ระบบดาต้าล็อกเกอร์

DATA LOGGER SYSTEM

สราวัตน์ บุญเกิด

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2555

ระบบดาต้าล็อกเกอร์
DATA LOGGER SYSTEM

สรารัตน์ บุญเกิด

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมระบบควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA LOGGER SYSTEM

SARARAT BOONKIRD

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN CONTROL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2555

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบดาต้าล็อกเกอร์

DATA LOGGER SYSTEM

ผู้จัดทำ นางสาวสรารัตน์ บุญเกิด 52011253

.....อาจารย์ที่ปรึกษา

(รองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบดาต้าล็อกเกอร์

โดย

นางสาวสรารัตน์ บุญเกิด 52011253

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์

ปีการศึกษา 2555

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอระบบดาต้าล็อกเกอร์ สำหรับแสดงผลและจัดเก็บบันทึกข้อมูล โดยโครงสร้างของระบบประกอบไปด้วยเซนเซอร์ การปรับแต่งสภาพสัญญาณ การต่อเชื่อมสัญญาณที่ได้รับจากเซนเซอร์ หน่วยประมวลผล และบันทึกข้อมูล ขั้นตอนการดำเนินการ เริ่มจากการออกแบบโครงสร้างของระบบดาต้าล็อกเกอร์ และหน่วยแสดงผลทางจอภาพ โดยใช้โปรแกรมวิชชวล ในการออกแบบ การทำงานของหน่วยประมวลผลสำหรับประมวลผลสัญญาณและส่งต่อไปยังหน่วยแสดงผล ได้การใช้ภาษาซี Arduino และการออกแบบหน่วยบันทึกข้อมูลจะใช้โปรแกรม SQL Server 2008 สำหรับสร้างฐานข้อมูลเพื่อช่วยในการจัดเก็บและรวมรวมข้อมูลที่ได้จากเซนเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA LOGGER SYSTEM

By

Miss Sararat Boonkird 52011253

Advisor

Assoc.Prof.Dr. Worapong Tangsrirat

Academic Year 2012

ABSTRACT

This project the implementation of the data logger for displaying, monitoring and recording the measured values obtained from sensors, signal conditioning circuits, microcontroller and database system. Procedure of process starts from design the structure of data logger system, design display & monitoring unit by visual program, design and program for data processing unit for processing incoming signal and sent to display & monitoring unit by using C arduino programming and design database system by using SQL server 2008 for collecting database from sensor unit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถที่จะสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับการช่วยเหลือเป็นอย่างดีจาก รองศาสตราจารย์ ดร.วรพงศ์ ตั้งศรีรัตน์ ที่ได้ให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งคอยถามถึงความคืบหน้าอยู่ตลอดเวลาและความช่วยเหลืออื่นๆ ที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและกราบขอบพระคุณอย่างสูง

ขอบคุณพี่กอล์ฟ ธงชัย พจน์เสถียร ที่ให้ความช่วยเหลือในส่วนของการเขียนโปรแกรม ซึ่งเป็นส่วนที่สำคัญของปริญญานิพนธ์นี้

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่สำคัญตลอดการทำงานรวมถึงสนับสนุนงบประมาณช่วยเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นางสาวสรารัตน์ บุญเกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน	1
1.4 รายละเอียดของปริญญานิพนธ์	2
บทที่ 2 ภาพรวมของระบบในโครงงาน	3
2.1 ส่วนของระบบเซนเซอร์	3
2.1.1 ส่วนของเซนเซอร์ตรวจวัดอุณหภูมิและความชื้น	3
2.1.2 ส่วนของเซนเซอร์ตรวจจับแสง	6
2.2 ส่วนของการปรับแต่งสภาพสัญญาณและการเชื่อมต่อสัญญาณที่ได้จากเซนเซอร์	6
2.2.1 การแปลงสัญญาณ A/D และ D/A	6
2.2.2 ส่วนของตัวทำหน้าที่เชื่อมต่อสัญญาณ	9
2.3 ส่วนของการแสดงผลและบันทึกข้อมูล	12
2.3.1 หน่วยแสดงผล (Output Unit)	12
2.3.2 ส่วนของฐานข้อมูลและจัดเก็บข้อมูลด้วยโปรแกรม SQL server R2	15
บทที่ 3 หลักการออกแบบ	16
3.1 การออกแบบตารางจัดเก็บข้อมูล	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2 การสร้างฐานข้อมูล (Data base)	18
3.3 การสร้างตาราง	22
3.3.1 การสร้างตารางแบบที่ 1	22
3.3.2 การสร้างตารางแบบที่ 2 ด้วยคำสั่ง CREATE	24
3.4 การเพิ่มข้อมูลด้วยคำสั่ง INSERT INTO	26
3.5 การเรียกดูข้อมูลด้วยคำสั่ง Select	28
บทที่ 4 การทดลองและผลการทดลอง	29
4.1 การทดลองโปรแกรม และผลที่ได้จากการทางาน	29
4.1.1 Open Interface	29
4.1.2 เริ่มทำการเชื่อมต่อกับฐานข้อมูล	29
4.1.3 เปิดฐานข้อมูลที่จัดเก็บด้วยนวัตกรรมของ SQL Server 2008 R2	30
4.1.4 รีเซ็ตข้อมูล	31
4.1.5 ตัดการเชื่อมต่อ	32
4.2 ภาคการแสดง และนำเสนอ	33
4.2.1 นำเสนอฐานข้อมูลเชิงกราฟ	33
4.2.2 นำเสนอฐานข้อมูลเชิงตาราง	36
บทที่ 5 บทวิจารณ์และสรุป	38
5.1 สรุป	38
5.2 ปัญหาที่พบและแนวทางแก้ไข	39
5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
ภาคผนวก ก ส่วนโปรแกรมภาษาซีที่ใช้ในโครงงาน	41
ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์	52
ข.1 เอกสารคู่มือการใช้งาน module – DHT11	52
ข.2 เอกสารคู่มือการใช้งาน บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2)	55
ข.3 เอกสารคู่มือการใช้งาน ET-USB/RS232 MINI	90
เอกสารอ้างอิง	105

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
2.1 องค์ประกอบรวมของระบบ	3
2.2 Humidity & temperaturesensor / module – DHT11	5
2.3 การต่อโมดุล DHT11 กับบอร์ด Arduino	5
2.4 แผงวงจรตรวจจับแสง : LDR Light sensor	6
2.5 บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2)	9
2.6 ET-USB/RS232 MINI	11
2.7 การเชื่อมต่ออุปกรณ์อุกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB	11
2.8 ตำแหน่งขาที่จะใช้งาน	11
2.9 หน่วยแสดงผล (Output Unit)	12
2.10 โปรแกรมวิชชวล (visual studio 2010)	15
2.11 โปรแกรมSQL server 2008 R2	15
3.1 เข้าใช้งานโปรแกรม	16
3.2 เข้าสู่หน้าของการ Login เพื่อใช้งานฐานข้อมูล	17
3.3 ผู้ใช้ NES-PC ต้องระบุ User name และ Password	17
3.4 Login สำเร็จจะเข้าสู่หน้าจอของโปรแกรม	18
3.5 การสร้างฐานข้อมูล (Data base)	18
3.6 New Database	19
3.7 Database Name	19
3.8 Select a page	20
3.9 Databases	20
3.10 แก้ไขภาษาให้กับฐานข้อมูล	21
3.11 แสดงหน้า Database Properties > Select a page	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.12 การสร้างตาราง (New Table)	22
3.13 ระบุ Column Name, Data Type และ Allow Nulls	23
3.14 สร้างชื่อฟิลด์ครบถ้วน	23
3.15 แสดงตาราง	24
3.16 การสร้างตารางใหม่ (Create New Table) ในฐานข้อมูล	25
3.17 ตารางที่ต้องการเพิ่มข้อมูล	25
3.18 เพิ่มข้อมูลให้ตรงกับคอลัมน์ที่สร้างไว้	26
3.19 การเพิ่มข้อมูลด้วยคำสั่ง INSERT INTO	26
3.20 โครงสร้างของฐานข้อมูล	27
3.21 เปลี่ยนค่าของ VALUES	27
3.22 การเรียกดูข้อมูลด้วยคำสั่ง Select	28
4.1 เปิดหน้า Interface	29
4.2 การ com port และการ เชื่อมต่อ	30
4.3 ทำการเชื่อมต่อ SQL Server 2008 R2	30
4.4 การเปิดไฟล์	30
4.5 เรียกไฟล์ที่ทาการเก็บข้อมูลมารับค่าและบันทึกผล	31
4.6 ทำการรีเซ็ตข้อมูล	31
4.7 หน้าต่างที่เก็บข้อมูล	32
4.8 ตัดการเชื่อมต่อ	32
4.9 แสดงข้อมูลเชิงกราฟ	33
4.10 ข้อมูลเชิงกราฟ	33
4.11 กราฟแสดงอุณหภูมิ	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

รูปที่	หน้า
4.12 กราฟแสดงความชื้น	34
4.13 กราฟแสดงความเข้มแสง	35
4.14 ปิดโปรแกรม	35
4.15 เลือกแสดงข้อมูลเชิงรายงานนำเสนอ	36
4.16 รายงานนาเสนอ	36
4.17 ปิดหน้า Interface	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 กล่าวนำ

ดาต้าล็อกเกอร์เป็นอุปกรณ์อิเล็กทรอนิกส์ที่บันทึกข้อมูลในช่วงเวลาที่กำหนด โดยเชื่อมต่อผ่านกันระหว่างฐานข้อมูลกับหน่วยที่ใช้ประมวลผล และสร้างฐานข้อมูลไว้สาหรับจัดเก็บรวบรวมข้อมูลที่ได้ จะบันทึกข้อมูลเมื่อเปิดระบบการใช้งานข้อมูลดาต้าล็อกเกอร์

ทั้งนี้ก็เพื่อช่วยให้เราได้รับข้อมูลที่ครอบคลุมสภาวะแวดล้อมได้ถูกต้องตามข้อมูลที่มีการติดตามผล เช่น อุณหภูมิ ความชื้นสัมพัทธ์ และความเข้มแสง เพื่อช่วยในการวิเคราะห์และคาดคะเนการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และความเข้มแสงแสงได้ เพื่อทำการประเมินสภาวะแวดล้อมกับสมดุลที่เกิดขึ้นรอบรอบบริเวณที่ทำการบันทึกผล เพื่อนาค่าที่วัดได้จากเซนเซอร์มาวิเคราะห์และประยุกต์ไปใช้งานจริงได้

โดยระบบดาต้าล็อกเกอร์ มีองค์ประกอบที่สาคัญด้วยกันสามส่วนพื้นฐาน คือ เซนเซอร์ หน่วยประมวลผล (ไมโครคอนโทรลเลอร์) และการแสดงผลพร้อมทั้งบันทึกค่าที่ได้จากเซนเซอร์

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เพื่อศึกษาการเปลี่ยนแปลงของอุณหภูมิ ความชื้น และความเข้มของแสง

2. เพื่อจัดสร้างฐานข้อมูลสาหรับจัดเก็บความสัมพันธ์ของอุณหภูมิ ความชื้น และความเข้มของแสง ในรูปแบบตารางอิเล็กทรอนิกส์ หรือ ฐานข้อมูล (Database) เพื่อบันทึกค่าจากข้อมูลจริง

3. เพื่อออกแบบหน้าต่างที่ไว้สาหรับติดต่อสื่อสารกันระหว่างผู้ดูแลระบบและผู้เยี่ยมชมฐานข้อมูล ที่สามารถเอาข้อมูลจริงมานาเสนอออกมาทางหน้าจอภาพได้

1.3 ขั้นตอนการศึกษาและการจัดทำโครงงาน

1. การศึกษาถึงข้อมูลจากเอกสารโครงงงานต่างๆ รวมทั้งออกแบบรูปแบบโครงงาน

2. ศึกษาการจัดทำฐานข้อมูล และการจัดทำฐานข้อมูลในรูปแบบการนำเสนอ

3. การนำเสนอข้อมูลเพื่อใช้ในการเปรียบเทียบการเปลี่ยนแปลงข้อมูล เพื่อศึกษาถึงสภาวะการเปลี่ยนแปลงของสภาพแวดล้อมในเชิงรายงานนำเสนอ และเชิงกราฟ

4. การออกแบบหน้าต่างของการจัดนำเสนอข้อมูล พร้อมทั้งเขียนโปรแกรมเพื่อควบคุมการทางานของโปรแกรมการนำเสนอข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำรายงานผลการทดลอง และสรุปผลการทำงานทั้งหมด

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึง วัตถุประสงค์ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์แต่ละบท

บทที่ 2 ภาพรวมของระบบในโครงงาน กล่าวถึง ส่วนประกอบหลักของโครงงาน หน้าที่ และหลักการทำงานของส่วนประกอบต่างๆ ภายในระบบ

บทที่ 3 การทำงาน กล่าวถึง ออกแบบ และการสร้างระบบ DATA LOGGER

บทที่ 4 การทดลอง กล่าวถึง การทดลองการทำงานของระบบ

บทที่ 5 บทวิจารณ์และสรุป กล่าวถึง บทสรุปของโครงงาน ปัญหาที่เกิดขึ้น แนวทางในการแก้ไข และแนวทางในการปรับปรุงโครงงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ภาพรวมของระบบในโครงงาน

รูปที่ 2.1 องค์ประกอบรวมของระบบ

สำหรับระบบดาต้าล็อกเกอร์นั้นเราได้แบ่งส่วนประกอบต่างๆออกเป็นส่วนย่อยๆ ตามรูปที่ 2.1 ซึ่งในแต่ละส่วนจะมีหน้าที่ หลักการทำงาน และการสื่อสารที่แตกต่างกันโดยเราสามารถแบ่งส่วนประกอบต่างๆของระบบดาต้าล็อกเกอร์ ออกได้เป็น 3 ส่วน ดังนี้

2.1 ส่วนของระบบเซนเซอร์

2.1.1 ส่วนของเซนเซอร์ตรวจวัดอุณหภูมิและความชื้น

อุณหภูมิ (temperature) คือ ปริมาณทางฟิสิกส์ซึ่งเป็นหน่วยการวัดในการบ่งชี้ถึงระดับความร้อนหรือเย็นของอุปกรณ์ สสารใดๆ โดยขณะที่เรารู้สึกร้อนหรือเย็นนั้น สาเหตุก็เนื่องมาจากการเคลื่อนที่ของโมเลกุลภายในสสาร กล่าวได้อีกนัยหนึ่งว่า อุณหภูมิ คือ การวัดหรือบ่งชี้ถึงระดับพลังงานจลน์เฉลี่ย (average kinetic energy) ของโมเลกุลอันเนื่องมาจากการเคลื่อนที่ของโมเลกุลภายในสสารนั่นเอง หากระดับพลังงานจลน์ในการเคลื่อนที่มีค่าเพิ่มขึ้น จะทำให้อุณหภูมิของสสารนั้นมีค่าเพิ่มขึ้นตามไปด้วย ในทางกลับกันอุณหภูมิจะมีค่าลดลงเมื่อพลังงานจลน์ในการเคลื่อนที่ลดลง

การวัดอุณหภูมิโดยทั่วไปมักแสดงระดับของความร้อนหรือเย็นในอุปกรณ์ สสารใดๆเมื่อเทียบกับระดับอุณหภูมิอ้างอิงซึ่งเป็นค่าคงที่ที่ดั้งเอาไว้ โดยแสดงผลออกมาในหน่วยการวัดของอุณหภูมิซึ่งมีอยู่ 4 หน่วยการวัด คือ

- องศาฟาเรนไฮต์ (Fahrenheit,°F)
- องศาเซลเซียส (Celsius,°C
- องศาเคลวิน (Kelvin,°K)
- องศาแรงคิน (Rankine,°R)

ประวัติและวิวัฒนาการความเป็นมาของหน่วยวัดอุณหภูมิ

องศาฟาเรนไฮต์ คือชนิดสเกลค่าวัดอุณหภูมิชนิดหนึ่ง ที่ถูกตั้งชื่อตามนักฟิสิกส์ชาวเยอรมัน เกเบรียล ฟาเรนไฮต์ (1686-1736) โดยที่ค่าสเกลองศาฟาเรนไฮต์นี้ มีจุดเยือกแข็งอยู่ที่ 32 องศาฟาเรนไฮต์ โดยปกติจะเขียนว่า 32 °F และมีจุดเดือดที่ 212 องศาฟาเรนไฮต์ โดยที่มีระยะห่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างจุดเยือกแข็งกับจุดเดือดของน้ำคือ 180 องศา โดยที่ 1 องศาในสเกลองศาฟาเรนไฮต์นี้ มีค่าเท่ากับ 5/9 ของ 1 เคลวิน (ซึ่งก็คือ 1 องศาเซลเซียส) และที่ลบ 40 องศาฟาเรนไฮต์เท่ากับติดลบ 40 องศาเซลเซียส

องศาเซลเซียส (อังกฤษ: degree Celsius, สัญลักษณ์ °C) เป็นหน่วยวัดอุณหภูมิหน่วยหนึ่งในระบบเอสไอ กำหนดให้จุดเยือกแข็งของน้ำคือ 0 °C และจุดเดือดคือ 100 °C หน่วยนี้ตั้งตามชื่อของนาย แอนเดอร์ เซลเซียส (Anders Celsius มีชีวิตอยู่ระหว่าง ค.ศ. 1701 ถึง 1744) นักดาราศาสตร์ชาวสวีเดน เขาเป็นคนแรกที่เสนอระบบที่ใกล้เคียงกับระบบนี้ในปี พ.ศ. 2285 (ค.ศ. 1742) แต่แรกกำหนดให้อุณหภูมิจุดเยือกแข็งของน้ำคือ 0 องศา และจุดเดือดของน้ำคือ 100 องศาที่ระดับความดันบรรยากาศมาตรฐาน แต่ได้มีการสลับสเกลต่อมาหลังจากที่เซลเซียสเสียชีวิตไประยะหนึ่งในปี พ.ศ. 2491 หน่วยนี้มีชื่อเรียกต่างกันสามแบบ ได้แก่

- เซนติเกรด (centigrade)
- องศาเซนเทสซิมัล (centesimal degree)
- องศาเซลเซียส

แต่นับจากนั้นมา ชื่อที่ถูกเลือกเป็นทางการคือ องศาเซลเซียส จากการประชุม General Conference on Weights and Measures ครั้งที่ 9 โดยมีจุดประสงค์เพื่อระลึกถึงเซลเซียส และเป็นการตัดปัญหาความสับสนที่อาจเกิดขึ้นจากการใช้คำนำหน้า centi- กับระบบเอสไอ และจากการที่นิยามเดิมไม่เหมาะสมที่จะใช้เป็นมาตรฐานอย่างเป็นทางการ เนื่องจากอุณหภูมิขึ้นกับนิยามของความดันบรรยากาศมาตรฐาน ซึ่งความดันบรรยากาศมาตรฐานนี้ก็ขึ้นกับอุณหภูมิอีกทีหนึ่ง จึงกำหนดนิยามใหม่ให้อุณหภูมิ 0.01 °C ตรงกับจุดสามสถานะ (triple point) ของน้ำ และให้สเกลหนึ่งองศามีค่าเป็น 1/273.16 เท่าของผลต่างระหว่างอุณหภูมิที่จุดสามสถานะของน้ำกับจุดศูนย์องศาสัมบูรณ์ นิยามนี้ได้รับการยอมรับในการประชุม General Conference on Weights and Measures ครั้งที่ 10 เมื่อปี พ.ศ. 2497

เคลวิน (อังกฤษ: kelvin, สัญลักษณ์: K) เป็นหน่วยวัดอุณหภูมิหนึ่ง และเป็นหน่วยพื้นฐานหนึ่งในเจ็ดของระบบเอสไอ นิยามให้เท่ากับ 1/273.16 เท่าของอุณหภูมิเทอร์โมไดนามิกของจุดสามสถานะของน้ำ เคลวินตั้งชื่อเพื่อเป็นเกียรติแด่นักฟิสิกส์และวิศวกรชาวอังกฤษ วิลเลียม ทอมสัน โดยหาความสัมพันธ์ระหว่างอุณหภูมิและความเร็วของอิเล็กตรอนที่เคลื่อนที่รอบนิวเคลียส โดยสังเกตว่าถ้าให้ความร้อนกับสสารมากขึ้น อิเล็กตรอนจะมีพลังงานมากขึ้น ทำให้เคลื่อนที่มีความเร็วมากขึ้น ในทางกลับกันถ้าลดความร้อนให้กับสสาร อิเล็กตรอนก็จะมีพลังงานน้อยลง ทำให้การเคลื่อนที่ลดลง และถ้าสามารถลดอุณหภูมิลงจนถึงจุดที่อิเล็กตรอนหยุดการเคลื่อนที่ ณ จุดนั้น จะไม่มีอุณหภูมิหรือพลังงานในสสารเลย และจะไม่มีการแผ่รังสีความร้อนจากวัตถุ จึงเรียกอุณหภูมิ ณ จุดนี้ว่า ศูนย์สัมบูรณ์ (0 K)

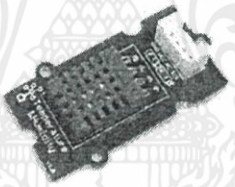
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

องศาโรเมอร์ (อังกฤษ: Réaumur scale/degree; ย่อ:°Ré, °Re, °R) คือหน่วยวัดอุณหภูมิที่คิดค้นขึ้นโดย เรอเน่ อังตวน เฟโชต์ เดอ โรเมอร์ (René Antoine Ferchault de Réaumur) นักวิทยาศาสตร์และนักประดิษฐ์ชาวฝรั่งเศส ในปี ค.ศ. 1731 โดยกำหนดให้จุดเยือกแข็งของน้ำอยู่ที่ 0 องศาโรเมอร์ และจุดเดือดของน้ำอยู่ที่ 80 องศาโรเมอร์ ดังนั้นช่วงอุณหภูมิ 1 องศาโรเมอร์จะเท่ากับ 1.25 องศาเซลเซียสหรือเคลวิน เทอร์โมมิเตอร์ของโรเมอร์นั้นจะบรรจุแอลกอฮอล์เจือจางและมีหลักการคือกำหนดให้อุณหภูมิจุดเยือกแข็งของน้ำอยู่ที่ 0 องศา และขยายตัวไปตามท่อเป็นทีละองศาซึ่งคือเศษหนึ่งส่วนพันของปริมาตรที่บรรจุไว้ในกระเปาะของหลอด ณ จุดศูนย์องศา

