ผลที่คาดว่าจะได้รับจากโครงงาน	4
บทที่ 2 ทฤษฎีพื้นฐาน	5
2.1 ทฤษฎีที่เกี่ยวข้องกับไมโครคอนโทรเลอร์ (dsPIC30F4011)	5
2.1.1 โครงสร้างบอร์ด dsPIC30F4011	5
2.1.2 คุณสมบัติของบอร์ด dsPIC30F4011	7
2.2 รีเลย์ (Relay)	9
2.2.1 ชนิดของวงจรรีเลย์ (Relay)	9
2.2.2 โครงสร้างภายในของวงจรรีเลย์ (Relay)	10
2.3 ไอซี DS1820	12
2.3.1 คุณสมบัติของไอซี DS1820	13
2.4 ไอซี MAX 232	14
2.4.1 การทำงานของไอซีปรับระดับแรงดัน MAX 232	15
2.5 การมอดูเลตเชิงความถี่แบบดิจิตอล FSK (Frequency Shift Keying)	16
2.5.1 ประเภทการใช้งานของเอฟเอสเค (FSK)	17
2.5.2 ข้อดีของระบบมอดูเลตแบบเอฟเอสเค (FSK)	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

2.5.3 ข้อเสียของระบบมอดูเลตแบบเอฟเอสเค (FSK) 17

บทที่ 3 การออกแบบปริญญานิพนธ์ 18

3.1 การออกแบบในส่วนของฮาร์ดแวร์ (Hardware) 18

3.1.1 เครื่อง Main 18

3.1.2 เครื่อง Client 20

3.2 การออกแบบในส่วนของ Software 21

3.2.1 รูปแบบการส่ง 22

3.2.2 การเซทค่าก่อนใช้งาน 29

3.2.3 ซอร์ฟแวร์บนตัว PC 29

บทที่ 4 การทดลองและการใช้งาน 31

4.1 อุปกรณ์ที่ใช้ 31

4.2 ขั้นตอนการทดลอง 33

4.2.1 แสดงการทดสอบวงจรเอฟเอสเค (FSK) 34

4.2.2 แสดงการทดสอบซอฟต์แวร์โดยติดต่อของ Client ตัวที่หนึ่ง และ Client ตัวที่สอง 35

บทที่ 5 บทสรุปและวิจารณ์ผลการทดลอง 41

5.1 สรุปผลการทดลอง 41

5.2 ปัญหาที่พบในระหว่างการดำเนินโครงงาน 41

5.3 แนวทางการแก้ไขปัญหาและพัฒนาโครงงาน 41

บรรณานุกรม 42

ภาคผนวก 43

ภาคผนวก ก.

ภาคผนวก ข.

ภาคผนวก ค.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
1.1	ขั้นตอนการดำเนินงาน	3
2.1	บอร์ดไมโครคอนโทรลเลอร์ dsPIC30F4011	5
2.2	แสดงการจัดขาสัญญาณของ dsPIC30F4011	7
2.3	รีเลย์ควบคุม (Control Relay)	9
2.4	โครงสร้างภายในของรีเลย์ (Relay)	10
2.5	โครงสร้างหน้าสัมผัสภายในของรีเลย์ (Relay)	10
2.6	ไอซี DS1820	12
2.7	การต่อใช้งาน DS1820 กับ ไมโครคอนโทรลเลอร์	13
2.8	แสดงตำแหน่งขาของไอซี MAX232	14
2.9	ตำแหน่งขาสัญญาณและวงจรภายในไอซีปรับระดับแรงดัน MAX 232	14
2.10	การต่อใช้งาน MAX232 กับ ไมโครคอนโทรลเลอร์	15
2.11	การผสมสัญญาณของข้อมูล 01101 ด้วยวิธีการ Frequency Shift Keying ในกรณีนี้ บิทเรท (Bit Rate) เท่ากับ บอดเรท (Baud Rate)	16
2.12	Spectrum ของสัญญาณ FSK ที่มีความถี่ของสัญญาณพาหะ fc0 และ fc1 และอัตราเร็วของข้อมูล Baud (Nbaud)	16
3.1	แสดงโครงสร้างของเครื่อง Main	18
3.2	แสดงวงจรเอฟเอสเค มอดูเลชั่น (FSK MODULATOR)	19
3.3	แสดงรูปโครงสร้างของเครื่อง Client	20
3.4	แสดงไดอะแกรม ซอฟแวร์ ควบคุมการรับส่งข้อมูล ของตัว Client	21
3.5	แสดงไดอะแกรมของส่วน PC	22
3.6	แสดงรูปแบบของ Command format	23
3.7	แสดงเส้นทางการวิ่งไปอ่านข้อมูลของ Client ตัวหมายเลขสี่	25
4.1	โปรแกรม GUI	31
4.2	เครื่อง Main	31
4.3	เครื่อง Client ที่หนึ่ง	32
4.4	เครื่อง Client ที่สอง	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

4.5 วงจรเอฟเอสเค (FSK) 32

4.6 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ PC ที่ USER ใช้ 34

4.7 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ วงจรเอฟเอสเค (FSK) แล้วทำการต่อเข้ากับลำโพง 34

4.8 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ วงจรเอฟเอสเค (FSK) 35

4.9 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ Client ทั้งสองตัว 35

4.10โปรแกรม GUI 36

4.11 เลือก Client ตัวที่หนึ่ง 36

4.12 เลือกให้ Client ตัวที่หนึ่งแสดงการอ่านอุณหภูมิอย่างเดียว 37

4.13 เลือกให้ Client ตัวที่หนึ่งแสดงการอ่านอุณหภูมิควบคุมวงจร Relay ทั้งสองตัว 37

4.14 แสดงการควบคุมวงจร Relay ทั้งสองตัว ของ Client ตัวที่หนึ่ง 38

4.15 เลือก Client ตัวที่สอง 38

4.16 เลือกให้ Client ตัวที่สองแสดงการอ่านอุณหภูมิอย่างเดียว 39

4.17 เลือกให้ Client ตัวที่สองแสดงการอ่านอุณหภูมิควบคุมวงจร Relay ทั้งสองตัว 39

4.18 แสดงการควบคุมวงจร Relay ทั้งสองตัว ของ Client ตัวที่สอง 40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การอ่านค่าอุณหภูมิของไอซี DS1820	13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

บทนี้เป็นการกล่าวถึงแนวคิด ที่มา ของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน ขั้นตอนการดำเนินงาน และผลที่คาดว่าจะได้รับจากการทำโครงงานนี้

1.1 บทนำ

ในปัจจุบันการตรวจสอบการบันทึกค่าการเปลี่ยนแปลงต่างๆ ติดต่อกันเป็นช่วงเวลานานๆ เพื่อคาดการณ์เหตุการณ์ วิเคราะห์ข้อมูลของเหตุการณ์ที่อาจจะเกิดขึ้นได้ในอนาคต เป็นสิ่งจำเป็นต่อการพิจารณาเพื่อหาแนวทางแก้ไข เพื่อบรรเทาความร้ายแรงของเหตุการณ์ต่างๆ ที่อาจจะเกิดขึ้นในอนาคต ตัวอย่างเช่น การติดตาม ค่าอุณหภูมิ ค่าระดับน้ำ ค่าความชื้น เมื่อทำการเก็บค่าสถิติเหล่านี้เป็นเวลาต่อเนื่องนั้น จะสามารถคาดการณ์เหตุการณ์ต่างๆ ได้ โดยใช้หลักสถิติความน่าจะเป็น เช่น ปัญหาอุณหภูมิโลกที่สูงขึ้นทุกปี เป็นต้น

1.2 แนวคิดและที่มาของโครงงาน

ปัจจุบันการตรวจสอบบันทึกค่าการเปลี่ยนแปลงต่างๆ ของธรรมชาติ ติดต่อกันในช่วงเวลานานๆ เพื่อการคาดการณ์หรือวิเคราะห์เหตุการณ์อันอาจจะเกิดขึ้นได้ในอนาคต เป็นสิ่งจำเป็นมากต่อการพิจารณาเพื่อหาแนวทางแก้ไขหรือบรรเทาปัญหาต่างๆที่อาจเกิดขึ้นได้ ตัวอย่าง เช่น การติดตาม ค่าอุณหภูมิ ซึ่งหากได้มีการตรวจสอบอย่างต่อเนื่องไปเป็นระยะนานๆ ก็สามารถนำค่าเหล่านั้น มาหาสถิติความน่าจะเป็นต่างๆ เพื่อจะได้วิเคราะห์หาวิธีการแก้ไขปัญหาเหล่านั้นได้ตรงจุดที่สุดต่อไป ดังเช่น ปัญหาน้ำท่วมซ้ำซากเป็นต้น แต่การวัดค่าเหล่านี้จำเป็นต้องมีการนำเอาเซนเซอร์ไปติดตั้งไว้ในสถานที่ต่างๆที่เราต้องการจะทราบข้อมูลเป็นจำนวนมากและส่วนมากก็จะเป็นสถานที่ๆ ห่างไกลจากความเจริญ เช่น แนวภูเขา ในป่ารกทึบ ซึ่งจะไม่มีแม้สัญญาณโทรศัพท์ ในพื้นที่นั้นๆ

ดังนั้น หากต้องการทราบว่าค่าที่ต้องการวัดนั้นมีค่าเท่าใด ก็จำเป็นต้องออกเดินป่า เข้าไปจดค่าเหล่านั้น ซึ่งจะศูนย์เสียทั้งเวลาและงบประมาณ, กำลังบุคคล ในการบันทึกค่าเหล่านั้นในแต่ละครั้ง ยิ่งไปกว่านั้น หากต้องมีการเก็บค่าเหล่านั้นเป็นระยะบ่อยครั้งเท่าใดก็จะยิ่งทำให้ เสียค่าใช้จ่ายมากขึ้นตามไปด้วยเท่านั้น ซึ่งได้มีงานวิจัยเกี่ยวกับ การนำเสนอการออกแบบและสร้างระบบ (เครื่อง)ตรวจสอบวัดค่าข้อมูลของอุณหภูมิระยะไกลแบบใหม่ออกมาแล้วเพื่อแก้ปัญหาในระดับหนึ่งแล้ว ซึ่งโครงงานงานวิจัยชิ้นนี้เหมือนกับเป็นการต่อยอดให้ระบบมีประสิทธิภาพใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานมากยิ่งขึ้นกว่าเดิม โดยโครงสร้างการทำงานของระบบยังคงอ้างอิงจากระบบเดิมคือค่าต่างๆที่ต้องการวัดนั้นจะถูกวัดเก็บไว้โดยการทำงานของระบบเซนเซอร์และไมโครคอนโทรลเลอร์ ที่บรรจุโปรแกรมจัดการการทำงานเพื่อตรวจวัดค่าต่างๆอยู่ในเครื่องดังกล่าวแล้ว โดยผู้ใช้งานสามารถจะตรวจดูค่าผลของการวัดเหล่านั้นที่เป็นแบบเป็นแบบ GUIที่ทำงานบนคอมพิวเตอร์ทั่วไป ดังนั้นจึงง่ายต่อการใช้งานและสามารถเข้าไปดูข้อมูลได้ทุกจุดทุกเวลาได้ตามต้องการ และโดยการรับส่งข้อมูลของแต่ละจุดจะใช้ระบบสื่อสารที่เป็นวิทยุที่ต่อเป็นระบบอัตโนมัติที่มีการติดตั้งให้ครอบคลุมกว่าเดิม โดยการเพิ่มจุดที่ใช้ระบบสื่อสารที่เป็นวิทยุในพื้นที่ที่ห่างออกไป และยังเป็นการสำรองไว้หากวิทยุตัวใดตัวหนึ่งเกิดข้อขัดข้องขึ้น ผู้ใช้ระบบก็จะสามารถตรวจสอบค่าที่ต้องการได้เหมือนเดิม จากวิทยุที่ได้ทำการสำรองไว้ ซึ่งในการใช้ระบบสื่อสารที่เป็นวิทยุแทนระบบ GSM หรือระบบอินเตอร์เน็ตนั้นเป็นการแก้ปัญหาเนื่องจากอาจไม่มีสัญญาณในพื้นที่ดังกล่าว ดังนั้นระบบที่นำเสนอนี้จึงอำนวยความสะดวกคือไม่จำเป็นที่จะต้องส่งบุคคลเข้าไปจดบันทึกข้อมูลโดยตรงและในเรื่องของความสามารถในการวัดจะมีความครอบคลุมในพื้นที่มากยิ่งขึ้นกว่าเดิม

1.3 วัตถุประสงค์ของโครงงาน

ในการพัฒนาระบบเครื่องวัดสภาพสิ่งแวดล้อมระยะไกลผ่านระบบวิทยุสื่อสารนั้นมีจุดประสงค์ดังนี้

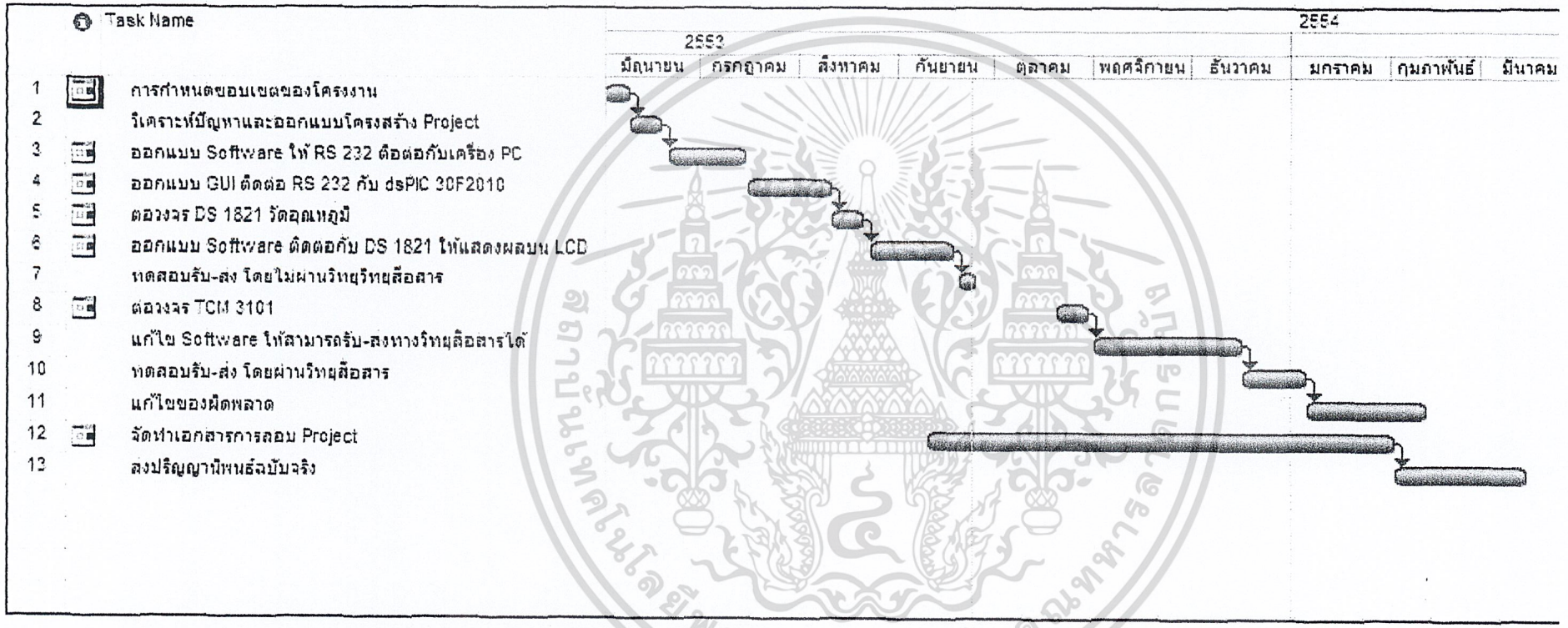
1. สร้างเครื่องวัดอุณหภูมิระยะไกล ให้สามารถตรวจสอบวัดค่าได้ระยะทางครอบคลุมมากขึ้น ทำให้ในการเก็บค่าสถิติต่างๆอาจจะมีประสิทธิภาพมากยิ่งขึ้น
2. สร้างเครื่องวัดอุณหภูมิระยะไกลที่ให้ผู้ใช้ GUI นั้น มีความสะดวกในการใช้งานระบบมากยิ่งขึ้น

1.4 ขอบเขตของโครงงาน

1. สร้างเครื่องตรวจวัดอุณหภูมิระยะไกลโดยผ่านวิทยุสื่อสาร
2. สร้างการรับส่งข้อมูลจาก เครื่อง Main ไปถึง เครื่อง Client จากวงจรเอฟเอสเค (FSK)
3. สร้างโปรแกรมควบคุม dsPIC30F4011 ด้วย โปรแกรม MPLAB
4. สร้าง User Interface ซึ่งเป็นลักษณะของ GUI ด้วย โปรแกรม Visual C#2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ขั้นตอนการดำเนินงาน



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.6 ผลที่คาดว่าจะได้รับจากโครงงาน

1. ในการเก็บข้อมูลสถิติต่างๆ สามารถวัดได้ตลอดเวลาและสามารถติดตั้งได้ทุกสถานที่

2. ระบบที่ได้พัฒนานั้นสามารถลดค่าใช้จ่ายในการเก็บค่าสถิติต่างๆ ในพื้นที่ห่างไกล

3. ความเร็วรวดในการทราบข้อมูลต่างๆ ที่ผ่านซอฟท์แวร์แบบ GUI ที่ทำงานบนคอมพิวเตอร์ทั่วไป ทำให้สามารถคำนวณเหตุการณ์ที่เกิดขึ้นในอนาคตอันใกล้อย่างรวดเร็ว และแม่นยำ

4. สามารถนำไปใช้งานในองก์กร ต่างๆ เช่น ฟาร์มขนาดใหญ่ต่างๆ กรมชลประทาน และกรมอุตินิยมวิทยา หรือหน่วยงานที่มีการเฝ้าระวังภัยธรรมชาติ เหล่านี้เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

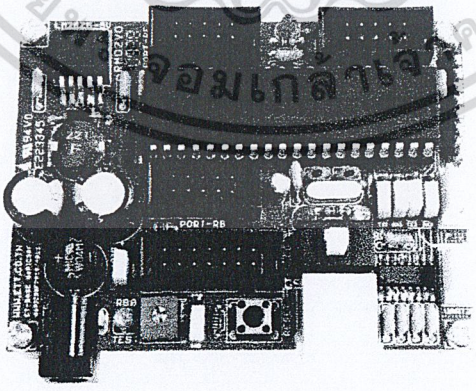
บทที่ 2
ทฤษฎีพื้นฐานที่ใช้

ในหัวข้อนี้จะกล่าวถึงทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการทำโครงงานเครื่องตรวจวัดอุณหภูมิระยะไกลผ่านระบบวิทยุสื่อสาร

2.1 ทฤษฎีที่เกี่ยวข้องกับไมโครคอนโทรเลอร์ (dsPIC30F4011)

dsPIC30F4011 เป็นไมโครคอนโทรเลอร์ในตระกูล dsPIC30F ขนาด 40 Pin ของ Microchips เป็น MCU ประจำบอร์ด โดย dsPIC30F4011 เป็น MCU ซึ่งใช้การประมวลผลข้อมูลแบบ 16 บิต ซึ่งมีจุดเด่นในด้านของความสามารถในการประมวลผลข้อมูลสัญญาณแบบดิจิตอลเหมาะอย่างยิ่งสำหรับนำไปประยุกต์ใช้ในงานควบคุมต่างๆ โดยโครงสร้างภายในจะเป็นการผสมผสานระหว่างไมโครคอนโทรลเลอร์ (MCU) และวงจร DSP (Digital Signal Processing) รวมเข้าไว้ด้วยกัน หรืออาจเรียก MCU ตระกูล dsPIC30F ว่าเป็น DSC หรือ Digital Signal Controller ก็ได้ โดยโครงสร้างของบอร์ดไมโครคอนโทรเลอร์ dsPIC30F4011 ได้รับการออกแบบให้บอร์ดมีขนาดเล็ก เหมาะต่อการนำไปประยุกต์ใช้งานเป็นหลัก โดยภายในบอร์ดได้บรรจุเอาวงจรที่จำเป็นต่อการใช้งาน และสะดวกต่อการพัฒนาโปรแกรม มีความยืดหยุ่นสามารถปรับเปลี่ยนสัญญาณ I/O เพื่อนำไปประยุกต์ใช้งานในลักษณะต่างๆ ให้สอดคล้องและเหมาะสมกับความต้องการใช้งานได้ในหลายๆ ลักษณะตามต้องการ

2.1.1 โครงสร้างบอร์ด dsPIC30F4011



รูปที่ 2.1 บอร์ดไมโครคอนโทรลเลอร์ dsPIC30F4011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟ 7-20VAC/DC

- IC Regulate แบบ Switching ขนาด 5V/1A

- LED TEST สำหรับทดสอบการทำงานของบอร์ด โดยควบคุมจาก RB0

- Jumper สำหรับ ตัด ต่อ สัญญาณ RB0 กับ LED TEST

- VR ปรับค่า สำหรับใช้ปรับความสว่างของหน้าจอแสดงผล LCD

- ขั้วต่อ 14PIN IDE สำหรับเชื่อมต่อกับ LCD แบบ Character

- Jumper สำหรับเลือกรูปแบบการควบคุมขา R/W ของ LCD

- สวิตช์ Reset สำหรับ Reset การทำงานของ MCU เมื่ออยู่ในโหมด Run

- ขั้วต่อ ICD2 สำหรับใช้เชื่อมต่อกับเครื่องโปรแกรมและดีบักตามมาตรฐาน ICD2

- สวิตช์ สำหรับเลือกโหมดการทำงานระหว่าง Run(RUN) และ Program(PGM)

- LED สีแดง แสดงสถานะ PGM เมื่อบอร์ดทำงานใน Program Mode

- LED สีเขียว แสดงสถานะ RUN เมื่อบอร์ดทำงานใน Run Mode

- ขั้วต่อ UART2 โดยเป็นสัญญาณแบบ RS232 โดยใช้ Pin ของ RF4(RX2) และ RF5(TX2) เป็นสัญญาณเชื่อมต่อ

- ขั้วต่อ UART1 โดยเป็นสัญญาณแบบ RS232 ซึ่งใช้ Pin ของ RC13(TX1),RC14(RX1) เป็นสัญญาณเชื่อมต่อ

- Jumper สำหรับเลือกการเชื่อมต่อสัญญาณ RC13,RC14,RF4,RF5 ว่าจะใช้ขาสัญญาณดังกล่าวทำหน้าที่เป็นขาสัญญาณรับส่งของ RS232 หรือ GPIO สำหรับใช้งานทั่วไป

- ขั้วต่อสัญญาณ RC13,RC14,RD0,RD1,RD2 และ RD3 สำหรับใช้งาน

- ขั้วต่อสัญญาณ Port-RF ซึ่งถ้าเป็น MCU รุ่น 40 Pin จะมี 7 บิต คือ RF[0..6]

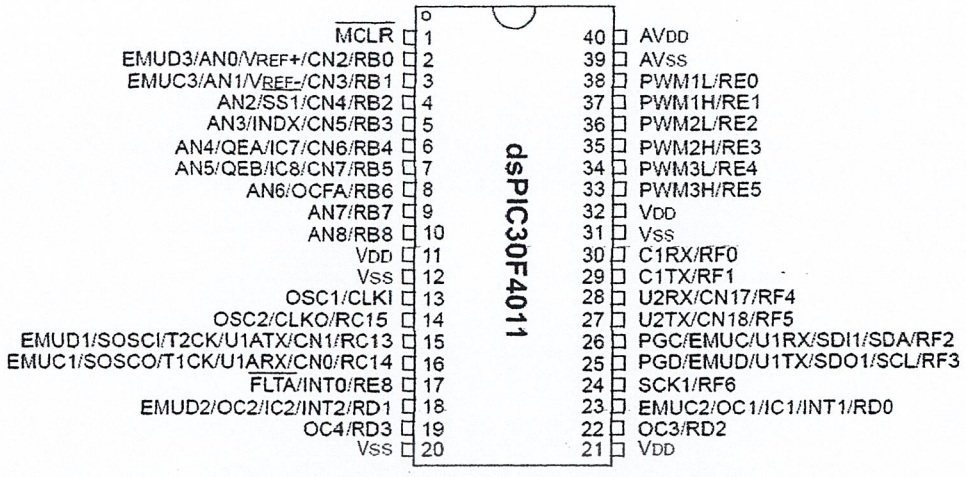
- LED สำหรับแสดงสถานะ ของแหล่งจ่ายไฟ +5V ของบอร์ด

- ขั้วต่อสัญญาณ Port-RE ซึ่งจะมี 7 บิต คือ RE[0..6 และ 8]

- ขั้วต่อสัญญาณ Port-RB ซึ่งถ้าเป็น MCU40Pin จะมี 8 บิต คือ RB[0..7]

- MCU ประจำบอร์ด เป็น รุ่น 40Pin ใช้เบอร์ dsPIC30F4011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการจัดขาสัญญาณของ dsPIC30F4011

2.1.2 คุณสมบัติของบอร์ด dsPIC30F4011

☐ ใช้ MCU ตระกูล dsPIC30F4011 ของ Microchips เป็น MCU ประจำบอร์ด โดยคุณสมบัติเด่นๆของ MCU ได้แก่

- o มีหน่วยความจำ Flash 48 KByte
- o มีหน่วยความจำ RAM ขนาด 2 KByte
- o มีหน่วยความจำ EEPROM ขนาด 1 KByte สำหรับเก็บข้อมูลใช้งาน
- o มีพอร์ต I/O ขนาด 29 Bit
- o มี 16Bit Timer/Counter จำนวน 5 ชุด
- o มี Input Capture จำนวน 4 ช่อง
- o มี Output Compare จำนวน 4 ช่อง
- o มี ADC 10Bit/500Ksps จำนวน 9 ช่อง
- o มี PWM Motor Control จำนวน 6 ช่อง พร้อม Quadrature Encode Interface(QEI)
- o มี UART จำนวน 2 ช่อง
- o มี SPI จำนวน 1 ช่อง และ มี I2C จำนวน 1 ช่อง
- o มีวงจร Watchdog, Power-ON Reset, PWM

☐ ใช้ Crystal ความถี่ 7.3728 MHz สามารถใช้ PLL คูณความถี่เพื่อ Run ความถี่ 29.4912 MHz ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

☐ มีพอร์ตสื่อสารอนุกรม UART แบบ RS232 จำนวน 2 ช่อง พร้อม Jumper สำหรับเลือกใช้งาน UART หรือ GPIO ได้ตามต้องการ โดยใช้ขั้วต่อ UART แบบ CPA-4 Pin มาตรฐาน อีทีที

☐ มีขั้ว ICSP มาตรฐาน ICD2 แบบ RJ11 สำหรับใช้ร่วมกับชุดพัฒนาโปรแกรมและ Debugger ที่

รองรับการทำงานตามมาตรฐาน ICD2 ของ Microchips เช่น ICD2 หรือ Pickit2 ได้

☐ มี Switch สำหรับสลับสัญญาณระหว่าง Program/Debug(PGM) และ ใช้งานปรกติ(RUN) พร้อม

LED แสดงโหมดการทำงานของบอร์ด

☐ มีขั้วต่อสัญญาณ I/O แบบ Header ขนาด 2x5 จำนวน 3 ชุด และ Header 1x8 Pin อีก 1 ชุด

☐ Header 14Pin สำหรับ Character LCD พร้อม VR ปรับความสว่าง

☐ มี Switch Reset สำหรับสั่ง Reset การทำงานของ MCU ภายในบอร์ด

☐ มี LED สำหรับทดสอบการทำงาน โดยใช้ RB0 ในการควบคุม พร้อม Jumper ตัดต่อสัญญาณ

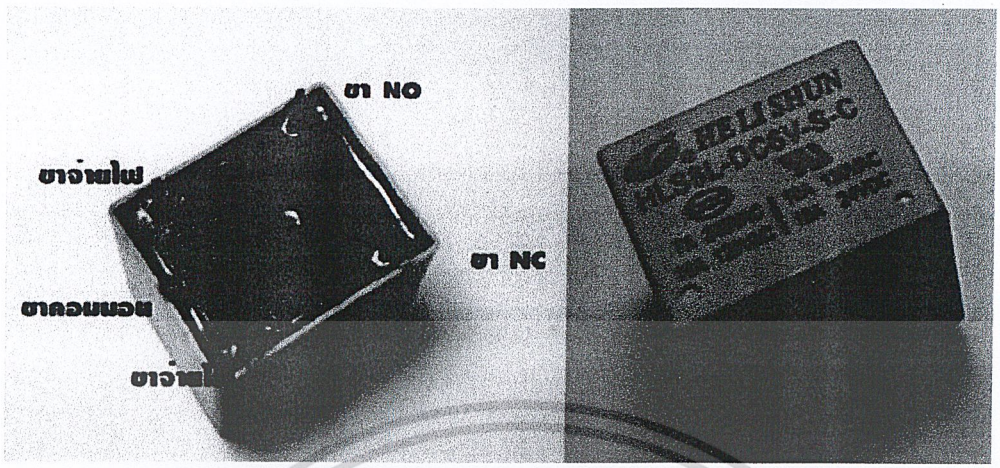
☐ Power AC/DC Input พร้อม Regulate แบบ Switching เบอร์ LM2575 ขนาด 5V/1A ลดปัญหา

ความร้อนจากวงจร Regulate และ LED แสดงสถานะแหล่งจ่าย Power

☐ ขนาด PCB Size เล็กเพียง 8 x 6 cm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 รีเลย์ (Relay)



รูปที่ 2.3 รีเลย์ (Relay) ควบคุม

รีเลย์ (Relay) เป็นอุปกรณ์แม่เหล็ก Magnetics device ที่ใช้กันมานาน ปัจจุบันก็ยังคงนิยมใช้กันอยู่ แต่ในปัจจุบันนี้ รีเลย์ (Relay) ถูกพัฒนาให้มีคุณภาพดีกว่าสมัยก่อนมาก แต่ยังคงหลักการ และโครงสร้างเดิม เอาไว้

2.2.1 ชนิดของวงจรรีเลย์ (Relay)

รีเลย์ (Relay) เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์ เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่า คอนแทกเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

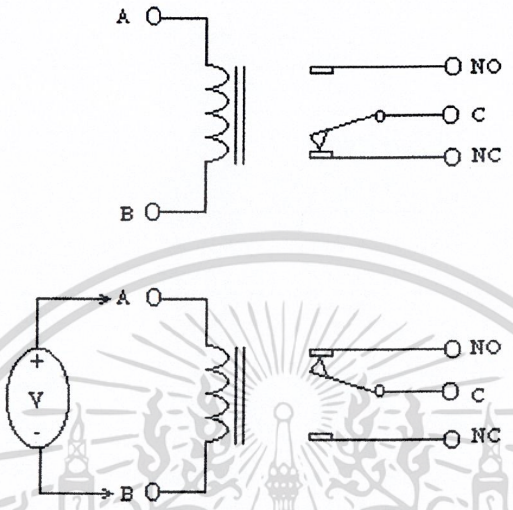
2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้า ต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อ การควบคุมรีเลย์ (Relay) หรือคอนแทกเตอร์(Contactor) ขนาดใหญ่ รีเลย์ควบคุม (Control Relay) บางทีเรียกกันง่าย ๆ ว่า "รีเลย์ (Relay)"

หน้าที่ของคอนแทกเตอร์ (Contactor) คือ การใช้กำลังไฟฟ้าจำนวนน้อยเพื่อไปควบคุมการตัดต่อกำลังไฟฟ้าจำนวนมาก คอนแทกเตอร์ (Contactor) ทำให้เราสามารถควบคุมกำลังไฟฟ้าในตำแหน่งอื่นๆ ของระบบไฟฟ้าได้ สายไฟควบคุมให้รีเลย์กำลังหรือคอนแทกเตอร์ (Contactor) ทำงานเป็นสายไฟฟ้าขนาดเล็กต่อเข้ากับสวิตช์ควบคุมและคอล์ยของคอนแทกเตอร์ (Contactor) กำลังไฟฟ้าที่ป้อนเข้าคอล์ยอาจจะเป็นไฟฟ้ากระแสตรง หรือไฟฟ้ากระแสสลับก็ได้ขึ้นอยู่กับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบ การใช้คอนแทกเตอร์ (Contactor) ทำให้สามารถควบคุมวงจรจากระยะไกล (Remote) ได้ ซึ่งทำให้เกิดความปลอดภัยกับผู้ปฏิบัติงานในการควบคุมกำลังไฟฟ้า

2.2.2 โครงสร้างภายในของรีเลย์ (Relay)



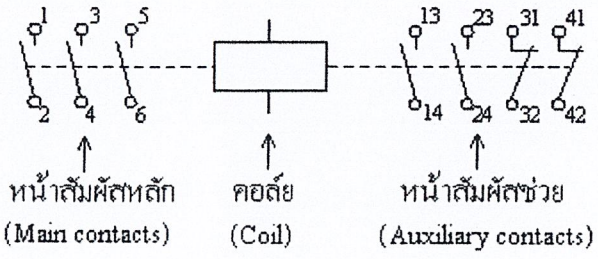
รูปที่ 2.4 โครงสร้างภายในของรีเลย์ (Relay)

ภายในโครงสร้างของ รีเลย์ จะประกอบไปด้วยขดลวด (Coil) 1 ชุด และ หน้าสัมผัส (Contactor) ซึ่งในหน้าสัมผัส 1 ชุด จะประกอบไปด้วย

1. หน้าสัมผัสแบบปกติปิด (Normally Close หรือ NC.) ซึ่งในสภาวะปกติ ขานี้จะต่ออยู่กับขาร่วม (Common)

2. หน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO.) ขานี้จะต่อเข้ากับขาร่วม (Common) เมื่อขดลวดมีแรงดันตกคร่อม หรือกระแสไหลผ่าน ในปริมาณที่เพียงพอ

ใน รีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด เช่น 2 ชุด, 4 ชุด แล้วแต่ผู้ผลิต



รูปที่ 2.5 โครงสร้างหน้าสัมผัสภายในของรีเลย์ (Relay)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนแทกเตอร์ (Contactor) นอกจากจะมีหน้าสัมผัสทั้งส่วนเคลื่อนที่ และหน้าสัมผัสส่วนที่อยู่กับที่แล้วหน้าสัมผัสภายในของคอนแทกเตอร์ (Contactor) ยังแบ่งออกเป็น 2 ส่วนตามลักษณะของการทำงาน ซึ่งแบ่งออกเป็น 2 ส่วนดังนี้ คือ

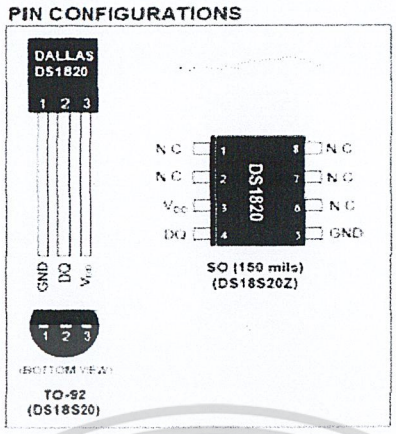
1. หน้าสัมผัสหลัก (Main Contacts) โดยปกติแล้วหน้าสัมผัสหลักมี 3 อัน สำหรับส่งผ่านกำลังไฟฟ้า 3 เฟสเข้าไปสู่มอเตอร์ หรือโหลดที่ใช้แรงดันไฟฟ้า 3 เฟส หน้าสัมผัสหลักของคอนแทกเตอร์ (Contactor) มีขนาดใหญ่ทนแรงดันและกระแสได้สูง หน้าสัมผัสหลักเป็นชนิดปกติเปิด (Normally open;N.O. contact)

อักษรกำกับ หน้าสัมผัสด้านแหล่งจ่ายคือ 1, 3, 5 หรือ L1, L2, L3 และด้านโหลดคือ 2, 4, 6 หรือ T1, T2, T3 ดังรูป

2. หน้าสัมผัสช่วย (Auxiliary Contacts) หน้าสัมผัสชนิดนี้ติดตั้งอยู่ด้านข้างทั้งสองด้านของตัวคอนแทกเตอร์ (Contactor) มีขนาดเล็กทนกระแสได้ต่ำทำหน้าที่ช่วยการทำงานของวงจร เช่น เป็นหน้าสัมผัสที่ทำให้คอนแทกเตอร์ (Contactor) ทำงานได้ตลอดเวลา หรือเรียกว่า "Holding" หรือ "Maintaining Contact" หน้าสัมผัสช่วยนี้จะเป็นหน้าสัมผัสแบบโยกได้สองทาง โดยจะถูกดึงขึ้น-ลงไปตามจังหวะการดูด-ปล่อยของคอนแทกเตอร์ (Contactor) อักษรกำกับหน้าสัมผัสช่วย จะเป็น 13, 14 สำหรับคอนแทกเตอร์ (Contactor) ที่มีหน้าสัมผัสช่วยแบบปกติเปิด 1 ชุด ถ้ามี N.O. ชุดที่ 2จะเป็น 23, 24 และหน้าสัมผัสช่วยแบบปกติปิดจะมีอักษรกำกับเป็น 31, 32 และ 41, 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ไอซี DS1820



PIN	SYMBOL	Description
1	GND	Ground
2	DQ	Data Input/ Output pin
3	Vdd	Optional Vdd pin

รูปที่ 2.6 ไอซี DS1820

เนื่องจากการการออกแบบระบบการวัดอุณหภูมิระยะไกลผ่านวิทยุสื่อสารนั้น เราได้ใช้ไมโครคอนโทรลเลอร์ dsPIC30f4011 เป็นตัวสำหรับเขียนโปรแกรมติดต่อกับไอซี DS1820 ซึ่งไอซี DS1820 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาด และใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกามาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลจะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลาซึ่งเรียกว่า ไทม์สล็อต (Time Slot) โดยคาบเวลาต่ำสุดและสูงสุดของสถานะต่าง ๆ ในการสื่อสารข้อมูลในแต่ละไทม์สล็อต (Time Slot) มีการกำหนดขอบเขตไว้อย่างชัดเจนการถ่ายทอดข้อมูลจะเกิดขึ้นในแต่ละไทม์สล็อต (Time Slot) นั้น รูปแบบการถ่ายทอด ข้อมูลจะเป็นแบบอะซิงโครนัส (Asynchronus) ในระดับบิต ไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน

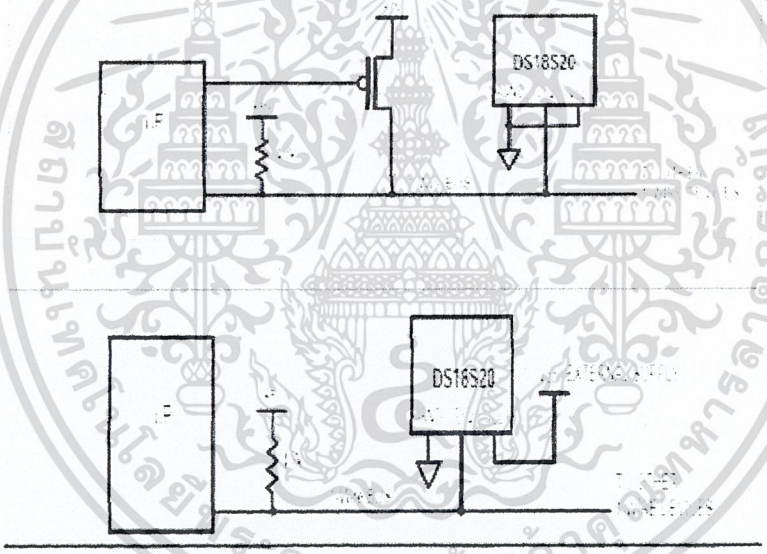
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การอ่านค่าอุณหภูมิของไอซี DS1820

TEMPERATURE/DATA RELATIONSHIPS Table 1

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125 C	00000000 11111010	00FA
+25 C	00000000 00110010	0032h
$+\frac{1}{2}$ C	00000000 00000001	0001h
+0 C	00000000 00000000	0000h
$-\frac{1}{2}$ C	11111111 11111111	FFFFh
-25 C	11111111 11001110	FFCEh
-55 C	11111111 10010010	FF92h

ข้อมูลอุณหภูมิที่อ่านจาก DS1820 จะมีขนาด 2 bytes แสดงผลได้ดังตาราง ก็คือ ถ้า byte แรกเป็น 0 แสดงว่าอุณหภูมิเป็น + ถ้าเป็น FFh ก็หมายความว่าอุณหภูมิ เป็นลบ



รูปที่ 2.7 การต่อใช้งาน DS1820 กับ ไมโครคอนโทรลเลอร์

2.3.1 คุณสมบัติของไอซี DS1820

1) DS1820 สามารถ Interface โดยใช้สายสัญญาณเพียงเส้นเดียว

2) DS1820 เพียงตัวเดียว สามารถวัดอุณหภูมิได้โดยไม่ต้องต่ออุปกรณ์ร่วม

3) DS1820 มีย่านวัดอยู่ที่ +125 ถึง -55 C

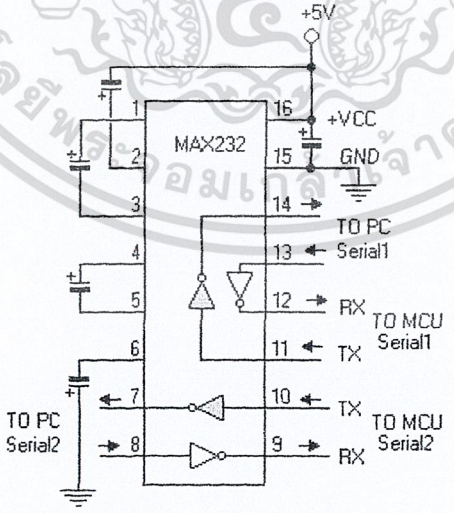
4) DS1820 มีความละเอียดในการวัดได้ 0.5 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไอซี MAX 232

ไอซีปรับระดับแรงดัน MAX 232 เป็นไอซี 16 ขาที่ใช้ทำหน้าที่ในการรับส่งข้อมูลแบบอนุกรม (Serial) ระหว่างอุปกรณ์ TTL/CMOS กับพอร์ท RS-232 โดยไอซีปรับระดับแรงดัน MAX 232 มีช่องทางในการรับข้อมูลแบบอนุกรม (Serial) 2 ช่องทางและมีช่องทางในการส่งข้อมูลแบบอนุกรม (serial) 2 ช่องทาง โครงสร้างภายในของไอซีปรับระดับแรงดัน MAX 232 ยังมีวงจรทวีแรงดัน (Voltage doubler) และวงจรอินเวอร์สแรงดัน (Voltage inverter) วงจรทวีแรงดันเป็นวงจรที่ทำหน้าที่ยกระดับแรงดันจาก 5 โวลท์เป็น 10 โวลท์ และวงจรอินเวอร์ส (Invert)แรงดันเป็นวงจรที่ทำหน้าที่กลับสัญญาณจาก +10 โวลท์เป็น –10 โวลท์ วงจรทั้งสองทำหน้าที่หลักก็คือขยายสัญญาณก่อนที่จะส่งออกไปยังพอร์ท RS 232

รูปที่ 2.8 แสดงตำแหน่งขาของไอซี MAX232



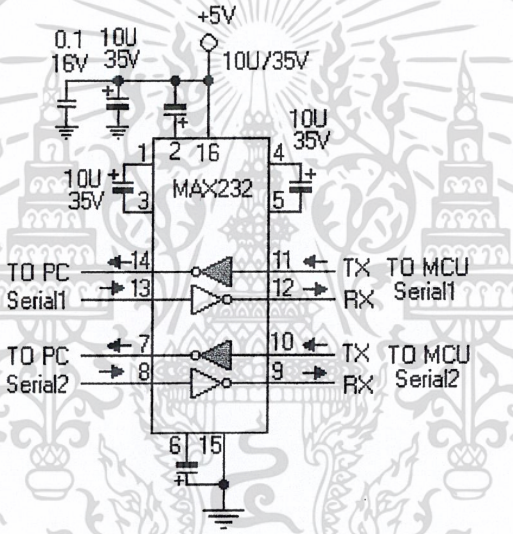
รูปที่ 2.9 ตำแหน่งขาสัญญาณและวงจรภายในไอซีปรับระดับแรงดัน MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การทำงานของไอซีปรับระดับแรงดัน MAX 232

1. **การส่งข้อมูลออก** โดยไอซีปรับระดับแรงดัน MAX 232 จะรับข้อมูลจากอุปกรณ์ TTL/CMOS เข้ามาที่ขา TX IN จากนั้นไอซีปรับระดับแรงดัน MAX 232 จะทำการยกระดับแรงดันของข้อมูลให้เป็น +10 โวลท์หรือ –10 โวลท์ และส่งออกมาที่ขา TX OUT ผ่านพอร์ท RS 232 เพื่อส่งข้อมูลต่อไปยังอุปกรณ์ภายนอก

2. **การรับข้อมูลเข้า** โดยไอซีปรับระดับแรงดัน MAX 232 จะรับข้อมูลที่ส่งมาจากอุปกรณ์ภายนอกผ่านพอร์ท RS 232 มายังขา RX IN ของไอซีปรับระดับแรงดัน MAX 232 จากนั้นไอซีปรับระดับแรงดัน MAX 232 จะทำการแปลงระดับแรงดันของข้อมูลให้เป็น 0 โวลท์ หรือ 5 โวลท์ และส่งออกที่ขา RX OUT เพื่อทำการส่งข้อมูลต่อไปที่อุปกรณ์ TTL/CMOS

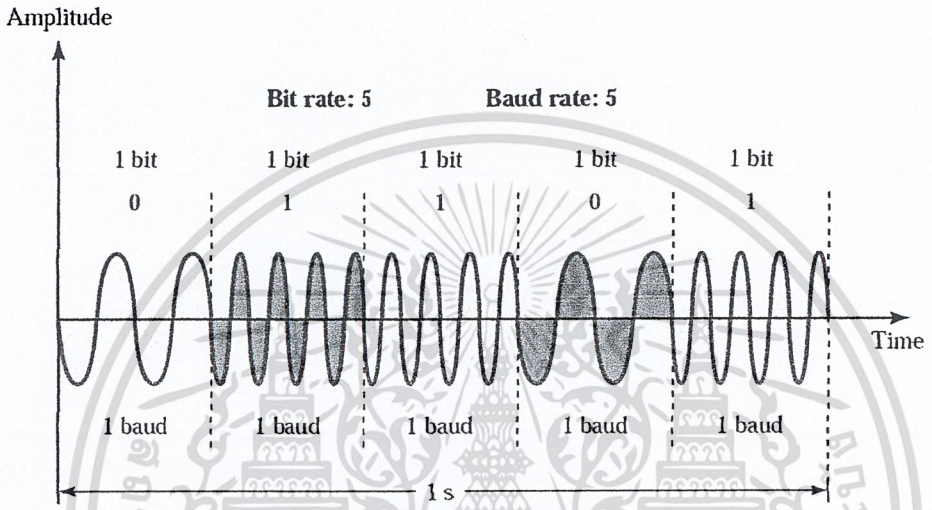


รูปที่ 2.10 การต่อใช้งาน MAX 232 กับ ไมโครคอนโทรลเลอร์

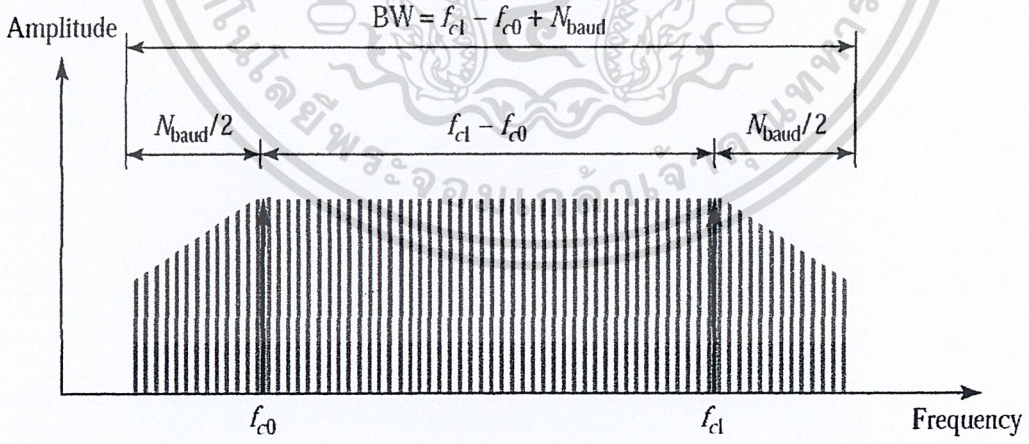
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การมอดูเลตเชิงความถี่แบบดิจิตอล FSK (Frequency Shift Keying)

ระบบการมอดูเลตเชิงความถี่แบบดิจิตอล (Frequency Shift Keying, FSK) เป็นการนำสัญญาณข่าวสารดิจิตอลไปเปลี่ยนแปลงความถี่ของสัญญาณพาหะ ซึ่งเป็นหลักการเดียวกับระบบเอฟเอ็ม (FM) เพื่อทำให้ระบบมอดูเลตนี้สามารถใช้ภาคจำกัดระดับสัญญาณได้สัญญาณเอฟเอสเค (FSK) เป็นสัญญาณที่มีแอมปลิจูด (Amplitude)คงที่ และมีความถี่เปลี่ยนแปลงไปตามระดับลอจิกของสัญญาณข่าวสารดิจิตอล (Digital)



รูปที่ 2.11 การผสมสัญญาณของข้อมูล 01101 ด้วยวิธีการ Frequency Shift Keying ในกรณีนี้บิทเรท (Bit Rate) เท่ากับบอดเรท (Baud Rate)



รูปที่ 2.12 Spectrum ของสัญญาณ FSK ที่มีความถี่ของสัญญาณพาหะ fc0 และ fc1 และอัตราเร็วของข้อมูล Baud (Nbaud)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.5.1 ประเภทการใช้งานของ FSK

สามารถแบ่งการนำไปใช้งานได้ 2 ประเภทใหญ่ๆ คือ

1. การนำไปใช้งานย่านความถี่เสียง เพื่อใช้ในการมัลติเพล็กซ์บนช่องสื่อสารของระบบโทรศัพท์ 4 kHz

2. การนำไปใช้ในงานย่านความถี่สูง ตั้งแต่ย่าน HF (High frequency) หรือ VHF (Very High Frequency) สำหรับระบบส่งวิทยุโทรเลข (Radio teletype)

2.5.2 ข้อดีของระบบมอดูเลตแบบเอฟเอสเค (FSK)

1. *สัญญาณเอฟเอสเค (FSK) เป็นสัญญาณที่มีกรอบ (Envelope) ของสัญญาณคงที่* ดังนั้นจึงไม่มีความไวต่อการเปลี่ยนแปลงต่อขนาดหรืออัตราขยายของสัญญาณข่าวสาร

2. การตรวจจับสัญญาณเอฟเอสเค (FSK) สามารถอาศัยหลักการการเปลี่ยนแปลงความถี่ระหว่างสถานะ 2 สถานะ ซึ่งไม่จำเป็นต้องใช้ความแม่นยำ หรือเที่ยงตรงของความถี่มาก การประยุกต์ใช้งานระบบเอฟเอสเค (FSK) ในปัจจุบันระบบมอดูเลตแบบเอฟเอสเค (FSK) ได้มีการนำไปประยุกต์ใช้อย่างแพร่หลาย

2.5.3 ข้อเสียของระบบมอดูเลตแบบเอฟเอสเค (FSK)

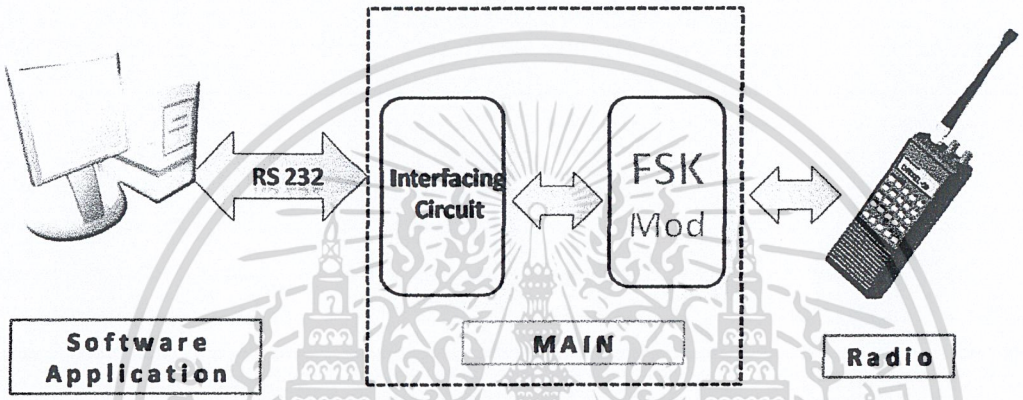
1. ต้องการแบนวิดธ์ (Bandwidth)กว้างกว่าวิธีมอดูเลชั่น (Modulation) แบบเอเอสเค (ASK) เมื่อส่งข้อมูลที่มีบอดเรท (Baud Rate) เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119534

บทที่ 3
การออกแบบปริญญานิพนธ์

3.1 การออกแบบในส่วนของฮาร์ดแวร์ (Hardware)

3.1.1 เครื่อง Main



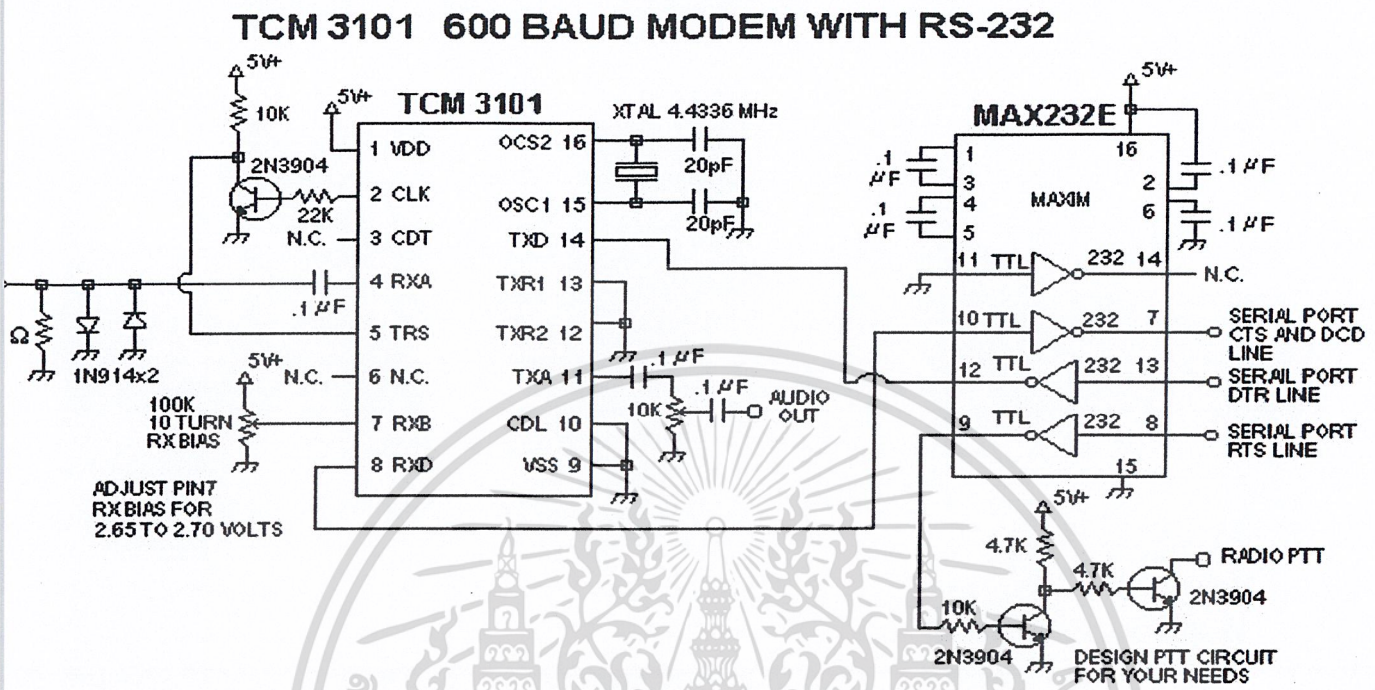
รูปที่ 3.1 แสดงโครงสร้างของเครื่อง Main

หลักการทำงานของเครื่อง Main

ส่วนนี้จะเป็นฮาร์ดแวร์ที่ใช้ในการติดต่อรับ-ส่งข้อมูลหรือคำสั่งจากตัวเครื่องคอมพิวเตอร์ (PC) กับตัวเครื่องที่บรรจุเซนเซอร์วัดอุณหภูมิไว้ หรือที่เรียกว่า Client เมื่อผู้ใช้ (User) ทำการเลือกสิ่งที่ต้องการวัดค่าผ่านทางหน้าจอคอมพิวเตอร์ หลังจากนั้นเครื่องคอมพิวเตอร์จะทำการส่งคำสั่งไปยังวิทยุสื่อสาร (Walkie Talkie) โดยผ่าน RS 232 แล้วเข้าสู่อินเตอร์เฟสเซอร์กิท (Interfacing circuit) เพื่อแปลงแรงดันไฟที่มาจากคอมพิวเตอร์ 12 โวลท์ ไปยัง FSK เพื่อทำการแปลงสัญญาณจากดิจิตอลเป็นอนาล็อกเข้าสู่วิทยุสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร เอฟเอสเค มอดูเลเตอร์ (FSK MUDULATOR)



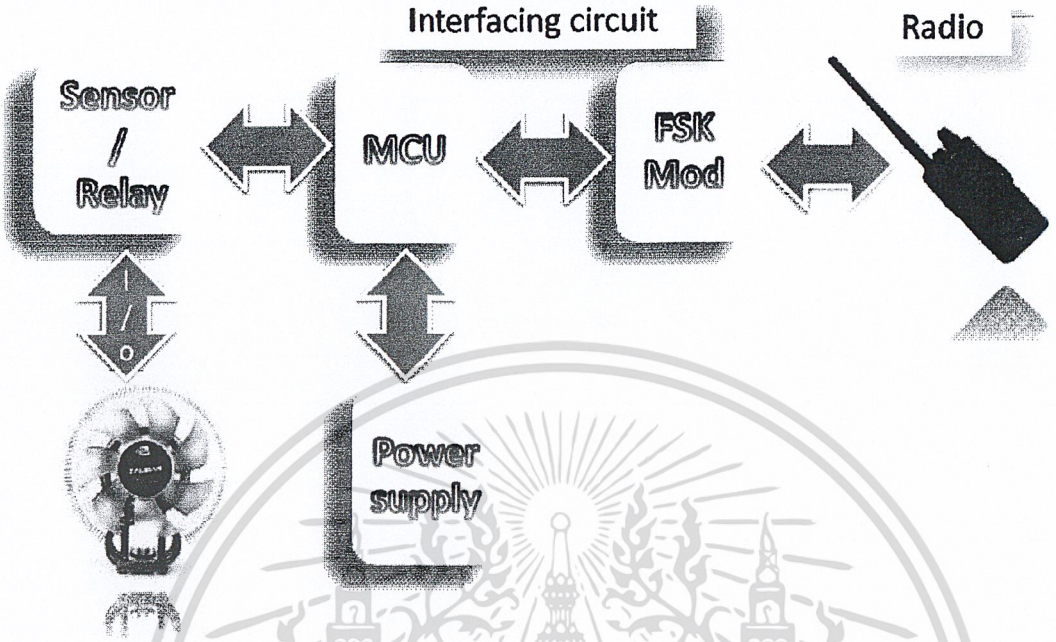
รูปที่ 3.2 แสดงวงจรเอฟเอสเค มอดูเลเตอร์ FSK MODULATOR

หลักการทำงานของวงจรเอฟเอสเค (FSK)

วงจร FSK มีไอซีทั้งหมด 2 ตัว ได้แก่ MAX 232 และ TCM 3101 ซึ่งไอซี MAX 232 ทำหน้าที่ในการแปลงสัญญาณ เพื่อให้คอมพิวเตอร์สามารถติดต่อกับพอร์ต RS 232 ได้ และในส่วนไอซี TCM 3101 ทำหน้าที่ในการรับ-ส่งสัญญาณ โดยมีขาสัญญาณเข้าที่ขา 4 และส่งสัญญาณออกที่ขา 14 และขา 14 จะต่อเข้ากับขา 12 ของไอซี MAX 232 สัญญาณที่ออกจะแปลงจาก สัญญาณดิจิตอล Digital signal)เป็นสัญญาณอนาล็อก (Analog signal) ออกมาในรูปแบบของสัญญาณเสียง และมีขา 5, 12 และ 13 เป็นตัวกำหนดค่าบอดเรท (Baud rate)ในการส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 เครื่อง Client



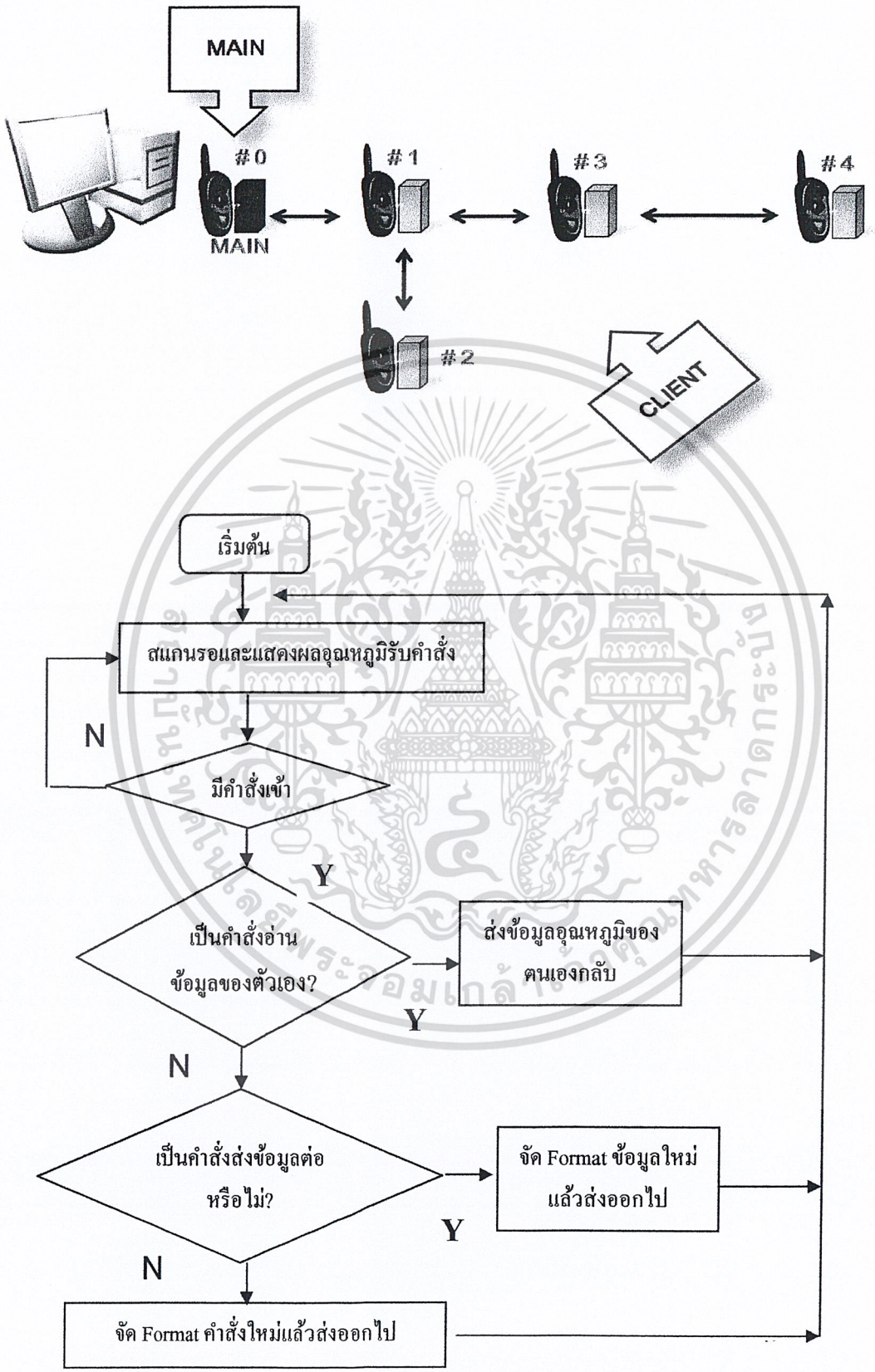
รูปที่ 3.3 แสดงรูปโครงสร้างของเครื่อง Client