ในโครงงานนี้เราได้ใช้ชุดเซนเซอร์วัดความชื้นและอุณหภูมิแบบดิจิตอลด้วยHumidity & temperaturesensor/module – DHT11 ดังรูปที่ 2.2 ซึ่งมีคุณสมบัติตามที่เราต้องการ โดยเซนเซอร์โมดุล DHT11 เป็นชุดที่สามารถวัดการเปลี่ยนแปลงทั้งอุณหภูมิและความชื้นสามารถดูจากดาต้าชีทในภาคผนวก

มีคุณลักษณะที่สำคัญดังนี้

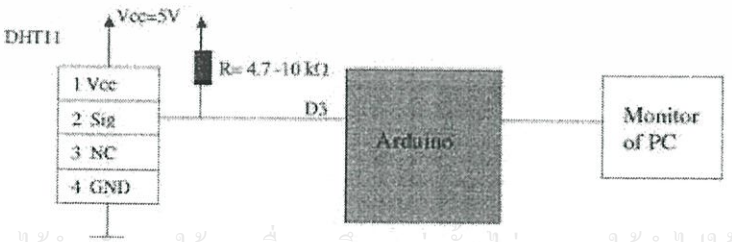
- Power supply : 3-5.5V DC
- Output signal : digital signal via single-bus
- Measuring range : humidity 20-90%RH temperature 0-50 Celsius
- Accuracy : humidity +-4%RH (Max +-5%RH) ,temperature +-2.0Celsius



รูปที่ 2.2 Humidity & temperaturesensor/module – DHT11

การต่อวงจร

สำหรับการต่อ DHT11 กับ Arduino นั้นใช้สายไฟเพียง 3 เส้นเท่านั้นคือ ขา 1 ต่อ Vcc ขา 2 ต่อ D3 และ ขา 4 ต่อ GND เป็นดังรูปที่ 2.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 การต่อโมดุล DHT11 กับบอร์ด Arduino

2.1.2 ส่วนของเซนเซอร์ตรวจจับแสง

เซนเซอร์แสงเป็นอุปกรณ์ที่เปลี่ยนความเข้มแสงให้เป็นสัญญาณทางไฟฟ้า เพื่อส่งสัญญาณให้กับส่วนควบคุมของเครื่องไฟฟ้าหรืออุปกรณ์ทำงานต่าง ๆ เซนเซอร์แสงนั้นภายในจะประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ แต่อุปกรณ์ที่สำคัญในการเปลี่ยนความเข้มแสงนั้นก็คืออุปกรณ์ที่มีชื่อว่า แอลดีอาร์

- ตัวต้านทานแปรค่าตามแสง หรือแอลดีอาร์

ตัวต้านทานแปรค่าตามแสง หรือแอลดีอาร์ (LDR : Light Dependent Resistor) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่เปลี่ยนค่าความเข้มแสงไปเป็นค่าความต้านทานทางไฟฟ้า มีรูปร่างและขนาดหลายแบบ โดยเมื่อมีความเข้มแสงมาตกกระทบสูงจะทำให้มีค่าความต้านทานทางไฟฟ้าต่ำ แต่เมื่อมีความเข้มแสงที่ตกกระทบน้อยจะทำให้มีค่าความต้านทานทางไฟฟ้าสูง ทำให้เราสามารถนำหลักการทำงานนี้ไปประยุกต์ใช้ในงานด้านต่าง ๆ ได้ โดยเราต้องต่อแอลดีอาร์ร่วมกับอุปกรณ์อื่น เช่น ทรานซิสเตอร์เพื่อช่วยในการทำงาน

ในโครงงานนี้เราใช้แผงวงจรตรวจจับแสง : LDR Light sensorใช้ตรวจจับแสงสว่าง ดังรูปที่ 2.4 เลือกเอาต์พุตได้ 2 แบบคือ

- แรงดันเอาต์พุตเพิ่ม เมื่อแสงตกกระทบ
- แรงดันเอาต์พุตลดลง เมื่อแสงตกกระทบ

รูปที่ 2.4 แผงวงจรตรวจจับแสง : LDR Light sensor

2.2 ส่วนของการปรับแต่งสภาพสัญญาณและการเชื่อมต่อสัญญาณที่ได้จากเซนเซอร์

ในส่วนของการปรับแต่งสัญญาณและการเชื่อมต่อสัญญาณที่ได้จากเซนเซอร์จะแยกออกเป็น 2 ส่วน คือ ส่วนการแปลงสัญญาณ ส่วนที่สองเป็นส่วนของการเชื่อมต่อสัญญาณ

2.2.1 การแปลงสัญญาณ A/D และ D/A

การสื่อสารข้อมูลด้วยคอมพิวเตอร์ สามารถสื่อสารข้อมูลได้ทุกประเภทประกอบด้วยเสียง (Voice) อักขระข้อความ (Text), ภาพ (Image) และข้อมูลคอมพิวเตอร์ (Data) ซึ่งแต่ละข้อมูล มีลักษณะเฉพาะของสัญญาณที่แตกต่างกัน แบ่งการกระทำของข้อมูลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Analog Computer

สัญญาณอนาลอกคือ สัญญาณข้อมูลแบบต่อเนื่อง (Continuouse Data) มีขนาดของสัญญาณไม่คงที่ การเปลี่ยนแปลงขนาดของสัญญาณแบบค่อยเป็นค่อยไปแปรผันตามเวลาเป็นสัญญาณที่มนุษย์สามารถสัมผัสได้ เช่น แรงดันของน้ำ

- Digital Computer

สัญญาณดิจิตัล คือ สัญญาณข้อมูลแบบไม่ต่อเนื่อง (Discrete Data) มีขนาดของสัญญาณคงที่ การเปลี่ยนแปลงขนาดของสัญญาณเป็นแบบทันที ทันใด ไม่แปรผันตามเวลาเป็นสัญญาณที่มนุษย์ไม่สามารถสัมผัสได้ เช่น สัญญาณไฟฟ้า

ความสัมพันธ์ของสัญญาณอะนาลอก ดิจิตอล และตัวแปลงสัญญาณ

สัญญาอะนาลอก (Analog) และสัญญาณดิจิตอล (Digital)ทั้งสองสัญญาณ เกี่ยวข้องกับตัวแปลงสัญญาณ (Transducer)การเชื่อมต่อระบบอนาลอกเข้าสู่ระบบคอมพิวเตอร์ จะต้องมีตัวกลางใน การแปลงเปลี่ยนจากAnalogให้เป็นสัญญาณทางอิเล็กทรอนิกส์ เรียกว่า “ทรานส์ดิวเซอร์”(Transducer) การแปลงสัญญาณกลับไปกลับมาระหว่างสัญญาณ Analog และ Digital อาศัย "ตัวเปลี่ยนสัญญาณข้อมูล Converter"

การแปลงสัญญาณมี 2 วิธีคือ

- การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล
- การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

Analog to Digital Converter (A/D)ทำหน้าที่แปลงสัญญาณข้อมูลที่ มนุษย์รับรู้ สัมผัสได้เป็นข้อมูลทางไฟฟ้า เพื่อป้อนเข้าสู่การประมวลผล จึงเป็นขบวนการหนึ่งของการรับข้อมูล (Input Unit)เป็นกระบวนการอีเลคโทรนิคส์ ที่สัญญาแปรผันต่อเนื่อง (analog) ได้รับการแปลงให้เป็นสัญญาณดิจิตอล โดยไม่มีการลบข้อมูลสำคัญผลลัพธ์ของ ADC มีลักษณะตรงข้าม คือ กำหนดระดับหรือสถานะ ตัวเลขของสถานะมักจะเป็นการยกกำลังของ 2 คือ 2, 4, 8, 16 เป็นต้น สัญญาณดิจิตอลพื้นฐานมี 2 สถานะและเรียกว่า binary ตัวเลขทั้งหมดสามารถแสดงในรูปของไบนารี ในฐานะข้อความของ หนึ่งและศูนย์

วงจรที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลมีมากมายหลายชนิด โดยทั่วไปแล้ววงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D converters) มีใช้งานอยู่ประมาณ 7 ชนิดคือ

- Parallel Comparator, Simultaneous, หรือ Flash A/D converter
- Single – Ramp หรือ Single – Slope A/D converter
- Dual – Slope A/D converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Charge balance A/D converter
- A/D converters using Counters and D/A converters
- Tracking A/D converters
- Successive – Approximation A/D converters
- Counting Converter

เป็นการแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิตอล โดยใช้อัลกอริทึม การนับค่าเพิ่มขึ้นเรื่อยๆ แล้วนำผลที่ได้จากการนับไปเปรียบเทียบกับค่าที่ต้องการที่ตั้งไว้ การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล มีประโยชน์มากในการควบคุมอุปกรณ์สวิตชิ่ง ซึ่งมีลักษณะการแปลงสัญญาณได้หลายวิธี แต่ละวิธีจะมีอัลกอริทึม ความรวดเร็วในการทำงาน และการใช้อุปกรณ์ฮาร์ดแวร์ต่างกันด้วย

ข้อบ่งเฉพาะของการแปลงสัญญาณ A/D (A/D SPECIFICATIONS)

ข้อบ่งเฉพาะจะบอกถึงขีดความสามารถของ converter โดยทั่วไปแล้วจะมีอยู่หลายคำ เช่น ความแม่นยำ,ความเที่ยงตรง และความเที่ยงตรงเป็นเส้นตรง ซึ่งค่าเหล่านี้จะขึ้นอยู่กับลักษณะของแต่ละวงจร แต่มีข้อบ่งเฉพาะอีกข้อหนึ่งที่ไม่ขึ้นอยู่กับลักษณะของวงจรคือ ค่าผิดพลาดระหว่างค่าจริงของสัญญาณอะนาลอก กับค่าของดิจิตอลที่ใช้แทนค่า (ค่าของ Output ของ A/D converter) ซึ่งเรียกว่า Quantizing error จะมีค่าอยู่ประมาณ +1/2 digit ต่ำสุด (LSB)ของการแปลงสัญญาณซึ่งก็เป็นการบ่งถึงความแม่นยำได้อีกทางหนึ่งด้วยค่า พารามิเตอร์ที่สำคัญอีกตัวหนึ่งสำหรับ A/D converter คือ conversiontime หรือค่าเวลาสำหรับการแปลงสัญญาณ ซึ่งมีช่วงเวลาอยู่ประมาณ 10-9 วินาที ถึง 10-3 วินาที ซึ่งขึ้นอยู่กับชนิดของconverterและจำนวน bit

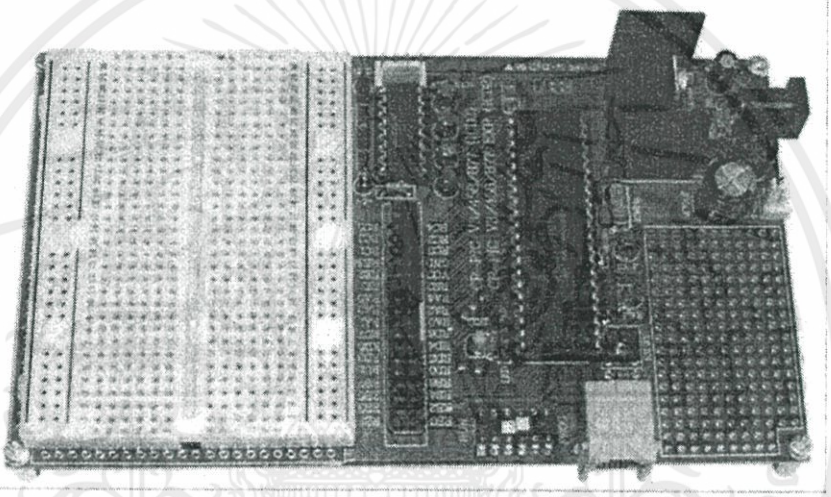
การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

Digital to Analog Converter (D/A) ทำหน้าที่แปลงข้อมูลผลลัพธ์จากการประมวลผลเป็นสัญญาณไฟฟ้า ให้เป็นสัญญาณที่มนุษย์รับรู้ได้ สัมผัสได้ เป็นการแสดงผลข้อมูล (Output Unit)digital-to-analog conversion เป็นกระบวนการซึ่งสัญญาณมีการกำหนดระดับ หรือสถานะจำนวนหนึ่ง (ปกติ คือ 2 สถานะ) หรือสัญญาณดิจิตอล ให้เป็นสัญญาณที่ไม่จำกัดจำนวนของสถานะหรือสัญญาณอนาลอก ตัวอย่าง กระบวนการของโมเด็มในการแปลงข้อมูลคอมพิวเตอร์ เป็นความถี่เสียง ให้สามารถส่งผ่านสายโทรศัพท์ twisted pair ในวงจรที่ทำงานให้กับฟังก์ชันนี้ เรียกว่า digital-to-analog converter (DAC) โดยพื้นฐาน digital-to-analog conversion ตรงข้ามกับ analog-to-analog conversion ถ้า analog-to-analog converter (ADC) วางอยู่ในวงจรการสื่อสารต่อจาก DAC สัญญาณดิจิตอลส่งออก จะตรงกับสัญญาณดิจิตอลนำเข้า ในกรณีที่ DAC วางอยู่ในวงจรต่อจาก ADC สัญญาณอะนาล็อกส่งออกจะเป็นตรงกับสัญญาณอะนาล็อกนำเข้าสัญญาณดิจิตอล แบบ binary จะปรากฏเป็นข้อความขนาดยาว ของ 1 และ 0 ซึ่งจะไม่มีความหมายต่อการอ่าน แต่เมื่อ DAC ใช้ถอดรหัสสัญญาณดิจิตอลแบบ binary จึงปรากฏผลลัพธ์ที่มีความหมาย ซึ่งอาจจะเป็น เสียง ภาพ เสียงดนตรี และกลไกการเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้เราใช้ CP-PIC V3.0 (ICD2) ดังรูปที่ 2.5 เป็นบอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกแบบวงจรเฉพาะส่วนพื้นฐานที่จำเป็นเช่น แหล่งจ่ายไฟ วงจรรีเซ็ต วงจรกำเนิดความถี่สัญญาณนาฬิกา พอร์ตสำหรับ Download โปรแกรม และวงจรสื่อสารอนุกรม ส่วนวงจร I/O ภายนอกนั้น จะไม่ได้จัดเตรียมไว้ให้ด้วย แต่จะทำการต่อสัญญาณ I/O ต่างๆจาก CPU มาไว้ยังขั้วต่อConnector สำหรับให้ผู้ใช้นำไปเชื่อมต่อกับอุปกรณ์ I/O ภายนอกได้โดยง่าย และยังมีพื้นที่เอนกประสงค์สำหรับให้ผู้ใช้ออกแบบวงจร I/O และต่อวงจร I/O เพิ่มเติมได้เอง เหมาะสำหรับผู้ใช้ที่ต้องการนำบอร์ดไปใช้พัฒนางานต้นแบบโดยการสร้าง I/O ต่างๆขึ้นมาใช้งานเอง ซึ่งในบอร์ดจะประกอบด้วย

- RS – 232 1 แชนเนล
- ETT CON 34PIN (ET BUS I/O 34PIN)
- 5 Volt Regulator On Board



รูปที่ 2.5 บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 (ICD2)

2.2.2 ส่วนของตัวทำหน้าที่เชื่อมต่อสัญญาณ

Serial Port คือ พอร์ตอนุกรม ในการสื่อสารข้อมูลนั้นพอร์ตอนุกรมจะมีความเร็วในการสื่อสารที่ช้ากว่าแบบ ขนาน เพราะการเคลื่อนย้ายข้อมูลแบบอนุกรมนั้นเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลทีละหลายๆ บิต พร้อมๆ กันได้ แต่ข้อดีของการสื่อสารข้อมูลแบบอนุกรมคือ สามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน และใช้สายสัญญาณที่น้อยกว่าการสื่อสารข้อมูลแบบขนาน

มาตรฐาน RS-232-C เป็นมาตรฐาน RS-232 ที่มีการปรับปรุงแก้ไขจากมาตรฐานเดิม ซึ่งเราอาจคุ้นเคยกับชื่อนี้มากกว่า RS-232-A หรือ RS-232-B อันที่จริงแล้วยังมีมาตรฐาน RS- 232-D ที่ใหม่กว่า RS-232-C โดยที่มีการเพิ่มข้อกำหนดของคอนเน็กเตอร์แบบ DB เข้าไปด้วย เช่น DB-25 ซึ่งในขณะนั้นสิทธิบัตรของตัวคอนเน็กเตอร์แบบนี้ได้หมดอายุลงพอดี

ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งจริงๆแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งสองฝั่งจะเป็นอะไรก็ได้ การสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) หรือไม่ก็ได้ มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น และถ้ามีสัญญาณรบกวนมากๆ เช่นในโรงงาน หรือบริเวณใกล้เครื่องจักรที่เป็นแบบมีการสวิทช์สัญญาณไฟฟ้าที่กระแสสูงๆ ก็จะทำให้ต้องมีการลดความเร็วในการส่งสัญญาณลงหรือใช้สายที่สั้นลงแสดงการจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ

ประเภทของการสื่อสารแบบอนุกรมแบ่งตามลักษณะสัญญาณในการส่งแบ่งได้ 2 แบบคือ

- การสื่อสารแบบซิงโครนัส (Synchronous) เป็นการสื่อสารข้อมูลโดยใช้สัญญาณนาฬิกาในการควบคุมจังหวะของการรับส่งสัญญาณ
- การสื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการสื่อสารที่ใช้สายข้อมูลเพียงตัวเดียว จะใช้รูปแบบของการส่งข้อมูล(Bit Pattern) เป็นตัวกำหนดว่าส่วนไหนเป็นส่วนเริ่มต้นข้อมูล ส่วนไหนเป็นตัวข้อมูล ส่วนไหนจะเป็นตัวตรวจสอบความถูกต้องของข้อมูลและส่วนไหนเป็นส่วนปิดท้ายของข้อมูลโดยต้องกำหนดให้สัญญาณนาฬิกาเท่ากันทั้งภาคส่งและภาครับ

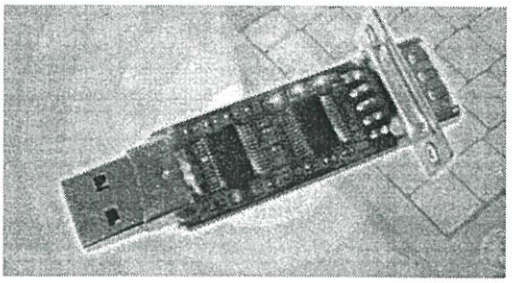
ในโครงงานนี้เราใช้ ET-USB/RS232 MINI ดังรูปที่ 2.6 ที่ตรงตามความต้องการของโครงงาน
คุณสมบัติของ ET-USB/RS232 MINI

- รองรับมาตรฐาน USB 1.1 และ USB 2.0
- อัตราการรับส่งข้อมูล 300 bps ถึง 128 kbps
- ใช้ไฟเลี้ยงจาก USB Port โดยตรงไม่ต้องต่อเพิ่มภายนอก
- USB Connector แบบ TYPE A, RS232 Connector แบบ DP9 ตัวผู้ ดังรูปที่ 2.7
- แสดงสถานการณ์ทำงานด้วย LED 3 สี คือ การรับ (RX) สีเขียว, การส่ง (TX) สีเหลือง และ Power (PWR) สีแดง
- Driver รองรับ Windows 98SE/ME/2000/XP
- ขนาด กว้าง 3 cm x ยาว 6.8 cm

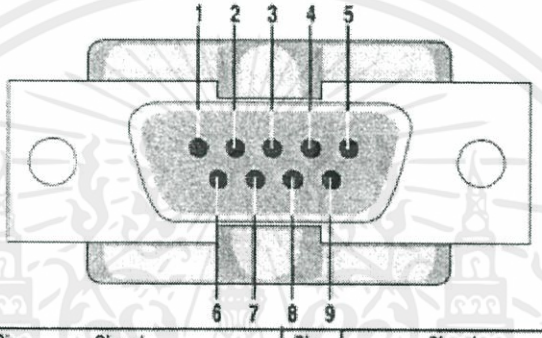
การต่อใช้งาน ET-USB/RS232 MINI

- เสียบ ET-USB/RS232 MINI เข้ากับ Port USB ของคอมพิวเตอร์ ขณะนี้จะสังเกตเห็นว่าไฟแสดงสถานะ PWR จะยังไม่ติดสว่าง
- ติดตั้ง Driver สำหรับ ET-USB/RS232 MINI โดยจากคู่มือการติดตั้ง Driver เมื่อติดตั้งเรียบร้อยแล้ว ไฟแสดงสถานะ PWR จะติดสว่างแสดงว่าพร้อมที่จะใช้งานแล้ว
- ต่อสายพอร์ตอนุกรม เพื่อใช้งานโดยตำแหน่งขาใช้งาน จะเป็นดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

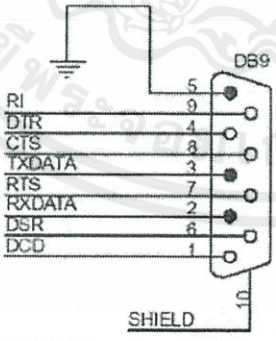


รูปที่ 2.6 ET-USB/RS232 MINI



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

รูปที่ 2.7 การเชื่อมต่ออุปกรณ์อุกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9



Pin	Name	Dir	Description
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	—	Signal Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

รูปที่ 2.8 ตำแหน่งขาที่จะใช้งาน

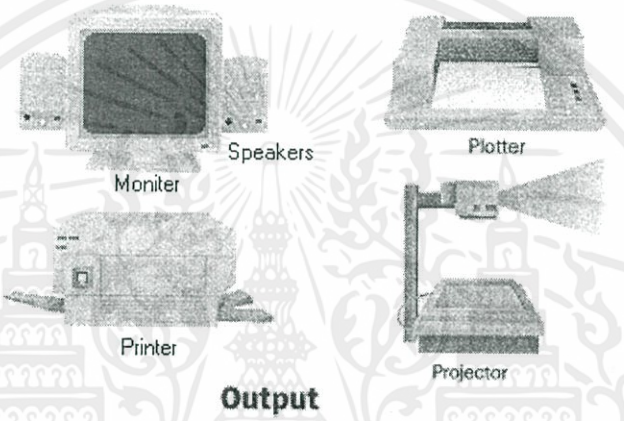
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนของการแสดงผลและบันทึกข้อมูล

ในส่วนของการแสดงผลและบันทึกข้อมูลจะเป็นหัวใจหลักของดาต้าล็อกเกอร์ เราจะใช้โปรแกรมวิชชวล (visual studio 2010) ในการแสดงผล และใช้โปรแกรมSQL ในการสร้างฐานข้อมูลแล้วทำการบันทึกจัดเก็บ

2.3.1 หน่วยแสดงผล (Output Unit)

หน่วยแสดงผลชั่วคราว (Soft Copy) และ หน่วยแสดงผลถาวร (Hard Copy) ดังรูปที่ 2.9



รูปที่ 2.9 หน่วยแสดงผล (Output Unit)

หน่วยแสดงผลชั่วคราว (Soft Copy)

หมายถึงการแสดงผลออกมาให้ผู้ใช้ได้รับทราบในขณะนั้น แต่เมื่อเลิกการทำงานหรือเลิกใช้แล้วผลนั้นก็จะหายไป ไม่เหลือเป็นวัตถุให้เก็บได้ แต่ถ้าต้องการเก็บผลลัพธ์นั้นก็สามารถส่งถ่ายไปเก็บในรูปของข้อมูลในหน่วยเก็บข้อมูลสำรอง เพื่อให้สามารถใช้งานในภายหลัง หน่วยแสดงผลที่จัดอยู่ในกลุ่มนี้ คือ

จอภาพ (Monitor)

ใช้แสดงข้อมูลหรือผลลัพธ์ให้ผู้ใช้เห็นได้ทันที มีรูปร่างคล้ายจอภาพของโทรทัศน์ บนจอภาพประกอบด้วยจุดจำนวนมากมาย เรียกจุดเหล่านั้นว่า พิกเซล (pixel) ถ้ามีพิเซลจำนวนมากก็จะทำให้ผู้ใช้มาองเห็นภาพบนจอได้ชัดเจนมากขึ้น จอภาพที่ใช้ในปัจจุบันแบ่งได้เป็นสองประเภท คือ จอซีอาร์ที (Cathode Ray Tube) นิยมใช้กับเครื่องไมโครคอมพิวเตอร์ส่วนมากในปัจจุบัน ใช้หลักการยิงแสงผ่านหลอดภาพคล้ายกับโทรทัศน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จอแอลซีดี (Liquid Crystal Display) นิยมใช้เป็นจอภาพของเครื่องคอมพิวเตอร์แบบพกพาทำให้เป็นจอภาพที่มีความหนาไม่มาก มีน้ำหนักเบาและกินไฟน้อยกว่า

จอภาพซีอาร์ที แต่มีราคาสูงกว่าเทคโนโลยีจอแอลซีดีในปัจจุบันจะมีสองแบบคือ Passive Matrix ซึ่งมีราคาต่ำ แต่ขาดความคมชัดและอาจมองไม่เห็นภาพเมื่อผู้ใช้มองจากบางมุม ส่วน Aciive Matrix หรือบางครั้งอาจเรียกว่า Thin Film Transistor (TFT) จะให้ภาพที่คมชัดกว่าแต่จะมีราคาสูงกว่ามาก

สมัยก่อนมีจอภาพระบบขาวดำ ซึ่งเรียกว่าจอโมโนโครม (Monochrome) แต่ปัจจุบันนี้ซอฟต์แวร์ส่วนมากจะใช้ร่วมกับจอภาพชนิดสีเท่านั้นซึ่งจะมีจอภาพอยู่หลายชนิดให้เลือกโดยแตกต่างกันในส่วนของความละเอียด(Resolution) จำนวนสี (color) และ ขนาดของจอภาพ (size)

ในส่วนความละเอียดของจอภาพ ในปัจจุบันจะนิยมใช้จอภาพชนิดสีแบบ Super Video Graphic Adapter หรือเรียกสั้น ๆ ว่า ซูเปอร์วีจีเอ (Super VGA) ซึ่งมีความละเอียด 800x600 พิกเซล สำหรับจอภาพที่มีความละเอียดต่ำ (low resolution) และสำหรับจอภาพที่มีความละเอียดสูง จะนิยมใช้ความละเอียดที่ 1024x768,1280x1024 หรือ 1600x1200 พิกเซล ซึ่งจะให้ความคมชัดที่สูงมาก สิ่งที่เป็นปัจจัยอีกอันหนึ่งที่ทำให้ภาพดูคมชัดมากขึ้นถึงแม้ว่าจะมีจำนวนพิกเซลเท่ากันก็คือ ระยะห่างระหว่างพิกเซล (dot pitch) โดยระยะห่างระหว่างพิกเซลน้อยก็จะให้ความละเอียดได้มากกว่า จอภาพที่มีขายในท้องตลาดปัจจุบันมีระยะห่างระหว่างพิกเซลอยู่ระหว่าง 0.25-0.28 หน่วย ซึ่งระยะห่างระหว่างพิกเซลนี้เป็นสิ่งที่ติดมากับเครื่องไม่สามารถเปลี่ยนแปลงได้

ในส่วนของจำนวนสีนั้น ณ ขณะใดขณะหนึ่งแต่ละพิเซลจะแสดงสีได้เพียงสีเดียวเท่านั้น ซึ่งสีต่าง ๆ จะถูกแทนด้วยตัวเลข ดังนั้นถ้าจอภาพแสดงได้ 16 สี เลขเหล่านั้นก็จะแทนด้วย 4 บิต ถ้าต้องการแสดงถึง 256 สีก็จะต้องใช้ 8 บิตแทนรหัสสีนั้น ๆ

ปัจจุบันนี้ผู้ใช้มักจะแสดงภาพกราฟิก ภาพจากโทรทัศน์ ภาพเคลื่อนไหวบนจอภาพคอมพิวเตอร์ จึงต้องการจอภาพที่มีขนาดใหญ่มากขึ้น จอภาพที่นิยมใช้ในปัจจุบันมีขนาด 14 นิ้ว 15 นิ้ว และ 17 นิ้ว ส่วนจอภาพซึ่งมีจนาดใหญ่กว่านี้จะนิยมใช้กับงานที่เน้นกราฟฟิก เช่น งานออกแบบ (CAD/CAM) เป็นต้น .

- จอภาพคอมพิวเตอร์และแผงวงจรกราฟฟิก

การต่อจอภาพเข้ากับเครื่องไมโครคอมพิวเตอร์นั้นจะต้องมี แผงวงจรกราฟฟิก (Graphic Adapter Board) หรือเรียกอีกอย่างหนึ่งว่า การ์ดวีดีโอ (video card) ซึ่งจอภาพแต่ละชนิดก็ต้องการแผงวงจรที่ต่างกัน แผงวงจรกราฟฟิกจะถูกเสียบเข้ากับช่องขยายเพิ่มเติม (expansion slot) ในคอมพิวเตอร์ แผงวงจรกราฟฟิกมักจะมีหน่วยความจำเฉพาะที่เรียกว่า หน่วยความจำวีดีโอ (video memory) เพื่อให้ใช้โปรแกรมด้านกราฟฟิกได้สวยงามและรวดเร็ว ซึ่งหน่วยความจำนี้อาจใช้แรมธรรมดาหรือแรมแบบพิเศษต่าง ๆ เพื่อให้สามารถทำงานได้เร็วขึ้น เช่น วีดีโอแรม (video RAM) ซึ่งบางครั้งเรียกว่า วีแรม (VRAM) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ่งที่เป็นปัจจัยข้อหนึ่งที่ผู้ใช้จอภาพต้องคำนึงคือ อัตราการเปลี่ยนภาพ (refresh rate) ของการ์ดวีดีโอ โดยภาพที่แสดงบนจอภาพแต่ละภาพนั้นจะถูกลบและแสดงภาพใหม่เริ่มจากบนลงล่าง หาก อัตรการเปลี่ยนภาพในแนวดิ่ง (Vertical-refresh rate) เป็น 60 ครั้งต่อวินาที หรือ 60 Hz จะเกิดการกระพริบทำให้ผู้ใช้ปวดศีรษะได้ มีผู้วิจัยพบว่าอัตราเปลี่ยนภาพในแนวดิ่งไม่ควรต่ำกว่า 70 Hz จึงจะไม่เกิดการกระพริบ และทำให้ผู้ใช้ดูจอภาพได้อย่างสบายตานอกจากนี้ยังมีอุปกรณ์สำหรับถอดรหัสภาพแบบ MPEG (Motion Picture Experts) ซึ่งอาจอยู่ในรูปซอฟต์แวร์หรือฮาร์ดแวร์ที่ติดอยู่บนการ์ดวีดีโอ อันจะทำให้สามารถแสดงภาพเคลื่อนไหว เช่น ภาพยนตร์ต่าง ๆ บนจอคอมพิวเตอร์ได้อย่างต่อเนื่อง

- อุปกรณ์ฉายภาพ (Projector)

เป็นอุปกรณ์ที่นิยมใช้ในการเรียนการสอนหรือการประชุม เนื่องจากสามารนำเสนอข้อมูลให้ผู้ชมจำนวนมากเห็นพร้อม ๆ กัน อุปกรณ์ฉายภาพในปัจจุบันจะมีอยู่หลายแบบ ทั้งที่สามารถต่อสัญญาณจากคอมพิวเตอร์โดยตรง หรือใช้อุปกรณ์พิเศษในการวางลงบนเครื่องฉายภาพข้ามศีรษะ (Overhead Projector) ธรรมดา เหมือนกับอุปกรณ์นั้นเป็นแผ่นใส่แผ่นหนึ่ง

อุปกรณ์ฉายภาพจะมีข้อแตกต่างกันมากในเรื่องของกำลังส่องสว่าง เนื่องจากยิ่งมีกำลังส่องสว่างสูง ภาพที่ได้ก็จะชัดเจนมากขึ้น กำลังส่องสว่างมีหน่วยวัดค่าอยู่ 3 แบบ คือ LUX , LUMEN และ ANSI LUMEN โดยการวัดค่าแบบ LUX จะวัดค่าความสว่างที่จุดกึ่งกลางของภาพ จึงได้ค่าความสว่างสูงที่สุดเมื่อเทียบกับอีก 2 แบบ การวัดแบบ จะแบ่งภาพออกเป็น 3 ส่วน คือ บน กลางและล่าง และแต่ละส่วนจะถูกแบ่งออกเป็น 3 จุด คือ ริมซ้าย กลาง และริมขวา รวมจุดภาพทั้งหมด 9 จุด แล้วจึงใช้ค่าเฉลี่ยของความสว่างทั้ง 9 จุดคิดออกมาเป็นค่า LUMEN ส่วนการวัดแบบ ANSI LUMEN จะมีมาตรฐานสูงสุด โดยใช้วิธีเดียวกับ แต่จะกำหนดขนาดจอภาพไว้คงที่คือ 40 นิ้ว (หากไม่กำหนด การวัดค่าความสว่างจะสูงขึ้นเมื่อจอภาพมีขนาดเล็กลง)

หน่วยแสดงผลถาวร (Hard Copy)

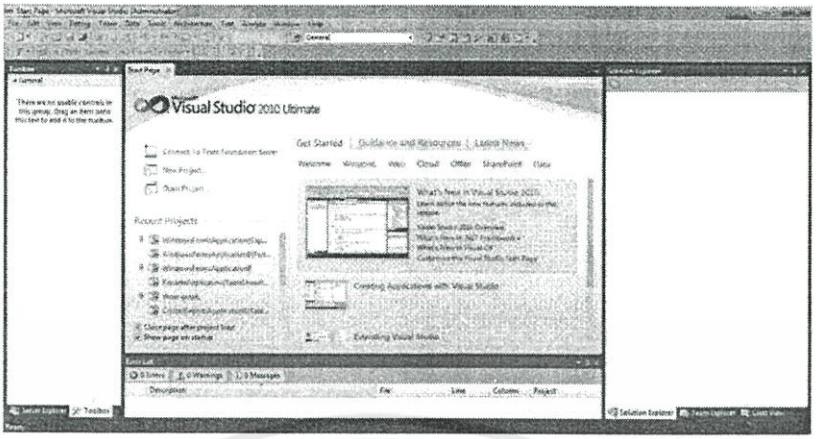
หมายถึงการแสดงผลที่สามารถจับต้อง และเคลื่อนย้ายได้ตามต้องการ มักจะออกมาในรูปของกระดาษ ซึ่งผู้ใช้สามารถนำไปใช้ในที่ต่าง ๆ หรือให้ผู้ร่วมงานดูในที่ใด ๆ ก็ได้ อุปกรณ์ที่ใช้ เช่น

เครื่องพิมพ์ (Printer)

เป็นอุปกรณ์ที่นิยมใช้กันมากและมีให้เลือกหลายชนิดขึ้นอยู่กับคุณภาพของตัวอักษรความเร็วในการพิมพ์ และเทคโนโลยีที่ใช้งาน

ในโครงงานนี้เราใช้โปรแกรมวิชชวล (visual studio 2010) ใช้ส่วนของบันทึกข้อมูลจะดังรูปที่ 2.10 ในการแสดงผลออกทางจอภาพ

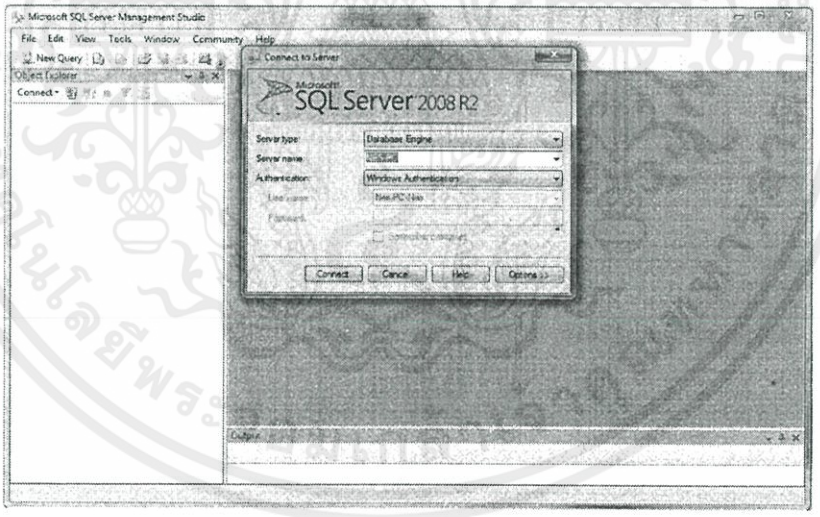
เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆ ทั้งสิ้น ... ไม่อนุญาตให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 โปรแกรมวิชชวล (visual studio 2010)

2.3.2 ส่วนของฐานข้อมูลและจัดก็บข้อมูลด้วยโปรแกรม (SQL)

ส่วนของบันทึกข้อมูลจะใช้โปรแกรม SQL server 2008 R2 ดังรูปที่ 2.11 ในการสร้างฐานข้อมูลแล้วทำการบันทึกจัดเก็บ



รูปที่ 2.11 โปรแกรมSQL server 2008 R2

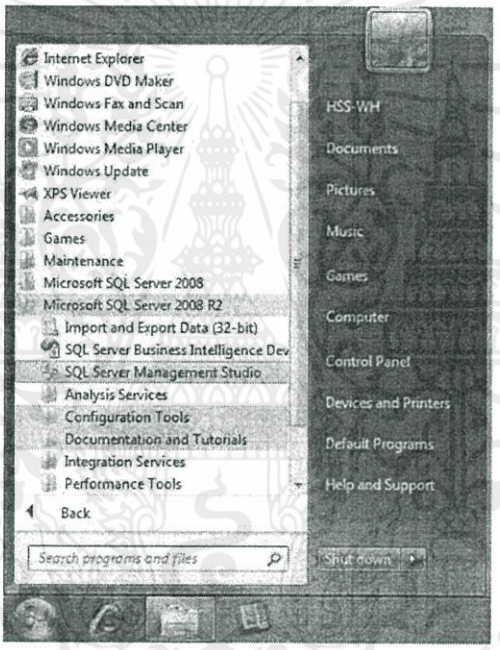
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการออกแบบ

การออกแบบระบบดาต้าล็อกเกอร์ ในเบื้องต้นนั้นต้องสร้างฐานข้อมูลก่อน เพื่อใช้ในการใช้ในการประมวลผลภาพจะเป็นลำดับถัดไป

3.1 การออกแบบตารางจัดเก็บข้อมูล

- เข้าใช้งานโปรแกรม SQL Server โดยเข้าที่ Start>All Program>Microsoft SQL Server2008 R2> SQL Server Management Studio ดังรูปที่ 3.1

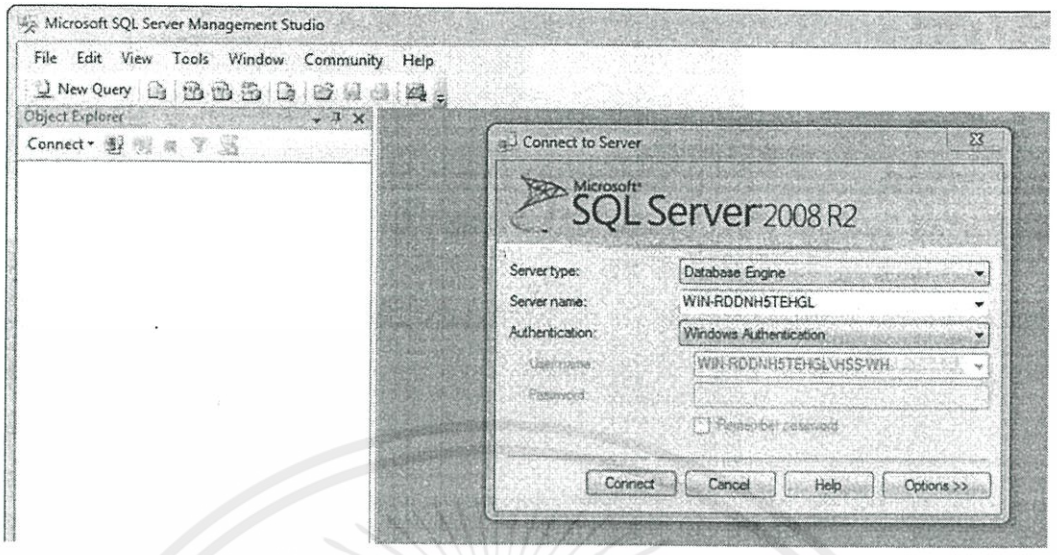


รูปที่ 3.1

- จากนั้นเข้าสู่หน้าของการ Login เพื่อใช้งานฐานข้อมูล ซึ่งสามารถจำแนกได้ 2 แบบ ดังนี้

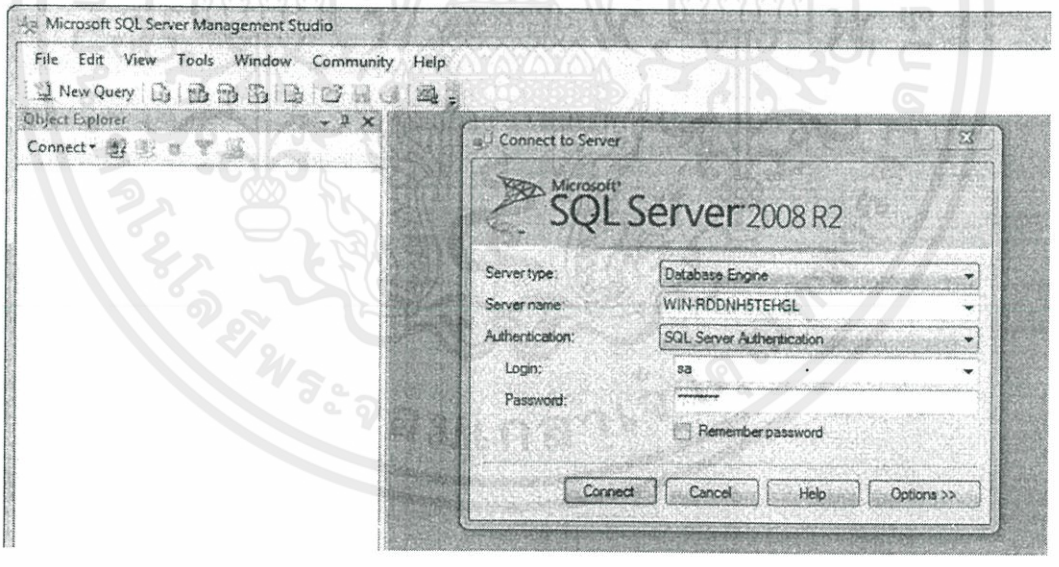
Windows Authentication ซึ่งผู้ใช้ไม่ต้องระบุ User name และ Password เนื่องจากใช้ Login เดียวกับการ Login เข้า Window จากนั้นกดปุ่ม Connect ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2

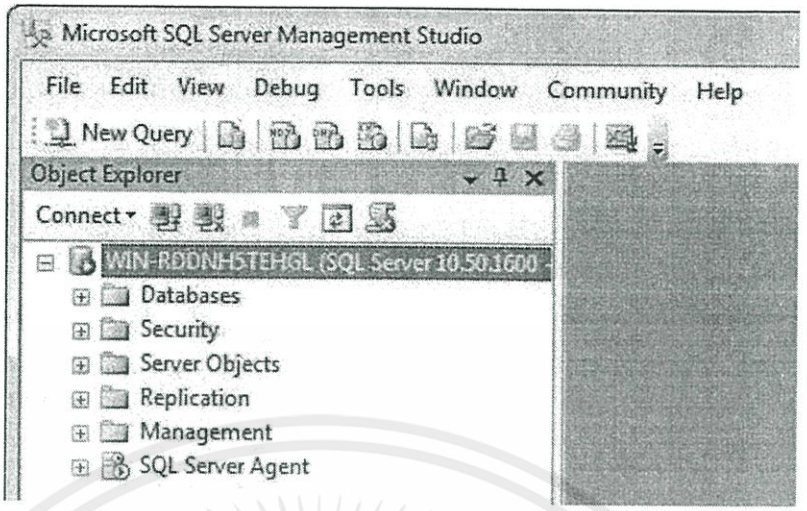
SQL Server Authentication ผู้ใช้ NES-PC ต้องระบุ User name และ Password สำหรับ Login เข้าใช้งานฐานข้อมูล จากนั้นกดปุ่ม Connect ดังรูปที่ 3.3



รูปที่ 3.3

- เมื่อ Login ด้วย 2 แบบข้างต้นแล้ว หาก Login สำเร็จจะเข้าสู่หน้าจอของโปรแกรม ดังรูปที่ 3.4