หลักการทำงานของเครื่อง Client

ในส่วนของฮาร์ดแวร์ของเครื่อง Client ทำการรับสัญญาณอนาล็อก(Analog signal) จากวิทยุสื่อสารที่มาจากเครื่อง Main โดยมีเอฟเอสเค (FSK) แปลงสัญญาณจากอนาล็อก(Analog)เป็นดิจิตอล (Digital)แล้วทำการส่งคำสั่งไปให้ dsPIC30F 4011 ตรวจสอบว่าเป็นเครื่อง Client ที่ต้องการหรือไม่ ถ้าใช่ก็จะทำการส่งค่าจากเซนเซอร์(Sensor)ไปยังเครื่อง Main โดยผ่านวิทยุสื่อสาร หากไม่ใช่ก็ทำการส่งสัญญาณให้ Client ที่อยู่ในบริเวณใกล้เคียง ทำอย่างนี้ไปเรื่อยๆ จนกว่าจะพบ Client ที่ต้องการ แล้วทำการส่งข้อมูลกลับมายัง Client เดิมที่ทำการส่งสัญญาณมาให้ ทำอย่างนี้ไปเรื่อยๆจนถึงเครื่อง Main

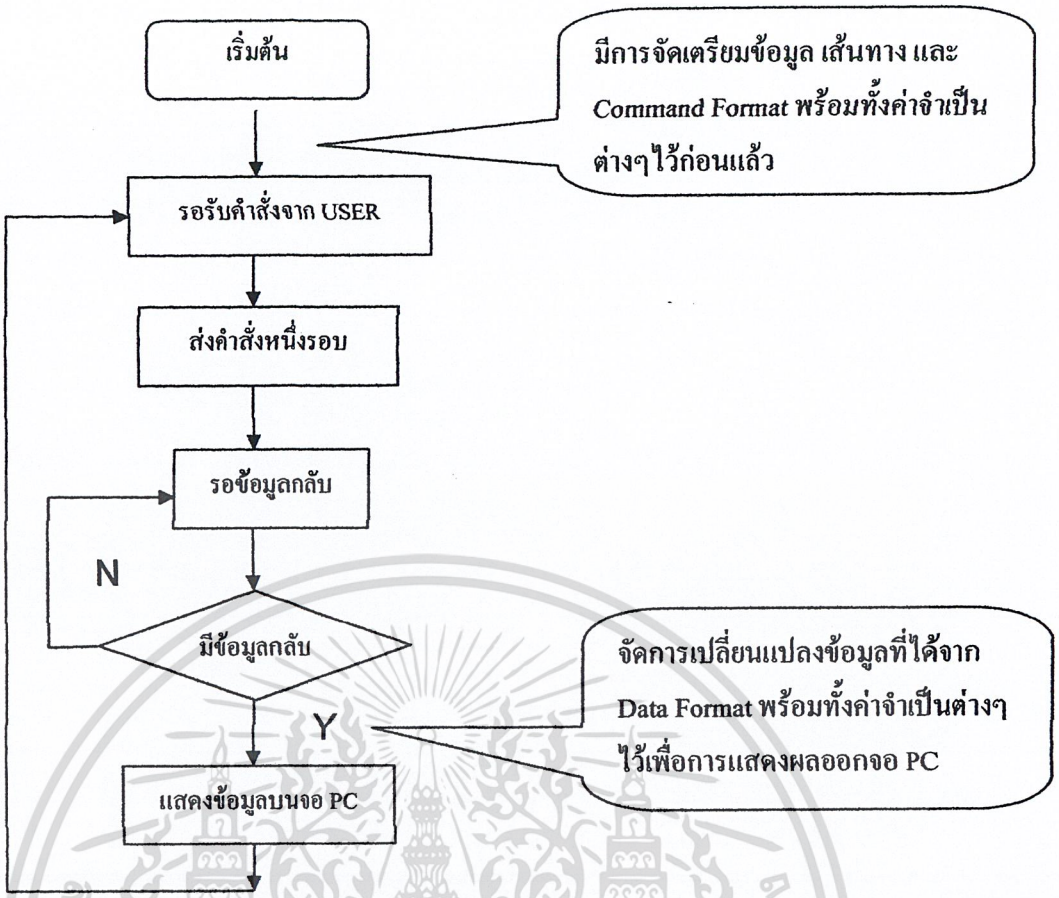
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบในส่วนของ Software



รูปที่ 3.4 แสดงไดอะแกรมซอฟแวร์ควบคุมการรับส่งข้อมูลของตัว Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงไดอะแกรมของส่วน PC

3.2.1 รูปแบบการส่ง

Command Format (ส่งออกไปเพื่ออ่านข้อมูล) และ Data Format (ส่งกลับข้อมูลกลับไปสู่คอมพิวเตอร์)

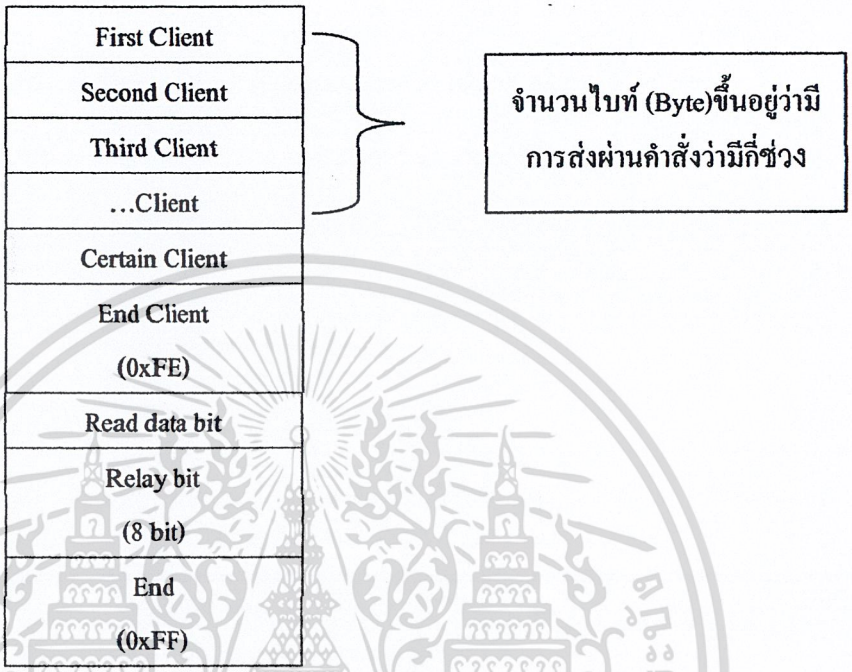
Command Format เป็นฟอร์แมท (Format) ที่ใช้ในการส่งจาก ตัว Main ไปสู่แต่ละช่วงของตัว Clients ที่เป็นตัวส่งผ่านไปจนถึงตัว Client ที่เป็นตัวที่ถูกระบุ (ตัวเป้าหมาย)ให้อ่านข้อมูลเพื่อส่งกลับต่อไปฟอร์แมทคำสั่ง (Command Format) จะประกอบไปด้วยไบท์ (Byte)ที่เรียงกันไปเป็นชั้นๆซึ่งแต่ละชั้นก็จะมีความหมายของตัวเองและจะมาน้อยกี่ชั้นเท่าไดนั้นก็แล้วแต่ความห่างไกลระหว่างตัว Main ถึง Client ที่ระบุให้อ่านข้อมูล ฟอร์แมท (Format)สามารถแสดงได้ ดังรูปที่ 3.6 ดังนี้

First Client ไบท์ (Byte) นี้จะใช้เก็บค่าที่บอกถึงหมายเลขของ Client ที่จะรับคำสั่งเพื่อทำการส่งต่อคำสั่งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Second Client ไบท์ (Byte)นี้จะเก็บค่าที่บอกถึงหมายเลขของ Client ตัวต่อมาจาก First Client ที่จะส่งต่อคำสั่งต่อไป

Certain Client ไบท์ (Byte)นี้จะเก็บค่าที่บอกถึงหมายเลขของตัวสุดท้ายหรือตัวเป้าหมายที่จะอ่านค่าข้อมูลเพื่อส่งกลับต่อไป



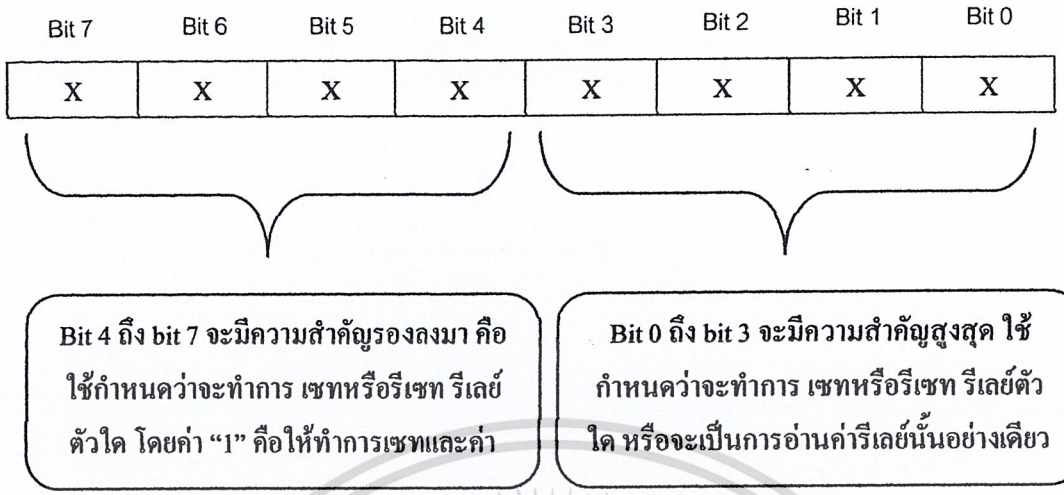
รูปที่ 3.6 แสดงรูปแบบของ Command format

End Client เป็นไบท์ที่ใส่ค่า 0xFE บ่งบอกถึงการจบของรหัสส่งต่อคำสั่ง

Read/Write Relay bit ในไบท์นี้แต่ละบิทจะบอกถึงให้ Client ตัวเป้าหมายนั้นอ่านข้อมูลในลักษณะต่างๆ เช่น นอกจากจะเป็นคำสั่งอ่านอุณหภูมิแล้ว ยังมีการกำหนดสั่งให้ รีเลย์ทำงาน ON/OFF อีกสองตัว หรือจะเป็นการอ่านอย่างเดียวก็ได้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายในแต่ละบิท (Bit) ของ Read/Write Relay bit



การใช้งานบิท (Bit)ต่างๆ

สี่บิท (Bit)ต่ำ คือ bit 0 ถึง bit 3 จะมีความสำคัญสูงสุด ซึ่งจะถูกตรวจสอบก่อนว่า แต่ละบิทมีค่าเป็นอะไร โดยแต่ละบิท (Bit)จะใช้งานคู่กันเป็นคู่ๆกับ สี่บิท (Bit)สูง ระหว่าง bit 0,1,2,3 และ bit 4,5,6,7 โดย

รีเลย์ (Relay)ตัวที่ศูนย์ จะใช้ bit 0 คู่กับ bit 4

รีเลย์ (Relay)ตัวที่หนึ่ง จะใช้ bit 1 คู่กับ bit 5

รีเลย์ (Relay)ตัวที่สอง จะใช้ bit 2 คู่กับ bit 6

รีเลย์ (Relay)ตัวที่สาม จะใช้ bit 3 คู่กับ bit 7

โดย หากบิท (Bit)ใดในสี่บิท (Bit)ต่ำนี้เป็น "0" หมายถึงเป็นคำสั่งที่กำหนดให้รีเลย์ของแต่ละบิท (Bit)นั้น มีสถานะเป็น ON หรือ OFF ตามค่าในสี่บิท (Bit)บน คือ bit 4, bit 5, bit 6, bit 7 เช่น หาก bit 0 มีค่าเป็น "0" และ bit 4 มีค่าเป็น "1" หมายถึงการสั่งรีเลย์ ON นั่นเอง

ตัวอย่างเช่น

0x30 หมายถึงมีการกำหนดให้รีเลย์ (Relay)ตัวที่ ศูนย์=ON, สอง=OFF, สาม=OFF

0x33 หมายถึงมีการกำหนดให้อ่านค่ารีเลย์ (Relay)ตัวที่ ศูนย์, หนึ่ง และให้ สาม=OFF, สาม=OFF

0x3F หมายถึงมีการกำหนดให้อ่านค่ารีเลย์ (Relay)ตัวที่ ศูนย์, หนึ่ง, สอง และ สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End เป็นไบท์ที่ใส่ค่า 0xFF บ่งบอกถึงการจบของรหัสคำสั่งในการส่ง Command Format นี้ไป ในแต่ละช่วงของตัว Client ที่ต้องส่งคำสั่งต่อก็จะต้องมีการทำการปรับ Command Format นี้ไปทุกช่วงจนถึงตัวเป้าหมาย Client ที่ต้องอ่านข้อมูลส่งกลับ ดังแสดงในตัวอย่างดังนี้ ในรูปที่ 3.7 แสดงเส้นทางตัวอย่างการวิ่งไปของ Command Format เพื่อไปอ่านข้อมูลของตัว Client ตัวที่สี่ กำหนดให้ตัว Client หมายเลขสี่ มีการวัดอุณหภูมิเพียงอย่างเดียวและมี Relay ที่ใช้งานอยู่ชุดเดียวในช่องที่ศูนย์ เมื่อเป็นดังนี้เราก็จะได้ค่าคำสั่งในส่วนของ ไบท์ Read Data bit และ Relay bit คือ

Read/Write bit = 0x10 (หากกำหนดให้รีเลย์ตัวที่ศูนย์มีการเซทค่าเป็น ON ส่วนตัวอื่นปิน OFF ส่วน ไบท์ (Byte)คำสั่งอื่นๆ ก็จะเป็นดังนี้

First Client = 0x00

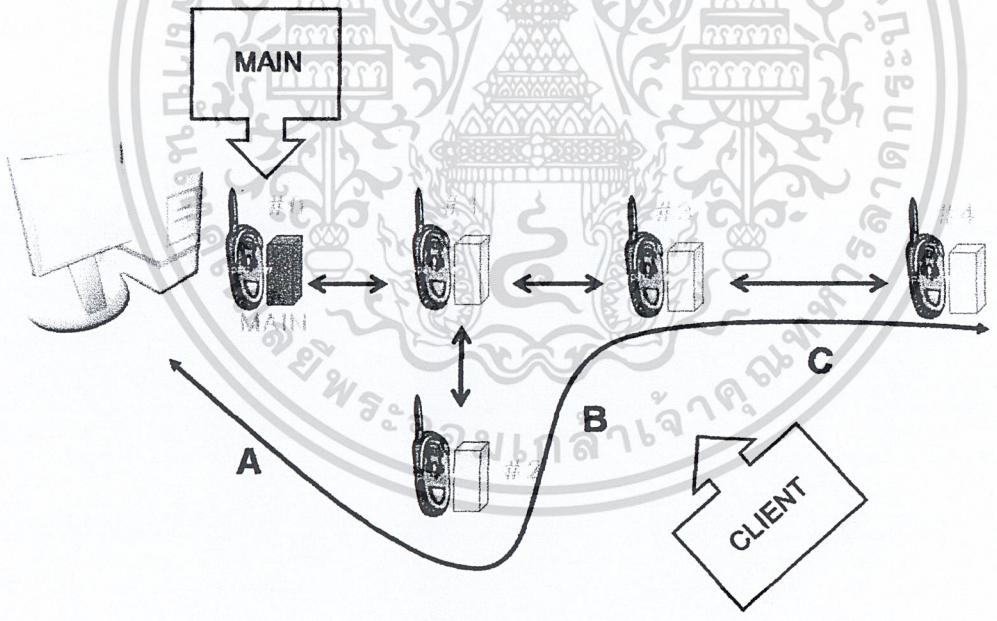
Second Client = 0x02

Third Client = 0x03

Certain Client = 0x04

.....

End = 0xFF



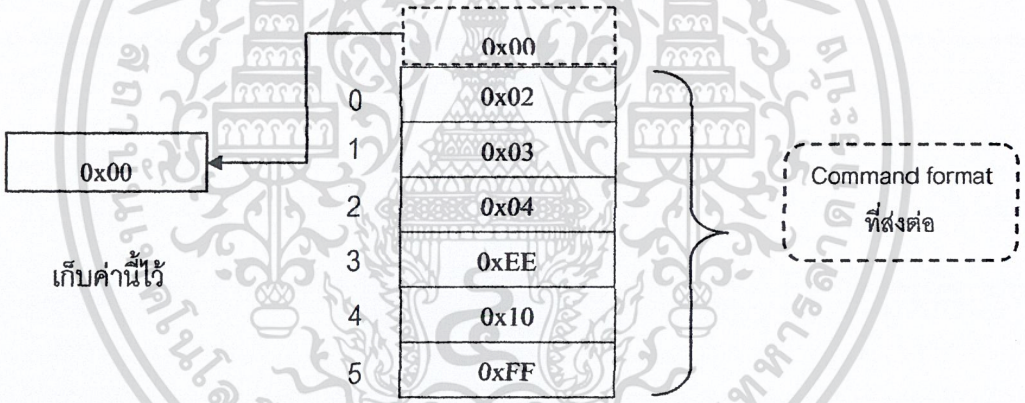
รูปที่ 3.7 แสดงเส้นทางการวิ่งไปอ่านข้อมูลของ Client ตัวหมายเลขสี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

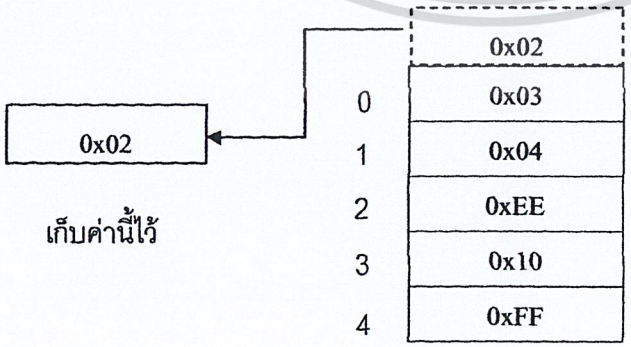
ดังนั้นเราก็จะได้ Command Format ชุดแรกที่ถูกส่งออกไปจากตัว Main (แสดงในรูป 2 จุดที่ A) คือ

0	0x00	จากไหน
1	0x02	ตัวเอง
2	0x03	ไปไหนที่หนึ่ง
3	0x04	ไปไหนที่สอง
4	0xEE	หมดไม่ไปต่อ
5	0x10	สั่งรีเลย์ (Relay)
6	0xFF	จบ command format

และ Command Format ก็จะถูกดัดแปลงเล็กน้อยแล้วส่งต่อไปสู่จุดต่อไปตามเส้นทางในรูป ซึ่งจะได้ Command Format ใหม่ที่จุด B คือ



จะเห็นได้ว่าเพียงแค่ส่วนบนหายไป หนึ่งไบท์ (Byte) ดังนั้น ในจุด C ซึ่งเป็นจุดสุดท้ายก็จะได้เป็นดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงตรงนี้ตัว Client สุดท้ายนี้ก็จะทำการตรวจสอบคำสั่ง ว่าจะได้ให้อ่านอุณหภูมิและสั่ง รีเลย์ตัวใด ให้ทำงาน ก็จะทำตามนั้นทันที

<u>หมายเหตุ</u> ใน Command format จะสังเกตว่าไบท์ที่ระบุหมายเลขตัวที่สามารถรับ Command format ได้ (ไบท์ที่หนึ่ง) จะมีไบท์ที่ศูนย์ที่แสดงถึงว่ามาจากตัวไหนก่อนและไบท์ที่สองที่ ระบุว่าจะต้องส่งไปไหนต่อ แต่หากตรวจไบท์นี้แล้วเจอรหัส 0xFE ก็แสดงว่าไม่มีการส่งต่อแล้วและตัวมันเองคือตัวเป้าหมายที่ต้องส่งค่าการวัดต่างๆ กลับในรูป Data format สู่ PC ต่อไป

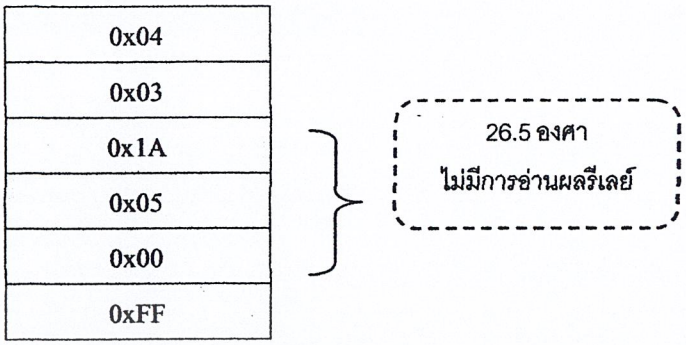
Data Format

เมื่อตัว Client ที่กำหนดได้ทำการอ่านค่าข้อมูลตามคำสั่งที่ต้องการแล้ว ต่อไปก็จะทำการส่งข้อมูลนั้นกลับสู่ PC อย่างถูกต้องตามเส้นทางที่กำหนดต่อไป ซึ่งจำเป็นต้องมีฟอร์แมท ที่สามารถทำงานได้ดังนี้ คือ *Data format เป็นฟอร์แมทที่ใช้ในการส่งข้อมูลกลับสู่ PC*
ในรูปที่ 3 แสดงถึง Format ของ Data Format ที่จะต้องส่งกลับไปตามเส้นทางที่มา จะเห็นได้ว่ามีไบท์ (Byte)แรกที่แสดงถึงหมายเลขผู้ส่ง ส่วนไบท์ (Byte)ที่สองคือ หมายเลขของ Client ที่จะส่งไปหา ตามด้วย ไบท์ (Byte)บอกถึงว่ามีการอ่านอุณหภูมิกี่ช่อง (ตัวอย่างนี้คือช่องเดียว) และตามด้วยข้อมูลของอุณหภูมินั้นซึ่งจะมี เลขหน้าจุดทศนิยม(8 bits) และเลขหลังจุดทศนิยม (8 bits) ตามด้วยไบท์ (Byte)ที่บอกสถานะของรีเลย์ (Relay)ทั้งแปดตัว(หากใช้เต็มที่) ว่าสถานะเป็น ON หรือ OFF (โดย ON จะแสดงด้วยบิท (Bit)ที่เป็น"1" และ OFF แสดงด้วยบิท (Bit)ที่เป็น "0") สุดท้ายก็ปิดด้วยคำสั่งจบ (End)

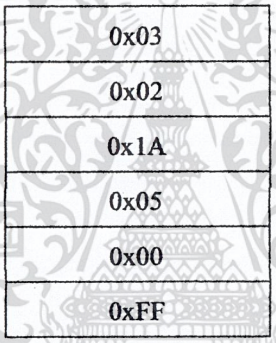
Own number (หมายเลขตัวเอง)
Send number (หมายเลขที่จะส่งไปหา)
อุณหภูมิเลขหน้าจุดทศนิยม (8 bits)
อุณหภูมิเลขหลังจุดทศนิยม (8 bits)
บอกสถานะของ รีเลย์
End (0xFF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจากตัวอย่างนี้ เราก็จะได้ Data Format ที่ส่งออกไปช่วงแรกคือ ที่จุด C คือ



เมื่อ Client หมายเลข 3 ได้รับ Data Format แล้ว มันจะรู้ว่าจะส่งข้อมูลทั้งหมดไปต่อโดยจะมีการดัดแปลง Data Format เล็กน้อยดังนี้



เป็น Data Format ที่จะถูกส่งออกไป ณ จุด B

<u>หมายเหตุ</u> หมายเลขแรกที่เป็นหมายเลข 3 นั้น ได้มาจาก ในตอนที่มันรับ Command Format นั้น มันจะเก็บหมายเลขนี้ไว้แล้ว เพื่อรอ Data Format กลับมาจะได้ใช้ในการส่งกลับได้อย่างถูกต้องนั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นในจุด A เราก็จะมี Data format เป็น

0x02
0x00
0x1A
0x05
0x00
0xFF

ถึงจุดนี้ เมื่อ Main ได้รับ Data format นี้แล้วก็จะทำให้สามารถแยกแยะค่าข้อมูลที่ต้องการ เพื่อส่งให้ PC ทำการแสดงผล หรือเก็บบันทึกไว้ตามต้องการ

3.2.2 การเซทค่าก่อนใช้งาน

ดังนั้นการ กำหนดคือต่างๆ เช่น เส้นทางการเดินทางของ Command/Data Format ไปสู่ตัว Client แต่ละตัวนั้นก็จะมีความจำเป็นมากที่จะต้องกำหนดไว้ก่อนพร้อมทั้งกำหนดถึงจำนวนของช่องอุณหภูมิ, รีเลย์ที่ใช้งานหรือ ช่องใช้งานอื่นๆเพิ่มเติมที่อาจมีมากขึ้นมาอีก ซึ่งหากมีมากขึ้นก็ต้องมีการแก้ไขเพิ่มเติม Command/Data format (Protocol) ใหม่ให้สอดคล้องกับความต้องการต่อไปซึ่งสามารถทำได้ไม่ยากนัก งานต่างๆทั้งหมดที่กล่าวมานั้นผู้สร้างเครื่องและออกแบบซอฟแวร์จะเป็นผู้ติดตั้งให้ผู้ใช้ สามารถนำไปใช้งานได้ทันที โดยที่ผู้ใช้ไม่ต้องทราบในส่วนนี้ก็ได้ แต่หาก มีวิธีการใดที่จะสามารถทำให้การกำหนดเส้นทางและเพิ่มลดช่องข้อมูลต่างๆนี้เป็นไปได้แบบอัตโนมัติหรือง่ายต่อการเซทค่า หรือแม้แต่หากสามารถมีวิธีทราบถึงว่า Client ตัวไหนไม่ทำงานได้แล้ว ก็จะเป็นประโยชน์อย่างสูงต่อการนำไปใช้งานจริงต่อไป

3.2.3 ซอร์ฟแวร์บนตัว PC

Command Format และ Data Format ทำการรับ-ส่งระหว่างเครื่องผู้ใช้ PC ไปสู่เครื่อง Main ก็จะมีลักษณะเช่นเดียวกันกับที่กล่าวมาข้างต้น (คือ fix ที่ 2 ไบท์ ในกรณี Command Format) ส่วน Data Format ก็คือการจัดรูปแบบ Command Format เพื่อส่งไปและเมื่อรับ Data Format แล้วต้องทำการแยกข้อมูลเพื่อจะได้แสดงผลออกหน้าจอของผู้ใช้หรือ ทำการจัดเก็บลง Hard disk ต่อไป ดังนั้นจึงต้องมีการออกแบบแต่ละส่วนดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบเป็น GUI หรือ เมนูให้ง่ายต่อการใช้งาน หากใช้ Visual Studio 2010

1. การออกแบบหน้าจอ

   - แสดงเวลาและวันเดือนปีปัจจุบัน
   - มีการเลือกการใช้งานคอมพอร์ต (Comport)
   - ช่องแสดงผลอุณหภูมิที่วัดได้ออกทางหน้าจอคอมพิวเตอร์
   - ปุ่มเลือกรีเลย์ (Relay), ปุ่มบันทึกอุณหภูมิ, ปุ่มลบหน้าจอ

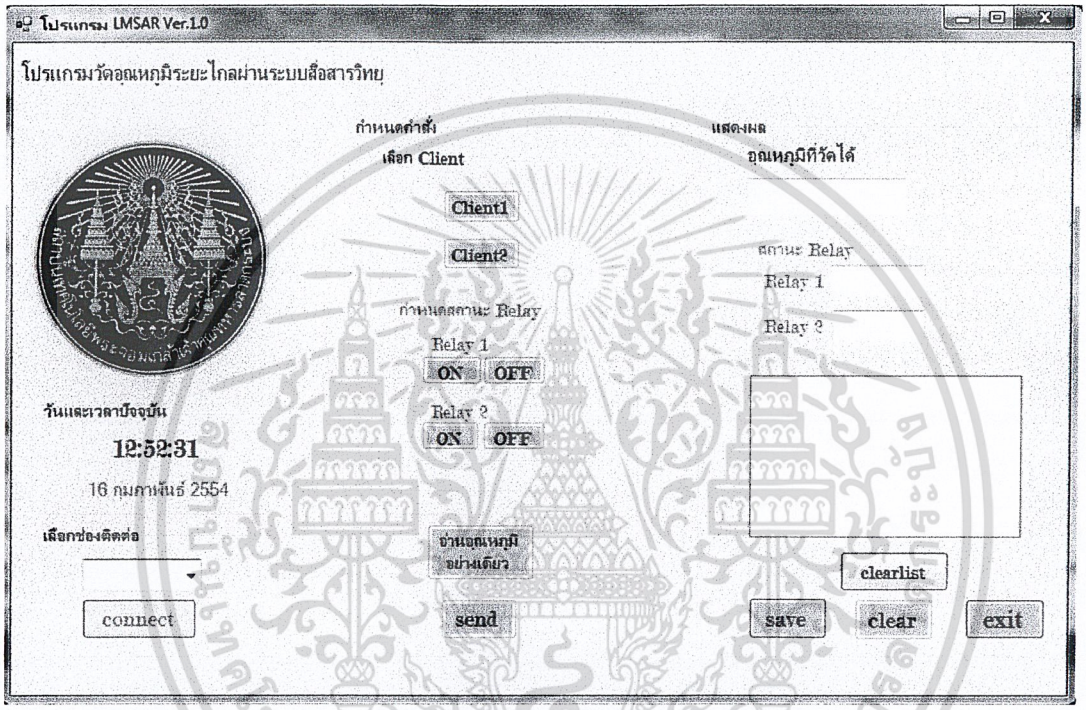
2. เขียนโปรแกรมการใช้งาน RS 232 เพื่อทำการติดต่อกับบอร์ด dsPIC30F 4011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