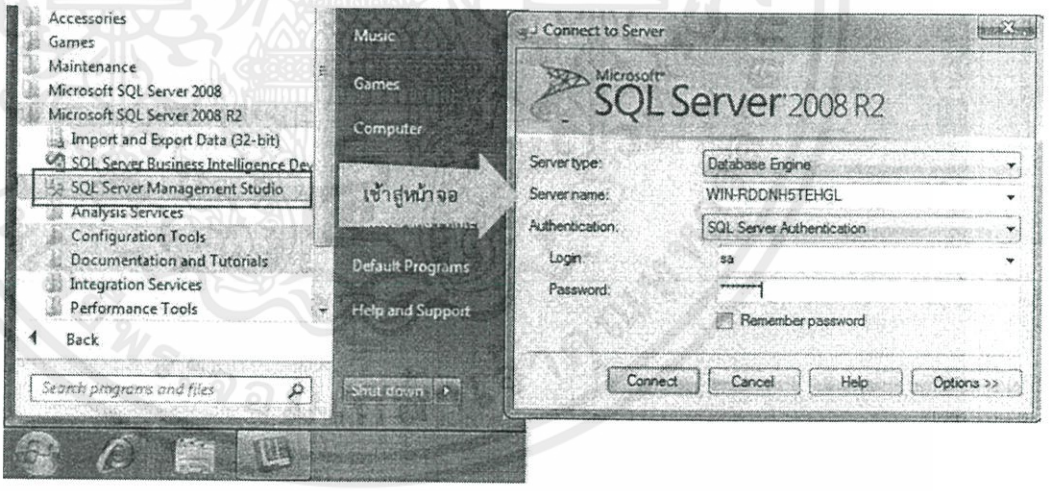
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4

3.2 การสร้างฐานข้อมูล (Data base)

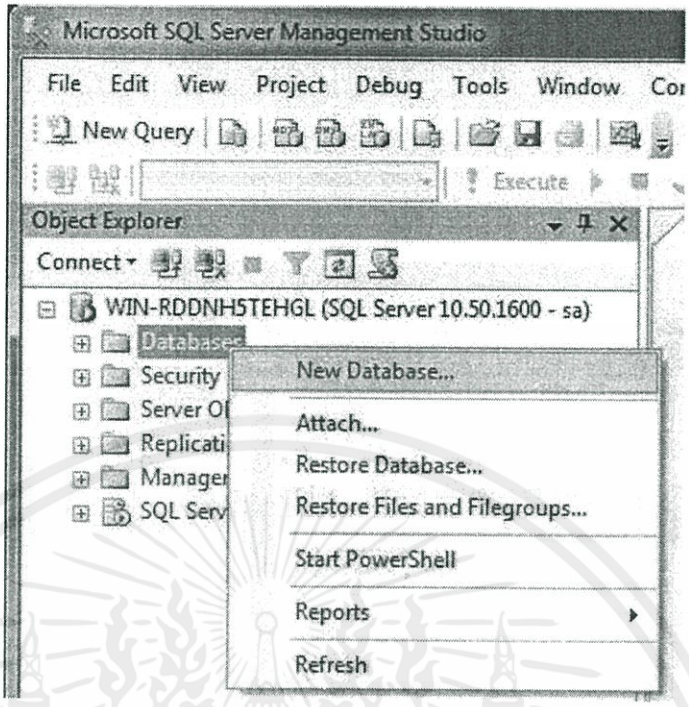
- เปิดโปรแกรม SQL Server Management Studio ไปที่ Start -> All Programs -> Microsoft SQL Server 2008 R2 -> SQL Server Management Studio > Login ด้วย User ชื่อ NES-PC ตามรูปที่ 3.5



รูปที่ 3.5

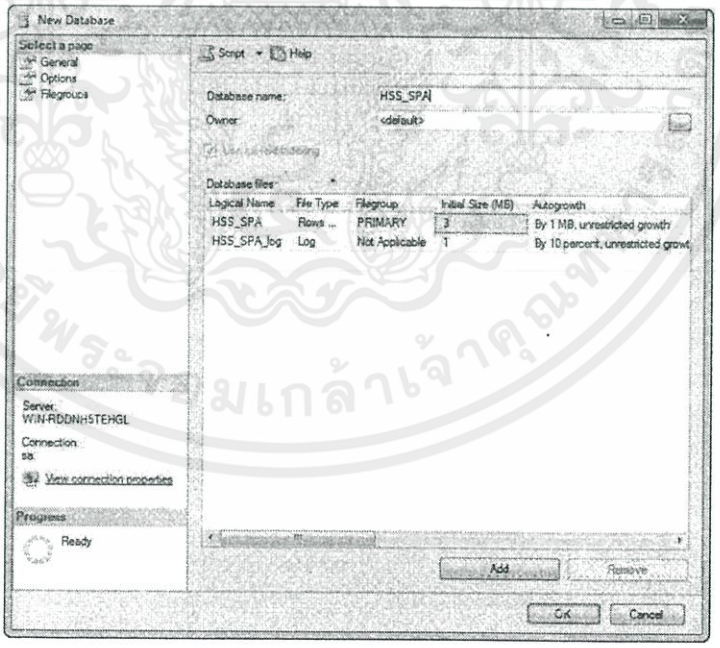
- Connect to Server
- คลิกขวาที่ Database เลือก New Database ตามรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6

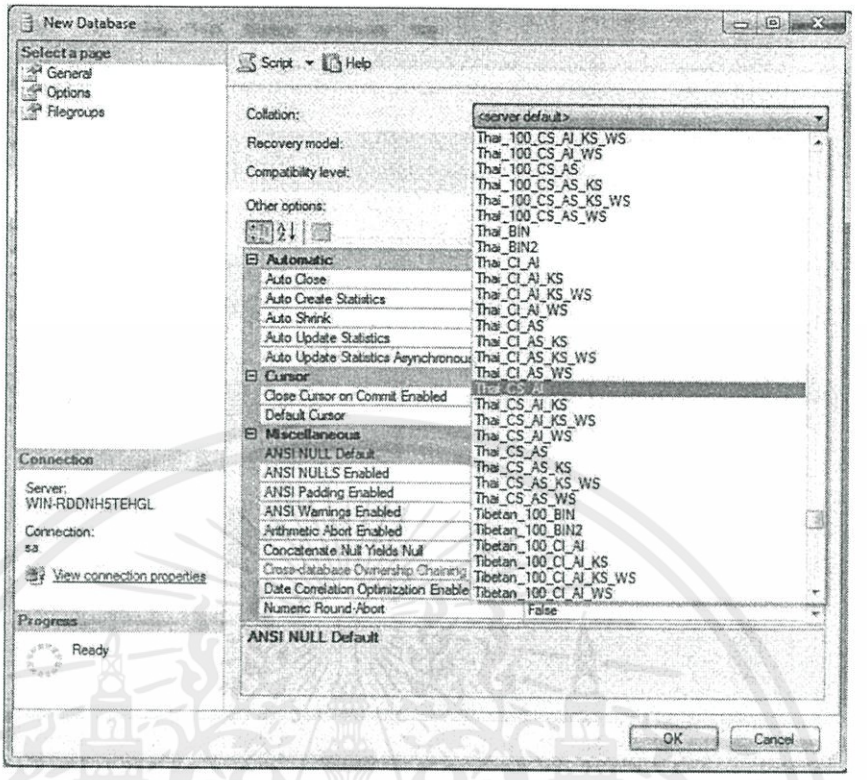
- ที่ Database Name ให้ใส่ชื่อ Database แล้วกด OK ตามรูปที่ 3.7



รูปที่ 3.7

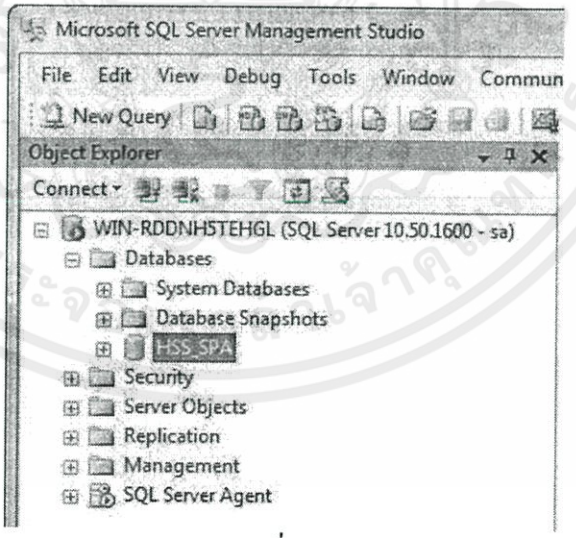
- Select a page เลือก Options ที่ช่อง Collation เลือกเป็น Thai_CS_AI แล้วกด OK เป็นการเซตให้ใช้ภาษาไทยได้ ไม่กลายเป็นเครื่องหมาย ??? ตามรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น โดยไม่ได้รับอนุญาตจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8

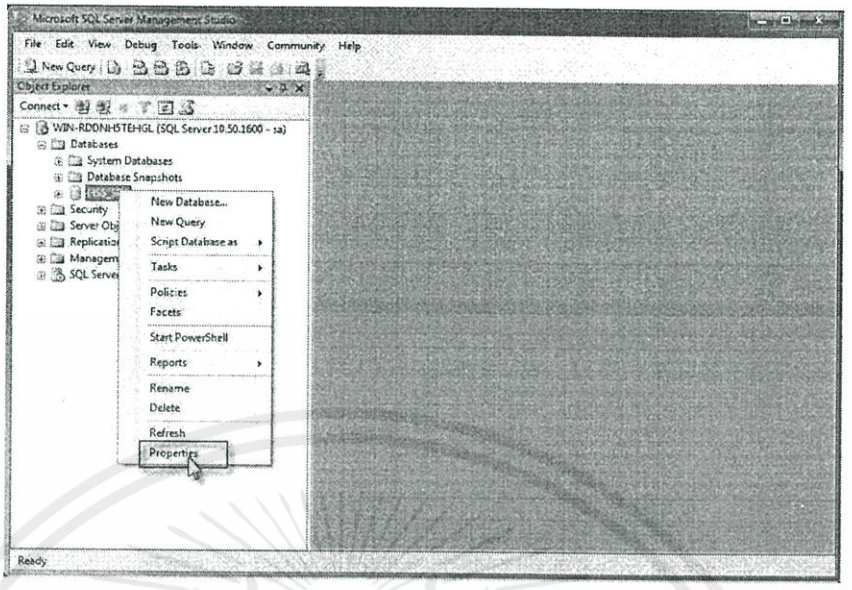
- คลิกที่ Databases จะเห็นว่ามี Database ที่เราสร้างขึ้นใหม่ชื่อว่า HSS_SPA ตามรูปที่ 3.9



รูปที่ 3.9

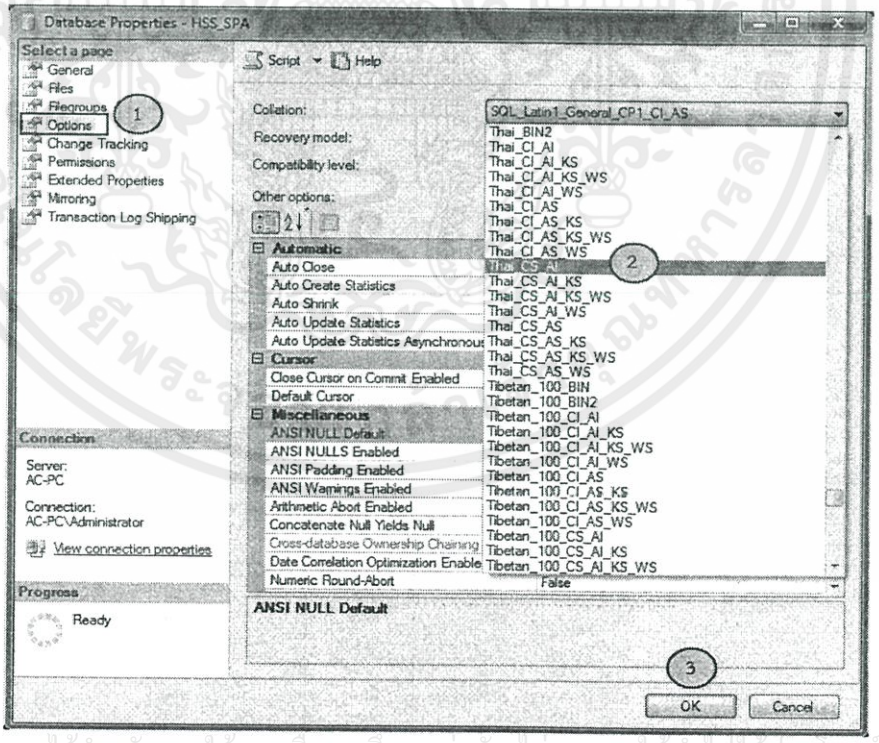
- หากต้องการแก้ไขภาษาให้กับฐานข้อมูล ทำได้โดยการคลิกขวาที่ HSS_SPA เลือก Properties ตามรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10

- จากนั้นแสดงหน้า Database Properties>Select a page ขึ้นมา เลือก Options ที่ช่อง Collation เลือกเป็น Thai_CS_AI แล้วกด OK เป็นการเซตให้ใช้ภาษาไทยได้ ไม่กลายเป็นเครื่องหมาย ??? ตามรูปที่ 3.11



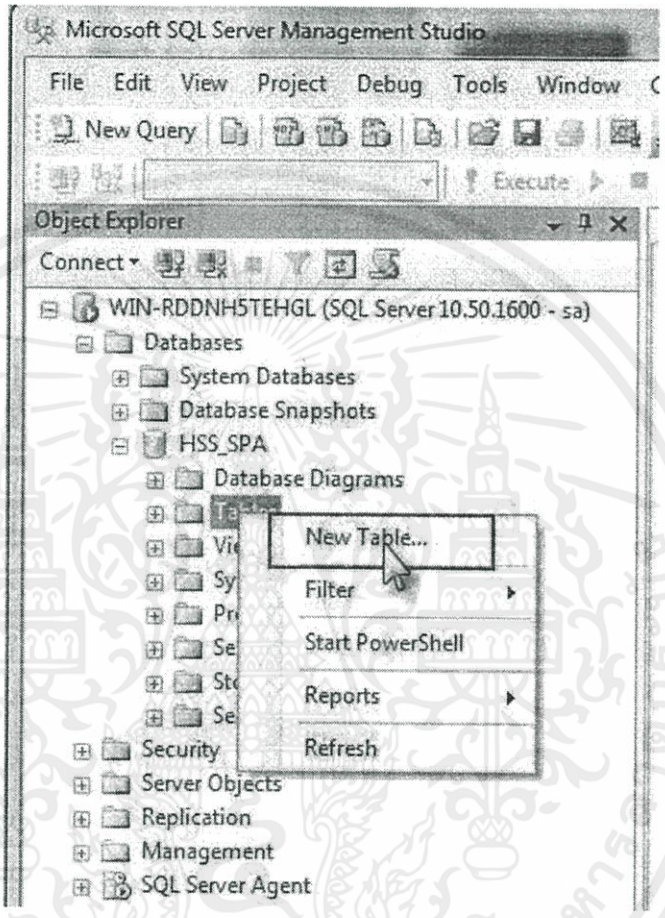
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11

3.3 การสร้างตาราง

3.3.1 การสร้างตารางแบบที่ 1

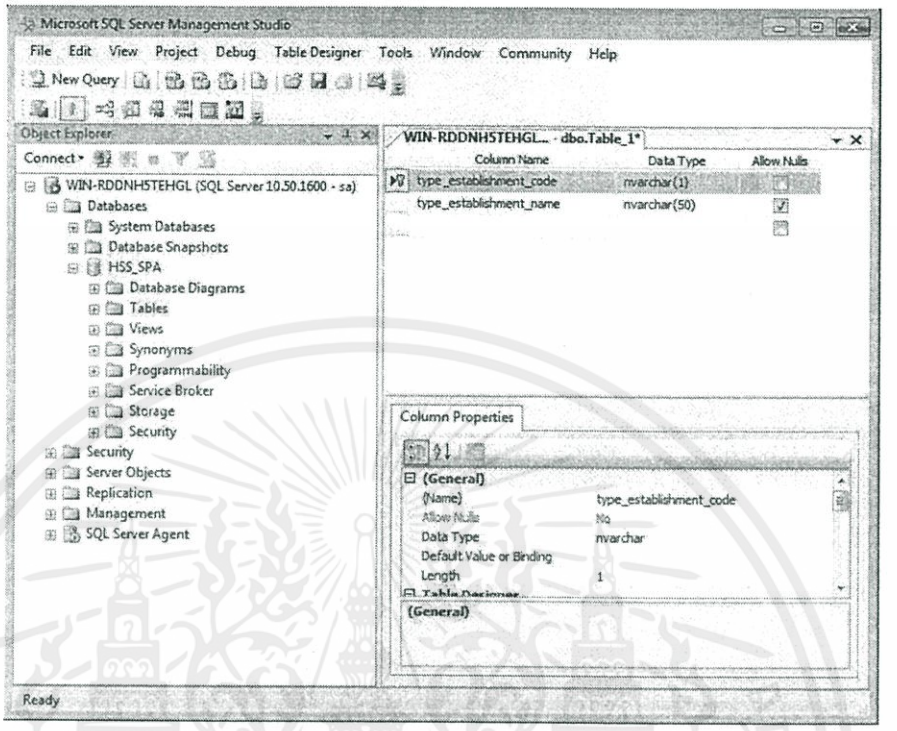
- คลิกขวาที่ Table เลือก New Table ตามรูปที่ 3.12



รูปที่ 3.12

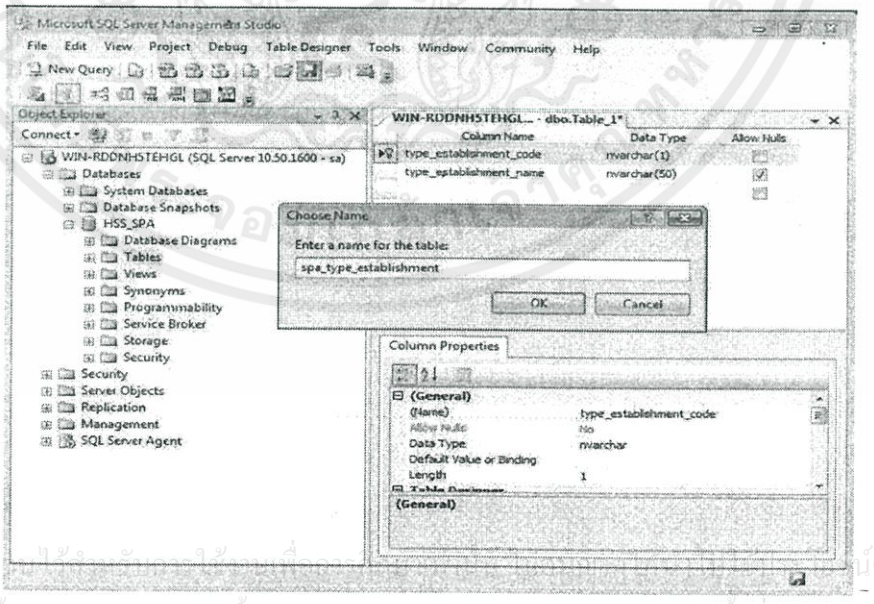
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบุ Column Name, Data Type และ Allow Nulls ตามรูปที่ 3.13



รูปที่ 3.13

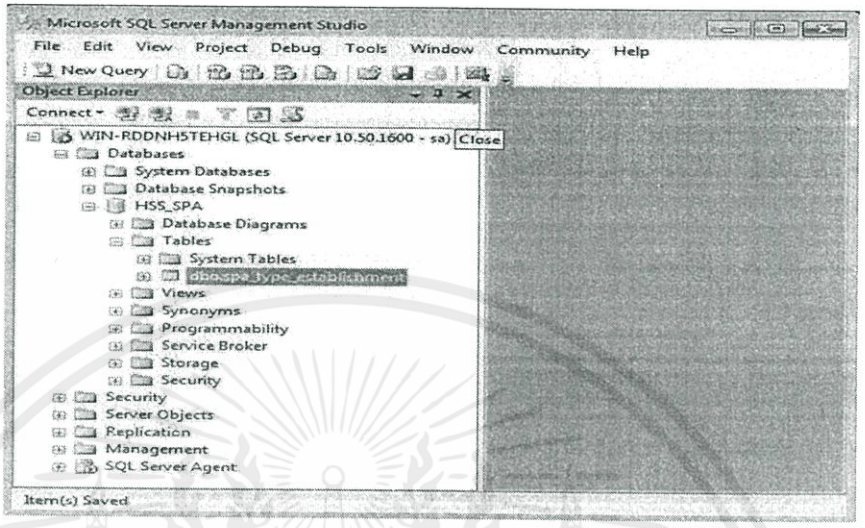
- เมื่อสร้างชื่อฟิลด์ครบถ้วนแล้ว กดปุ่ม จากนั้นแสดงกล่องข้อความให้ผู้ใช้ระบุชื่อตาราง และกดปุ่ม OK ตามรูปที่ 3.14



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14

- แสดงตาราง ตามรูปที่ 3.15



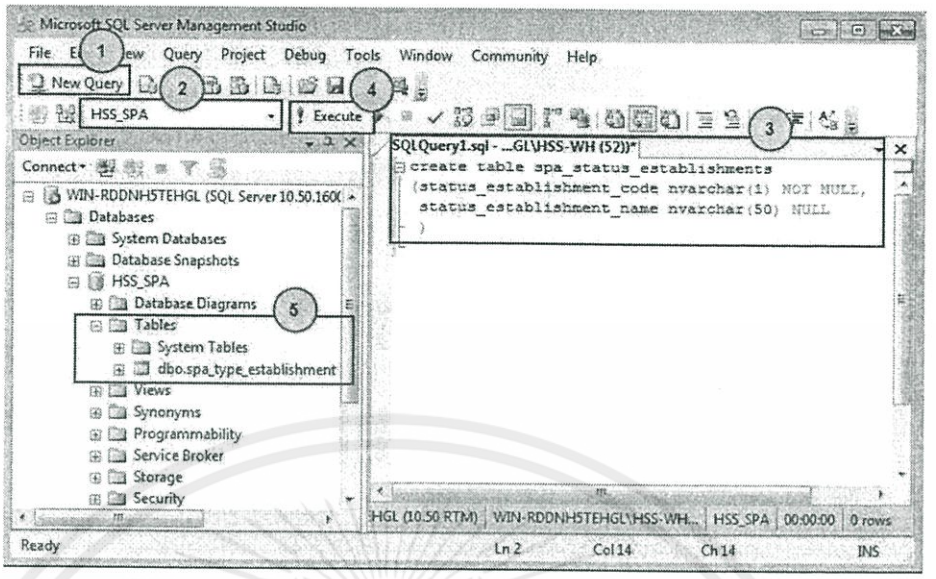
รูปที่ 3.15

3.3.2 การสร้างตารางแบบที่ 2 ด้วยคำสั่ง CREATE

เป็นคำสั่งที่ใช้สำหรับการสร้างตารางใหม่ (Create New Table) ในฐานข้อมูล ซึ่งมีรูปแบบดังนี้ CREATE TABLE ชื่อตาราง (ชื่อฟิลด์ ประเภทของข้อมูล(ขนาด),ชื่อฟิลด์ ประเภทของข้อมูล ตามรูปที่ 3.16