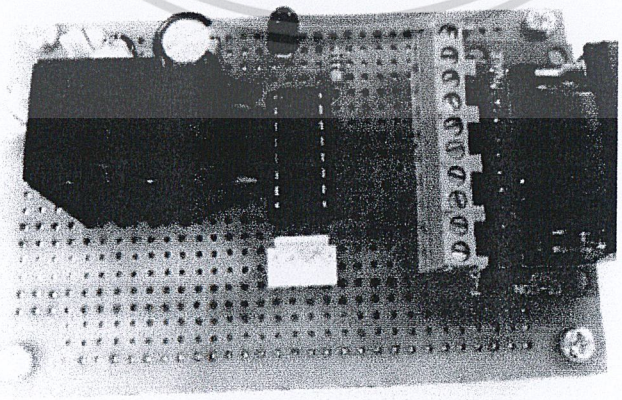
บทที่ 4
การทดลองและการใช้งาน

4.1 อุปกรณ์ที่ใช้

ในการทำงาน อุปกรณ์และส่วนเกี่ยวข้องที่ใช้ มีดังนี้

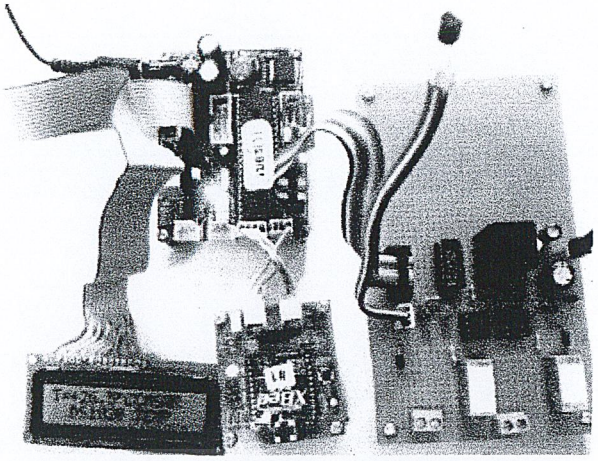


รูปที่ 4.1 โปรแกรม GUI

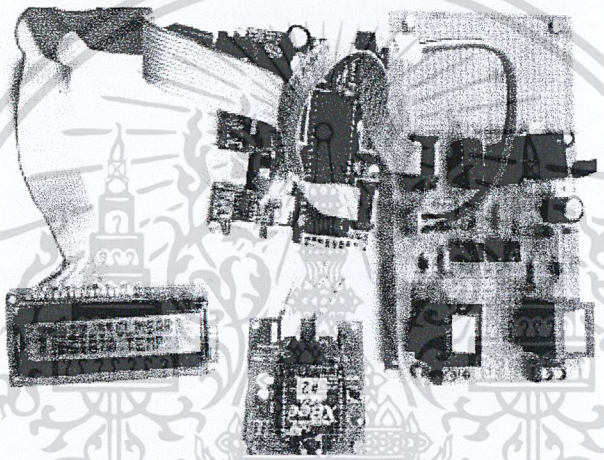


รูปที่ 4.2 เครื่อง Main

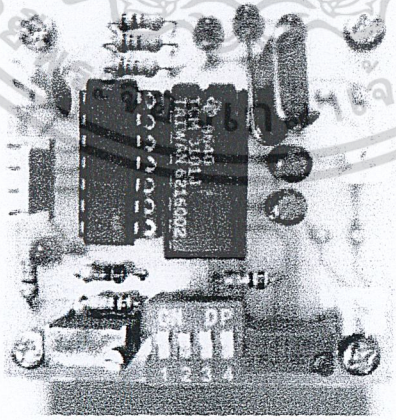
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 เครื่อง Client ตัวที่หนึ่ง



รูปที่ 4.4 เครื่อง Client ตัวที่สอง



รูปที่ 4.5 วงจรเอฟเอสเค (FSK)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ขั้นตอนการทดลอง

การทดลองในส่วนเครื่องตรวจสอบวัดค่าอุณหภูมิระยะไกลแบบใหม่ โดยรับ-ส่ง ข้อมูลระบบ เอฟเอสเค(FSK) ทำให้สามารถรับส่งด้วยระบบวิทยุได้ ซึ่งจะมีการตั้งเครื่องวัดของตัวลูก (Client) ได้ครอบคลุมพื้นที่มากขึ้นโดยไม่ต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุ แต่เนื่องจากมีงบประมาณจำกัดในการทดลองจึงได้มีการใช้ XBee PRO 2.4GHz แทนการส่งโดยใช้วิทยุสื่อสาร จึงได้มีการแบ่งการทดลองออกเป็น 2 ส่วน เพื่อเป็นการทดสอบซอฟต์แวร์ว่าสามารถส่งต่อกันเป็นทอดๆได้ และ ทดสอบวงจรเอฟเอสเค(FSK) เพื่อแสดงให้เห็นว่าได้มีการแปลงออกข้อมูลจาก Digital เป็น Analog แล้ว

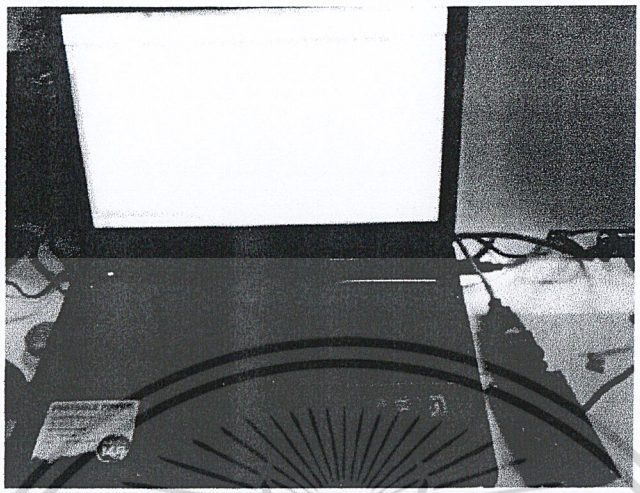
โดยการทดลองแรกเป็นการทดสอบวงจรเอฟเอสเค(FSK) ที่ได้ทำการออกแบบวงจร ว่าสามารถใช้งานได้จริง ที่บอดเรท (Baud rate) เท่ากับ 600 bps ในการใช้งานบนหน้าจอ GUI นั้น ข้อมูลจะถูกส่งไปในรูปแบบของสัญญาณดิจิตอล (Digital) โดยวงจร FSK จะทำการแปลงสัญญาณจาก ดิจิตอล (Digital) เป็น Analog (สัญญาณเสียง) และทำการส่งจากเครื่อง main ไปยัง Client ตัวที่หนึ่ง และส่งต่อไปยัง Client ตัวที่สองได้

และการทดลองที่สองเป็นการทดสอบว่าซอฟต์แวร์สามารถใช้งานได้จริง เนื่องจากไม่มีวิทยุสื่อสารจึงไม่สามารถวางอุปกรณ์ตามระยะจริงได้ ทำให้ต้องนำอุปกรณ์ XBee PRO 2.4GHz มาใช้แทนวิทยุสื่อสาร โดยจะต้องส่งต่อเป็นทอดๆ จาก Client ตัวที่หนึ่ง ไปยัง Client ตัวที่สอง โดยถ้าทำการถอด Client ตัวที่หนึ่งออกก่อนจะไม่สามารถอ่านอุณหภูมิและติดต่อจาก Client ตัวที่สองได้ ข้อมูลก็จะไม่แสดงผลบนหน้าจอ GUI ด้วย ซึ่งการทดลองนี้เป็นการแสดงให้เห็นว่าซอฟต์แวร์มีการส่งสัญญาณ ต่อเป็นทอดๆ โดยผ่าน Client ตัวที่หนึ่งจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

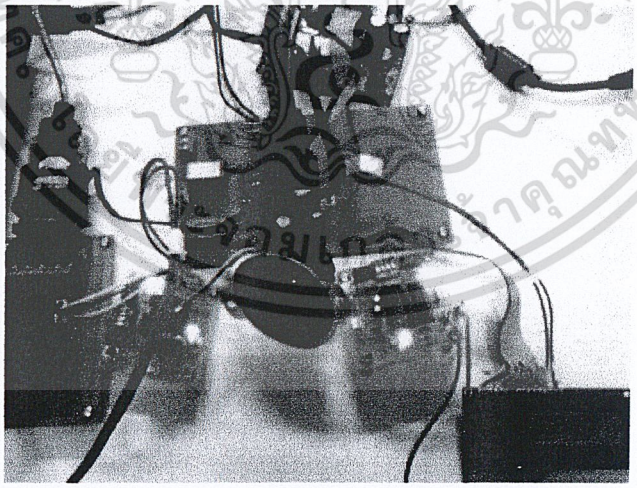
4.2.1 แสดงการทดสอบวงจรเอฟเอสเค (FSK)

1.ทำการเชื่อมต่อ PC ที่ USER ใช้กับเครื่อง Main



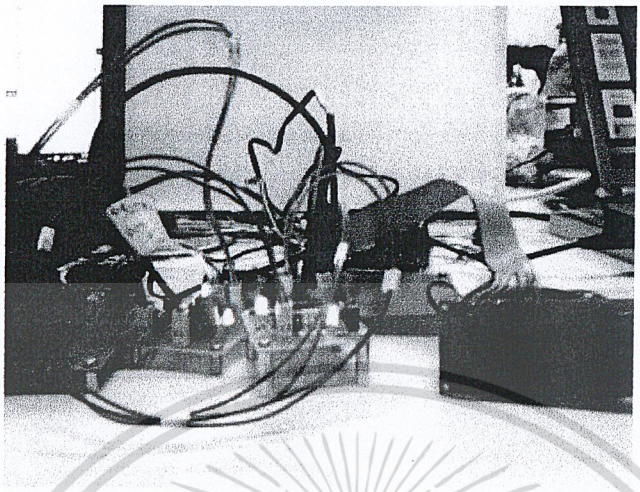
รูปที่ 4.6 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ PC ที่ USER ใช้

2. เชื่อมต่อเครื่อง Main กับ วงจรเอฟเอสเค (FSK) แล้วทำการต่อเข้ากับลำโพงเพื่อแสดงให้เห็นว่า Output ที่ออกมาเป็นเสียง เพื่อพิสูจน์ว่าสัญญาณดิจิตอล (Digital signal) ถูกแปลงเป็นสัญญาณอนาล็อก (Analog signal) แล้ว



รูปที่ 4.7 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ วงจรเอฟเอสเค (FSK) แล้วทำการต่อเข้ากับลำโพง

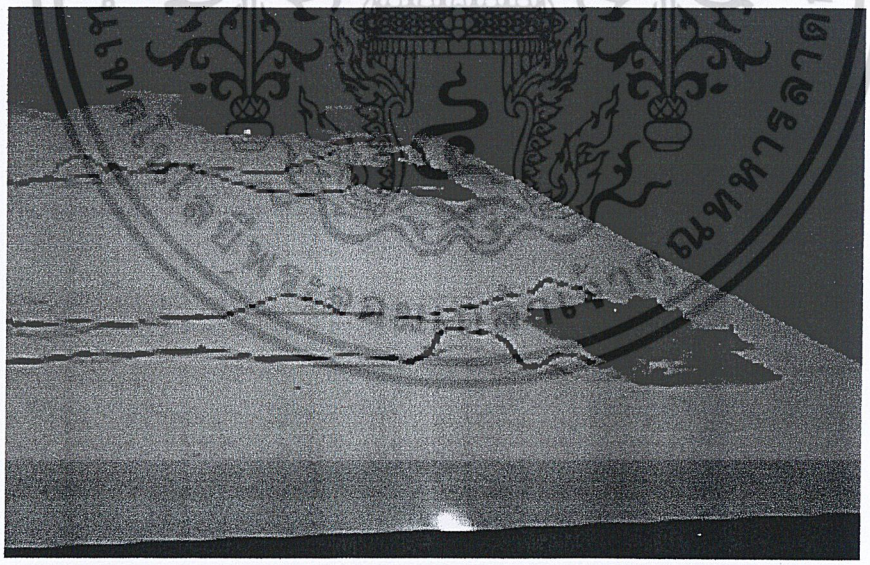
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ วงจรเอฟเอสเค (FSK)

4.2.2 แสดงการทดสอบซอฟต์แวร์โดยติดต่อของ Client ตัวที่หนึ่ง และ Client ตัวที่สอง

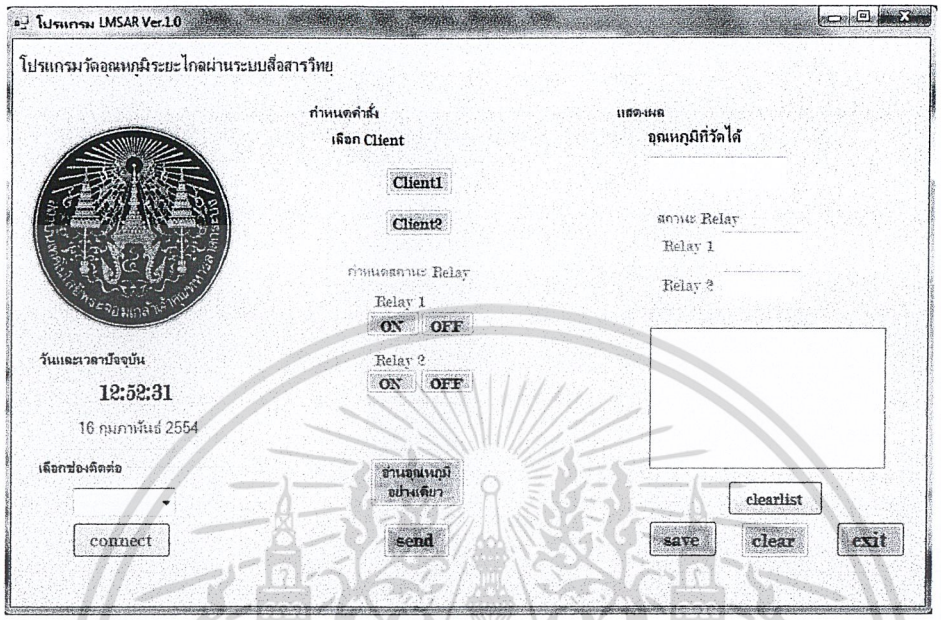
1. เชื่อมต่อเครื่อง Main กับ Client ทั้งสองตัว



รูปที่ 4.9 แสดงการเชื่อมต่อระหว่างเครื่อง Main กับ Client ทั้งสองตัว

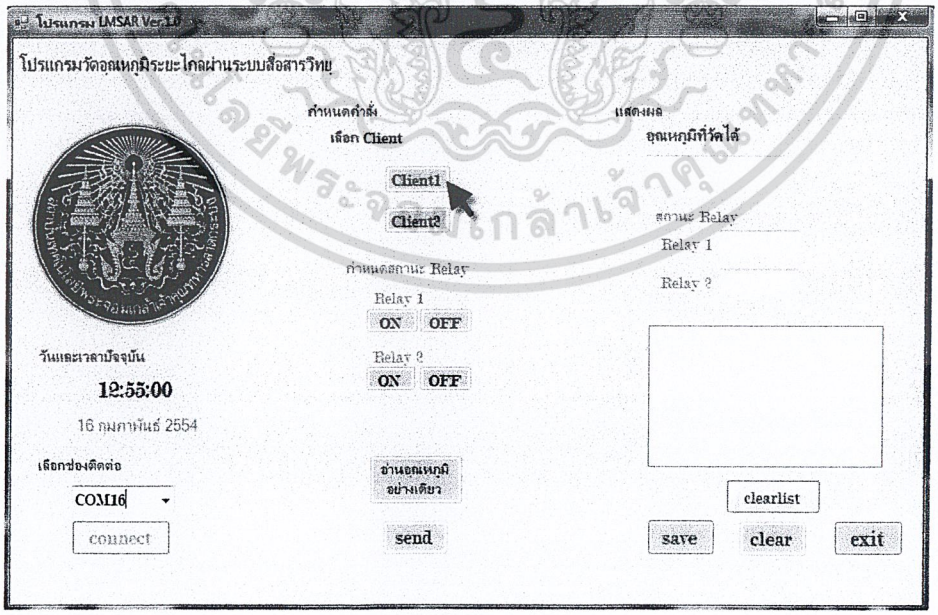
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เปิดโปรแกรม GUI ที่สร้างจากโปรแกรม Visual Studio C# ทำการเลือกช่องทางการติดต่อ และเลือกช่องการอ่านอุณหภูมิว่าต้องการอ่านอุณหภูมิ อย่างเดียวหรือสั่ง ON –OFF หรือการอ่านสถานะของรีเลย์ (Relay) ว่าอยู่ในสถานะใด แล้วจะแสดงกลับมายังหน้าจอ GUI



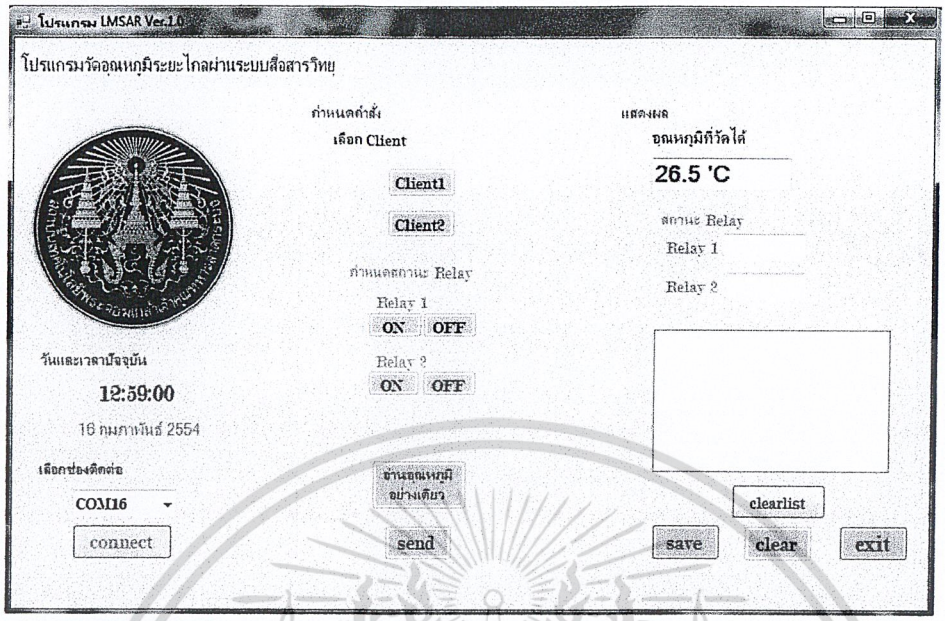
รูปที่ 4.10 โปรแกรม GUI

3. แสดงการทดสอบการทำงานของรีเลย์ (Relay) และวัดอุณหภูมิ ของ Client ตัวที่หนึ่ง

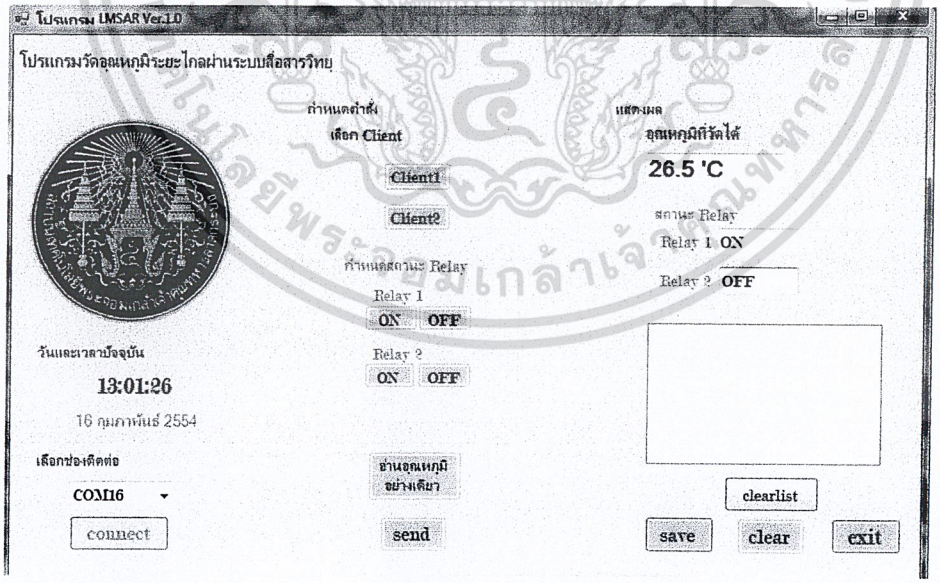


รูปที่ 4.11 เลือก Client ตัวที่หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

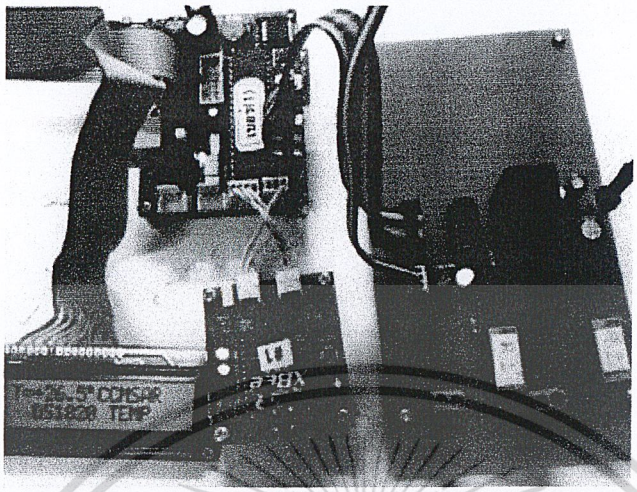


รูปที่ 4.12 เลือกให้ Client ตัวที่หนึ่งแสดงการอ่านอุณหภูมิอย่างเดียว



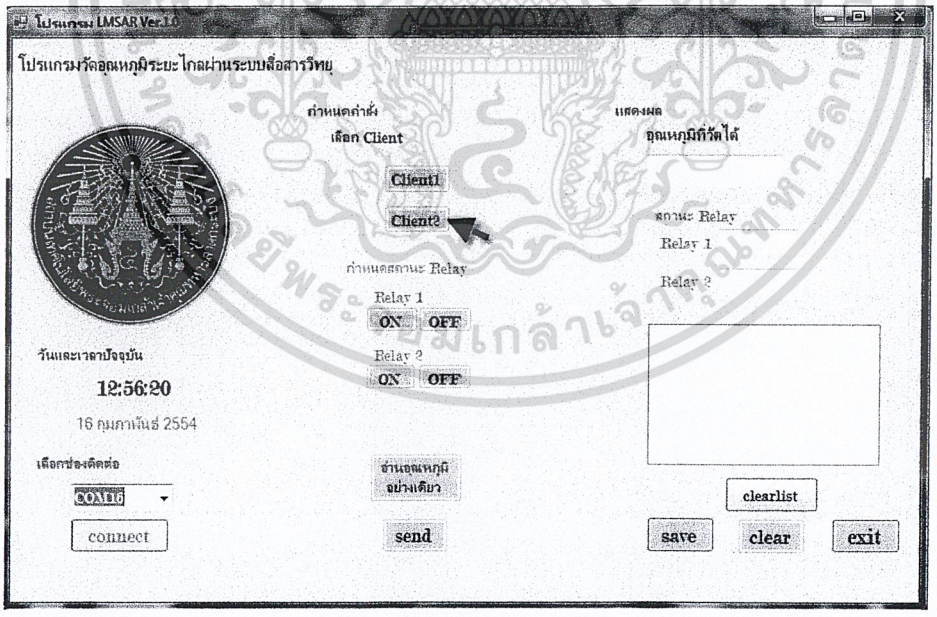
รูปที่ 4.13 เลือกให้ Client ตัวที่หนึ่ง แสดงการอ่านอุณหภูมิควบคุมวงจรรีเลย์ (Relay) ทั้งสองตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



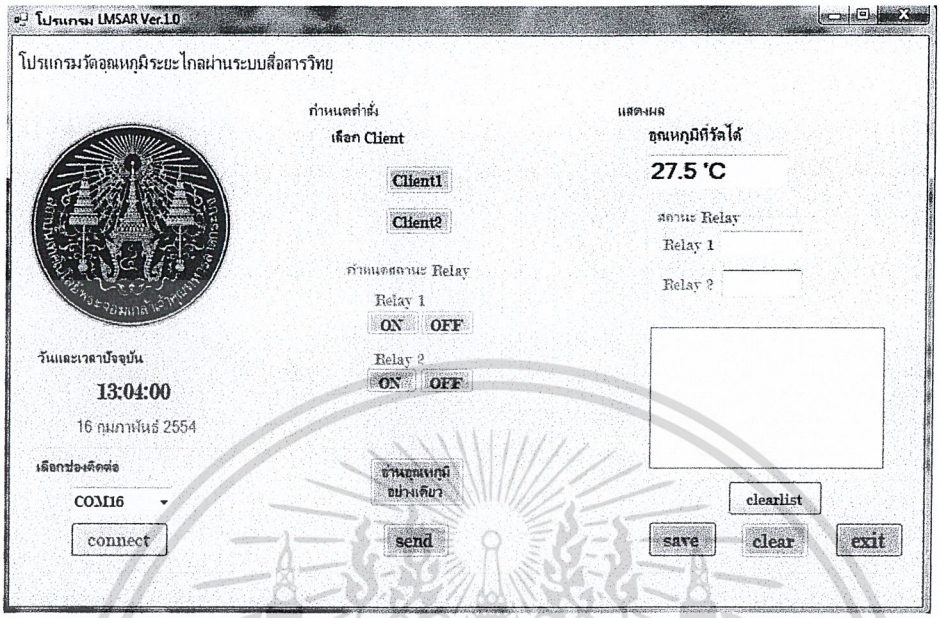
รูปที่ 4.14 แสดงการควบคุมวงจรรีเลย์ (Relay) ทั้งสองตัว ของ Client ตัวที่หนึ่ง

4. แสดงการทดสอบการทำงานของรีเลย์(Relay) และวัดอุณหภูมิ ของ Client ตัวที่สอง

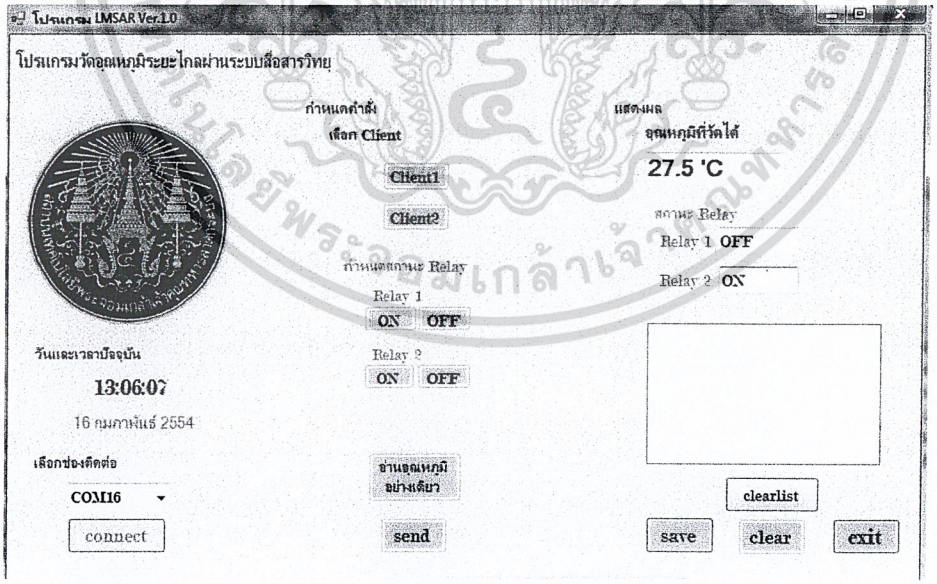


รูปที่ 4.15 เลือก Client ตัวที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

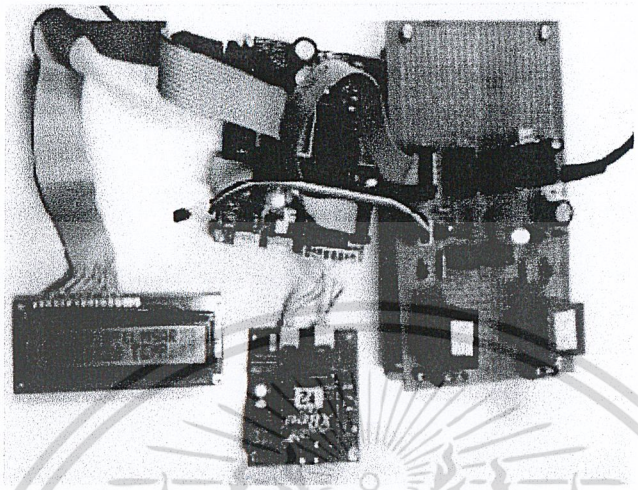


รูปที่ 4.16 เลือกให้ Client ตัวที่สอง แสดงการอ่านอุณหภูมิอย่างเดียว



รูปที่ 4.17 เลือกให้ Client ตัวที่สอง แสดงการอ่านอุณหภูมิควบคุมวงจร Relay ทั้งสองตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงการควบคุมวงจรรีเลย์ (Relay) ทั้งสองตัว ของ Client ตัวที่สอง

ซึ่งในการทดสอบว่าซอฟต์แวร์มีการส่งต่อเป็นทอดๆจริงนั้น ทำได้โดยการถอดเครื่อง Client ตัวที่หนึ่งออก แล้วสั่งหน้าจอ GUI ให้วัดอุณหภูมิ หรือ ควบคุมการทำงานของรีเลย์(Relay) ของ Client ตัวที่สอง จะไม่สามารถทำงานได้ เนื่องจากเราถอดเครื่อง Client ตัวที่หนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากข้างต้นได้มีการแบ่งการทดลองออกเป็น 2 ส่วนซึ่งทำให้ได้ผลการทดลองดังนี้

1. จากการทดลองรับส่งข้อมูลผ่านวงจรเอฟเอสเค (FSK) ที่ได้ออกแบบ โดยใช้บอทเรท (Baud rate) เท่ากับ 600 bps พบว่าจากการรับส่งสัญญาณ ข้อมูลที่ส่งไปและรับกลับมามีความถูกต้องแม่นยำและสามารถรับส่งข้อมูลได้จริงทุกครั้งที่ทำการทดลอง ซึ่งเป็นการแสดงให้เห็นว่าวงจรเอฟเอสเค (FSK) ที่ได้ออกแบบขึ้นมานั้นเหมาะสมและสามารถนำมาใช้ได้จริงกับการส่ง

2. สามารถวัดค่าอุณหภูมิจาก Client ตัวที่หนึ่ง และ Client ตัวที่สอง และส่งค่าที่วัดได้มาแสดงผลทางหน้าจอคอมพิวเตอร์ได้ถูกต้อง และเมื่อได้ทำการถอด Client ตัวที่หนึ่งออกทำให้ไม่สามารถติดต่อกับ Client ตัวที่สองได้ ซึ่งเป็นการพิสูจน์ให้เห็นว่าซอฟท์แวร์ (Software)สามารถใช้งานได้จริง

5.2 ปัญหาที่พบในระหว่างการดำเนินโครงงาน

1. เนื่องจากโครงงานนี้ต้องใช้ความรู้ในหลายๆ ด้านทั้งฮาร์ดแวร์และซอฟแวร์ รวมถึงเทคโนโลยีอื่นๆ ที่ใช้ในโครงงานนี้ ผู้ทำโครงงานมีความรู้ไม่พอจึงทำให้เสียเวลามากในการค้นหาข้อมูล

2. ปัญหาความไม่คงทนและชำรุดง่ายของอุปกรณ์อิเล็คทรอนิก เมื่อทำการทดลองมากๆ ทำให้เสียเวลาในการซ่อมแซม