- คลิกที่ New Query
- เลือกชื่อฐานข้อมูล
- พิมพ์คำสั่งสาหรับสร้างตาราง ซึ่งจากตัวอย่างในตารางประกอบด้วย 2 ฟิลด์ คือ status_establishment_code และ status_establishment_name เป็นต้น
- ทำการรันโปรแกรมโดยคลิกที่ Execute
- Refresh ตาราง โดยการคลิกขวาที่ Tables จากนั้นจะปรากฏชื่อตารางที่สร้างใหม่

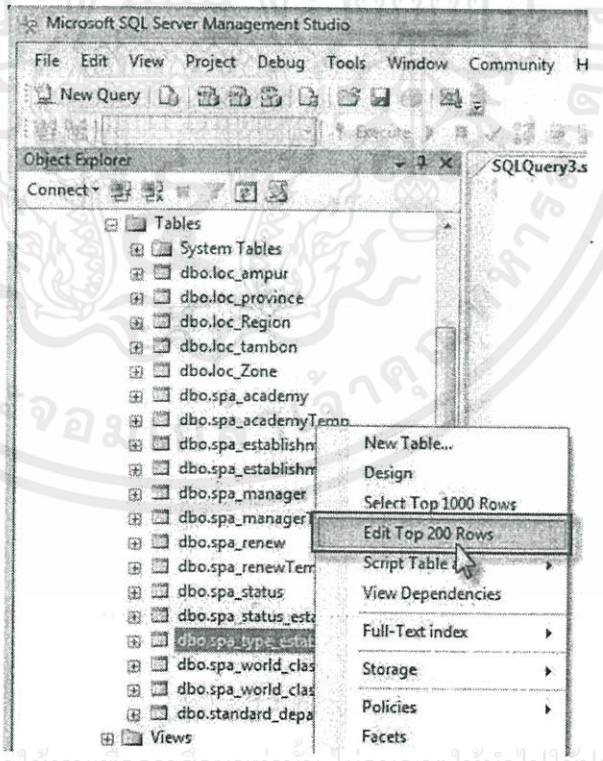
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16

การใส่ข้อมูลลงในตาราง

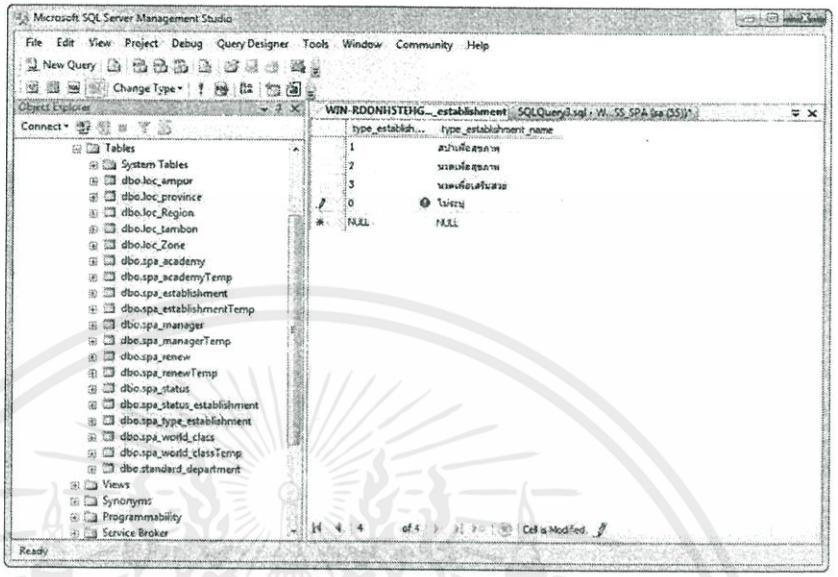
- การเพิ่มข้อมูลผ่านหน้า Edit
- คลิกขวาที่ตารางที่ต้องการเพิ่มข้อมูล เลือก Edit Top 200 Row ตามรูปที่ 3.17



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.17

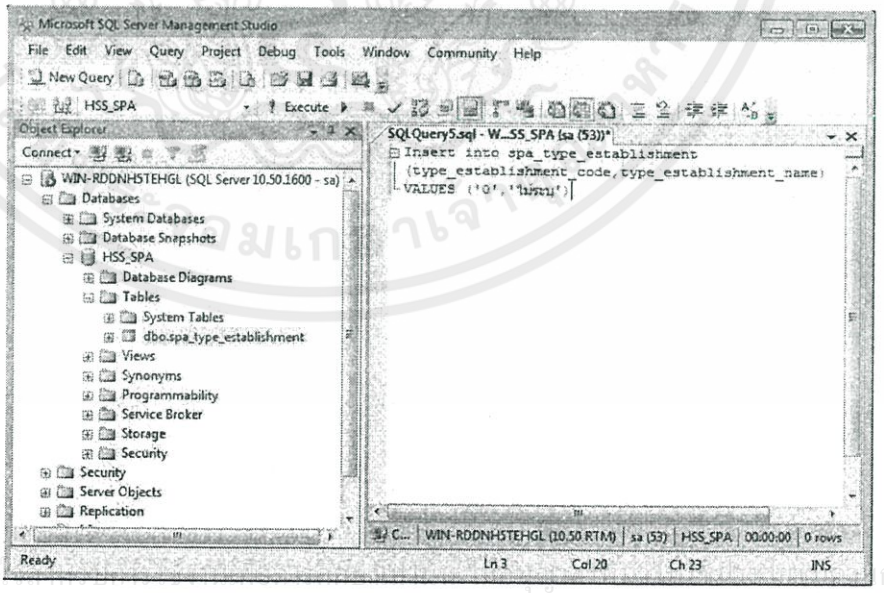
- จากนั้นเพิ่มข้อมูลให้ตรงกับคอลัมน์ที่สร้างไว้ ตามรูปที่ 3.18



รูปที่ 3.18

3.4 การเพิ่มข้อมูลด้วยคำสั่ง INSERT INTO มีรูปแบบดังนี้

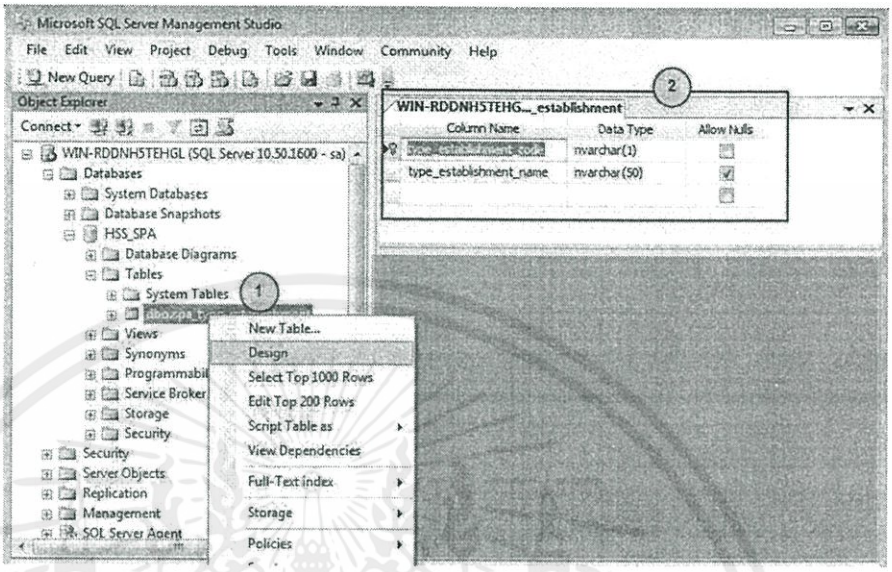
- คลิกที่ปุ่ม New Query ด้านบน ปรากฏหน้าจอ SQLQuery พิมพ์คำสั่ง INSERT INTO ลงไป ตามรูปที่ 3.19



เอกสารนี้เป็นเอกสารที่สงวน การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

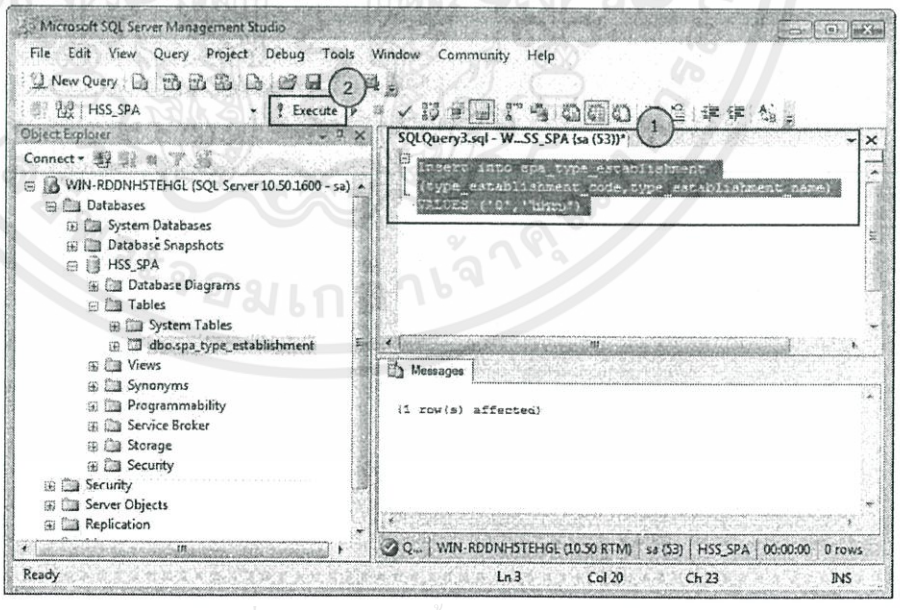
รูปที่ 3.19

- หากต้องการทราบโครงสร้างของฐานข้อมูลทำได้โดยการ คลิกขวาที่ตารางแล้วเลือก Design จะแสดงรายละเอียดของตาราง ตามรูปที่ 3.20



รูปที่ 3.20

- ลากเมาส์คลุมคำสั่งทั้งหมด แล้วคลิกปุ่ม Execute เมื่อ SQL Server ทาคาสั่งเสร็จแล้ว จะมี Messages ปรากฏที่ด้านล่าง ถ้าต้องการเพิ่มข้อมูลแถวใหม่ก็ใช้คำสั่งเดิมอีกครั้ง เปลี่ยนค่าของ VALUES ตามรูปที่ 3.21



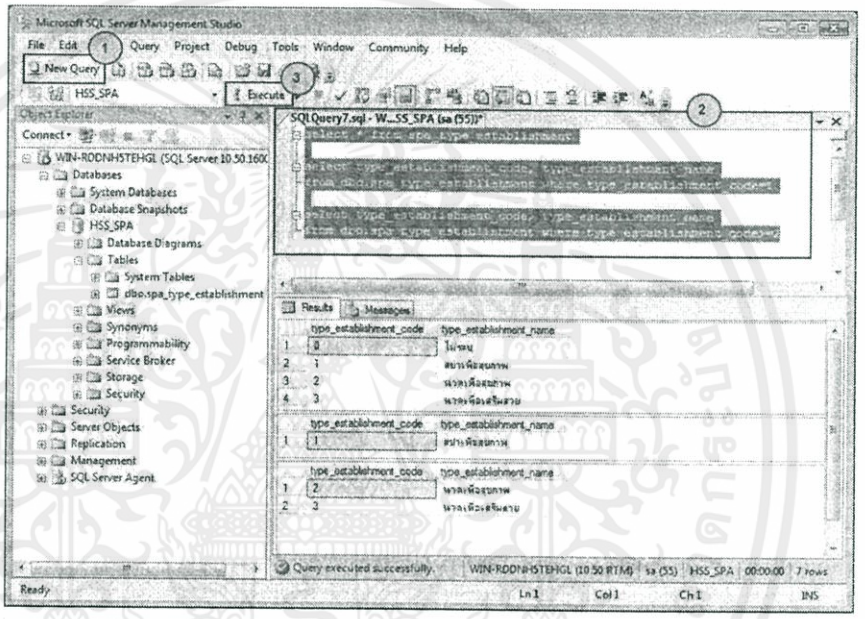
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.21

3.5 การเรียกดูข้อมูลด้วยคำสั่ง Select มีรูปแบบคำสั่งดังนี้

SELECT ชื่อคอลัมน์ FROM ชื่อตาราง หรือ SELECT * FROM ชื่อตาราง * หมายถึงทุกคอลัมน์ในตารางนั้น

- คลิกที่ “ New Query ”
- พิมพ์คำสั่ง ดังรูปที่ 3.22
- สำหรับการรันโปรแกรมนั้น ผู้ใช้สามารถรันทีละคำสั่งหรือหลายคำสั่งได้พร้อมกัน โดยการใช้เมาส์ลากที่คำสั่ง จากนั้นคลิกที่ Execute



รูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึง การทดลองโปรแกรมและผลที่ได้จากการทำงานของระบบรวบรวมและจัดเก็บข้อมูลโดยการรับ-ส่งคำสั่งจากคอมพิวเตอร์โดยมีรายละเอียดของการทดลองดังนี้

4.1 การทดลองโปรแกรม และผลที่ได้จากการทำงาน

ในส่วนนี้ เป็นการศึกษาเฉพาะส่วนโปรแกรม Visual C# 2010 และ ผลที่ได้จากการสั่งจากโปรแกรม ซึ่งผลการทดลองมีดังนี้

หลังจากทำการรันโปรแกรมจาก Visual basic 2010

4.1.1 Open Interface

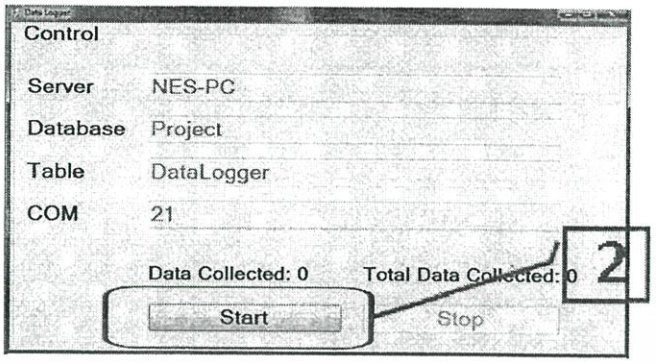
เรียกหน้า Interface มาแสดงจากนั้นเราจะทำการเชื่อมต่อกับฐานข้อมูลตามหมายเลข 1 ดังรูปที่ 4.1

รูปที่ 4.1 เปิดหน้า Interface

4.1.2 เริ่มทำการเชื่อมต่อกับฐานข้อมูล

Interface จะแสดงเงื่อนไขการติดต่อกับฐานข้อมูล ตามหมายเลข 2 ดังรูปที่ 4.2

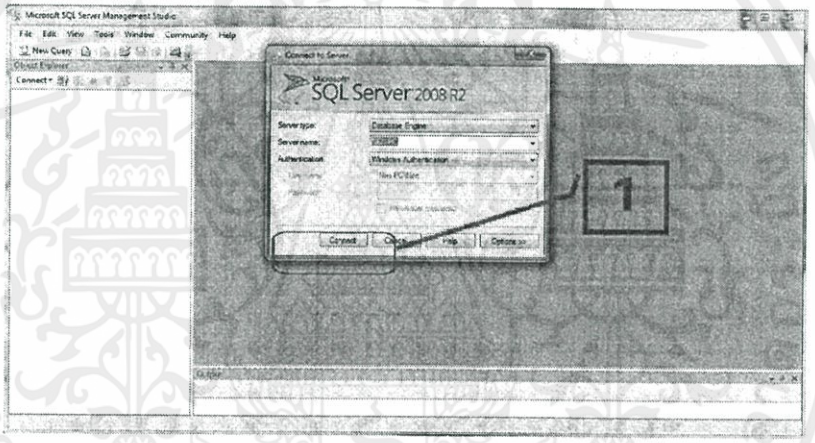
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



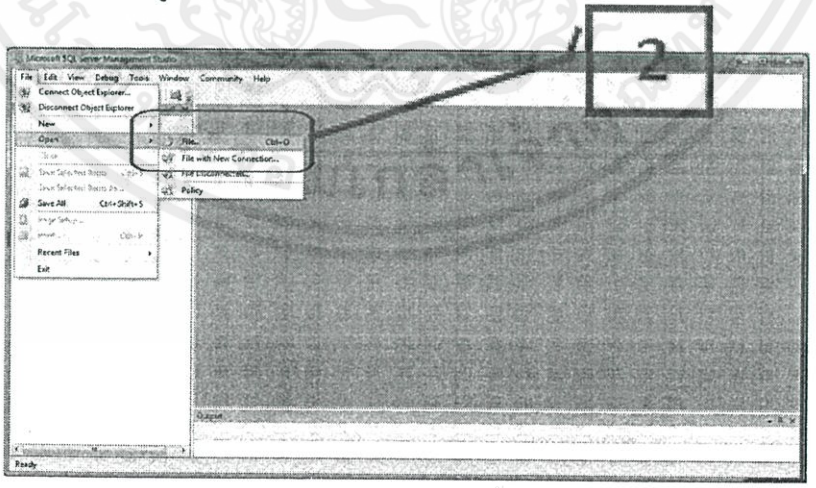
รูปที่ 4.2 การ com port และการ เชื่อมต่อ

4.1.3 เปิดฐานข้อมูลที่จัดเก็บด้วยนวัตกรรมของ SQL Server 2008 R2

เรียก SQL Server 2008 R2

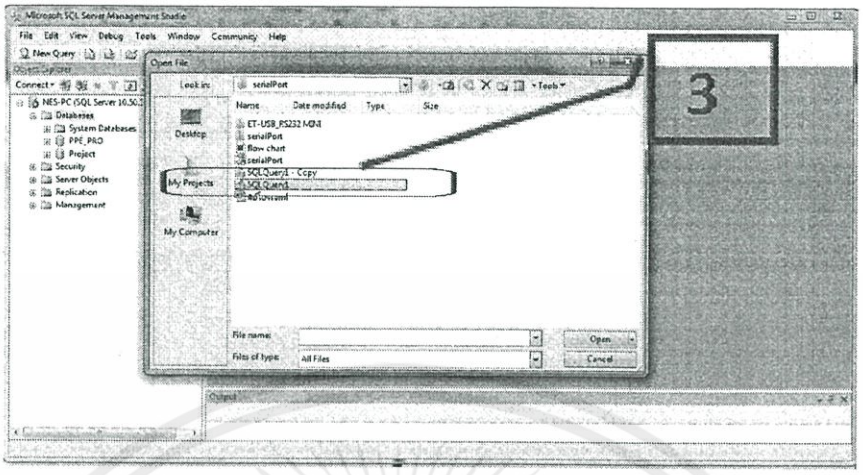


รูปที่ 4.3 ทำการเชื่อมต่อ SQL Server 2008 R2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

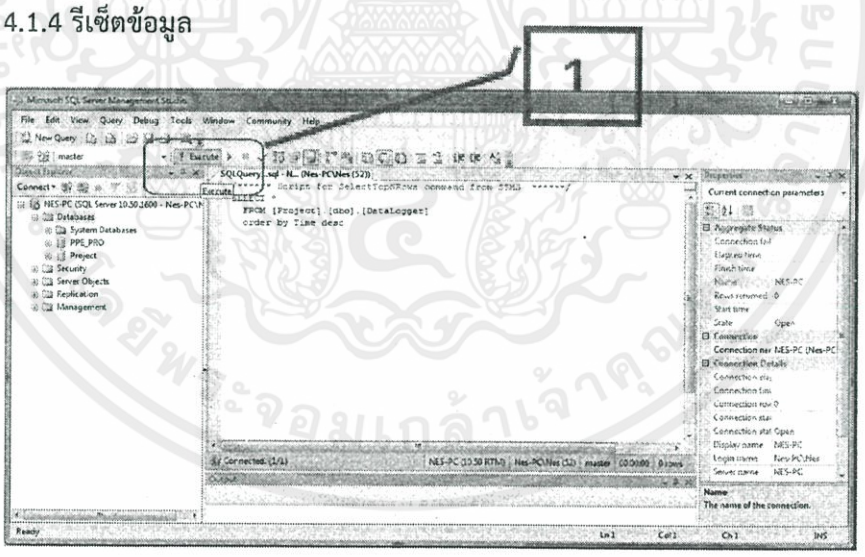
รูปที่ 4.4 การเปิดไฟล์



รูปที่ 4.5 เรียกไฟล์ที่ทำการเก็บข้อมูลมารับค่าและบันทึกผล

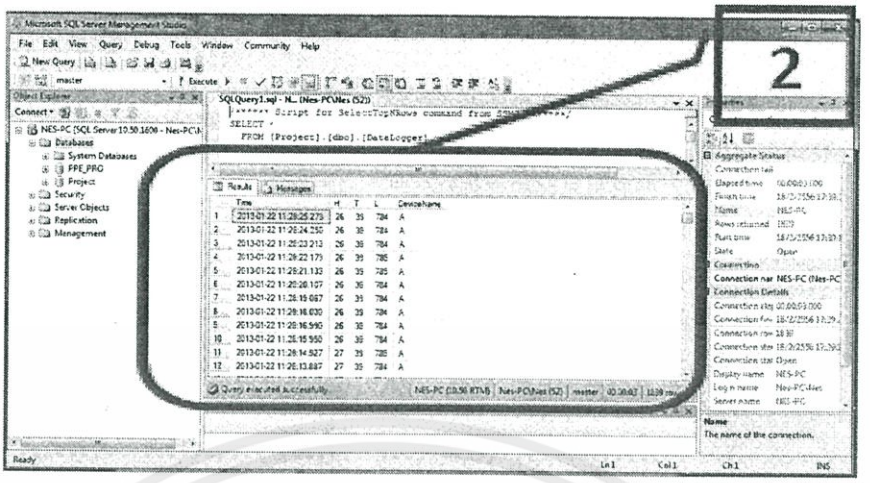
- หมายเลข 1 ทำการ Connect ดังรูปที่ 4.3
- หมายเลข 2 เปิดไฟล์ที่สร้างฐานข้อมูล ดังรูปที่ 4.4
- หมายเลข 3 เลือกไฟล์ข้อมูลที่เราต้องการ ดังรูปที่ 4.5

4.1.4 รีเซ็ตข้อมูล



รูปที่ 4.6 ทำการรีเซ็ตข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

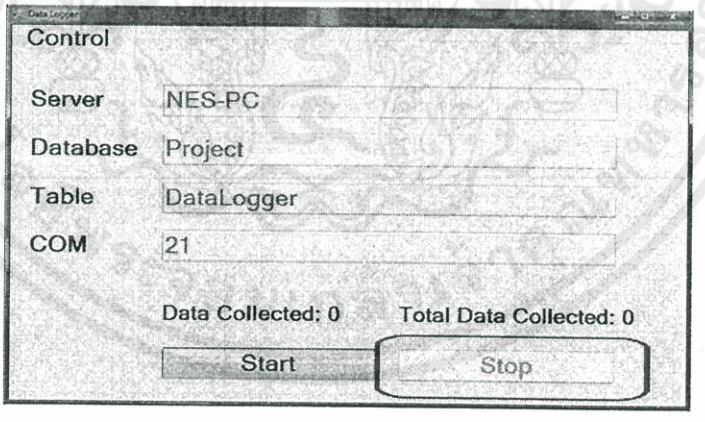


รูปที่ 4.7 หน้าต่างที่เก็บข้อมูล

- หมายเลข 1 กดปุ่ม Execute เพื่อเป็นการรีเซ็ตค่าที่เก็บให้มีการอัปเดตเมื่อมีข้อมูลเข้ามา ดังรูปที่ 4.6
- หมายเลข 2 หน้าต่างที่เก็บข้อมูล จะโชว์ค่าที่รับมาล่าสุดอยู่ข้างบนฐานข้อมูล ดังรูปที่ 4.7

4.1.5 ตัดการเชื่อมต่อ

เมื่อรับข้อมูลตามที่ต้องการ ต้องทำการหยุด (Stop) ทุกครั้ง เมื่อเลิกใช้งาน ดังรูปที่ 4.8



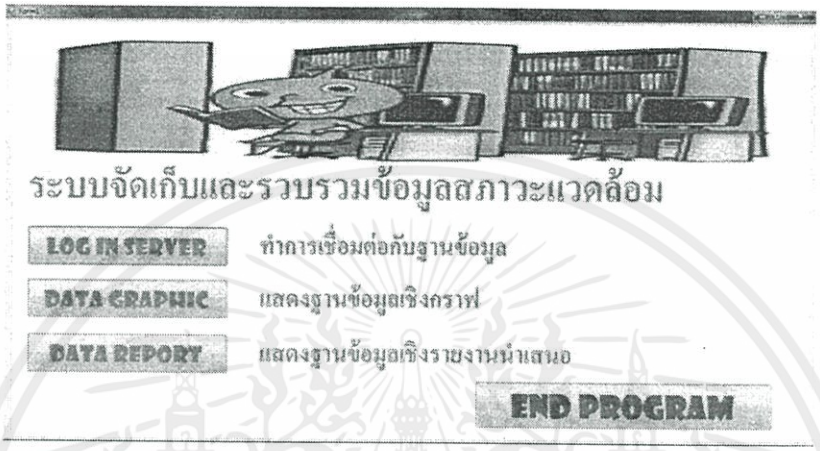
รูปที่ 4.8 ตัดการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

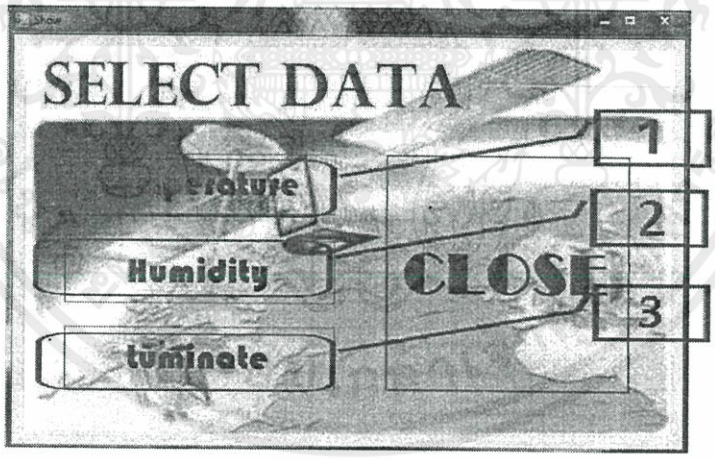
4.2 ภาคการแสดง และนำเสนอ

ในส่วนนี้เป็นการแสดงผลที่ถูกจัดเก็บและรวบรวมผล เพื่อนำมาแสดงค่าและนำเสนอเป็นรายงาน ดังต่อไปนี้

4.2.1 นำเสนอฐานข้อมูลเชิงกราฟ



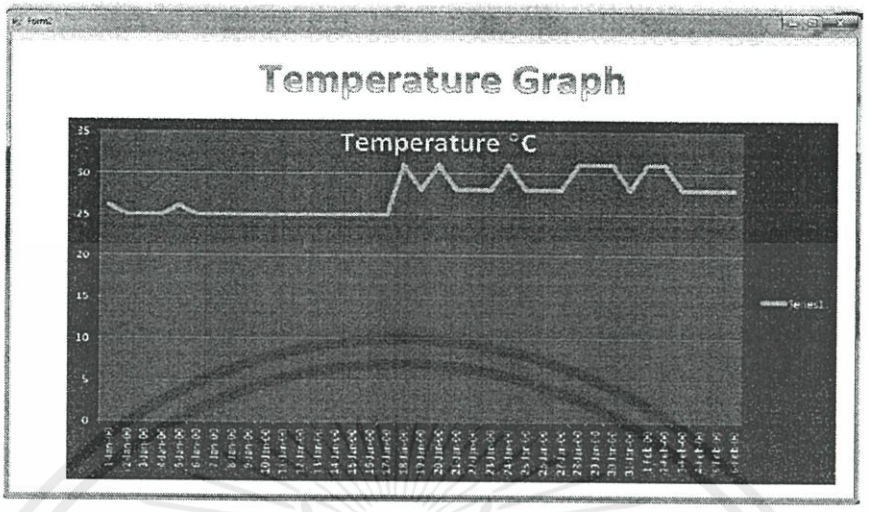
รูปที่ 4.9 แสดงข้อมูลเชิงกราฟ



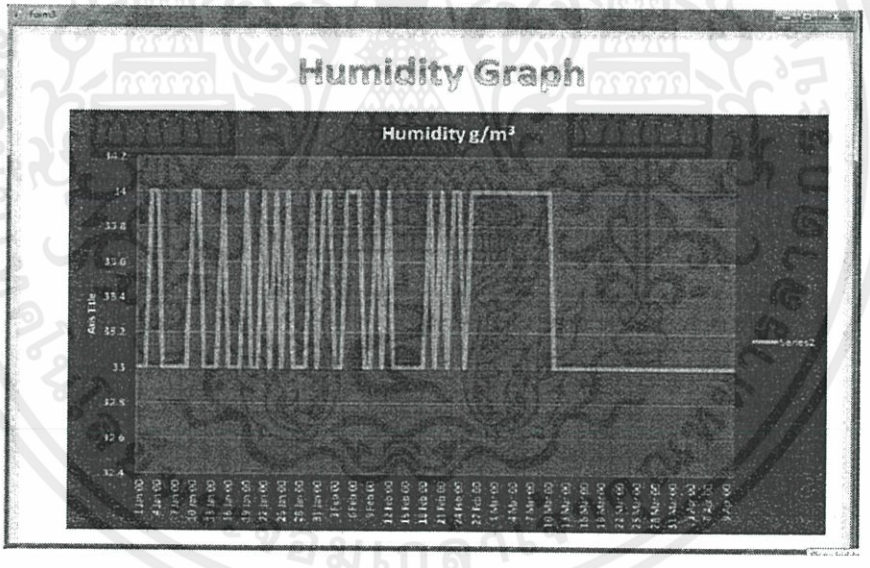
รูปที่ 4.10 ข้อมูลเชิงกราฟ

- หมายเลข 1 เลือกการแสดงกราฟอุณหภูมิ จะได้ดังรูปที่ 4.11
- หมายเลข 2 เลือกการแสดงกราฟความชื้น จะได้ดังรูปที่ 4.12
- หมายเลข 1 เลือกการแสดงกราฟความเข้มแสง จะได้ดังรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

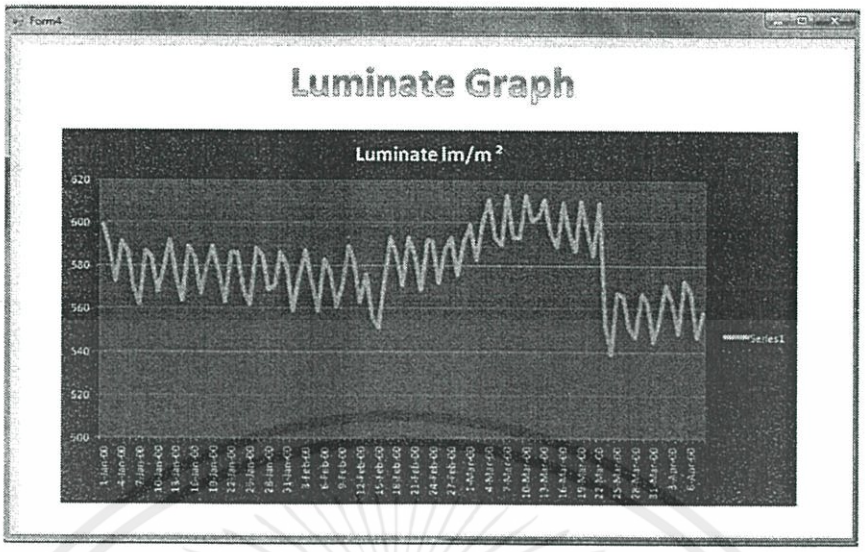


รูปที่ 4.11 กราฟแสดงอุณหภูมิ



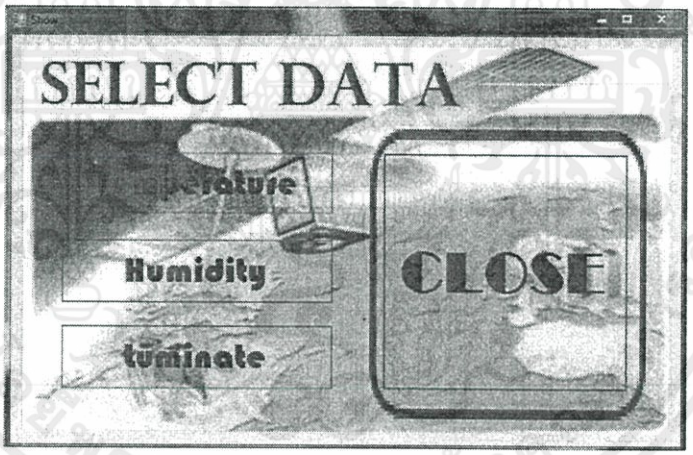
รูปที่ 4.12 กราฟแสดงความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 กราฟแสดงความเข้มแสง

ปิดการแสดงผล คลิกที่ปุ่ม Close

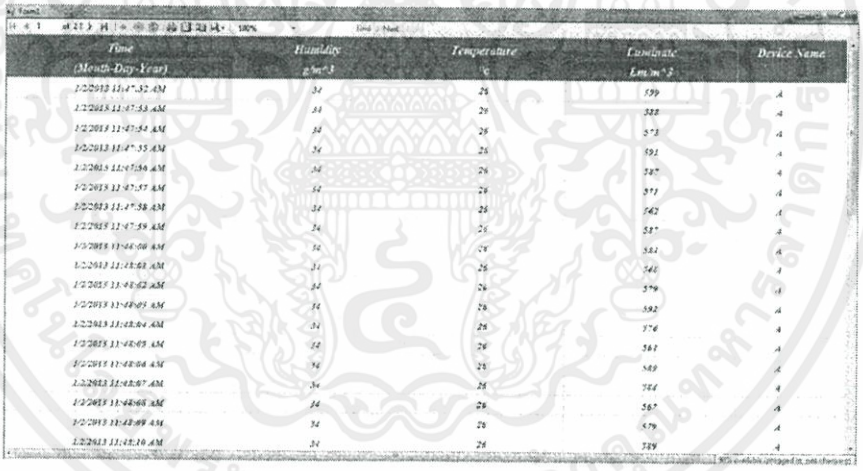


รูปที่ 4.14 ปิดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

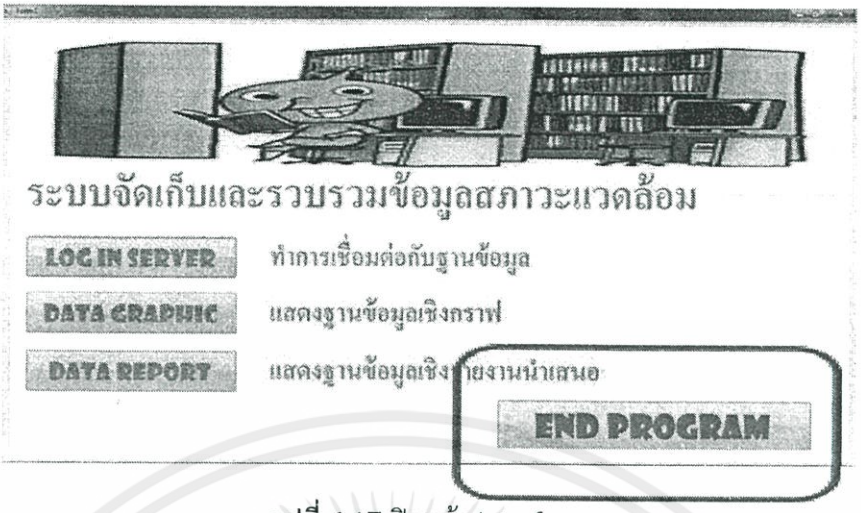
4.2.2 นำเสนอฐานข้อมูลเชิงตาราง

รูปที่ 4.15 เลือกแสดงข้อมูลเชิงรายงานนำเสนอ



รูปที่ 4.16 รายงานนำเสนอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ปิดหน้า Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุป

จากการออกแบบฐานข้อมูลโดยใช้โปรแกรมSQL server แล้วทดลองโหลดข้อมูลจาก MCU ผ่าน Port อนุกรม RS232 เข้าสู่คอมพิวเตอร์ ซึ่งใช้โปรแกรมวิชชวล (Visual C#) โดยเปิดทิ้งไว้สักพัก จากนั้นการทำการเปิดฐานข้อมูลจากไฟล์.SQL ค่าที่ถูกดึงมาเก็บไว้ในฐานข้อมูลเป็นรหัสที่ผ่านการแปลจากสัญญาณอะนาล็อกเป็นดิจิตอล เพื่อนำมาใช้ในการแสดงผลออกทางจอภาพของคอมพิวเตอร์ ทั้งในรูปแบบกราฟและรายงานนำเสนอ

ความสามารถและการใช้งานดาต้าล็อกเกอร์ (Hardware)

- แหล่งจ่ายไฟขนาด 12 VDC – 30 VDC
- เมื่อเปิดเครื่องจะพร้อมใช้งานได้ เมื่อมีการติดต่อกับ Hyper Terminal เรียบร้อย
- เมื่อต้องการจัดเก็บให้ทำการ capture text แล้วทำการบันทึกในไฟล์.txt

ความสามารถและการใช้งานดาต้าล็อกเกอร์ (Software)

ในที่นี้หมายถึงกระบวนการ Load ข้อมูลซึ่งมีความสามารถละการใช้งานดังนี้

- เมื่อรันโปรแกรม แล้วให้เลือก Comport เริ่มรับข้อมูลเมื่อรับข้อมูลเสร็จแล้วก็จบกระบวนการ
- การนำข้อมูลที่รับมาใช้ประโยชน์
- สามารถดูข้อมูลในรูปแบบตารางได้
- สามารถดูข้อมูลในรูปแบบกราฟ
- การดูข้อมูลสามารถดูได้โดย
- เรียกเวลา จะเป็นการดูข้อมูลในวันใดวันหนึ่งเท่านั้น
- เรียกเดือน จะเป็นการดูข้อมูลในเดือนใดเดือนหนึ่งเท่านั้น
- เรียกวันปี จะเป็นการดูข้อมูลในปีใดปีหนึ่งเท่านั้น
- ระบุเป็นช่วง โดยแบ่งเป็น วัน เดือน ปี แต่สามารถดูข้อมูลโดยการระบุ ช่วงเริ่มต้นและสิ้นสุด
- เมื่อต้องการจบการทำงานให้กดหยุดการรับค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบและแนวทางแก้ไข

จากการศึกษาพบว่าปัญหาส่วนใหญ่อาจเกิดการผิดพลาดจากเหตุต่อไปนี้

- ตัวแปลข้อมูลที่รับข้อมูลมาไม่ครบ
- ตัวเซนเซอร์ที่ต่อเข้ากับไมโครคอนโทรลเลอร์นั้นอาจหลุดออกจากกันได้
- ต่อเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์นั้นจะใช้ได้เฉพาะสถานที่ใกล้
- ถ้านำออกไปใช้กลางแจ้งจะไม่สามารถตัวเชื่อมต่อRS232

5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

- ตัวแปลสัญญาณมาไม่ครบต้องทำการเขียนโปรแกรมและทดสอบตัวไมโครคอนโทรลเลอร์อีกครั้ง
- เนื่องจากการทดลองนี้ยังต้องมีการวิจัย จึงไม่สามารถทำการบัดกรีได้แต่ให้ใช้ในการทดสอบอย่างระมัดระวัง
- การเชื่อม (RS232) สามารถใช้ได้กับไมโครคอนโทรลเลอร์เพียงตัว เราก็สามารถใช้การเชื่อมต่อตัวอื่นที่มีคุณสมบัติตรงตามความต้องการของเรา
- การทดสอบนั้นถ้าเป็นสถานที่ใกล้ไกลเราอาจต่อไฟกระแสสลับ (AC 220 V) แต่ถ้าต้องออกไปทดสอบกลางแจ้งหรือที่ไกลไกลเราสามารถจ่ายแหล่งพลังงานอื่นอย่างแบตเตอร์รี่ให้กับตัวไมโครคอนโทรลเลอร์ได้
- และเรายังสามารถเก็บข้อมูลแบบไร้สายได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ส่วนโปรแกรมภาษาซีที่ใช้ในโครงงาน

ส่วน Data Logger

โปรแกรม VC#

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.IO.Ports;

using System.Data.SqlClient;

namespace serialPort

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private static string COM = "COM";

private static string msg_total = "Total Data Collected: ";

private static string msg_round = "Data Collected: ";

private int total = 0;

private int round = 0;

private SerialPort port;

private string serverName;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private string databaseName;

private string tableName;

private SqlConnection dbcon;

private void Form1_Load(object sender, EventArgs e)

{

//Console.WriteLine("Incoming Data:");

//// Attach a method to be called when there

//// is data waiting in the port's buffer

//port = new SerialPort(COM + text_com.Text, 9600, Parity.None, 8, StopBits.One);

//port.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived);

//dbcon = ConnectToSql(serverName, databaseName);

//// Begin communications

//port.Open();

//// Enter an application loop to keep this thread alive

//Application.Run();

////int i =0;

////for (; i < 100000; i++);

//// port.Close();

text_server.Text = "NES-PC";

text_database.Text = "Project";

text_table.Text = "DataLogger";

text_com.Text = "21";

port = new SerialPort(COM + text_com.Text, 9600, Parity.None, 8, StopBits.One);

port.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived);

lb_round.Text = msg_round + round.ToString();

lb_total.Text = msg_total + total.ToString();

}

/// <serial port>

String buffer; ตัวเปิดport

private void port_DataReceived(object sender, SerialDataReceivedEventArgs e)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
// Show all the incoming data in the port's buffer
String s = port.ReadExisting();
//Console.WriteLine(s);
if (s.IndexOf("@") != -1)
{//string start
buffer = s;
if (s.IndexOf("#") != -1)
{//string end
//put into database
Console.WriteLine(buffer);
insertData(dbcon, tableName, buffer);
addCounter();
buffer = "";
}
}
else
{//string continue
//concat string
buffer += s;
if (s.IndexOf("#") != -1)
{//string end
//put into database
Console.WriteLine(buffer);
insertData(dbcon, tableName, buffer);
addCounter();
buffer = "";
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/// </serial port>
///
/// <database> คำสั่งSqlสำหรับเก็บ "Select"
private void insertData(SqlConnection dbcon, string tableName, string data)
{
try
{
int posADD = data.IndexOf("@");
int posH = data.IndexOf("H");
int posT = data.IndexOf("T");
int posL = data.IndexOf("L");
int posSH = data.IndexOf("#");
//Console.WriteLine("pos@ " + posADD);
//Console.WriteLine("posH " + posH);
//Console.WriteLine("posT " + posT);
//Console.WriteLine("posL " + posL);
//Console.WriteLine("pos# " + posSH);
string name = data.Substring(posADD + 1, posH - posADD - 1);
int H = Convert.ToInt32(data.Substring(posH + 1, posT - posH - 1));
int T = Convert.ToInt32(data.Substring(posT + 1, posL - posT - 1));
int L = Convert.ToInt32(data.Substring(posL + 1, posSH - posL - 1));
//Console.WriteLine(name + " " + H + " " + T + " " + L);
string sqlIns = "INSERT INTO " + tableName + " VALUES (@val1, @val2, @val3, @val4, @val5)";
SqlCommand cmdIns = new SqlCommand(sqlIns, dbcon);
//cmdIns.Parameters.AddWithValue("@val1", DateTime.Now.ToString("yyyy-MM-dd-HH-mm-ss"));
cmdIns.Parameters.AddWithValue("@val1", DateTime.Now);
cmdIns.Parameters.AddWithValue("@val2", H);
cmdIns.Parameters.AddWithValue("@val3", T);
cmdIns.Parameters.AddWithValue("@val4", L);
cmdIns.Parameters.AddWithValue("@val5", name);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cmdIns.ExecuteNonQuery();

}

catch (Exception ex)

{

throw new Exception(ex.ToString(), ex);

}

}

private SqlConnection ConnectToSql() ต่อdatabase

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = "Server = " + serverName + "; Database = " + databaseName + ";
Trusted_Connection = True;";

try

{

conn.Open();

//MessageBox.Show("Successfully connect to Database");

}

catch (Exception ex)

{

MessageBox.Show("Failed to connect to data source");

MessageBox.Show(ex.ToString());

}

return conn;

}

/// </database>

///

///

/// <interface>

private void btn_start_Click(object sender, EventArgs e)