3. ปัญหาในการสร้างวงจร FSK และวงจร RELAY ที่เกิดปัญหาอย่างมากเนื่องจากจะต้องมีการออกแบบวงจรที่ให้มีความเหมาะสมกับการใช้งานมากที่สุดทำให้ต้องมีการทดลองหลายครั้งทำให้เสียทั้งงบประมาณและเวลา

5.3 แนวทางการแก้ไขปัญหาและพัฒนาโครงงาน

1. พัฒนาโครงข่ายของตัว Client โดยใช้วิธีการส่งต่อกันเป็นทอดๆ เพื่อให้ครอบคลุมพื้นที่ที่ต้องการมากยิ่งขึ้น

2. สร้างวงจรควบคุมรีเลย์เพื่อทำการเปิดปิดอุปกรณ์ตามต้องการ

3. พัฒนาส่วนหน้าจอ GUI ให้สะดวกต่อการใช้งานมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] สัจจะ จรัสรุ่งรวีวร. 2550. **คู่มือ Visual C# 2005 ฉบับสมบูรณ์.** ครั้งที่ 1. นนทบุรี: ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์

[2] พีรภัทร์ สว่างเพียร. 2545. **เทคนิคการเขียนโปรแกรมและเกมด้วย Visual C#.** กรุงเทพ : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

[3] นคร ภักดีชาติ และคณะ. **คู่มือการทดลอง dsPIC Microcontroller เบื้องต้นด้วยโปรแกรมภาษาซี** . กรุงเทพ : บริษัท อินโนเวตีฟ เอ็กพอริเมนต์ จำกัด

[4] http://www.goodsclick.com

[5] http://eestud.kku.ac.th

[6] http://www.rmcs.buu.ac.th

[7] http://www.te.kmutnb.ac.th

[8] http://www.thaimicrotron.com

[9] http://www2.egat.co.th

[10] http://www.ett.co.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

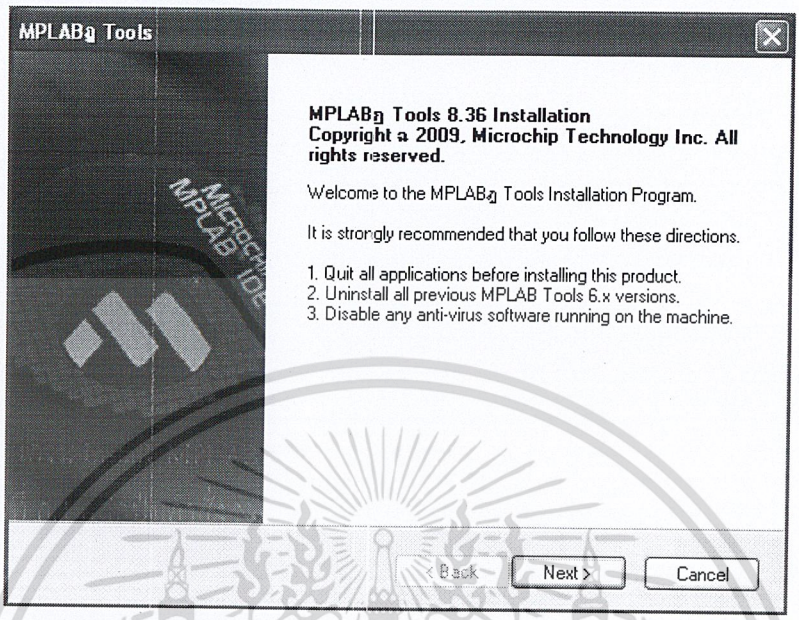
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

การติดตั้งโปรแกรม MP LAB

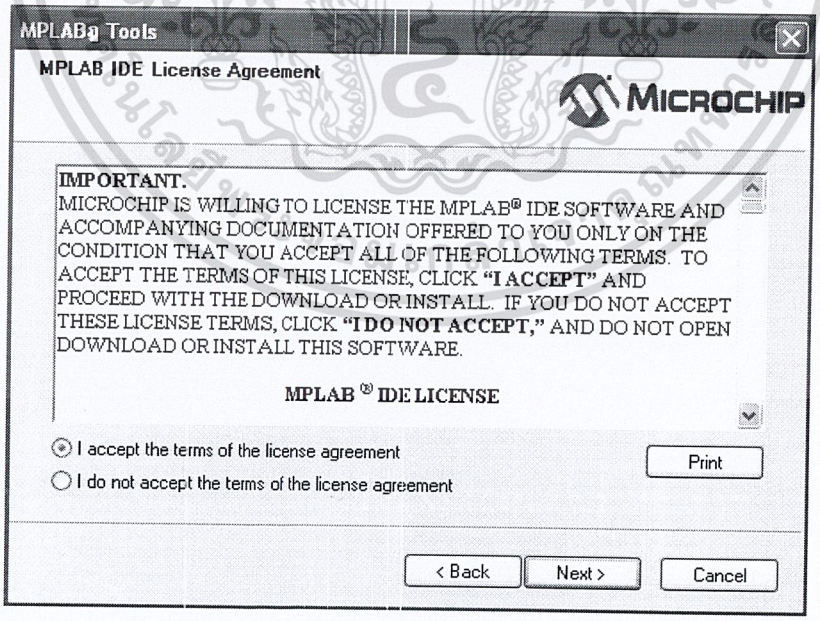
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คลิกเรียกโปรแกรม Install_MPLAB_v8.36.exe หลังจากที่เรียกโปรแกรมแล้วจะได้หน้าต่างการติดตั้ง ดังรูปที่ ก.1 คลิกปุ่มคำสั่ง Next



รูปที่ ก.1 ไดอะล็อก MPLAB Tools เริ่มต้นการติดตั้งโปรแกรม

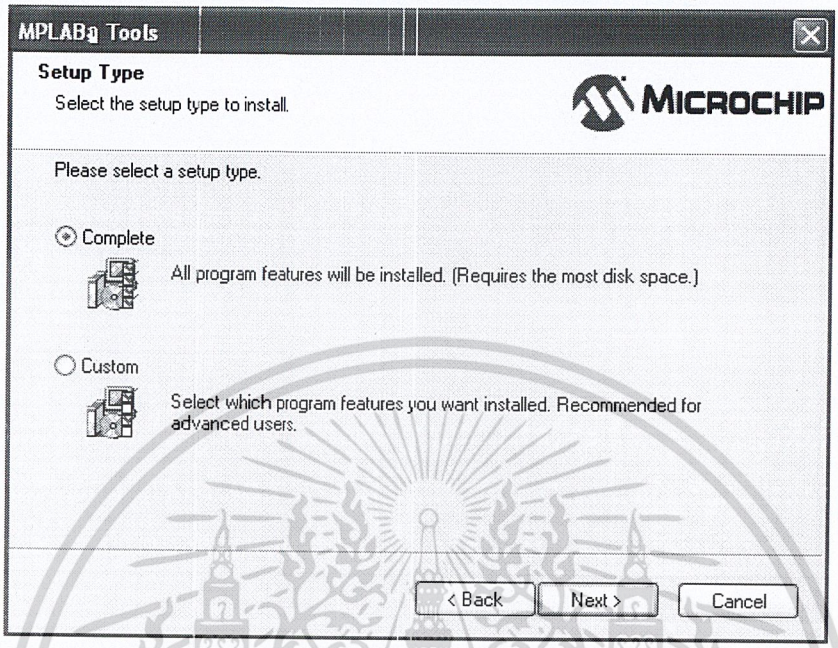
2. คลิกเลือกปุ่มออปชัน "I accept.." คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.2



รูปที่ ก.2 ไดอะล็อกแสดงรายละเอียด License Agreement

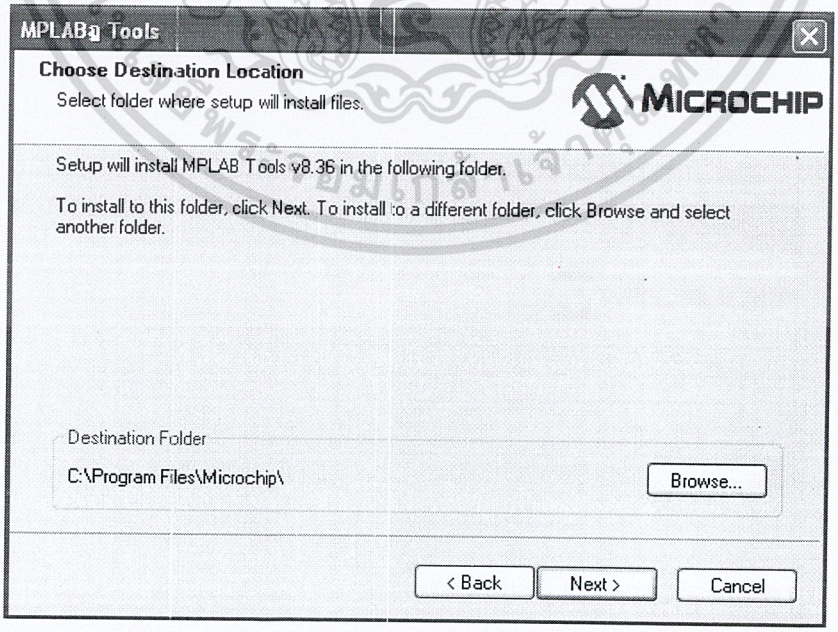
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกเลือกปุ่มออปชันเลือกรูปแบบการติดตั้ง ผู้อ่านสามารถที่จะเลือก complete เพื่อติดตั้งโปรแกรมที่มีมาพร้อมกับ MPLAB Tools ทั้งหมดหรือเลือก Custom เพื่อเลือกโปรแกรมที่จะติดตั้งด้วยตนเอง (แนะนำให้เลือกแบบ Complete) คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.3



รูปที่ ก.3 เลือกรูปแบบการติดตั้งแบบ Complete

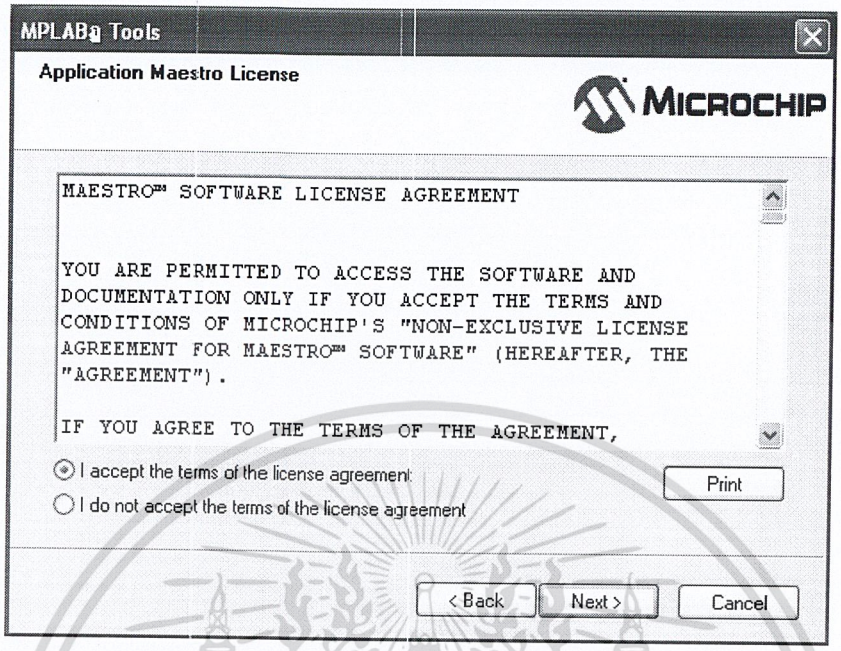
4. กำหนดโฟลเดอร์ของโปรแกรมที่จะทำการติดตั้ง คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.4



รูปที่ ก.4 กำหนดการติดตั้งไปที่ C:\Program Files\Microchip\

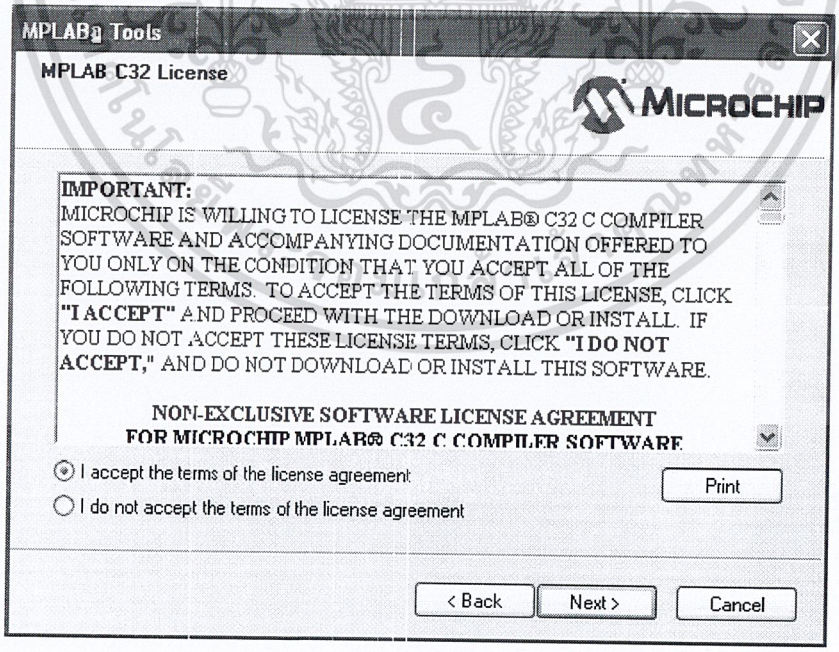
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คลิกเลือกปุ่มออปชัน “I accept..” คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.5



รูปที่ ก.5 แสดงรายละเอียดโปรแกรม Maestro License

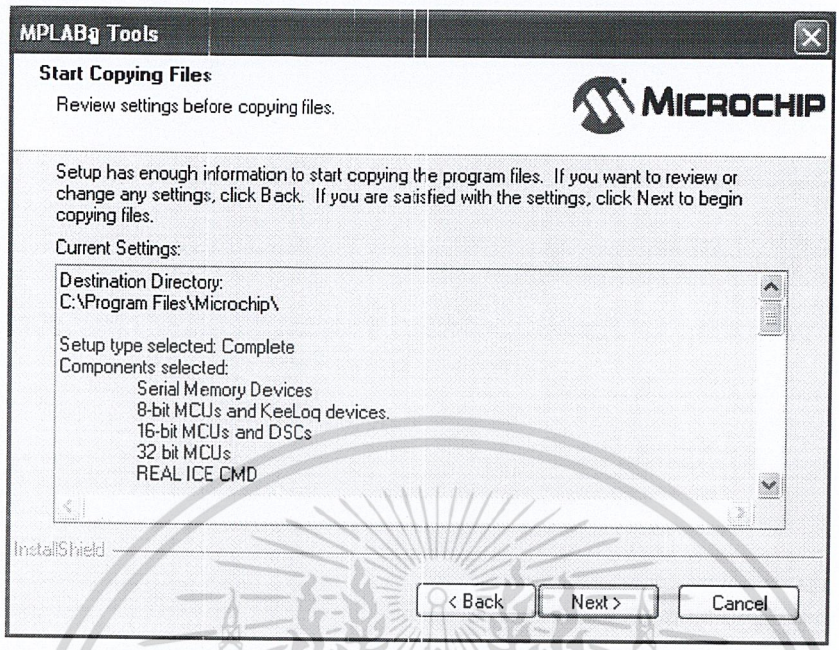
6. คลิกเลือกปุ่มออปชัน “I accept..” คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.6



รูปที่ ก.6 แสดงรายละเอียด MPLAB C30 License

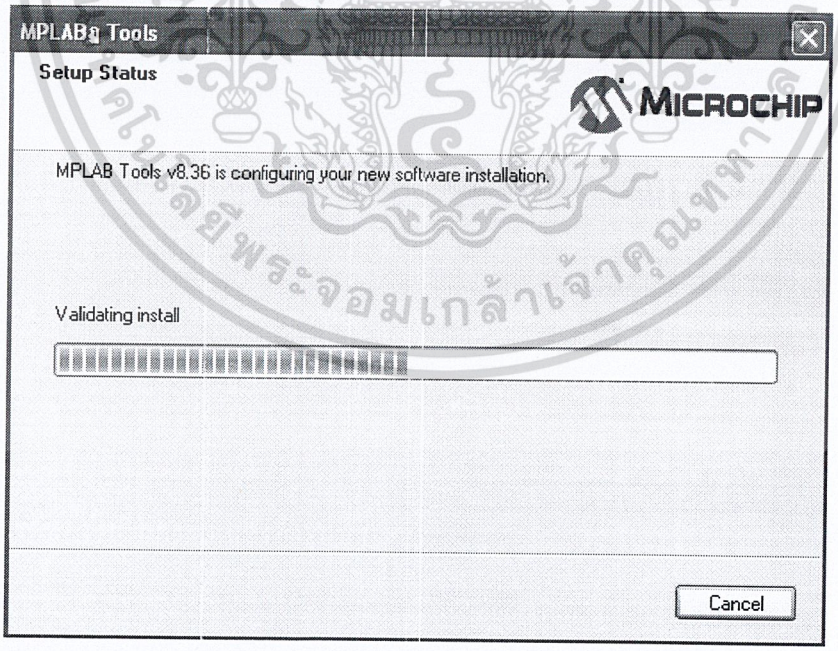
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. แสดงรายละเอียดของโปรแกรมที่จะถูกติดตั้ง คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.7



รูปที่ ก.7 รายละเอียดโปรแกรมที่จะถูกติดตั้ง

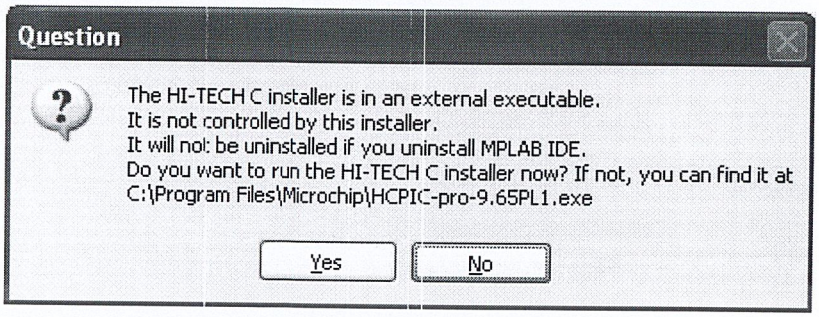
8. เริ่มต้นการติดตั้งโปรแกรม แสดงดังรูปที่ ก.8



รูปที่ ก.8 กำลังดำเนินการติดตั้งโปรแกรม

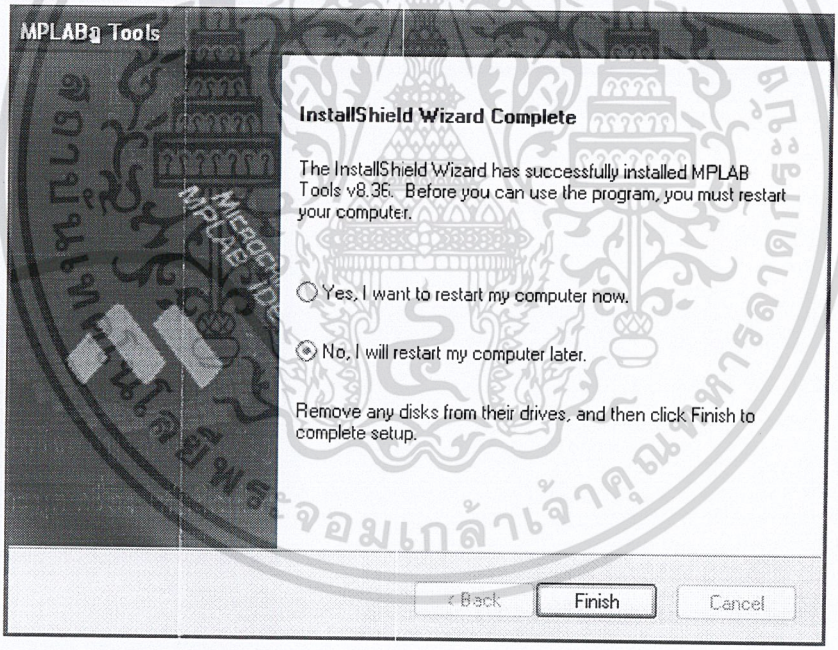
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. แสดงไดอะล็อกแจ้งว่าจะติดตั้งโปรแกรม HI-TECH C หรือไม่ ให้คลิกปุ่มคำสั่ง No ดังรูปที่ ก.9



รูปที่ ก.9 โปรแกรม HI-TECH C

10. การติดตั้งเสร็จสมบูรณ์ แสดงดังรูปที่ ก.10 หากต้องการบูตเครื่องใหม่ คลิกปุ่มคำสั่ง Finish หากไม่ต้องการบูตเครื่อง ให้คลิกเลือกปุ่มออปชัน "No..." ก่อน แล้วจึงคลิกปุ่มคำสั่ง Finish



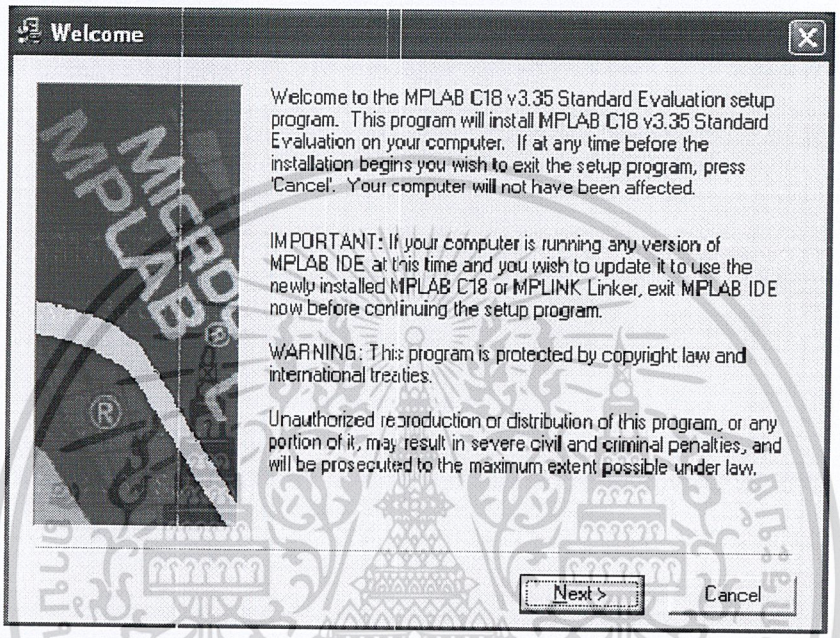
รูปที่ ก.10 แสดงการติดตั้งโปรแกรมเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้ติดตั้งโปรแกรม MPLAB Tools เสร็จแล้ว ในขั้นตอนต่อไปจะติดตั้งโปรแกรม MPLAB C30 Compiler หรือ MPLAB C30

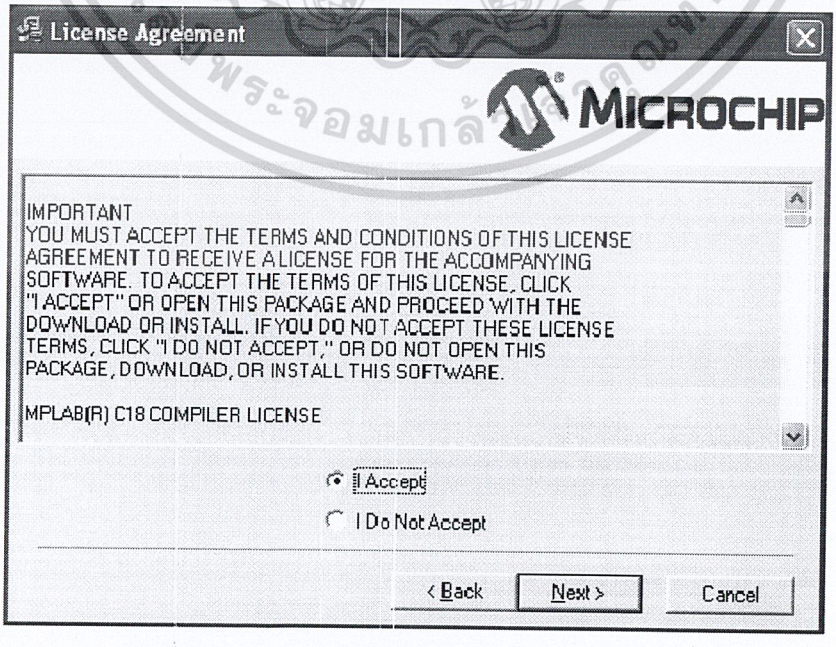
ติดตั้ง MPLAB C30 Compiler

1. แตกไฟล์ MPLAB-C30-Std-Eval-v3_35.zip คลิกเรียกโปรแกรม MPLAB-C30-Std-Eval-v3_35.exe หลังจากที่เรียกโปรแกรมแล้วจะได้หน้าต่างการติดตั้ง ดังรูปที่ ก.11



รูปที่ ก.11 แสดงรายละเอียดเวอร์ชัน MPLAB C30 Standard Evaluation

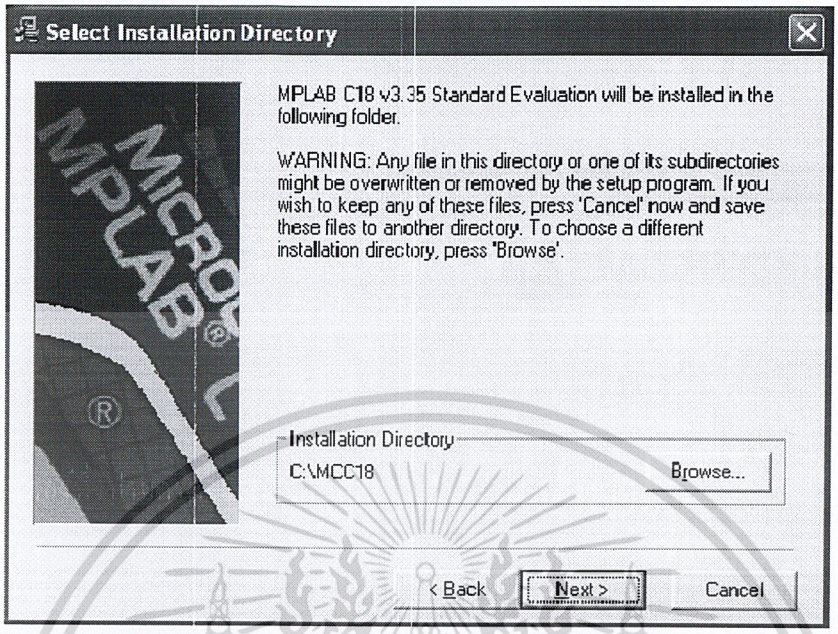
2. คลิกเลือกปุ่มออปชัน “I Accept” คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.12



รูปที่ ก.12 แสดงรายละเอียด License Agreement

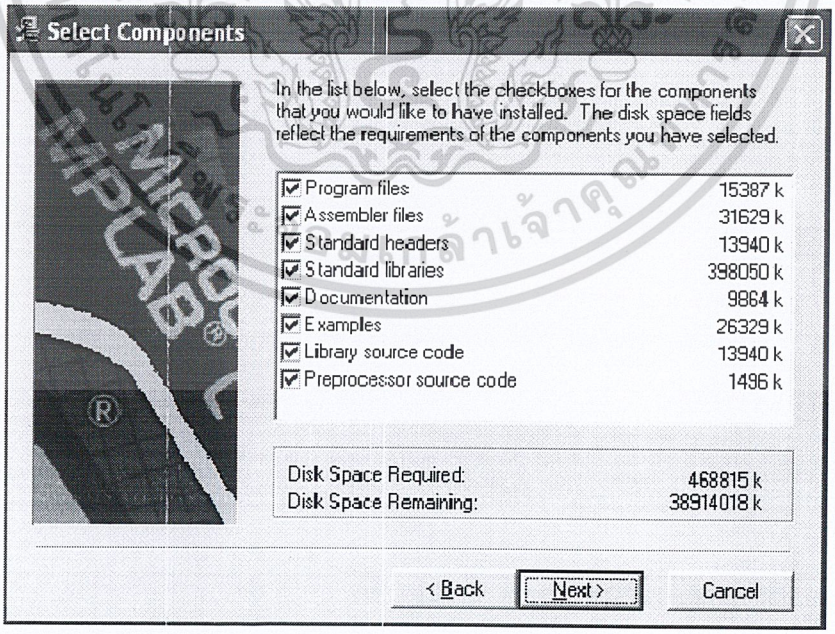
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แสดงโฟลเดอร์ที่โปรแกรมจะถูกติดตั้ง คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.13



รูปที่ ก.13 โฟลเดอร์ที่โปรแกรมจะถูกติดตั้ง

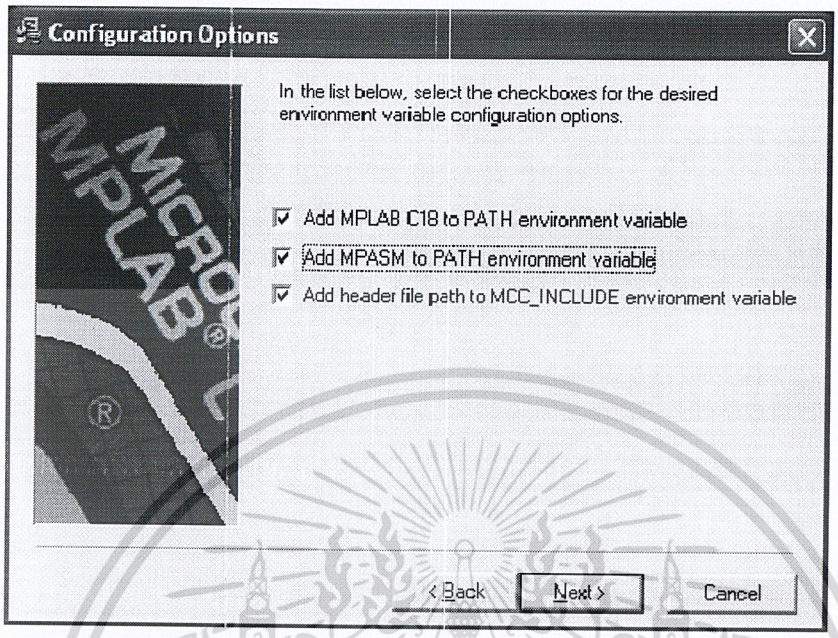
4. แสดงรายละเอียดของไฟล์ที่จะติดตั้ง คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.14



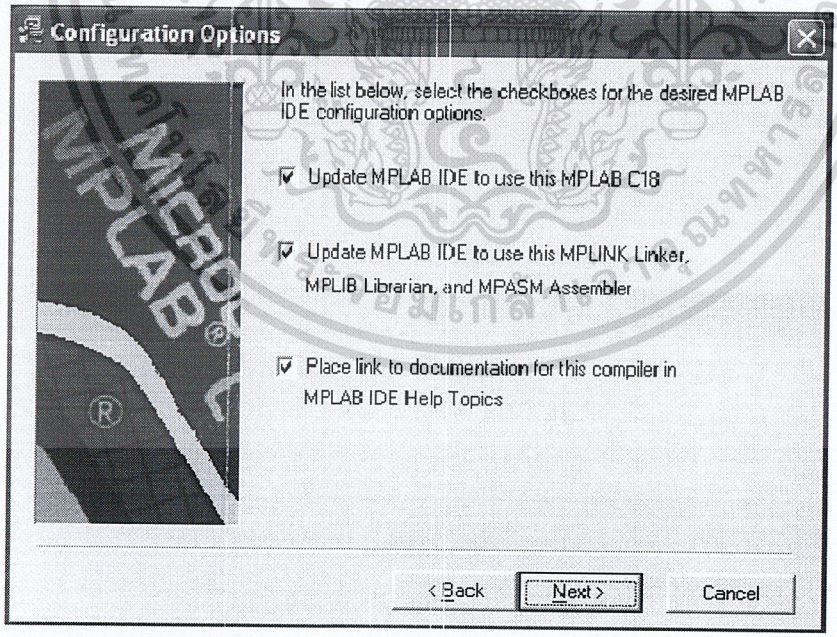
รูปที่ ก.14 แสดงรายละเอียดไฟล์ที่จะติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. คลิกเช็คบ็อกซ์ทั้งหมด คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.15 และ ก.16



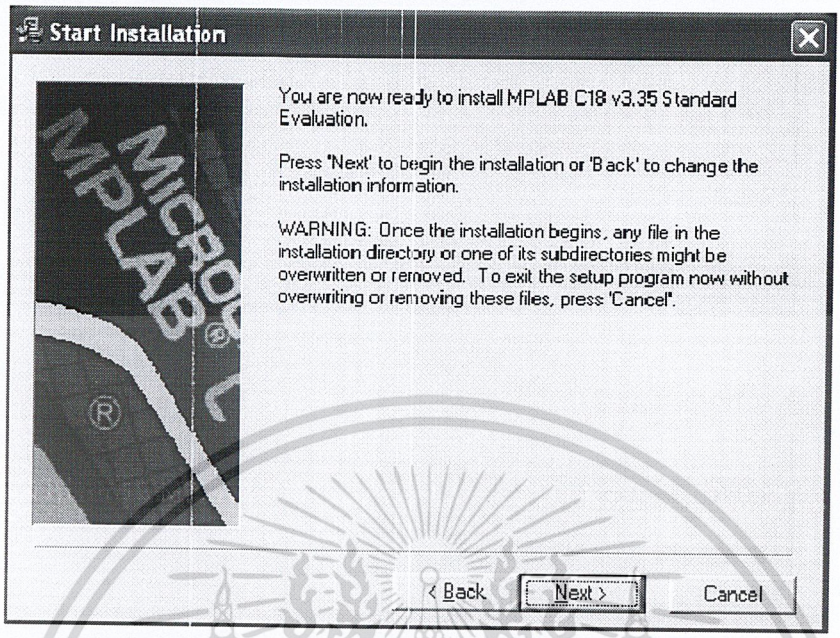
รูปที่ ก.15 กำหนดรายละเอียดของ PATH ไฟล์



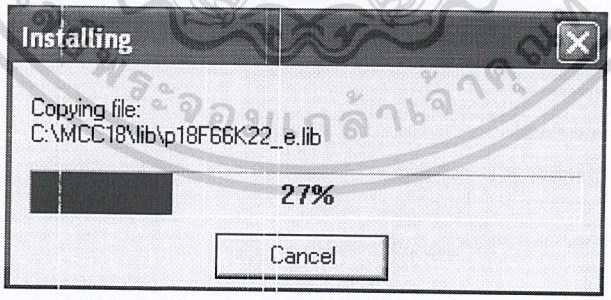
รูปที่ ก.16 ปรับปรุงโปรแกรม MPLAB ให้ใช้งาน MPLAB C30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เริ่มต้นติดตั้งโปรแกรม MPLAB C30 Compiler คลิกปุ่มคำสั่ง Next ดังรูปที่ ก.17 และ ก.18



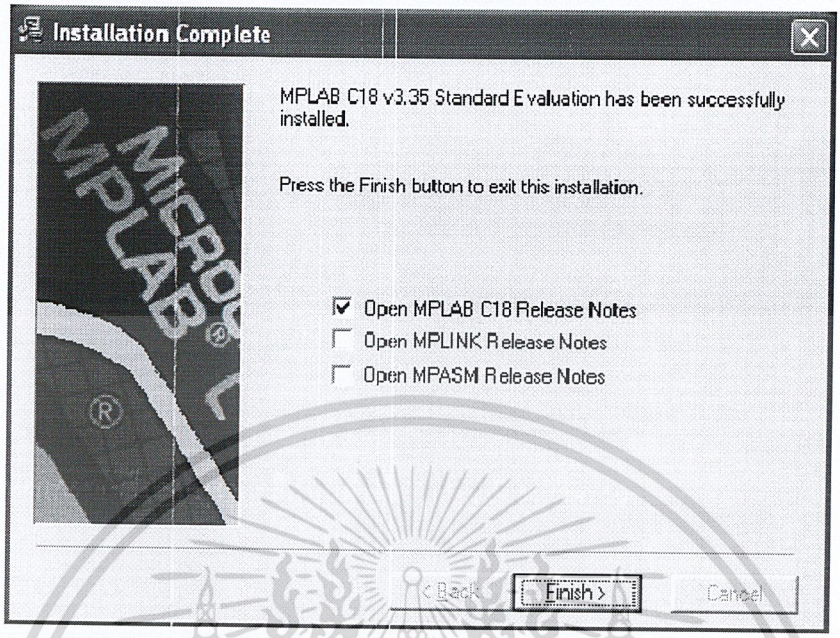
รูปที่ ก.17 แสดงการเริ่มต้นติดตั้งโปรแกรม



รูปที่ ก.18 กำลังดำเนินการติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การติดตั้งโปรแกรมเสร็จสมบูรณ์ คลิกปุ่มคำสั่ง Finish ดังรูปที่ ก.19



รูปที่ ก.19 การติดตั้งโปรแกรมเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโค้ดโปรแกรม MP LAB

```
/*****************************************************/
#include <p30f4011.h>                          // dsPIC30F4011 MPU Register
#include "stdio.h"                             // Used "sprintf" Function
#include "uart.h"
/* Setup Configuration For ET-BASE dsPIC30F4011 */
_FOSC(CSW_FSCM_OFF & XT_PLL16);
_FWDT(WDT_OFF);                                                    _FBORPOR(PBOR_OFF & PWRT_64 &
MCLR_EN);                                              _FGS(CODE_PROT_OFF);

#define CodeStop      0xFF      // Key Code บอกจบคำสั่งหรือข้อมูล
//#define MY_ID        0x02     //กำหนดเลขหมายตัวเองหมายเลข '2'
#define MY_ID         0x01     //กำหนดเลขหมายตัวเองหมายเลข '1'
//---- LED (Relay#0)
#define TRIS_Rly0 TRISFbits.TRISF0
#define Relay0  LATFbits.LATF0
//---- LED (Relay#1)
#define TRIS_Rly1 TRISFbits.TRISF1
#define Relay1  LATFbits.LATF1
// Character LCD Interface Pins
#define TRIS_DATA_PIN_4    TRISBbits.TRISB0                // Direction D4
#define TRIS_DATA_PIN_5    TRISBbits.TRISB1                // Direction D5
#define TRIS_DATA_PIN_6    TRISBbits.TRISB2                // Direction D6
#define TRIS_DATA_PIN_7    TRISBbits.TRISB3                // Direction D7
#define TRIS_RS          TRISBbits.TRISB4                  // Direction RS
#define TRIS_E           TRISBbits.TRISB5                  // Direction E
#define DATA_PIN_4       LATBbits.LATB0                    // RB0 = D4 LCD
#define DATA_PIN_5       LATBbits.LATB1                    // RB1 = D5 LCD
#define DATA_PIN_6       LATBbits.LATB2                    // RB2 = D6 LCD
#define DATA_PIN_7       LATBbits.LATB3                    // RB3 = D7 LCD
#define RS_PIN           LATBbits.LATB4                    // RB4 = RS LCD
#define E_PIN            LATBbits.LATB5                    // RB5 = E  LCD
/* Display ON/OFF Control */
#define DON              0x0F                              // Display on
#define DOFF             0x0B                              // Display off
#define CURSOR_ON        0x0F                              // Cursor on
#define CURSOR_OFF       0x0D                              // Cursor off
#define BLINK_ON         0x0F                              // Cursor Blink
#define BLINK_OFF        0x0E                              // Cursor No Blink
/* Cursor or Display Shift */
#define SHIFT_CUR_LEFT    0x13                             // Cursor shifts to the left
#define SHIFT_CUR_RIGHT   0x17                             // Cursor shifts to the right
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define SHIFT_DISP_LEFT    0x1B                    // Display shifts to the left
#define SHIFT_DISP_RIGHT   0x1F                    // Display shifts to the right
/* Function Prototypes */
void Initial_4bitLCD(void);                        // Initial LCD Interface
void SetCGRamAddr(unsigned char);
void SetDDRamAddr(unsigned char);                  // Set Cursor Address
void WriteCmdLCD(unsigned char);                   // Write Command
void WriteDataLCD(unsigned char);                  // Write Data
void PutsLCD(unsigned char*);                      // Print Message
void Delay_tW_LCD(void);                           // Enable Pulse Delay
void Busy_LCD(void);                               // Wait LCD Busy
void Delay(unsigned long int);                     // Delay Time Function
//---- For DS1820
//#define TRIS_DS1820    TRISEbits.TRISE0          // Direction E0
//#define DOUT_DS1820    LATEbits.LATE0            // OUT
//#define DIN_DS1820     PORTEbits.RE0             // IN
#define TRIS_DS1820    TRISFbits.TRISF2            // Direction F2
#define DOUT_DS1820    LATFbits.LATF2              // OUT
#define DIN_DS1820     PORTFbits.RF2               // IN
unsigned char tempresbit(void);
void tempwrbyte(unsigned char dat);
char temprdbyte(void);
char rdtemp(void);
void Forward(void);
void Backward(void);
char buf_Tempc[20]={'\0','\0','\0','\0','\0'       // Set buffer for display Temp
                    ,'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'};
//------------------------:Global variables
unsigned char Buf[80];          // Received Command/data is stored in array Buf
unsigned char data_tmp[80];
unsigned char ch;
unsigned char c_buf=0;
unsigned char addr_back;
unsigned char Read_black;
unsigned char Flag;
//=============================================================
//------------------------:Main Program:----------------------
//=============================================================
int main(void)
{
    int i,j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        unsigned char r_dat = 0;


        Flag=0;
        Read_black = 0;
 Initial_4bitLCD();
 Uart1_Init();
 TRIS_Rly0 = 0;
 Relay0 = 0;
 TRIS_Rly1 = 0;
 Relay1 = 0;
SetDDRamAddr(0x00);
// Start Cursor Line-1
 PutsLCD((unsigned char *)" PROGRAM LMSAR  ");
   SetDDRamAddr(0x40);
// Start Cursor Line-2
   PutsLCD((unsigned char *)"  DS1820 TEMP   ");
   Delay(2000000);
    for(;;)
        {
                        j = rdtemp();
                                if(j&0x01)
                                {
                                i =5;
                                j >>= 1;
                                }
                              else
                                {
                                 i = 0;
                                 j >>= 1;
                                }
        sprintf(buf_Tempc,"T=+%d.%d%cC",j,i,0xDF);
 SetDDRamAddr(0x00);
        PutsLCD((unsigned char *)buf_Tempc);
        Delay(20000);
    if (ch == CodeStop && Buf[1]== MY_ID)
        {
            ch = 0;
     c_buf=0;
      if (Buf[2]==0xEE
       {
              Flag = 0x01;
        Read_black = 0x01;
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else  if (Flag != 0x00)
  {
            Read_black = 0x00;
           }
           switch (Flag)
            {
                    case 0x00 : Forward();
                    Flag = !Flag;
                    break;
                    case 0x01 : Backward();
                    Flag = !Flag;
                            break;
                    default   : break;
     }
  }
  if (ch == CodeStop && Buf[1]!= MY_ID)
        {
                ch = 0;
          c_buf=0;
         }
                          }
 return 0;
}
//==================== Sub routine ======================
void Forward()
{
          unsigned char i=1;
                  addr_back = Buf[0];
                          while(BusyUART1());
               WriteUART1(Buf[i]);
           while(Buf[i]!=0xFF)
            {
                    i++;
                    while(BusyUART1());
               WriteUART1(Buf[i]);
                  }
}
void Backward()
{
          unsigned char relay_dat;
  int i,j;
      if(Read_black==0x01)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            j = rdtemp();
if(j&0x01)
  {
                    i =5;
                    j >>= 1;
                    }
        else
                    {
                            i = 0;
                            j >>= 1;
                    }
              data_tmp[0]= MY_ID;
              data_tmp[1]= Buf[0];
              data_tmp[2]=(unsigned char) j;
              data_tmp[3] =(unsigned char) i;
               if( 0x02 & Buf[3])
               { //แสดงว่าให้อ่านอย่างเดียว
                        relay_dat = Relay1;
                        relay_dat <<= 1;
 } else
                         {
                          if(0x20 & Buf[3])
                            Relay1 = 1;
                            else
                            Relay1 = 0; //OFF Relay0
                         }
                        if( 0x01 & Buf[3])
             {

                        relay_dat |= Relay0;
                       } else
                           {
                            if(0x10 & Buf[3])
                               Relay0 = 1;
                               else
                               Relay0 = 0;
                           }
                    data_tmp[4]= relay_dat;
                   data_tmp[5]= 0xFF
            i=0;
           while (data_tmp[i]!=0xFF)
           {
             while(BusyUART1());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                WriteUART1(data_tmp[i]);
        i++;
                }
                while(BusyUART1());
                WriteUART1(0xFF);
               }
               else
               {
                while(BusyUART1());
                WriteUART1(MY_ID);
                while(BusyUART1());
                WriteUART1(addr_back);
        i=2;

        /*
        if(addr_back==0x00)
         { SetDDRamAddr(0x00);
   PutsLCD((unsigned char *)" addr = 0  ");
        }
        else
         { SetDDRamAddr(0x00);
   PutsLCD((unsigned char *)" addr = 1  ");
        }
        */
              while (Buf[i]!=0xFF)
               {
                while(BusyUART1());
                WriteUART1(Buf[i]);
        i++;
               }
                while(BusyUART1());
                WriteUART1(0xFF);
              }
}
//================ GENERAL SUBROUTINE=====================
//=======================================================
// UART1 Receiver Interrupt Service Routine
void _ISR _U1RXInterrupt(void)
{
 _U1RXIF = 0;   // Clear interrupt flag RX

 // Read the receive buffer till atleast one or more character can be read
 while (DataRdyUART1()) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ch = ReadUART1();    // Read buffer rx
    //putcUART1(ch);      // print character rx
    Buf[c_buf++] = ch;   // wirte to buf
   }
 }
 //------------------------------------------------------:U1TX ISR
 // UART1 Transmitter Interrupt Service Routine
 void _ISR _U1TXInterrupt(void)
 {
  _U1TXIF = 0; // Clear interrupt flag TX
 }
 //------------------------------------------------------:Delay MS
 // Delay 1 ms (XT w/PLL 4x)
 void Delay_MS(unsigned int ms)
 {
  unsigned int i;
  for (; ms>0; ms--)
    for (i=0; i<728; i++)
      Nop();              // delay 1 mch cycle
 }


 //------------------------------------------------------:Uart1_Init
 // Initialize UART1
 void Uart1_Init(void)
 {
  unsigned int U1MODEvalue, U1STAvalue, BaudRate;
  CloseUART1();                          // Close UART1
  // Config Interrupt UART1
  ConfigIntUART1(UART_RX_INT_EN &        // Receive interrupt enabled
           UART_RX_INT_PR6 &       // Priority RX interrupt 6
           UART_TX_INT_EN &        // transmit interrupt enabled
           UART_TX_INT_PR2 );      // Priority TX interrupt 2
  U1MODEvalue = UART_EN &              // Module enable
          UART_IDLE_CON &         // Work in IDLE mode
          //UART_RX_TX &            // Communication through the normal pins
          //UART_ALTRX_ALTTX &      // Communication through ALT pins
          UART_DIS_WAKE &         // Disable Wake-up on START bit Detect during SLEEP Mode bit
          UART_DIS_LOOPBACK &     // Loop back disabled
          UART_DIS_ABAUD &        // Input to Capture module from ICx pin
          UART_NO_PAR_8BIT &      // no parity 8 bit
          UART_1STOPBIT;          // 1 stop bit
 U1STAvalue          = UART_INT_TX_BUF_EMPTY &  // Interrupt on TXBUF becoming empty
          UART_TX_PIN_NORMAL &    // UART TX pin operates normally
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        UART_INT_RX_CHAR &      // Interrupt on every char received
        UART_ADR_DETECT_DIS &   // address detect disable
        UART_RX_OVERRUN_CLEAR;  // Rx buffer Over run status bit clear
      // ET-BASE dsPIC30F4011 Hardware Board
                                // XTAL = 7.3728MHz
                                // Fosc = 7.3728 MHz x 16 = 117.9648 MHz
                                // Fcy(UART) = Fosc / 4
                                //       = 117.9648 / 4 = 29.4912 MHz
                                // U1BRG = [Fcy/(16xBaud)]-1
                                //     = [29.4912 MHz / (16x9600)] - 1
                                //     = 191 = BFH   // BaudRate 9600 pbs
                                //  BaudRate = 3070; //for 600 pbs
  BaudRate = 1535;    //  BaudRate 1200 pbs

  // Open UART1
  OpenUART1(U1MODEvalue, U1STAvalue, BaudRate);
}
//-----------------------------------------------------:Uart1 PrintStr
// print string to uart1
void Uart1_PrintStr(unsigned char *str_uart)
{
  putsUART1 ((unsigned int *)str_uart);
  while (BusyUART1());       // wait for transmission to complete
}
//--------------- SUBrountine ------------------------
void Initial_4bitLCD(void)
{
  TRIS_DATA_PIN_4 = 0;
  TRIS_DATA_PIN_5 = 0;
  TRIS_DATA_PIN_6 = 0;
  TRIS_DATA_PIN_7 = 0;
  TRIS_RS = 0;
  TRIS_E = 0;
  RS_PIN = 0;                                                    //
Instruction Select
  E_PIN = 0;                                                     //
Disable Strobe
  Delay(50000);                                                  //
Power-ON Delay (minimum of 15ms)
    WriteCmdLCD(0x33);                                           // Initial 4 Bit
Interface Format
  WriteCmdLCD(0x32);
  WriteCmdLCD(0x28);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  (DL=0 4-Bit,N=1 2 Line,F=0 5X7)
  WriteCmdLCD(0x0C);
  (Entry Display,Cursor off,Cursor not Blink)
   WriteCmdLCD(0x06);
  (I/D=1 Increment,S=0 Cursor Shift)
   WriteCmdLCD(0x01);
 Display  (Clear Display,Set DD RAM Address=0)
 }
 void SetCGRamAddr(unsigned char address)
 {
  RS_PIN = 0;                                                    //
 Select Instruction
  DATA_PIN_7 = 0;
 CGRam ADDR  DATA_PIN_6 = 1;
 CGRam ADDR
  DATA_PIN_5 = ((address & 0x20)>>5);
  DATA_PIN_4 = ((address & 0x10)>>4);
  E_PIN = 1;
  Delay_tW_LCD();
 Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
  RS_PIN = 0;                                                    //
 Select Instruction
  DATA_PIN_7 = ((address & 0x08)>>3);
  DATA_PIN_6 = ((address & 0x04)>>2);
  DATA_PIN_5 = ((address & 0x02)>>1);
  DATA_PIN_4 = ((address & 0x01)>>0);
  E_PIN = 1;                                                     //
 Strobe Low Nibble
  Delay_tW_LCD();                                                // Enable Pulse
 Delay
  E_PIN = 0;
  Busy_LCD();
 }
 void SetDDRamAddr(unsigned char address)
 {
  RS_PIN = 0;                                                    // Select
Instruction
  DATA_PIN_7 = 1;                                                // DB7 Must be 1
For Setting DDRam ADDR
  DATA_PIN_6 = ((address & 0x40)>>6);
  DATA_PIN_5 = ((address & 0x20)>>5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  DATA_PIN_4 = ((address & 0x10)>>4);
  E_PIN = 1;                                                                    //
 Strobe High Nibble
  Delay_tW_LCD();
 // Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
  RS_PIN = 0;
 // Select Instruction
  DATA_PIN_7 = ((address & 0x08)>>3);
  DATA_PIN_6 = ((address & 0x04)>>2);
  DATA_PIN_5 = ((address & 0x02)>>1);
  DATA_PIN_4 = ((address & 0x01)>>0);
  E_PIN = 1;
 // Strobe Low Nibble
  Delay_tW_LCD();
 // Enable Pulse Delay
  E_PIN =0;
  Busy_LCD();
 }
 void WriteCmdLCD(unsigned char cmd)
 {
  RS_PIN = 0;
 // Select Instruction
  DATA_PIN_7 = ((cmd & 0x80)>>7);
  DATA_PIN_6 = ((cmd & 0x40)>>6);
  DATA_PIN_5 = ((cmd & 0x20)>>5);
  DATA_PIN_4 = ((cmd & 0x10)>>4);
  E_PIN = 1;                                                                    //
 Strobe High Nibble
  Delay_tW_LCD();
 // Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
  RS_PIN = 0;
 // Select Instruction
  DATA_PIN_7 = ((cmd & 0x08)>>3);
  DATA_PIN_6 = ((cmd & 0x04)>>2);
  DATA_PIN_5 = ((cmd & 0x02)>>1);
  DATA_PIN_4 = (cmd & 0x01);
  E_PIN = 1;                                                                    //
Strobe Low Nibble
  Delay_tW_LCD();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 // Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
  Delay(50000);                                                    //
 1.64mS Delay
 }
 void WriteDataLCD(unsigned char data)
 {
  RS_PIN = 1;                                                      //
 Select Data
  DATA_PIN_7 = ((data & 0x80)>>7);
  DATA_PIN_6 = ((data & 0x40)>>6);
  DATA_PIN_5 = ((data & 0x20)>>5);
  DATA_PIN_4 = ((data & 0x10)>>4);
  E_PIN = 1;                                                       //
 Strobe High Nibble
  Delay_tW_LCD();
 // Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
  RS_PIN = 1;                                                      //
 Select Data
  DATA_PIN_7 = ((data & 0x08)>>3);
  DATA_PIN_6 = ((data & 0x04)>>2);
  DATA_PIN_5 = ((data & 0x02)>>1);
  DATA_PIN_4 = (data & 0x01);
  E_PIN = 1;                                                       //
 Strobe Low Nibble
  Delay_tW_LCD();
 // Enable Pulse Delay
  E_PIN = 0;
  Busy_LCD();
 }
 void PutsLCD(unsigned char* buffer)
 {
  while(*buffer != '\0')
 // Loop Until End of Message
  {
   WriteDataLCD(*buffer);
 // Write Character to LCD
   buffer++;
  }
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Delay_tW_LCD(void)
// Enable Pulse Delay
{
  int i;
  for(i=0;i<18;i++);
}

void Busy_LCD(void)                                                    // 40uS Delay
{
  unsigned int i;
  for(i=0;i<1800;i++);
}
/***********************/
/* Delay Time Function */
/*    1-4294967296     */
/***********************/
void Delay(unsigned long int count1)
{
  while(count1 > 0) {count1--;}
// Loop Decrease Counter
}
//***Function for DS1820 *********************************
unsigned char tempresbit()
{
            int j;
 TRIS_DS1820 = 0;  // set output
 DOUT_DS1820 = 0;  // Send LOW
 for(j=1764;j>0;j--)  //Delay 600 uS
    Nop();
 DOUT_DS1820 = 1;        /* end reset pulse */
 TRIS_DS1820 = 1;   // กำหนดเป็น INPUT
 for(j=176;j>0;j--)
    Nop();         // Delay 60 uS
 if(DIN_DS1820)
  return 0;
 for(j=1764;j>0;j--)  //Delay 600 uS
    Nop();
 return 1;
}
void tempwrbyte(unsigned char dat)
{
unsigned char i,j;
for(i=0;i<8;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
  TRIS_DS1820 = 0; // set OUTPUT
  DOUT_DS1820 = 0;
  for(j=0;j<44;j++)
    Nop();       // Delay 15 uS
  if(dat & 0x01)
  DOUT_DS1820=1;
  else
  DOUT_DS1820=0;

 for(j=176;j>0;j--)
    Nop();         // Delay 60 uS
 DOUT_DS1820 = 1;
 dat >>= 1;
  for(j=0;j<30;j++)
    Nop();       // Delay 10 uS
 }
}

char temprdbyte()
{
 unsigned char dat,i,j;
 for(i=0;i<8;i++)
 {
  dat >>= 1;
  TRIS_DS1820 = 0;
  DOUT_DS1820 = 0;
  for(j=0;j<44;j++)
   Nop();       // Delay 15 uS
  DOUT_DS1820 = 1;
     TRIS_DS1820 = 1;
  if(DIN_DS1820)
   dat |= 0x80;  // input = 1
   else
   dat &= 0x7F; // input = 0
   for(j=176;j>0;j--)
    Nop();         // Delay 60 uS

 }
 return dat;
}

char rdtemp()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 char dat;
 int j;

 if(tempresbit())
 {
  tempwrbyte(0xCC); //Skip rom
  tempwrbyte(0x44); //Temp Convert
   for(j=1470;j>0;j--)
    Nop();        // Delay 500 uS
    tempresbit();
    tempwrbyte(0xCC);
    tempwrbyte(0xBE);
        dat = temprdbyte();
   return dat;
}
return 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโค้ดโปรแกรม Visual studio

```
/*****************************************************/
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.IO.Ports;          //ใส่เพื่อใช้งาน RS-232 Port ได้
using System.IO;
using System.Windows.Forms;
namespace Test_txrx
{
  public partial class Form1 : Form
  {
    int counter = new int();
    int end_counter = new int();
    byte[] bbyte = new byte[20];  //สำหรับไว้รับข้อมูล
    byte[] tbyte = new byte[20];  // สำหรับไว้ส่งข้อมูล
    int send_num;
    String client;
    String data_out;
    String inbox;
    //  int i;
    int count;
    public Form1()
    {
      InitializeComponent();
    }
    private void Form1_Load(object sender, EventArgs e)
    {
      count = 0;
      serialPort1.DataReceived += new SerialDataReceivedEventHandler(serialPort1_DataReceived);
      //กำหนดช่องเลือกการติดต่อ
      string[] comStr = SerialPort.GetPortNames();
      int i = 0;
      foreach (string port in comStr)
      {
        comboBox1.Items.Add(comStr[i]);
        i++;
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //comboBox1.SelectedIndex = 0;
        //ให้ช่องว่างต่างๆยังไม่มีข้อมูลในเบื้องต้น
        textBox1.Text = "0";
        textBox2.Text = "0";
        textBox3.Text = "0";
        textBox4.Text = "0";
        textBox5.Text = "0";
        textBox6.Text = "0";
        textBox7.Text = "";
        textBox8.Text = "";
        textBox9.Text = "";
        textBox10.Text = "";
        textBox11.Text = "";
        textBox12.Text = "";
        //ปิดปุ่มที่ยังไม่ใช่ก่อน
        button3.Enabled = false; //ปิดปุ่ม "ส่ง"
        button4.Enabled = false; //ปิดปุ่ม "clear"
        button5.Enabled = false; //ปิดปุ่ม "Relay0 on"
        button6.Enabled = false; //ปิดปุ่ม "Relay0 off"
        button7.Enabled = false; //ปิดปุ่ม "ตัววัดที่ 1"
        button8.Enabled = false; //ปิดปุ่ม "ตัววัดที่ 2"
        button9.Enabled = false; //ปิดปุ่ม "อ่านอย่างเดียว"
        button10.Enabled = false; //ปิดปุ่ม "Relay0 on"
        button11.Enabled = false; //ปิดปุ่ม "Relay0 off"
        send_num = 0;       //กำหนดค่าเริ่มต้นของตัวนับคำสั่งเป็นศูนย์
        counter = 0;        //กำหนดค่าเริ่มต้นนับเป็นศูนย์
        end_counter = 20;    //กำหนดให้รับข้อมูลจำนวน 20 ไบท์
        timer1.Enabled = false;
        timer2.Enabled = false;
        timer3.Enabled = true;
        data_out = "";
        inbox = "";