{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
round = 0;
serverName = text_server.Text;
databaseName = text_database.Text;
tableName = text_table.Text;
dbcon = ConnectToSql();
btn_start.Enabled = false;
btn_stop.Enabled = true;
port.Open();
}
private void btn_stop_Click(object sender, EventArgs e)
{
port.Close();
dbcon.Close();
btn_start.Enabled = true;
btn_stop.Enabled = false;
}
/// <additional function>
private void addCounter() {
round++; total++;
set_lb_round(msg_round + round.ToString());
set_lb_total(msg_total + total.ToString());
}
public void set_lb_round(string text)
{
if (this.lb_round.InvokeRequired)
{
this.lb_round.Invoke(new Action<string>(set_lb_round), text);
}
else
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
this.lb_round.Text = text;
}
}
public void set_lb_total(string text)
{
if (this.lb_total.InvokeRequired)
{
this.lb_total.Invoke(new Action<string>(set_lb_total), text);
}
else
{
this.lb_total.Text = text;
}
}
/// </additional function>
/// </interface>
}
}
```

ส่วน Interface

โปรแกรม VC#

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
namespace WindowsFormsApplication_Gap_Nes_

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button4_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1_Click(object sender, EventArgs e)

{

System.Diagnostics.Process.Start(@"C:\Users\Nes\Desktop\serialPort\serialPort\obj\x86\Debug\serialPort.exe"
);

}

private void button2_Click(object sender, EventArgs e)

{

System.Diagnostics.Process.Start(@"C:\Users\Nes\Documents\Visual Studio 2010\Projects\show graph\show
graph\obj\x86\Debug\show graph.exe");

}

private void button3_Click(object sender, EventArgs e)

{

System.Diagnostics.Process.Start(@"C:\Users\Nes\Documents\Visual Studio
2010\Projects\ReportsApplication(TableShowReport)\ReportsApplication(TableShowReport)\obj\Debug\Rep
ortsApplication(TableShowReport).exe");

}

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Table Report

โปรแกรม VC#

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace ReportsApplication_TableShowReport_

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'ProjectDataSet.DataLogger' table. You can move, or remove it, as needed.

this.DataLoggerTableAdapter.Fill(this.ProjectDataSet.DataLogger);

this.reportViewer1.RefreshReport();

}

}

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน Design Graph

โปรแกรม VC#

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace show_graph

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button2_Click(object sender, EventArgs e)

{

Form f1 = new Form(); //หน้าฟอร์มแรกมีปุ่มที่1เลือก

Form2 f2 = new Form2(); //หน้าฟอร์มที่2 ถูกลิ้งไปยังหน้าฟอร์มแรก

f1.Close();

f2.Show();

}

private void button4_Click(object sender, EventArgs e)

{

Form f1 = new Form(); //หน้าฟอร์มแรกมีปุ่มที่3เลือก

Form4 f4 = new Form4(); //หน้าฟอร์มที่4 ถูกลิ้งไปยังหน้าฟอร์มแรก

f1.Close();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

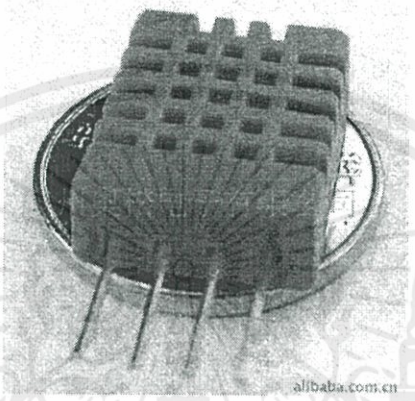
```
f4.Show();
}
private void button3_Click(object sender, EventArgs e)
{
Form f1 = new Form(); //หน้าฟอร์มแรกมีปุ่มที่2เลือก
Form3 f3 = new Form3(); //หน้าฟอร์มที่3 ถูกลิ้งไปยังหน้าฟอร์มแรก
f1.Close();
f3.Show();
}
private void button1_Click(object sender, EventArgs e)
{
this.Close();
}
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

ข.1 เอกสารคู่มือการใช้งาน module – DHT11

Digital-output relative humidity & temperature sensor/module – DHT11



Resisitive-type humidity and temperature module/sensor

1. Feature & Application:

* Full range temperature compensated * Relative humidity and temperature measurement
* Calibrated digital signal *Outstanding long-term stability *Extra components not needed
* Long transmission distance * Low power consumption *4 pins packaged and fully interchangeable

2. Description:

DHT11 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in OTP memory.

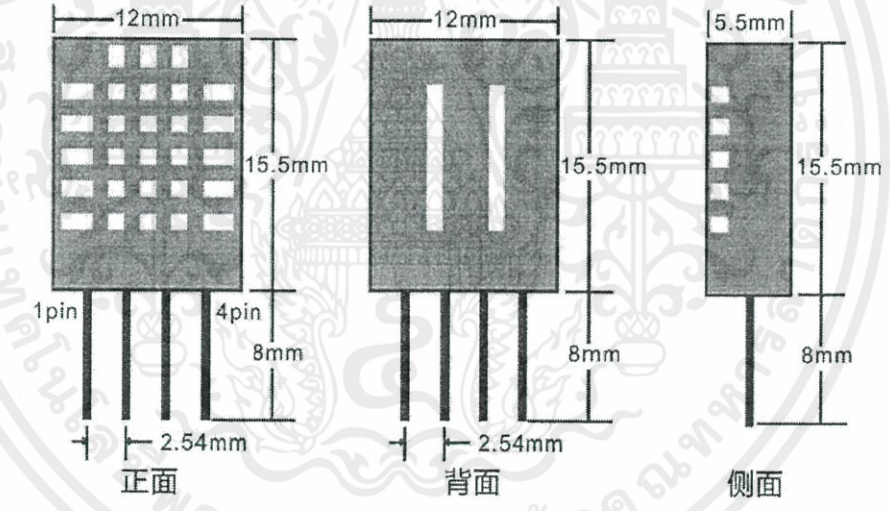
Small size & low consumption & long transmission distance(20m) enable DHT11 to be suited in all kinds of harsh application occasions. Single-row packaged with four pins, making the connection very convenient.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Technical Specification:

Model	DHT11
Power supply	3-5.5V DC
Output signal	digital signal via single-bus
Sensing element	Polymer resistor
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius
Accuracy	humidity +-4%RH (Max +-5%RH); temperature +-2.0Celsius
Resolution or sensitivity	humidity 1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-1Celsius
Humidity hysteresis	+-1%RH
Long-term Stability	+-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	size 12*15.5*5.5mm

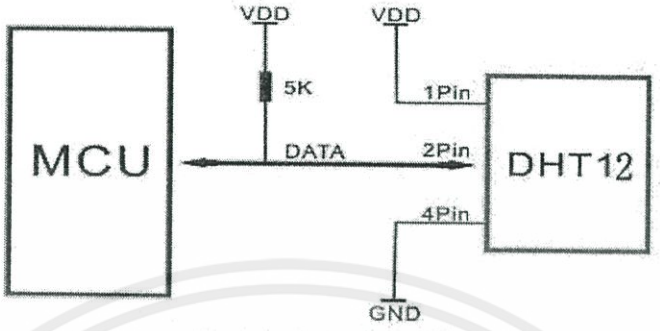
4. Dimensions: (unit----mm)



Front view **Back view** **Side view**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Typical application



典型应用电路

3Pin-NULL, MCU=Microcomputer or single-chip computer

6. Operating specifications:

(1) Power and Pins

Power's voltage should be 3-5.5V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

(2) Communication and signal

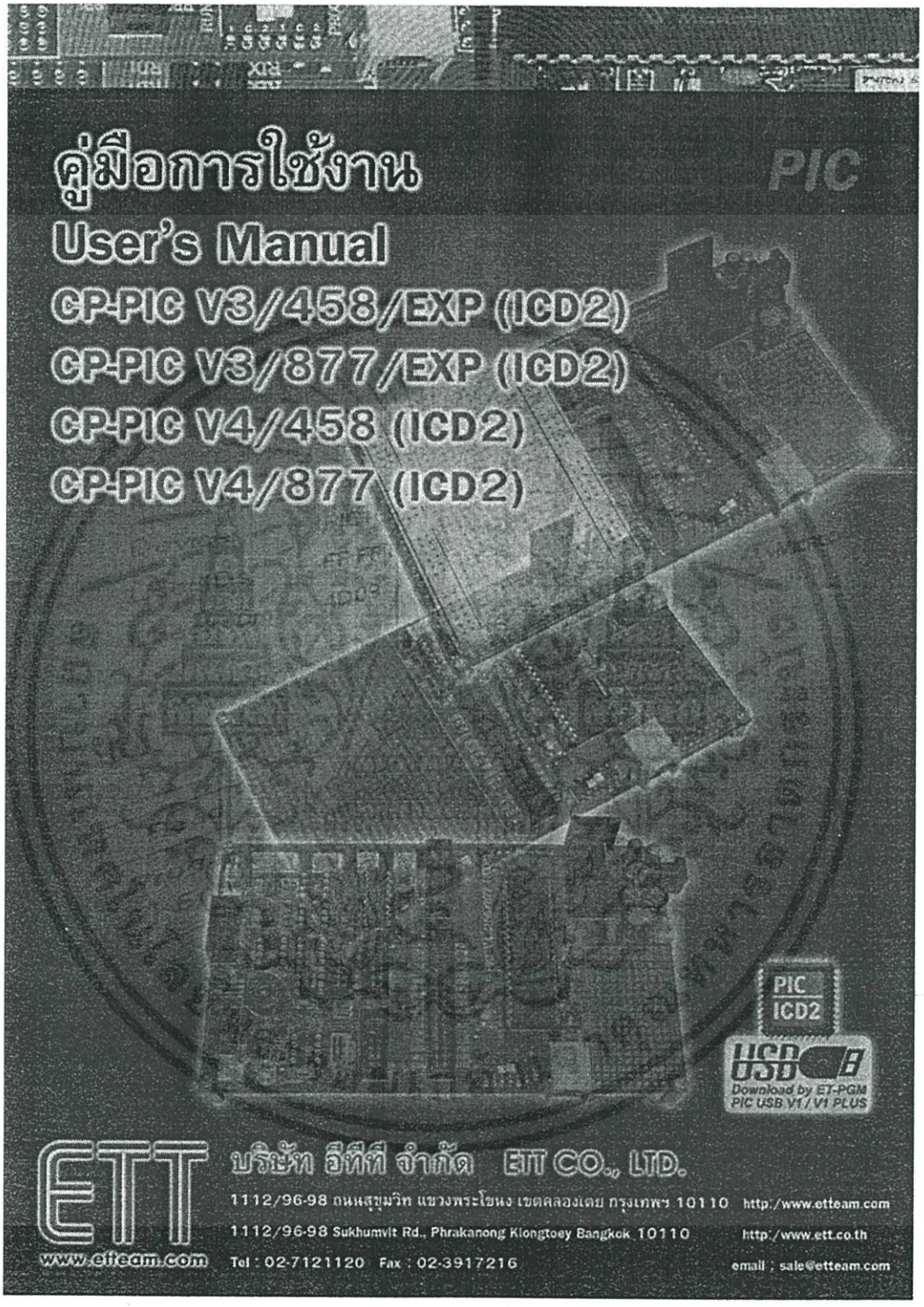
Single-bus data is used for communication between MCU and DHT11.

7. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 เอกสารคู่มือการใช้งาน บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0(ICD2)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะโดยทั่วไป

บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V3.0 & V4.0 ที่ได้ออกแบบนี้ เป็นบอร์ดที่ออกแบบไว้ใช้งานกับไมโครคอนโทรลเลอร์ในตระกูล PIC โดยจะสามารถใช้ได้ กับเบอร์ 16F877-20P,18F442 และ 18F458 หรือเบอร์อื่นๆ ที่มีโครงสร้างและตำแหน่งขาสัญญาณเหมือนกันโดย CPU แต่ละเบอร์ก็จะมีคุณสมบัติที่แตกต่างกันซึ่งสามารถสรุปคุณสมบัติของ CPU แต่ละเบอร์อย่างคร่าวๆ ดังตารางด้านล่าง

DEVICE	Program Memory	Data Memory		CAN Module	I/O (Bit)	OSC max (MHz)	Timers	PLL
	Flash	RAM (Bytes)	EEPROM (Byte)					
PIC 16F877	8K (14-Bit Words)	368	256	NO	33	20MHz	3	NO
PIC 18F442	16 Kbyte	768	256	NO	34	40MHz	4	YES
PIC 18F458	32 Kbyte	1536	256	YES	34	40MHz	4	YES

CPU ดังกล่าวจะบรรจุอยู่ภายในตัวถังแบบ DIP ขนาด 40 ขา และมีทรัพยากรต่างๆบรรจุไว้ภายในตัว CPU อย่างครบถ้วน ไม่ว่าจะเป็น ADC/TIMER/COUNTER/PWM หรือ PORT I/O ต่างๆ ซึ่งมีความเหมาะสมในการนำไปประยุกต์ใช้งานในลักษณะต่างๆได้เป็นอย่างดี

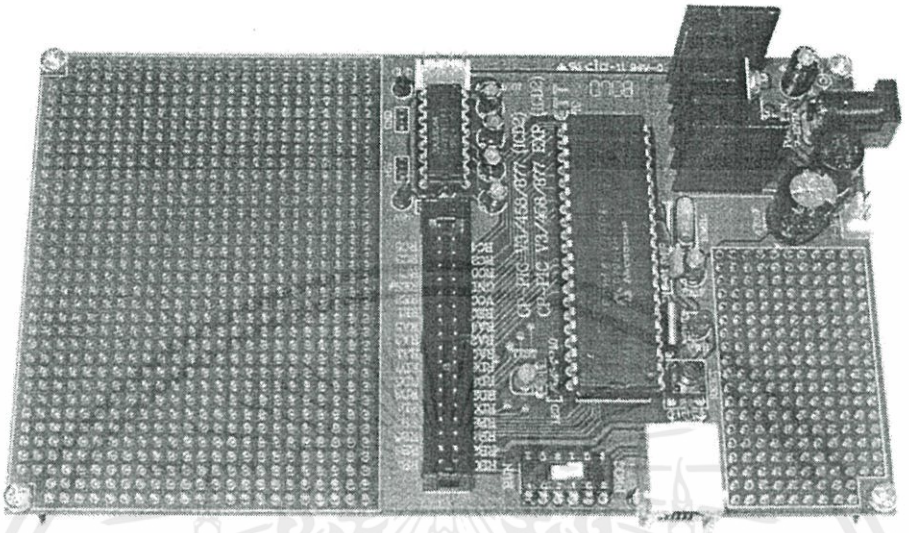
สำหรับอุปกรณ์ I/O ต่างๆ ซึ่งไม่ได้มีบรรจุไว้ในตัวCPU ด้วยทางทีมงานอีทีที ก็ได้จัดหาและทำการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ต่างๆที่มีความจำเป็นไว้ให้ด้วยแล้ว ไม่ว่าจะเป็นจอแสดงผลแบบ LCD ระบบฐานเวลา RTC วงจร Line Driver สำหรับการสื่อสารข้อมูลอนุกรมแบบ RS232 และ RS422/485 และยังสามารถให้ผู้ใช้ทำการเพิ่มเติมอุปกรณ์ I/O อื่นๆเข้าไปได้อีกตามความจำเป็นในการใช้งาน

โดยลักษณะของบอร์ดไมโครคอนโทรลเลอร์ที่ได้ออกแบบนี้ จะแบ่งออกเป็น 3 รุ่น แต่ละรุ่นก็จะมีทรัพยากรณ์บนบอร์ดแตกต่างกันไป เพื่อ ให้สามารถเลือกใช้งานกันตามความเหมาะสมกับงานดังนี้คือ

- CP-PIC V3.0(ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ ซึ่งออกแบบวงจรเฉพาะส่วนพื้นฐานที่จำเป็น เช่น แหล่งจ่ายไฟ วงจรรีเซ็ต วงจรกำเนิดความถี่สัญญาณนาฬิกา พอร์ตสำหรับ Download โปรแกรม และวงจรสื่อสารอนุกรม ส่วนวงจร I/O ภายนอกนั้น จะไม่ได้จัดเตรียมไว้ให้ด้วย แต่จะทำการต่อสัญญาณ I/O ต่างๆจาก CPU มาไว้ยังขั้วต่อ Connector สำหรับให้ผู้ใช้นำไปเชื่อมต่อกับอุปกรณ์ I/O ภายนอกได้โดยง่าย และยังมีพื้นที่เอนกประสงค์สำหรับให้ผู้ใช้ออกแบบวงจร I/O และต่อวงจร I/O เพิ่มเติมได้เอง เหมาะสำหรับผู้ใช้ที่ต้องการนำบอร์ดไปใช้พัฒนางานต้นแบบโดยการสร้าง I/O ต่างๆขึ้นมาใช้งานเอง ซึ่งในบอร์ดจะประกอบด้วย

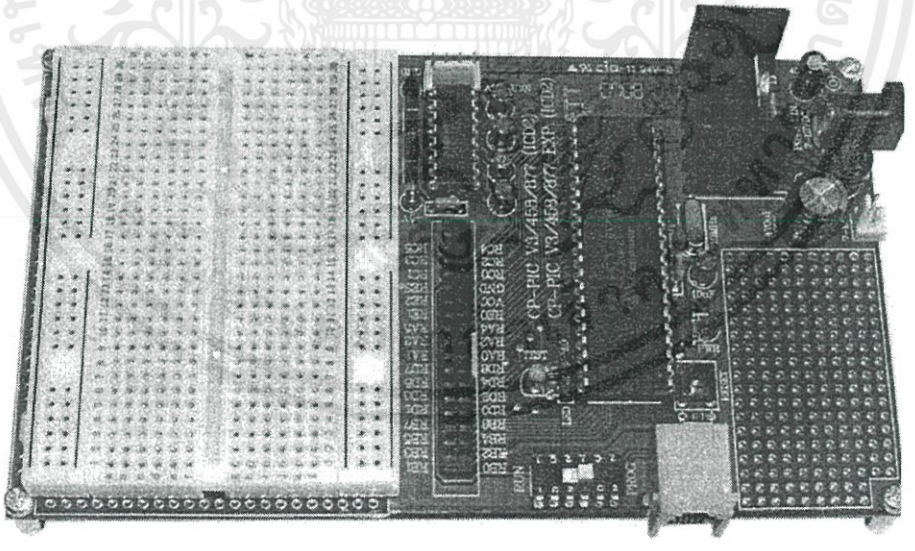
  - RS – 232 1 แชนเนล
  - ETT CON 34PIN (ET BUS I/O 34PIN)
  - 5 Volt Regulator On Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงบอร์ด CP-PIC V3/458/877 (ICD2)

- CP-PIC V3.0 EXPANSION(ICD2) จะมีลักษณะเดียวกันกับบอร์ด CP-PIC V3.0(ICD2) แต่จะมีแผง Photo Board สำหรับให้ผู้ใช้ต่อทดลองวงจร I/O อย่างง่ายๆได้เอง เหมาะสำหรับผู้ใช้ที่ต้องการศึกษาเรียนรู้และต้องการทดลองวงจร I/O ต่างๆ ร่วมกับ CPU อย่างง่ายๆ



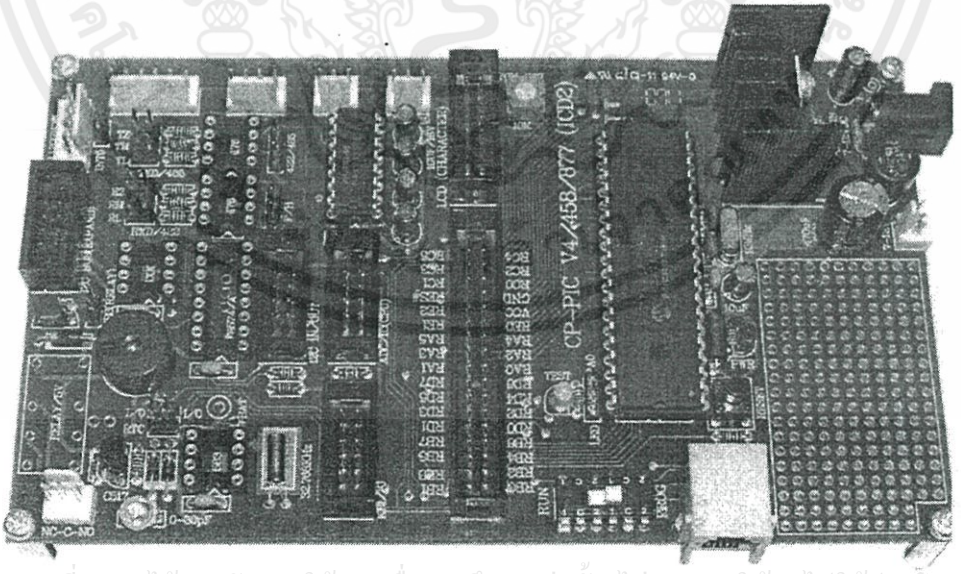
รูปแสดงบอร์ด CP-PIC V3/458/877 EXP (ICD2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

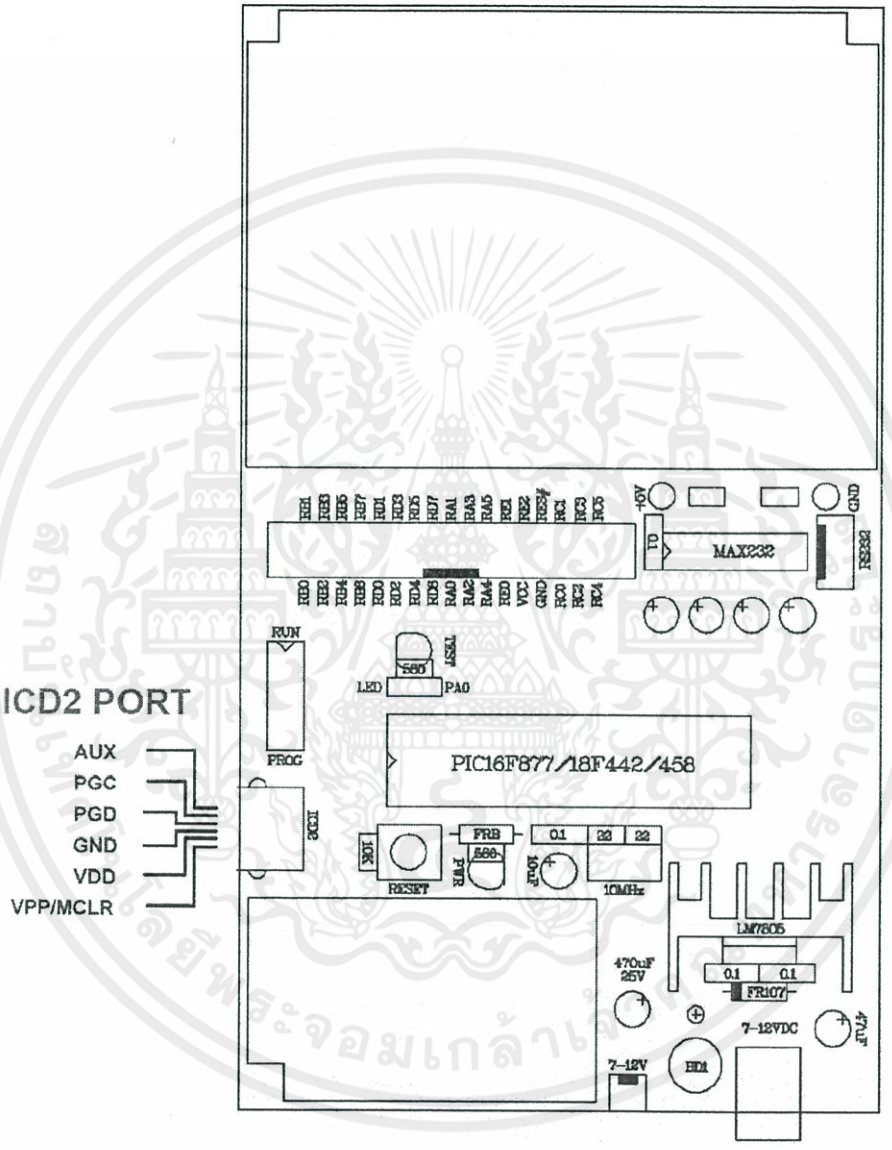
- CP-PIC V4.0 (ICD2) เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ I/O ภายนอกอื่นๆ ที่มีความจำเป็นไว้รองรับการใช้งานในลักษณะต่างๆ เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปใช้งานในลักษณะงานที่แตกต่างกันได้ โดยไม่ต้องดัดแปลงวงจร หรือ อาจดัดแปลงวงจรเพียงเล็กน้อยสำหรับงานบางอย่าง โดยประกอบด้วยส่วนต่างๆ ดังนี้