    }
    private void serialPort1_DataReceived(object sender, SerialDataReceivedEventArgs e)
    {
        while (counter < end_counter)
        {
            bbyte[counter] = (byte)serialPort1.ReadByte();
            counter = counter + 1;
        }
    }
    private void button1_Click(object sender, EventArgs e) //ปุ่มคอนเน็ก
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    /*
    if (comboBox1.Items.Count > 0)
    {
        if (!serialPort1.IsOpen)
        {
            serialPort1.PortName = comboBox1.Text;
            serialPort1.Open();
        }*/
    serialPort1.Open();
    button4.Enabled = true;
    button7.Enabled = true;
    button8.Enabled = true;
    // }
}
private void button3_Click(object sender, EventArgs e)//ปุ่มกดส่ง
{
    count++;
    for (int i = 0; i < 20; i++)
    {
        tbyte[i] = 0x00;
        bbyte[i] = 0x00;
    }
    //ใส่ค่าคำสั่งลงไป
    tbyte[0] = byte.Parse(textBox1.Text);
    tbyte[1] = byte.Parse(textBox2.Text);
    tbyte[2] = byte.Parse(textBox3.Text);
    tbyte[3] = byte.Parse(textBox4.Text);
    tbyte[4] = byte.Parse(textBox5.Text);
    tbyte[5] = byte.Parse(textBox6.Text);
    serialPort1.Write(tbyte, 0, send_num);    //ส่งคำสั่ง
    counter = 0;
    timer1.Enabled = true;    //สั่งเปิด Timer1 เพื่อแสดงผลข้อมูลที่รับเข้ามา
}
private void button2_Click(object sender, EventArgs e)//ปุ่มออก
{
    timer1.Enabled = false;
    serialPort1.Close();
    this.Close();
}
private void timer1_Tick(object sender, EventArgs e)
{
    int num = 19;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while ((bbyte[num] != 0xFF) && (num != 0))
{
   num--;
}
if (num > 0)
{
   textBox7.Text = bbyte[num - 5].ToString();
   textBox8.Text = bbyte[num - 4].ToString();
   textBox9.Text = bbyte[num - 3].ToString();
   textBox10.Text = bbyte[num - 2].ToString();
   textBox11.Text = bbyte[num - 1].ToString();
   textBox12.Text = bbyte[num].ToString();
   textBox13.Text = bbyte[num - 3].ToString() + "." + bbyte[num - 2].ToString() + " 'C";
   //if (counter == 6) counter = 0;   //ใส่เพื่อให้เป็นค่าเริ่มต้นศูนย์ทุกครั้งเพื่อการรับข้อมูลครบ
  if ((tbyte[1] == 0x01) && (tbyte[2] == 0xEE))
  {
      //client = "Client 1";
     if (bbyte[num - 1] == 0x03)
     {
        textBox14.Text = " ON";
        textBox15.Text = " ON";
     }
     else if (bbyte[num - 1] == 0x00)
     {
        textBox14.Text = "OFF";
        textBox15.Text = "OFF";
     }
     else if (bbyte[num - 1] == 0x01)
     {
        textBox14.Text = " ON";
        textBox15.Text = "OFF";
     }
     else if (bbyte[num - 1] == 0x02)
     {
        textBox14.Text = "OFF";
        textBox15.Text = " ON";
     }
  }
 else if ((tbyte[1] == 0x01) && (tbyte[2] == 0x02))
  {
     //client = "Client 2";
     if (bbyte[num - 1] == 0x03)
     {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                textBox14.Text = " ON";
                textBox15.Text = " ON";
            }
            else if (bbyte[num - 1] == 0x00)
            {
                textBox14.Text = "OFF";
                textBox15.Text = "OFF";
            }
            else if (bbyte[num - 1] == 0x01)
            {
                textBox14.Text = " ON";
                textBox15.Text = "OFF";
            }
            else if (bbyte[num - 1] == 0x01)
            {
                textBox14.Text = "OFF";
                textBox15.Text = " ON";
            }
        }
    }
    if ((tbyte[1] == 0x01) && (tbyte[2] == 0xEE))
    {
        client = "Client 1";
    }
    else
        client = "Client 2";
    data_out += client + ":" + label15.Text + "  " + label11.Text + "  " + bbyte[num - 3].ToString() + "." + bbyte[num - 2].ToString() + " 'C"
+ "  " + "Relay 1 :" + textBox14.Text + "  " + "Relay 2 :" + textBox15.Text + "\r\n";
    inbox = client + ":" + bbyte[num - 3].ToString() + "." + bbyte[num - 2].ToString() + " 'C" + "  " + "R1:" + textBox14.Text + " " +
"R2:" + textBox15.Text;
    listBox1.Items.Add(inbox);
    inbox = "";
    timer1.Enabled = false;
    //timer1.Enabled = false;
}
private void button4_Click(object sender, EventArgs e) //ปุ่มเคลียร์
{
    for (int i = 0; i < 20; i++)
    {
        tbyte[i] = 0x00;
        bbyte[i] = 0x00;
    }
    timer2.Enabled = true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    textBox1.Text = "0";
    textBox2.Text = "0";
    textBox3.Text = "0";
    textBox4.Text = "0";
    textBox5.Text = "0";
    textBox6.Text = "0";
    textBox14.Text = "0";
    textBox15.Text = "0";
    button7.Enabled = true;
    button8.Enabled = true;
    button9.Enabled = false;
    button5.Enabled = false;
    button6.Enabled = false;
    button3.Enabled = false;
    button10.Enabled = false;
    button11.Enabled = false;
}
private void button5_Click(object sender, EventArgs e)  //Relay#0 On
{
   byte R0_And = 0xFE;    //ค่าบิทที่ศูนย์เป็น"0"
   byte R0_Or = 0x10;    //ค่าบิทที่สี่เป็น"1"ช่องที่ศูนย์มีการสั่งให้ ON

   if (tbyte[2] == 0xEE)
   {
      tbyte[3] &= R0_And;   //กำหนดให้ช่องที่ศูนย์มีสถานะ ON
      tbyte[3] |= R0_Or;
      textBox4.Text = tbyte[3].ToString();
      tbyte[4] = 0xFF;
      textBox5.Text = tbyte[4].ToString();
      button5.Enabled = false;
      button6.Enabled = false;
   }
   else
   {
      tbyte[4] &= R0_And;   //กำหนดให้ช่องที่ศูนย์มีสถานะ ON
      tbyte[4] |= R0_Or;
      textBox5.Text = tbyte[4].ToString();
      tbyte[5] = 0xFF;
      textBox6.Text = tbyte[5].ToString();
      button5.Enabled = false;
      button6.Enabled = false;
   }
  button9.Enabled = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    button3.Enabled = true;
  }

  private void button6_Click(object sender, EventArgs e) //Relay#0 Off
  {
    byte R0_And = 0xEE; //ค่าสำหรับกรณี ช่องที่ศูนย์มีการสั่งให้ Off
    if (tbyte[2] == 0xEE)
    {
      tbyte[3] &= R0_And;   //กำหนดให้ช่องที่ศูนย์มีสถานะ Off
      textBox4.Text = tbyte[3].ToString();
      tbyte[4] = 0xFF;
      textBox5.Text = tbyte[4].ToString();
      button5.Enabled = false;
      button6.Enabled = false;
    }
    else
    {
      tbyte[4] &= R0_And;   //กำหนดให้ช่องที่ศูนย์มีสถานะ Off
      textBox5.Text = tbyte[4].ToString();
      tbyte[5] = 0xFF;
      textBox6.Text = tbyte[5].ToString();
      button5.Enabled = false;
      button6.Enabled = false;
    }
    button9.Enabled = false;
    button3.Enabled = true;
  }
  private void button9_Click(object sender, EventArgs e)  //กำหนดอ่านอุณหภูมิอย่างเดียว
  {

    if (tbyte[2] == 0xEE)
    {
      tbyte[3] = 0x0F;  //สี่บิทล่างเป็นหนึ่งหมดแสดงว่าให้อ่านสถานะรีเลย์ทั้งสี่ตัว
      textBox4.Text = tbyte[3].ToString();
      tbyte[4] = 0xFF;
      textBox5.Text = tbyte[4].ToString();
    }
    else
    {
      tbyte[4] = 0x0F;  //สี่บิทล่างเป็นหนึ่งหมดแสดงว่าให้อ่านสถานะรีเลย์ทั้งสี่ตัว
      textBox5.Text = tbyte[4].ToString();
      tbyte[5] = 0xFF;
      textBox6.Text = tbyte[5].ToString();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    button3.Enabled = true;
    button5.Enabled = false;
    button6.Enabled = false;
    button9.Enabled = false;
    button10.Enabled = false;
    button11.Enabled = false;
}
private void button7_Click(object sender, EventArgs e)  //เลือกตัวที่1
{
    send_num = 5;        //กำหนดจำนวนไบท์ของคำสั่งเป็น 5 ไบท์
    tbyte[0] = 0x00;
    textBox1.Text = tbyte[0].ToString();
    tbyte[1] = 0x01;
    textBox2.Text = tbyte[1].ToString();
    tbyte[2] = 0xEE;
    textBox3.Text = tbyte[2].ToString();
    //เปิดใช้งานปุ่มต่อมา
    button5.Enabled = true;
    button6.Enabled = true;
    button9.Enabled = true;
    button10.Enabled = true;
    button11.Enabled = true;
    button7.Enabled = false;
    button8.Enabled = false;
}
private void button8_Click(object sender, EventArgs e) //เลือกตัวที่2
{
    send_num = 6;        //กำหนดจำนวนไบท์ของคำสั่งเป็น 6 ไบท์
    tbyte[0] = 0x00;
    textBox1.Text = tbyte[0].ToString();
    tbyte[1] = 0x01;
    textBox2.Text = tbyte[1].ToString();
    tbyte[2] = 0x02;
    textBox3.Text = tbyte[2].ToString();
    tbyte[3] = 0xEE;
    textBox4.Text = tbyte[3].ToString();
    //เปิดใช้งานปุ่มต่อมา
    button5.Enabled = true;
    button6.Enabled = true;
    button9.Enabled = true;
    button10.Enabled = true;
    button11.Enabled = true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    button7.Enabled = false;
    button8.Enabled = false;
  }
  private void timer2_Tick(object sender, EventArgs e) //แสดงผลอุณหภูมิ,รีเลย์
  {
    //ลบข้อมูลเก่าในที่เก็บข้อมูลที่รับมา
    for (int i = 0; i < 20; i++)
    {
      tbyte[i] = 0x00;
      bbyte[i] = 0x00;
    }
    textBox7.Text = bbyte[0].ToString();
    textBox8.Text = bbyte[1].ToString();
    textBox9.Text = bbyte[2].ToString();
    textBox10.Text = bbyte[3].ToString();
    textBox11.Text = bbyte[4].ToString();
    textBox12.Text = bbyte[5].ToString();
    textBox13.Text = bbyte[2].ToString() + "." + bbyte[3].ToString() + " 'C ";
    timer2.Enabled = false;
    counter = 0;
  }
  private void button10_Click(object sender, EventArgs e)//ปุ่ม Relay#1 On
  {
    byte R1_And = 0xFD;    //ค่าบิทที่หนึ่งเป็น"0"
    byte R1_Or = 0x20;     //ค่าบิทที่สี่เป็น"2"ช่องที่หนึ่งมีการสั่งให้ ON
    if (tbyte[2] == 0xEE) //กรณีสั่ง ไปที่ตัวที่หนึ่ง
    {
      tbyte[3] &= R1_And;    //กำหนดให้ช่องที่หนึ่งมีสถานะ ON
      tbyte[3] |= R1_Or;
      textBox4.Text = tbyte[3].ToString();
      tbyte[4] = 0xFF;
      textBox5.Text = tbyte[4].ToString();
      button11.Enabled = false;
      button10.Enabled = false;
    }
    else  //กรณีสั่ง ไปที่ตัวที่สอง
    {
      tbyte[4] &= R1_And;    //กำหนดให้ช่องที่หนึ่งมีสถานะ ON
      tbyte[4] |= R1_Or;
      textBox5.Text = tbyte[4].ToString();
      tbyte[5] = 0xFF;
      textBox6.Text = tbyte[5].ToString();
      button11.Enabled = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        button10.Enabled = false;
    }
    button9.Enabled = false;
    button3.Enabled = true;
}

private void button11_Click(object sender, EventArgs e) //ปุ่ม Relay#1 Off
{
    byte R1_And = 0xDD; //ค่าสำหรับกรณี ช่องที่ศูนย์มีการสั่งให้ Off
    if (tbyte[2] == 0xEE) //กรณีสั่งไปตัววัดอุณหภูมิตัวที่หนึ่ง
    {
        tbyte[3] &= R1_And;    //กำหนดให้ช่องที่ศูนย์มีสถานะ Off
        textBox4.Text = tbyte[3].ToString();
        tbyte[4] = 0xFF;
        textBox5.Text = tbyte[4].ToString();
        button10.Enabled = false;
        button11.Enabled = false;
    }
    else //กรณีสั่งไปตัววัดอุณหภูมิตัวที่สอง
    {
        tbyte[4] &= R1_And;    //กำหนดให้ช่องที่ศูนย์มีสถานะ Off
        textBox5.Text = tbyte[4].ToString();
        tbyte[5] = 0xFF;
        textBox6.Text = tbyte[5].ToString();
      button10.Enabled = false;
        button11.Enabled = false;
    }
    button9.Enabled = false;
    button3.Enabled = true;
}
private void timer3_Tick(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToLongTimeString();
    label5.Text = DateTime.Now.ToLongDateString();
}
private void button12_Click(object sender, EventArgs e)
{
    saveFileDialog1.DefaultExt = "*.txt";
    saveFileDialog1.AddExtension = false;
    saveFileDialog1.Filter = "เท็กซ์ไฟล์ | *.txt";
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        string path = saveFileDialog1.FileName;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

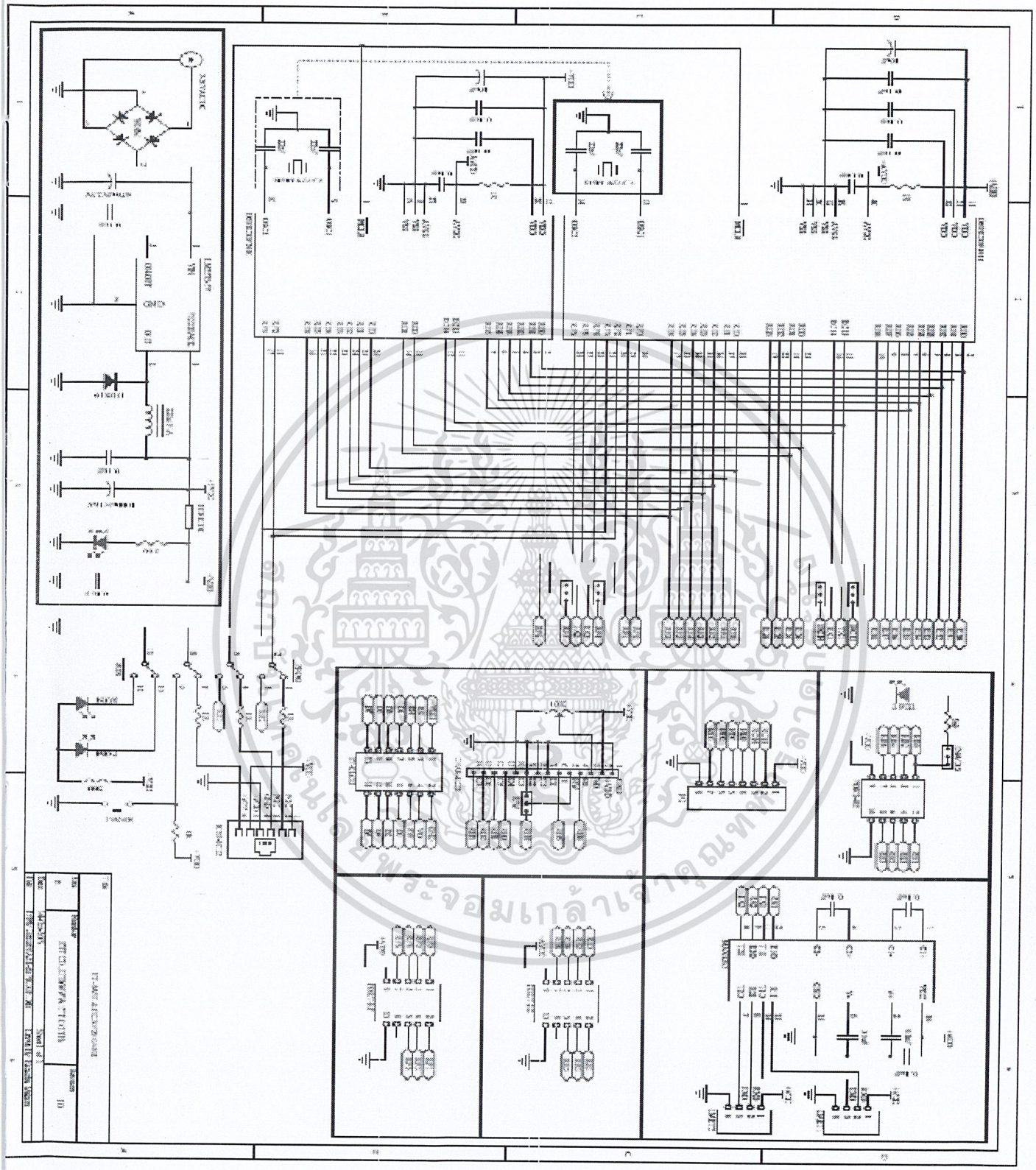
```
            StreamWriter sw = new StreamWriter(path);
             {
                String ss = data_out.Substring(0, 73);
                sw.WriteLine(ss);
                String sss = data_out.Remove(73);
                data_out = sss;
                ss = data_out.Substring(0, 73);
                sw.WriteLine(ss);
                int cl = data_out.IndexOf("cl");
                cl = data_out.IndexOf("cl");
                count = 0;
             }
            // sw.Write("    " + label1.Text);
            //  sw.Write("    " + label5.Text);
             // sw.Write("    " + textBox13.Text;*/
            sw.WriteLine(data_out);
            data_out = "";
            sw.Close();
        }
    }
    private void button13_Click(object sender, EventArgs e)
    {
        ClearListBox();
    }
    private void ClearListBox()
    {
        listBox1.Items.Clear();
    }
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

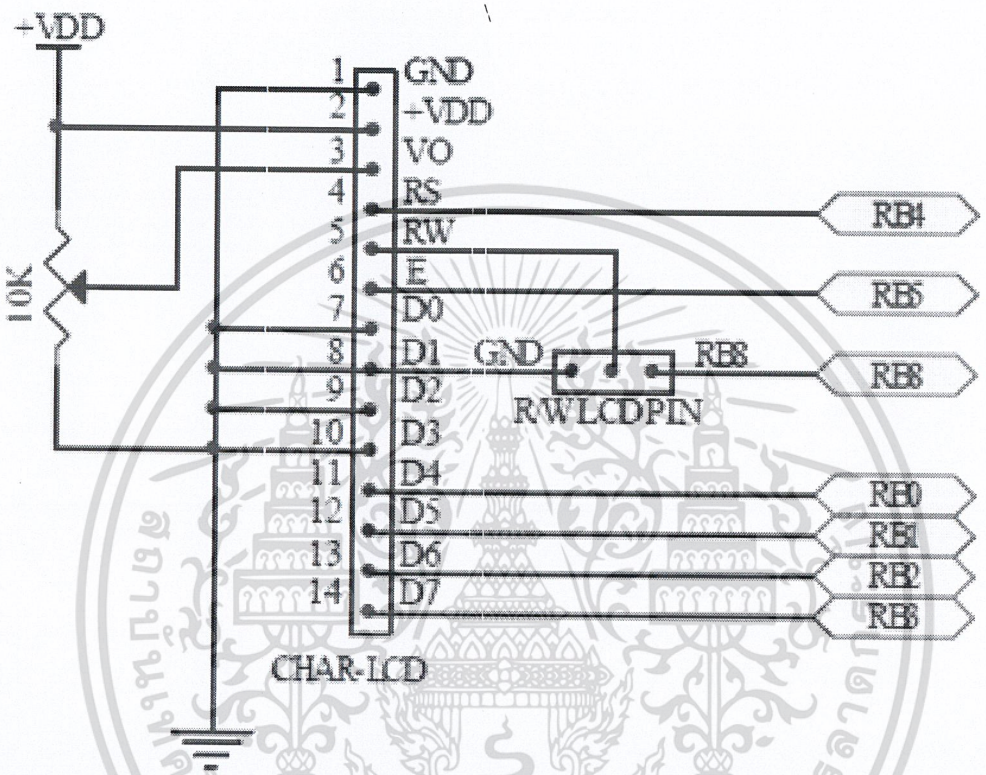
วงจรรวมของโครงงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



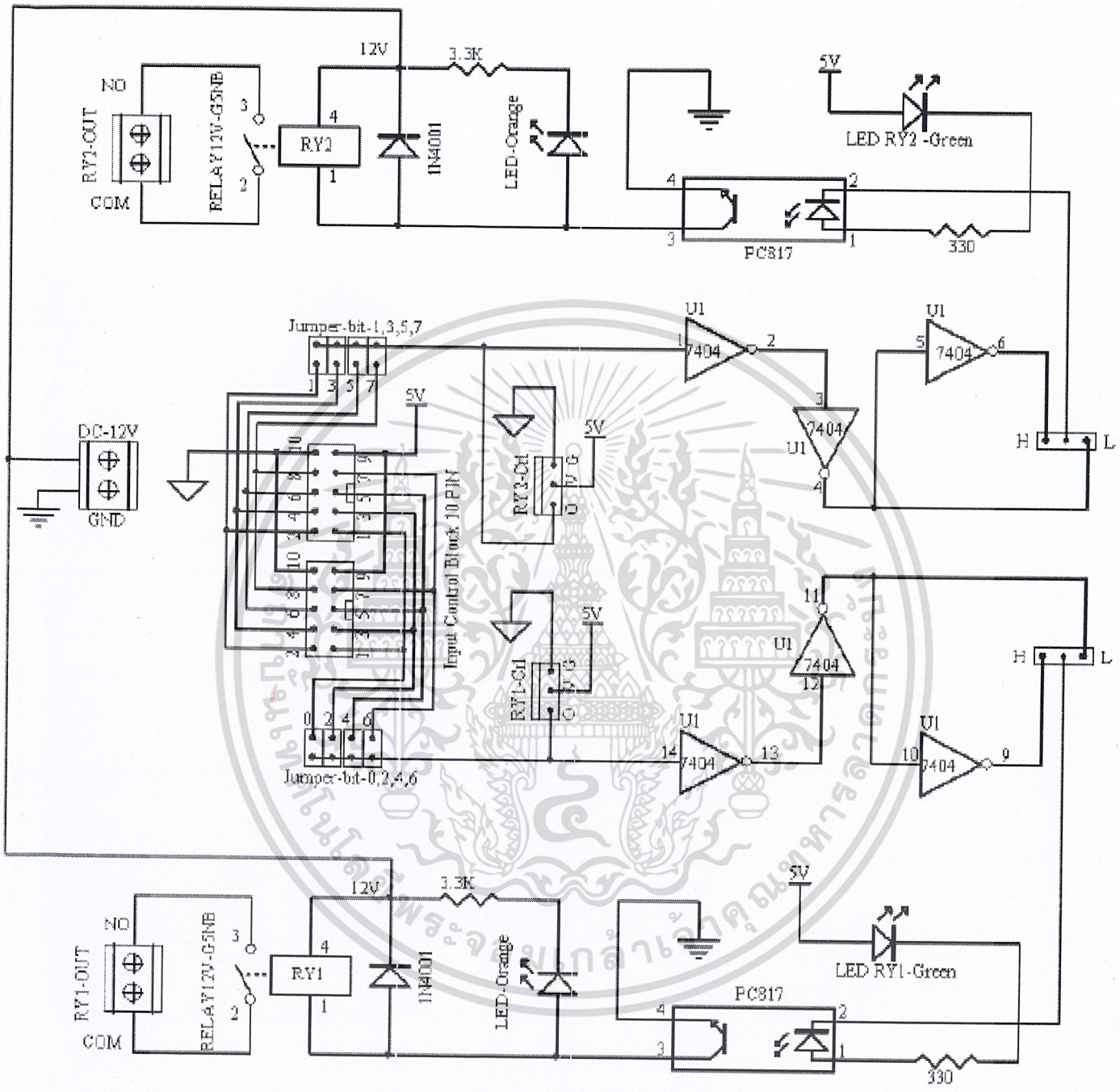
รูปที่ ข.1 แสดงลายวงจร ET-BASE dsPIC30F4011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



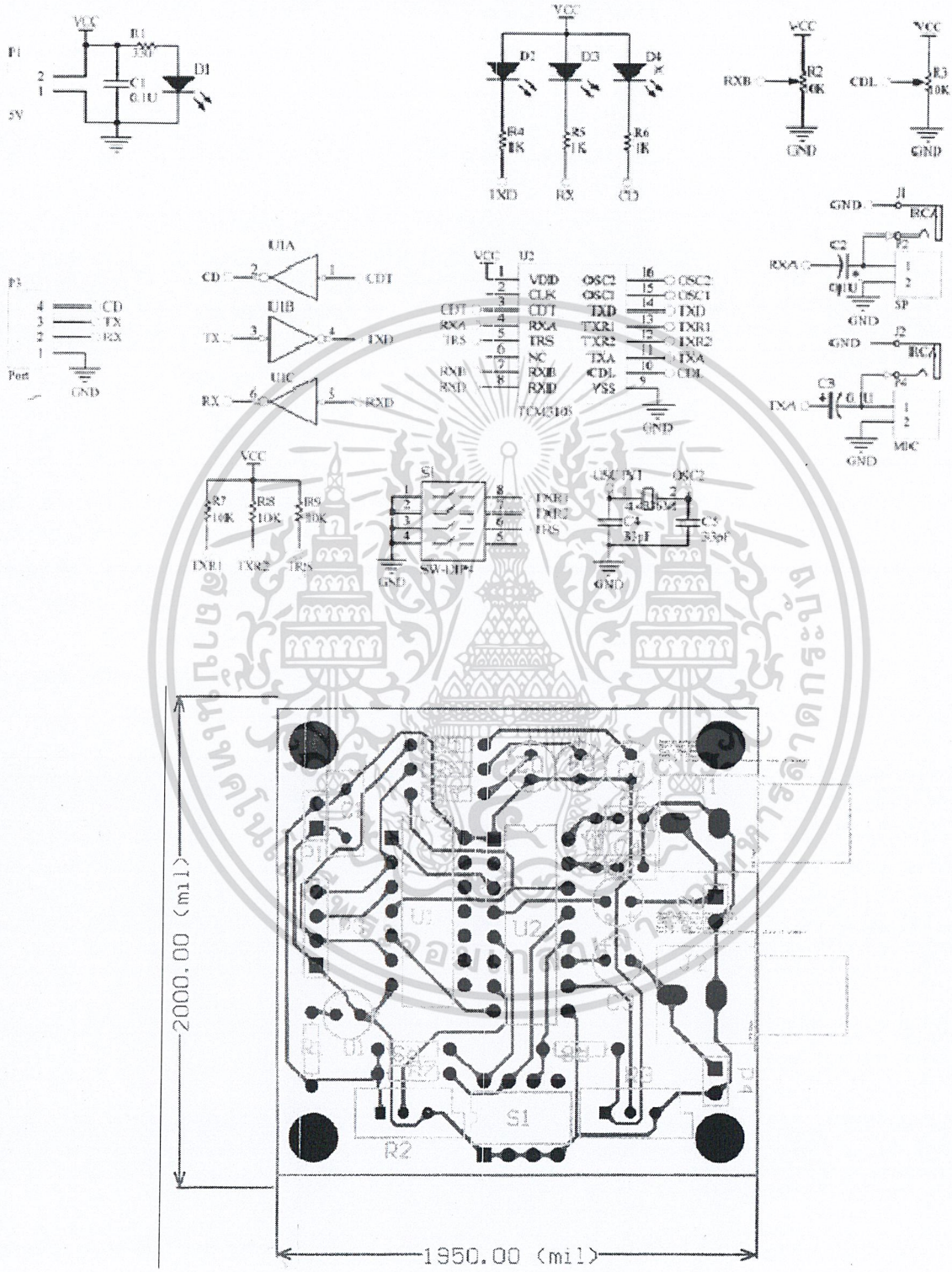
รูปที่ ข.2 ลายวงจรอุปกรณ์หน้าจอ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.3 ลายวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.4 ลายวงจร FSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

DATA SHEET

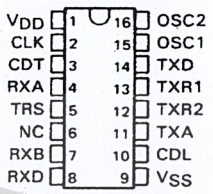
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D2866, AUGUST 1985—REVISED FEBRUARY 1986

NOT RECOMMENDED FOR NEW DESIGN

For new design, refer to TCM3105

J
DUAL-IN-LINE PACKAGE
(TOP VIEW)



NC—No internal connection

description

The TCM3101 is a single-chip asynchronous Frequency Shift Keying (FSK) modem that uses silicon gate CMOS technology to implement a switched capacitor architecture. It is pin selectable (TXR1, TXR2, and TRS inputs) for a wide range of transmit/receive baud rates and is compatible with the applicable Bell 202 or CCITT V23 specifications. Operation is fully reversible, thereby allowing both forward and backward channels to be used simultaneously.

The transmitter is a programmable frequency synthesizer that provides two output frequencies (on TXA), representing the 'marks' and 'spaces' of the digital signal present on the TXD input.

The receive section is responsible for the demodulation of the analog signal appearing at the RXA input and is based on the principle of frequency to voltage conversion. This section contains a group delay equalizer (to correct phase distortion), automatic gain control, carrier detect level adjustment, and bias distortion adjustment, thereby optimizing performance and giving the lowest possible bit error rate.

Carrier-detect information is given to the system by means of the carrier-detect circuits, which set a flag on the CDT output if the level of received in-band energy falls below a value set on the CDL input for a specified minimum duration.

Caution. These devices have limited built-in gate protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates. .

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1985, Texas Instruments Incorporated

TEXAS INSTRUMENTS
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

This Material Copyrighted By Its Respective Manufacturer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

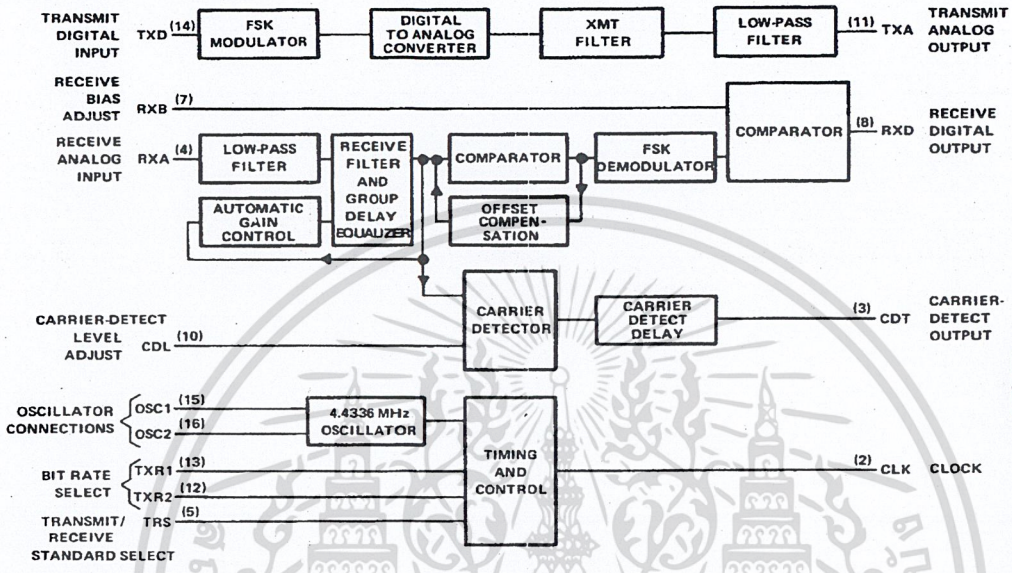
PIN NO.	PIN NAME	DESCRIPTION
1	V_{DD}	Positive supply voltage
2	CLK	Output for a continuous clock signal output at 19.11 kHz (9.56 kHz for CCITT V23 600 baud)
3	CDT	Carrier Detect Output, a low-level output that indicates carrier failure
4	RXA	Receive Analog Input to which the received line signal must be ac coupled
5	TRS	Transmit/Receive Standard Select Input, which, with TXR1 and TXR2, sets the standard bit rates and mark/space frequencies
6	NC	No internal connection
7	RXB	Receive Bias Adjust for external adjustment of the decision threshold of the final comparator to minimize bias distortion
8	RXD	Receiver Digital Output for the demodulated received data in positive logic. The high logic level is a mark and the low logic level is a space.
9	V_{SS}	Supply voltage (normally ground), connected to substrate
10	CDL	Carrier Detect Level Adjust for external adjustment of carrier detect threshold
11	TXA	Transmit Analog Output for the modulated signal, which must be ac coupled
12	TXR2	Bit Rate Select 2 input, which, along with TXR1 and TRS, sets the bit rates and mark/space frequencies
13	TXR1	Bit Rate Select 1 input, which, along with TXR2 and TRS, sets the bit rates and mark/space frequencies
14	TXD	Transmit Digital Input for input data to the transmitter in positive logic. The high logic level is a mark and the low logic level is a space. The data can be accepted at any speed from zero to the selected speed and may be totally asynchronous.
15	OSC1	Oscillator connections. The crystal (typically 4.4336 MHz) is connected to these pins. If an external clock is used, OSC1 is left open and the clock is connected to OSC2.
16	OSC2	

TEXAS INSTRUMENTS
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

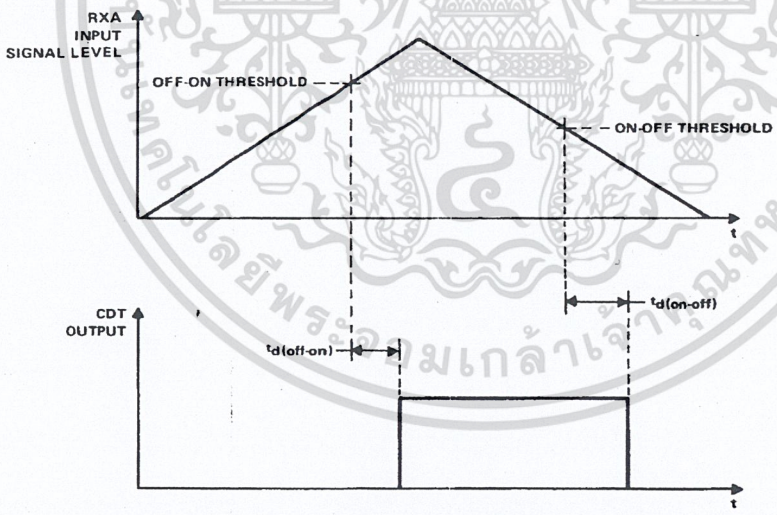
This Material Copyrighted By Its Respective Manufacturer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

functional block diagram



timing diagram



This Material Copyrighted By Its Respective Manufacturer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Supply voltage, V_{DD} (see Note 1) . . . . . −0.3 V to 10 V
Input voltage, V_I (any input) . . . . . −0.3 to V_{DD}
Operating free-air temperature range . . . . . −10°C to 70°C
Storage temperature range . . . . . −55°C to 150°C

NOTE 1: All voltage values are with respect to V_{SS}.

recommended operating conditions

	MIN	NOM	MAX	UNIT
Clock frequency, f_{clock}	4.4334	4.4336	4.4338	MHz
Supply voltage, V_{DD}	4	5	5.5	V
High-level input voltage, V_{IH}	2		V_{DD}	V
Low-level input voltage, V_{IL}	0		0.8	V
Analog input level, peak-to-peak (ac coupled)		0.3	0.78	V
Analog load impedance at TXA	50			kΩ
Operating free-air temperature range, T_A	−10		70	°C

2 Telecommunications Circuits

electrical characteristics over recommended ranges of operating free-air temperature and supply voltage (unless otherwise noted)

PARAMETER			TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-Level output voltage	RXD, CDT, CLK	I_{OH} = −100 μA,	V_{DD} = 5 V	2.4		V_{DD}	V
V_{OL}	Low-level output voltage	RXD, CDT, CLK	I_{OL} = 1.6 mA,	V_{DD} = 5 V	V_{SS}		0.4	V
	Analog output voltage level, peak-to-peak		V_{DD} = 4 V	R_L = 50 kΩ, C_L = 100 pF		1.3		V
			V_{DD} = 5 V		1.4	1.6	2	
			V_{DD} = 5.5 V			1.75		
	Adjust voltage	RXB	V_{DD} = 5 V		2.5	2.7	3	V
		CDL			2	3.2	4	
	Analog output dc offset					$V_{DD}/2$		V
	Digital input current		V_I = 0 to V_{DD}				±1	μA
	Analog input current						±15	μA
	Bias input current	RXB, CDL				100	150	μA
I_{DD}	Supply current		V_{DD} = 4 V			2	4	mA
			V_{DD} = 5 V			4	6	
			V_{DD} = 5.5 V			6	8	
C_i	Input capacitance, all inputs		f = 1 MHz			10		pF
C_o	Output capacitance, all outputs		f = 1 MHz			10		pF
	Phase jitter						200	μs
	Bias distortion†						15%	
	Carrier detect threshold, off-on‡				−45.5		−43	dBm
	Carrier detect threshold, on-off‡				−48		−45.5	dBm
	Carrier detect hysteresis				2.5	2.8		dBm
$t_{d(off-on)}$	Carrier detect delay time		RX = 600 or 1200 bps			12	25	ms
$t_{d(on-off)}$	Carrier detect delay time					12	20	ms

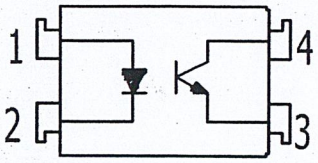
†Bias distortion is the departure from a 50% duty cycle when a series of alternating mark and space tones are received.
‡This is the threshold with the CDL input properly adjusted.

TEXAS INSTRUMENTS
POST OFFICE BOX 225012 • DALLAS, TEXAS 75265

This Material Copyrighted By Its Respective Manufacturer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Schematic:



For dimensions and pin-outs, see the last page of this document.

Features:

1. Current transfer ratio (CTR:MIN.50% at IF=5mA Vce=5V)
2. High isolation voltage between input and output (Viso:5300Vrms).
3. Compact dual-in-line package.

Ordering:

Suffix to Standard Part Number

- V = VDE Approved
- G = 10mm Lead Spread
- S = Surface Mount Lead-form
- T = Tape & Reel

Superior OPTO Part Number:

OPTO410

Absolute Maximum Ratings:

(Ta=25°C)

Parameter		Symbol	Rating	Unit
Input	Forward current	IF	50	mA
	Peak forward current	IFM	1	A
	Reverse voltage	VR	6	V
	Power dissipation	PD	70	mW
Output	Collector-emitter voltage	VCEO	60	V
	Emitter-collector voltage	VECO	6	V
	Collector current	IC	50	mA
	Collector power dissipation	PC	150	mW
Total power dissipation		Ptot	200	mW
Isolation voltage 1 minute		Viso	5300	Vrms
Operating temperature		Topr	-55 to +100	°C
Storage temperature		Tstg	-55 to +125	°C
Soldering temperature 10 second		Tsol	260	°C

Electrical Characteristics:

(Ta=25°C)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	VF	IF =20mA	—	1.2	1.4	V
	Peak forward voltage	VFM	IFM =0.5A	—	—	3.0	V
	Reverse current	IR	VR =4V	—	—	10	uA
	Terminal capacitance	Ct	V=0, f=1kHz	—	30	—	pF
Output	Collector dark current	ICEO	VCE =20V	—	—	0.1	uA
Transfer charac-teristics	Current transfer ratio	CTR	IF =5mA, VCE =5V	50	—	600	%
	Collector-emitter saturation voltage	VCE (sat)	IF =20mA, IC =1mA	—	0.1	0.2	V
	Isolation resistance	Riso	DC500V	$5X10^{10}$	10^{11}	—	ohm
	Floating capacitance	Cf	V=0, f=1MHz	—	0.6	1.0	pF
	Cut-off frequency	fc	VCC =5V, IC=2mA, RL=100ohm	—	80	—	kHz
	Respone time (Rise)		VCE =2V, IC=2mA, RL=100ohm	—	4	18	us
	Respone time (Fall)	tf		—	3	18	us

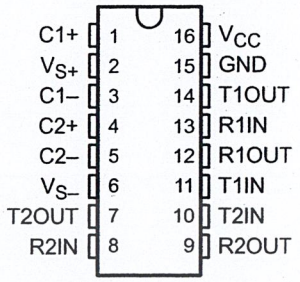
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

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

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



description/ordering information

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

ORDERING INFORMATION

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

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

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

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

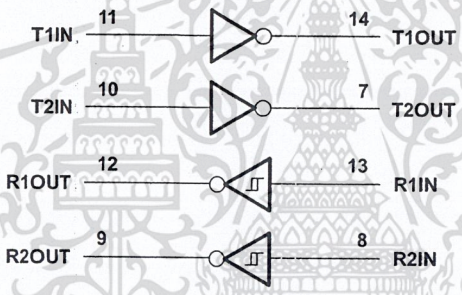
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FSK Modulator/Demodulator

GENERAL DESCRIPTION

The XR-210 is a highly versatile monolithic phase-locked loop system, especially designed for data communications. It is particularly well suited for FSK modulation/demodulation (MODEM) applications, frequency synthesis, tracking filters, and tone decoding. The XR-210 operates over a power supply range of 5V to 26V, and over a frequency band of 0.5 Hz to 20 MHz. The circuit can accommodate analog signals between 300μV and 3V, and can interface with conventional DTL, TTL, and ECL logic families.

FEATURES

Wide Frequency Range	0.5 Hz to 20 MHz
Wide Supply Voltage Range	5V to 26V
Digital Programming Capability	
RS-232C Compatible Demodulator Output	
DTL, TTL and ECL Logic Compatibility	
Wide Dynamic Range	300μV to 3V
ON-OFF Keying & Sweep Capability	
Wide Tracking Range	±1% to ±50%
Good Temperature Stability	200 ppm/°C
High-Current Logic Output	50 mA
Independent "Mark" and "Space" Frequency Adjustment	
VCO Duty Cycle Control	

APPLICATIONS

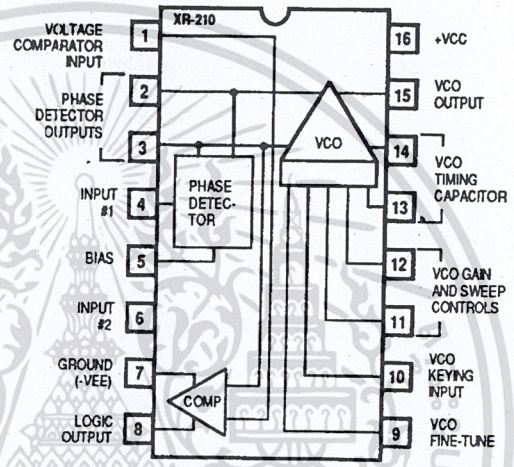
Data Synchronization
Signal Conditioning
FSK Generation
Tone Decoding
Frequency Synthesis
FSK Demodulation
Tracking Filter
FM Detection
FM and Sweep Generation
Wideband Discrimination

ABSOLUTE MAXIMUM RATINGS

Power Supply	26 Volts
Power Dissipation	750 mW
Derate Above +25°C	6.0 mW/°C
Storage Temperature	- 65°C to + 150°C

Rev-C

FUNCTIONAL BLOCK DIAGRAM



SYSTEM DESCRIPTION

The XR-210 is made up of a stable wide-range voltage- controlled oscillator (VCO), exclusive OR gate type phase detector, and an analog voltage comparator. The VCO, which produces a square wave as an output, is either used in conjunction with the phase detector to form a phase-locked loop (PLL) for FSK demodulation and tone detection or as a generator in FSK modulation schemes. The phase detector when used in the PLL configuration produces a differential output voltage with a 6 KΩ output impedance, which when capacitively loaded forms a single pole loop filter. The voltage comparator is used to sense the phase detector output and produces the output in the FSK demodulation connection.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

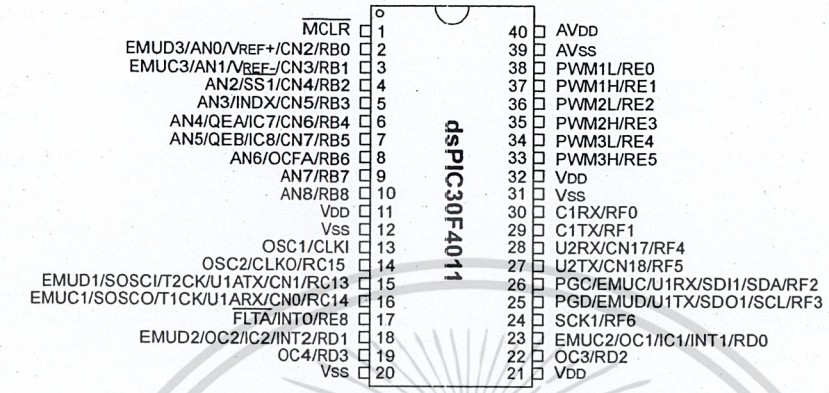
ELECTRICAL PERFORMANCE CHARACTERISTICS - XR-210

TEST	SYMBOL	CONDITIONS	TEMPERATURE	LIMITS MIN	LIMITS MAX	UNIT	GROUP A SUBGROUP
Supply Current	I_{CC}	V_{CC} = ±6V	T_A = +25°C	5.0	16.0	mA	1
			-55°C≤T_A≤+125°C	5.0	20.0	mA	2,3
Supply Current	I_{CC}	V_{CC} = ±13V	T_A = +25°C		26.0	mA	1
			-55°C≤T_A≤+125°C		26.0	mA	2,3
VCO Power Supply Stability	PSR	±6V≤V_{CC}≤±12V	T_A =+25°C		0.5	%/V	9
			-55°C≤T_A≤+125°C		1.0	%/V	10,11
VCO Sweep Range	FSW		T_A = +25°C	5:1			9
			-55°C≤T_A≤+125°C	3:1			10,11
VCO Duty Cycle Asymmetry	DC		T_A = +25°C		±3	%	9
			-55°C≤T_A≤+125°C		±10	%	10,11
Phase Detector Output Offset Voltage		Measured Across Pin 1 and Pin 3, VIN =0	T_A = +25°C		±150	mV	1
			-55°C≤T_A≤+125°C		±150	mV	2,3
Logic Output Leakage Current	I_{OH}	V_{CC} = ±12V	T_A = +25°C		10.0	µA	1
		V_{CC} = ±6V	-55°C≤T_A≤+125°C		100.0	µA	2,3
Logic Output Low Voltage	V_{OL}	I_L = 10 mA	T_A = +25°C		0.4	V	1
			-55°C≤T_A≤+125°C		0.7	V	2,3
Logic Output Sink Current	I_{SINK}	V_o≤1V	T_A =+25°C	30		mA	1
			-55°C≤T_A≤+125°C	25		mA	2,3

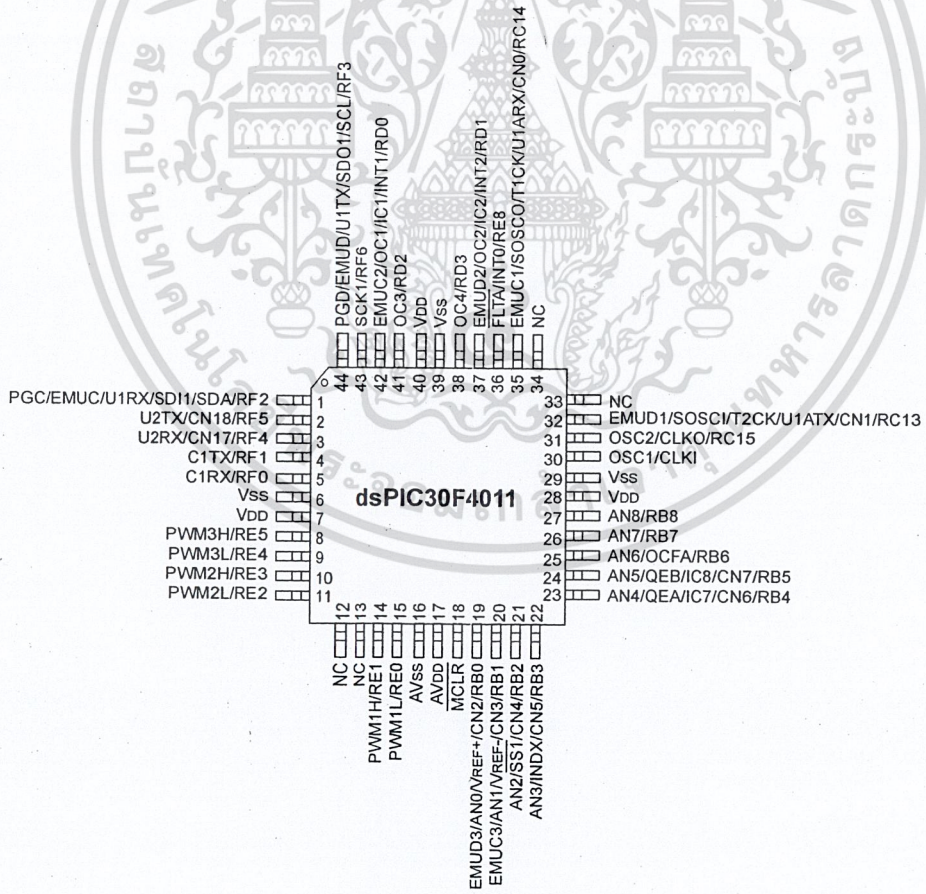
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagrams

40-Pin PDIP



44-Pin TQFP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1-1 provides a brief description of the device I/O pinout and the functions that are multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: dsPIC30F4011 I/O PIN DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN8	I	Analog	Analog input channels. AN0 and AN1 are also used for device programming data and clock inputs, respectively.
AVDD	P	P	Positive supply for analog module. This pin must be connected at all times.
AVSS	P	P	Ground reference for analog module.
CLKI CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN7 CN17-CN18	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
C1RX C1TX	I O	ST —	CAN1 bus receive pin. CAN1 bus transmit pin.
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2 EMUD3 EMUC3	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST ST ST	ICD Primary Communication Channel data input/output pin. ICD Primary Communication Channel clock input/output pin. ICD Secondary Communication Channel data input/output pin. ICD Secondary Communication Channel clock input/output pin. ICD Tertiary Communication Channel data input/output pin. ICD Tertiary Communication Channel clock input/output pin. ICD Quaternary Communication Channel data input/output pin. ICD Quaternary Communication Channel clock input/output pin.
IC1, IC2, IC7, IC8	I	ST	Capture inputs 1, 2, 7 and 8.
INDX QEA QEB	I I I	ST ST ST	Quadrature Encoder Index Pulse input. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode.
INT0 INT1 INT2	I I I	ST ST ST	External interrupt 0. External interrupt 1. External interrupt 2.
$\overline{FLTA}$ PWM1L PWM1H PWM2L PWM2H PWM3L PWM3H	I O O O O O O	ST — — — — — —	PWM Fault A input. PWM1 low output. PWM1 high output. PWM2 low output. PWM2 high output. PWM3 low output. PWM3 high output.
$\overline{MCLR}$	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active-low Reset to the device.
OCFA OC1-OC4	I O	ST —	Compare Fault A input (for Compare channels 1, 2, 3 and 4). Compare outputs 1 through 4.

Legend: CMOS = CMOS compatible input or output; Analog = Analog input
ST = Schmitt Trigger input with CMOS levels; O = Output
I = Input; P = Power

© 2008 Microchip Technology Inc.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-1: dsPIC30F4011 I/O PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin Type	Buffer Type	Description
OSC1	I	ST/CMOS	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
OSC2	I/O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.
PGD	I/O	ST	In-Circuit Serial Programming™ data input/output pin.
PGC	I	ST	In-Circuit Serial Programming clock input pin.
RB0-RB8	I/O	ST	PORTB is a bidirectional I/O port.
8RC13-RC15	8I/O	8ST	PORTC is a bidirectional I/O port.
RD0-RD3	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE5, RE8	I/O	ST	PORTE is a bidirectional I/O port.
RF0-RF6	I/O	ST	PORTF is a bidirectional I/O port.
SCK1	I/O	ST	Synchronous serial clock input/output for SPI1.
SDI1	I	ST	SPI1 data in.
SDO1	O	—	SPI1 data out.
$\overline{SS1}$	I	ST	SPI1 slave synchronization.
SCL	I/O	ST	Synchronous serial clock input/output for I^2C™.
SDA	I/O	ST	Synchronous serial data input/output for I^2C.
SOSCO	O	—	32 kHz low-power oscillator crystal output.
SOSCI	I	ST/CMOS	32 kHz low-power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
T1CK	I	ST	Timer1 external clock input.
T2CK	I	ST	Timer2 external clock input.
U1RX	I	ST	UART1 receive.
U1TX	O	—	UART1 transmit.
U1ARX	I	ST	UART1 alternate receive.
U1ATX	O	—	UART1 alternate transmit.
U2RX	I	ST	UART2 receive.
U2TX	O	—	UART2 transmit.
VDD	P	—	Positive supply for logic and I/O pins.
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog voltage reference (high) input.
VREF-	I	Analog	Analog voltage reference (low) input.

Legend: CMOS = CMOS compatible input or output; Analog = Analog input
ST = Schmitt Trigger input with CMOS levels; O = Output
I = Input; P = Power

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS04
Hex Inverting Gates

General Description

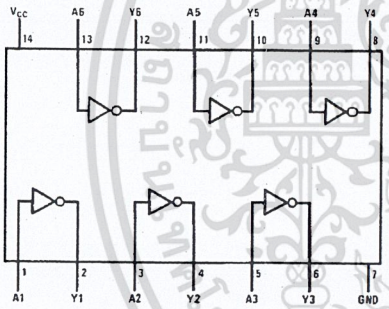
This device contains six independent gates each of which performs the logic INVERT function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS04M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS04SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS04N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

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

Connection Diagram



Function Table

$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H = HIGH Logic Level
L = LOW Logic Level

© 2000 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage	2			V
V_{IL}	LOW Level Input Voltage			0.8	V
I_{OH}	HIGH Level Output Current			−0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	µA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 3)	−20		−100	mA
I_{CCH}	Supply Current with Outputs HIGH	V_{CC} = Max		1.2	2.4	mA
I_{CCL}	Supply Current with Outputs LOW	V_{CC} = Max		3.6	6.6	mA

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R_L = 2 kΩ				Units
		C_L = 15 pF		C_L = 50 pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	3	10	4	15	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	3	10	4	15	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KEC
KOREA ELECTRONICS CO.,LTD.

SEMICONDUCTOR
TECHNICAL DATA

KIA7805AP/API~ KIA7824AP/API
BIPOLAR LINEAR INTEGRATED CIRCUIT

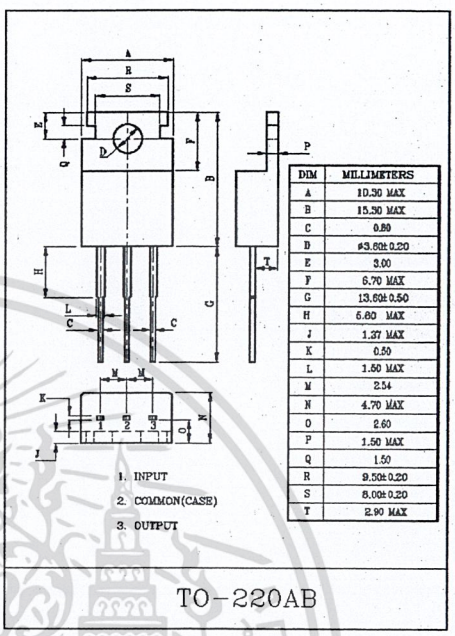
THREE TERMINAL POSITIVE VOLTAGE REGULATORS
5V, 6V, 8V, 9V, 10V, 12V, 15V, 18V, 20V, 24V.

FEATURES

- Suitable for C-MOS, TTL, the Other Digital IC's Power Supply.
- Internal Thermal Overload Protection.
- Internal Short Circuit Current Limiting.
- Output Current in Excess of 1A.
- Satisfies IEC-65 Specification. (International Electronical Commission).

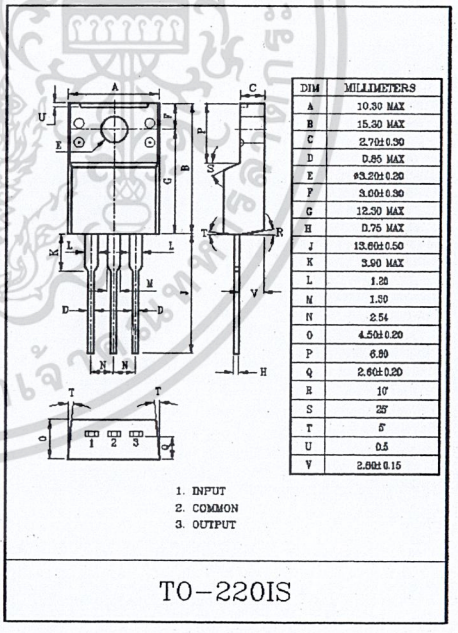
MAXIMUM RATINGS (Ta=25℃)

CHARACTERISTIC		SYMBOL	RATING	UNIT
Input Voltage	KIA7805AP/API~ KIA7815AP/API	V_{IN}	35	V
	KIA7818AP/API~ KIA7824AP/API		40	
Power Dissipation (Tc=25℃)		P_D	20.8	W
Power Dissipation (Without Heatsink)	KIA7805API~ KIA7824API	P_D	2.0	W
Operating Junction Temperature		T_j	-30~150	℃
Storage Temperature		T_{stg}	-55~150	℃



DIM	MILLIMETERS
A	10.30 MAX
B	15.30 MAX
C	0.80
D	ø3.60±0.20
E	3.00
F	6.70 MAX
G	13.60±0.50
H	5.60 MAX
J	1.27 MAX
K	0.50
L	1.50 MAX
M	2.54
N	4.70 MAX
O	2.60
P	1.50 MAX
Q	1.50
R	9.50±0.20
S	8.00±0.20
T	2.90 MAX

TO-220AB

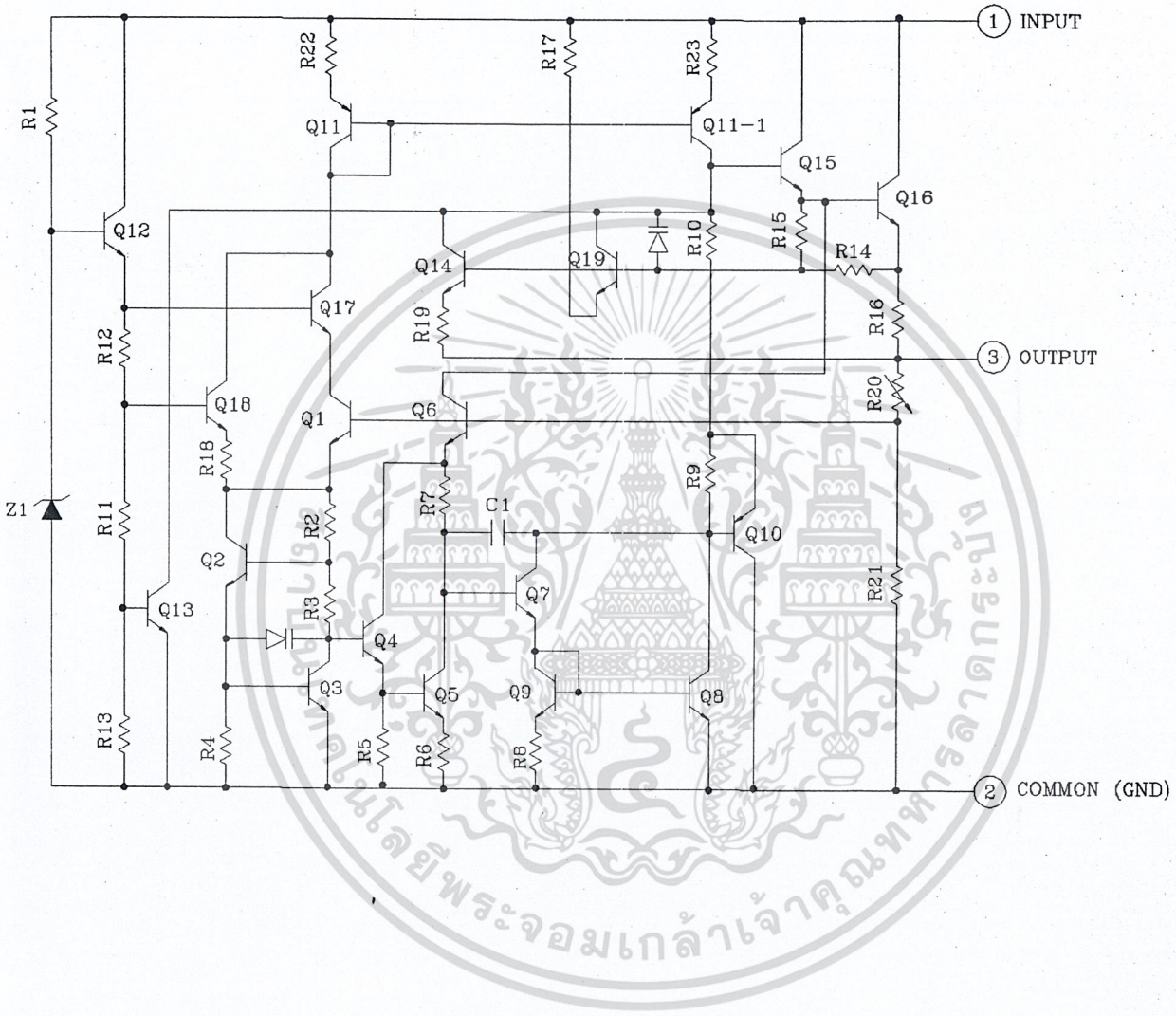


DIM	MILLIMETERS
A	10.30 MAX
B	15.30 MAX
C	2.70±0.30
D	0.85 MAX
E	ø3.20±0.20
F	3.00±0.30
G	12.30 MAX
H	0.75 MAX
J	13.60±0.50
K	3.90 MAX
L	1.20
N	1.30
N	2.54
O	4.50±0.20
P	6.80
Q	2.60±0.20
R	10°
S	25°
T	5°
U	0.5
V	2.60±0.15

TO-220IS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EQUIVALENT CIRCUIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-5474; Rev 8/10

DALLAS SEMICONDUCTOR MAXIM

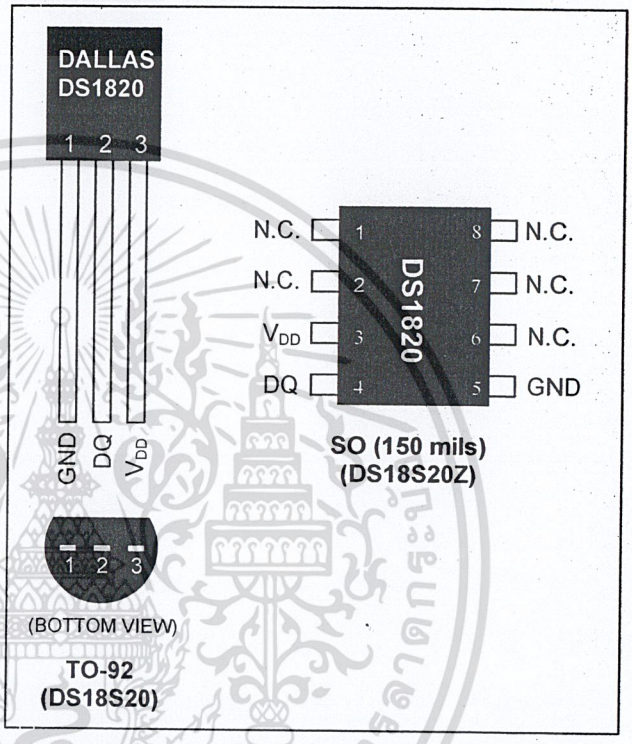
www.maxim-ic.com

DS18S20
High-Precision 1-Wire Digital Thermometer

FEATURES

- Unique 1-Wire® Interface Requires Only One Port Pin for Communication
- Each Device has a Unique 64-Bit Serial Code Stored in an On-Board ROM
- Multidrop Capability Simplifies Distributed Temperature Sensing Applications
- Requires No External Components
- Can Be Powered from Data Line. Power Supply Range is 3.0V to 5.5V
- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- 9-Bit Thermometer Resolution
- Converts Temperature in 750ms (max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Applications Include Thermostatic Controls, Industrial Systems, Consumer Products, Thermometers, or Any Thermally Sensitive System

PIN CONFIGURATIONS



DESCRIPTION

The DS18S20 digital thermometer provides 9-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +125°C and is accurate to ±0.5°C over the range of –10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1-Wire bus. Thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment, or machinery, and process monitoring and control systems.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE
DS18S20	–55°C to +125°C	3 TO-92
DS18S20+	–55°C to +125°C	3 TO-92
DS18S20/T&R	–55°C to +125°C	3 TO-92 (2000 Piece)
DS18S20+T&R	–55°C to +125°C	3 TO-92 (2000 Piece)
DS18S20-SL/T&R	–55°C to +125°C	3 TO-92 (2000 Piece)*
DS18S20-SL+T&R	–55°C to +125°C	3 TO-92 (2000 Piece)*
DS18S20Z	–55°C to +125°C	8 SO
DS18S20Z+	–55°C to +125°C	8 SO
DS18S20Z/T&R	–55°C to +125°C	8 SO (2500 Piece)
DS18S20Z+T&R	–55°C to +125°C	8 SO (2500 Piece)

+Denotes a lead(Pb)-free/RoHS-compliant package. A "+" appears on the top mark of lead(Pb)-free packages.
T&R = Tape and reel.
**TO-92 packages in tape and reel can be ordered with straight or formed leads. Choose "SL" for straight leads. Bulk TO-92 orders are straight leads only.*

PIN DESCRIPTION

PIN		NAME	FUNCTION
TO-92	SO		
1	5	GND	Ground
2	4	DQ	Data Input/Output. Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see the *Powering the DS18S20* section.)
3	3	V_{DD}	Optional V_{DD}. V_{DD} must be grounded for operation in parasite power mode.
—	1, 2, 6, 7, 8	N.C.	No Connection

OVERVIEW

Figure 1 shows a block diagram of the DS18S20, and pin descriptions are given in the *Pin Description* table. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L). The T_H and T_L registers are nonvolatile (EEPROM), so they will retain data when the device is powered down.

The DS18S20 uses Maxim's exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the *1-Wire Bus System* section.

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V_{DD}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้