  - RS – 232 1 แชนเนล
  - RS-422/458 1 แชนเนล (IC 75176 เป็น Option)
  - ETT CON 34PIN (ET BUS I/O 34PIN)
  - 5 Volt Regulator On Board
  - พอร์ตต่อสัญญาณ ICD2 สำหรับเชื่อมต่อสัญญาณโปรแกรมจากภายนอก
  - ADC/IO(CPU) พอร์ตสำหรับต่อ อินพุตอนาลอก 8 Channel
  - CLCD 14PIN พอร์ตสำหรับต่อ LCD (4 Bit Data)
  - RTC #PCF8583P (Option)
  - EEPROM ตั้งแต่เบอร์ #2432 ขึ้นไป (Option)
  - I^2C IN/OUT เป็นIC ขยายพอร์ต I/O #PCF8574AP (Option)
  - KBI/IO 10 Pin สำหรับต่อกับ Keyboard หรือ ใช้เป็น Input /Output Port
  - Relay Onboard 5V 1ตัว (Option)
  - MCRB02TTL ขั้วต่อ Macnetic Card Reader
  - Mini Speaker/Buzzer
  - I^2C BUS(EXPAND)
  - PWM1 และ PWM2 ขั้วต่อสำหรับใช้งาน Capture/Compare/PWM

#หมายเหตุ Option คือ ส่วนที่ออกแบบไว้ให้เป็น Socket เปล่าๆหากต้องการใช้งานต้องหาซื้อเพิ่มเอง

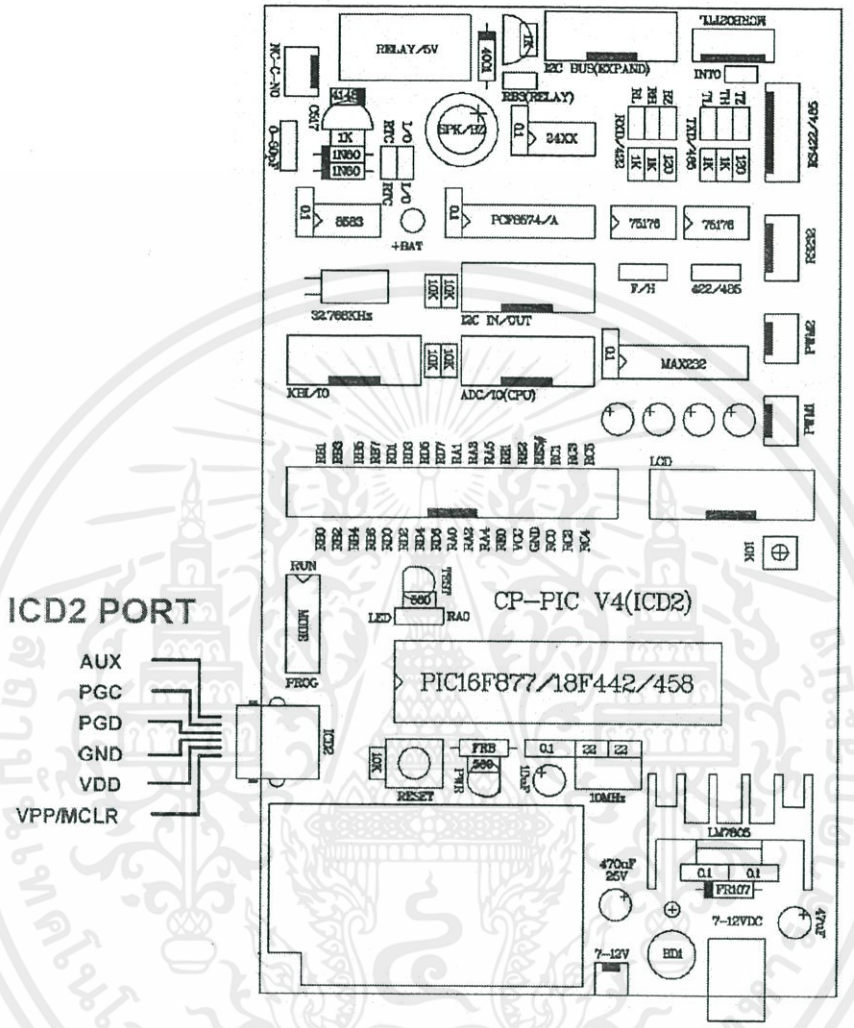


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง ลักษณะโครงสร้างของบอร์ด CP-PIC V3.0(ICD2) และ V3.0 EXPANSION (ICD2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง โครงสร้างของบอร์ด CP-PIC V4.0 (ICD2)

แหล่งจ่ายไฟ (POWER SUPPLY)

สำหรับแหล่งจ่ายไฟของบอร์ด CP-PIC ทั้ง V3.0 และ V4.0 นั้น จะสามารถต่อใช้งานได้ทั้งกับไฟกระแสตรงและกระแสสลับ เนื่องจากในบอร์ดได้จัดเตรียมวงจร RECTIFIER แบบ BRIDGE พร้อมวงจร FILTER และ REGULATOR ขนาด +5V ไว้ให้อย่างครบถ้วนอยู่แล้ว โดยผู้ใช้สามารถป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดัน 9 - 12 V โดยสามารถเลือกต่อกับขั้ว CONNECTOR แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านขั้ว CONNECTOR สำหรับ ADAPTER จ่ายไฟก็ได้เช่นกัน โดยการทำงานของแหล่งจ่ายไฟจะมีหลอดแสดงผล LED "VCC" สำหรับแสดงผลการทำงานให้ทราบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานของบอร์ด

การทำงานของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น สามารถกำหนดโหมดการทำงานของบอร์ดได้ 2 โหมด การทำงานด้วยกัน คือ โหมดการโปรแกรม (PROG) และโหมดการทำงานปรกติ (RUN)

การทำงานในโหมดการโปรแกรม (PROG)

ในโหมดนี้จะใช้สำหรับในกรณีที่ต้องการโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ ในบอร์ด CP-PIC V3.0&V4.0 นี้ ได้ออกแบบ พอร์ต ICD2 สำหรับเชื่อมต่อสัญญาณโปรแกรม (เชื่อมต่อสัญญาณกับเครื่องโปรแกรมภายนอก ET-PGM PIC USB) ในการออกแบบได้ใช้ SLIDE SWITCH เพื่อตัดต่อขาสัญญาณที่ใช้ในการโปรแกรม ดังนั้นเมื่อทำงานใน โหมดปกติขาสัญญาณต่างๆ ก็จะถูกแยกออกจากวงจรส่วนของการโปรแกรม ดังนั้นจึงสามารถใช้งานขาสัญญาณต่างๆ ได้ครบทั้งหมด

การเข้าสู่โหมดของการโปรแกรมทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่ง PROG จากนั้นทำการเชื่อมต่อสัญญาณดาวน์โหลด (ICD2 port) เข้ากับเครื่องโปรแกรม และ ทำการโปรแกรมได้เลย โดยรายละเอียดของการโปรแกรมศึกษาได้จากคู่มือการใช้งานของเครื่องโปรแกรม

สวิตช์เลือกโหมด RUN/PROG

การทำงานใน USER MODE หรือ RUN MODE

การทำงานในโหมดนี้ คือ การทำให้ CPU กระทำตามคำสั่งต่างๆ ตามโปรแกรมที่เราได้ออกแบบไว้ซึ่งการเข้าสู่โหมดนี้ ทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่ง RUN สวิตช์ Slide ก็จะทำการแยกขาสัญญาณต่างๆ ออกจากวงจรในส่วนของการโปรแกรม ฉนั้นในการใช้งาน I/O Port จึงสามารถนำมาใช้งานได้ครบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดสรร I/O ของบอร์ด CP-PIC V3.0 & V4.0 (ICD2)

บอร์ด CP-PIC V3.0&V4.0(ICD2) จะใช้ได้กับ CPU เบอร์ 16F877 และ 18F458 ทั้งนี้ขึ้นอยู่กับการเลือกใช้งาน โดยตัว CPU เหล่านี้จะมีขาสัญญาณที่สามารถนำมาใช้งานเป็น I/O Port ซึ่งในเบอร์ 18F458 จะมี I/O รวมทั้งสิ้น 34 เส้น ส่วน 16F877 มี 33 เส้นประกอบด้วย

- RA0-RA6 จำนวน 7 เส้นสัญญาณ (ส่วน 16F877 จะมีเพียง RA0-RA5 คือ 6 เส้นเท่านั้น)
- RB0-RB7 จำนวน 8 เส้นสัญญาณ
- RC0-RC7 จำนวน 8 เส้นสัญญาณ
- RD0-RD7 จำนวน 8 เส้นสัญญาณ
- RE0-RE2 จำนวน 3 เส้นสัญญาณ

โดยการออกแบบวงจรของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น ได้พยายามออกแบบวงจรโดยวางโครงสร้างของบอร์ด ให้มีความอ่อนตัวในการใช้งานมากที่สุด เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปประยุกต์ใช้งานในหลายๆ ลักษณะได้โดยไม่ต้องดัดแปลงโครงสร้างวงจรของบอร์ดไปจากเดิมมากนัก ดังนั้นจึงได้มีการจัดสรรขาสัญญาณ Port I/O ของ CPU ให้สามารถทำงานได้หลายหน้าที่ โดยให้ผู้ใช้สามารถเลือกได้ตามต้องการ ซึ่งหน้าที่การใช้งาน Port I/O ของ CPU ในบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น สามารถสรุปได้ดังต่อไปนี้

RA0-RA3 และ RA5 ขาสัญญาณเหล่านี้นอกจากจะใช้งานเป็น I/O ปกติได้แล้วยังทำหน้าที่เป็นขาอินพุตของสัญญาณอนาล็อก (AN0-AN4) อีกด้วยดังนั้นเราจึงต่อสายสัญญาณเหล่านี้เข้ากับขั้วต่อ ADC/IO(CPU) เพื่อให้สะดวกต่อการนำไปใช้งาน

RA4 จะใช้งานในส่วนของแอลซีดี ซึ่งจะต่อเข้าที่ขา 6 ของคอนเนคเตอร์ CLCD โดยทำหน้าที่เป็นขา Enable ให้กับแอลซีดี

RA6/OSC2/CLKO เป็นขาสัญญาณที่ทำหน้าที่ในหลายส่วน คือ เป็นขา OSC2 และ CLKO จะนำมาใช้เป็นขาสัญญาณ I/O ได้ก็ต่อเมื่อเราใช้คริสตอลออสซิลเลเตอร์แบบที่เป็นโมดูลสำเร็จสามารถต่อเข้ากับขา OSC1/CLKIN ได้เลยโดยที่ไม่ต้องต่อกับขา RA6/OSC2 ทำให้ ขา RA6 ว่างและนำไปใช้เป็น I/O ได้ แต่ในบอร์ดที่เราออกแบบจะใช้งานขา RA6/OSC2 ร่วมกับ OSC1 ในการรับสัญญาณนาฬิกาจากภายนอกดังนั้น ขา RA6 นี้จึงไม่สามารถต่อออกไปใช้งานได้

RB0-RB7 สำหรับขาสัญญาณเหล่านี้จะสามารถใช้งานเป็น I/O ได้ปกติ แต่จะมีคุณสมบัติพิเศษคือจะมีวงจรพูลอัพ (Pull-Up) ภายในและยังเป็นแหล่งกำเนิดสัญญาณอินเตอร์รัพท์ต่างๆ ดังนี้

- RB0/INT0 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 0
- RB1/INT1 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 1
- RB2/INT2 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 2
- RB3/INT3 เป็นขาสัญญาณอินเตอร์รัพท์ภายนอก 3 (เฉพาะเบอร์ 18F442)
- RB4-RB7 เป็นขาที่สามารถกำเนิดสัญญาณอินเตอร์รัพท์ได้หากมีการเปลี่ยนแปลงในขาสัญญาณดังกล่าวและมีการ Enable อินเตอร์รัพท์ประเภทนี้ไว้ จึงเหมาะกับการนำไปใช้งานในส่วนของ สวิตช์คีย์บอร์ด เนื่องจากมีทั้ง อินเตอร์รัพท์และวงจรพูลอัพในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากคุณสมบัติดังกล่าวเราจึงจัดสรรค์การใช้งานดังนี้

RB0/INT0 จะต่อกับขาสัญญาณอินเตอร์รัพท์ของ RTC เบอร์ PCF8583 โดยจะต่อผ่านจัมเปอร์จึงสามารถเลือกที่จะต่อหรือไม่ก็ได้ สามารถเลือกได้โดยการ Shot หรือ Open จัมเปอร์ INT0

RB1/INT1 จะต่อเข้ากับขาสัญญาณอินเตอร์รัพท์ของ PCF8574A และขาสัญญาณอินเตอร์รัพท์ของ Macnetic Card Reader (MCRB02TTL) ทั้งสองส่วนนี้จะต่อผ่านจัมเปอร์ (INT1) ทำให้สามารถเลือกที่จะต่อหรือไม่ต่อก็ได้ การใช้งานจะต้องเลือกใช้งานอย่างใดอย่างหนึ่งเท่านั้น ไม่สามารถใช้งานทั้งสองตัวพร้อมกันได้

RB2 เป็นขาสัญญาณที่ต่อเข้ากับ SPK/BUZZER เพื่อควมคุมการทำงานของ Speaker หรือ Buzzer

RB3 เป็นขาสัญญาณที่ต่อกับวงจรที่ควบคุมการทำงานของรีเลย์(Relay) โดยจะต่อผ่านจัมเปอร์ดังนั้นจึงสามารถเลือกใช้งานหรือไม่ก็ได้โดยการ Shot หรือ Open จัมเปอร์ RB3(RELAY)

RB4-RB7 จะต่อเข้ากับขั้วต่อ KBI/IO สามารถนำไปต่อกับ คีย์บอร์ดประเภท Matrix แบบ 4x4,4x3 หรือ จะใช้เป็น I/O ธรรมดาก็ได้ ในขา RB6 และ RB7 นั้นนอกจากจะต่อกับขั้วต่อ KBI/IO แล้วยังต่อกับสวิตช์ PROG/RUN เพื่อใช้เป็นสัญญาณในการโปรแกรมเมื่ออยู่ในโหมดของการโปรแกรม แต่เมื่ออยู่ในโหมด RUN สามารถนำมาใช้งานเป็น I/O ได้ปกติ

RC0 ขาสัญญาณนี้จะต่อเข้ากับขั้วต่อ แอลซีดี (CLCD) โดยจะต่อเข้าที่ขา 4 ของคอนเนคเตอร์ เพื่อทำหน้าที่เป็นขาสัญญาณ RS เพื่อควบคุมการทำงานของ LCD

RC1 เป็นขาสัญญาณที่ต่อเข้ากับขั้วต่อ OC1B เพื่อใช้งานในส่วนของ ขาสัญญาณอินพุตของ Timer 1 หรือใช้เป็นขาสัญญาณในส่วนของ Capture2 input /Compare2 Output/PWM2

RC2 เป็นขาสัญญาณที่ต่อเข้ากับขั้วต่อ OC1A เพื่อใช้เป็นขาสัญญาณในส่วนของ Capture1 input /Compare1 Output/PWM1

RC3 สำหรับขาสัญญาณ RC3 จะใช้ทำหน้าที่เป็นขาสัญญาณ SCL ในการติดต่อกับอุปกรณ์ I^2C Bus และจะต่อเข้ากับ ขั้วต่อ I^2C EXPAND เพื่อขยายพอร์ต I^2C BUS

RC4 สำหรับขาสัญญาณ RC4 จะใช้ทำหน้าที่เป็นขาสัญญาณ SDA ในการติดต่อกับอุปกรณ์ I^2C Bus และจะต่อเข้ากับ ขั้วต่อ I^2C EXPAND เพื่อขยายพอร์ต I^2C BUS

RC5 จะใช้เป็นสัญญาณควบคุมการรับส่งข้อมูลในการใช้งาน RS485 โดยจะควบคุมการทำงานของอุปกรณ์ที่เป็น Line Driver ก็คือ IC 75176

RC6 เป็นขาสัญญาณที่ทำหน้าที่ในการส่งข้อมูล (Tx) ในโหมดการสื่อสารอนุกรม RS232,RS422 และ RS485 โดยจะต่อเข้ากับ IC ที่เป็น Line Driver คือ Max 232 และ 75176

RC7 เป็นขาสัญญาณที่ทำหน้าที่ในการรับข้อมูล (Rx) ในโหมดการสื่อสารอนุกรม RS232,RS422 และ RS485 โดยจะต่อเข้ากับ IC ที่เป็น Line Driver คือ Max 232 และ 75176

RD0-RD3 สำหรับขาสัญญาณเหล่านี้จะต่อเข้ากับขั้วต่อ KBI/IO เพื่อใช้งานสำหรับการต่อ คีย์สวิตช์ 4x4 หรือ 4x3 ซึ่งเมื่อใช้งานเป็นคีย์บอร์ดดังกล่าวจะทำงานร่วมกับ พอร์ต RB4-RB7 หรือจะใช้งานเป็น I/O ก็ได้

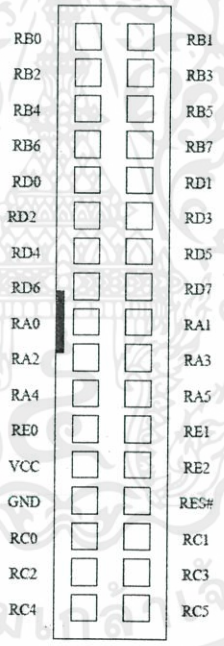
RD4-RD7 ขาสัญญาณเหล่านี้จะทำหน้าที่เป็นขาสัญญาณ Data ที่ใช้ติดต่อกับ LCD โดยจะถูกต่อไปที่คอนเน็คเตอร์ CLCD ซึ่งขั้วต่อ LCD ที่ได้ออกแบบนี้จะเป็นแบบ 4 Bit Data ฉนั้นในการรับส่งข้อมูลจะทำผ่านสายสัญญาณทั้ง 4 เส้น คือ RD4-RD7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RE0-RE2 ขาสัญญาณเหล่านี้สามารถใช้งานเป็น I/O ได้ตามปกติ แต่จะมีคุณสมบัติพิเศษคือขาสัญญาณดังกล่าวจะทำหน้าที่เป็นขาอินพุตอนาลอก (AN5-AN7) เมื่ออยู่ในโหมดของ Analog to Digital โดยเราจะนำไปต่อกับขั้วต่อ ADC/IO(CPU) ทำให้สามารถต่อออกไปใช้งานได้สะดวก

การใช้งานขั้วต่อ 34PIN (72IOZ80)

สำหรับขั้วต่อ Connector ขนาด 34 PIN ของบอร์ด CP-PIC V3.0&V4.0(ICD2) นั้น จะมีอยู่ด้วยกัน 2 แบบ คือ ถ้าเป็นบอร์ด CP-PIC V3.0(ICD2) และ CP-PIC V4.0(ICD2) นั้น จะเป็นขั้วแบบ IDE ขนาด 34 PIN ตัวผู้ ซึ่งขั้วต่อนี้ออกแบบไว้สำหรับให้ผู้ใช้เชื่อมต่อสัญญาณต่างๆของ CPU ออกไปใช้งานกับบอร์ดอื่นๆ ซึ่งอาจเป็นบอร์ดที่ผู้ใช้ออกแบบและสร้างขึ้นเอง หรืออาจใช้บอร์ด I/O ต่างๆที่ ทาง บริษัท อีทีที จำกัด สร้างขึ้นไว้สนับสนุนการใช้งานก็ได้ โดยวิธีการเชื่อมต่อนั้นขอแนะนำให้ใช้สายแพร์ขนาด 34 PIN จะสะดวกที่สุดเพราะสามารถทำการเชื่อมต่อหรือแยกบอร์ดออกจากกันได้ง่าย ส่วนในกรณีที่ใช้บอร์ดรุ่น CP-PIC V3.0 EXPANSION(ICD2) นั้น ขั้วต่อ 34 PIN จะเป็นแบบ IDE ตัวเมีย เพื่อให้ผู้ใช้สามารถใช้สาย Jumper ต่อสัญญาณต่างๆจากขั้วต่อนี้ไปยังแผงทดลอง Photo Board เพื่อต่อร่วมกับวงจรต่างๆได้โดยง่าย โดยลักษณะการจัดเรียงสัญญาณเป็นดังนี้



รูปแสดงลักษณะของการจัดเรียงสัญญาณของขั้ว 34PIN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานขั้วต่อ KBI/IO

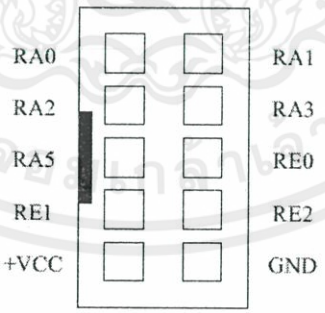
พอร์ต KBI/IO ถูกจัดไว้ที่ขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะ ในบอร์ด รุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อนี้จะเชื่อมต่อสัญญาณมาจาก PORTB(RB4-RB7) และ PORTD(RD0-RD3) ของ CPU ทั้งหมด 8 เส้น ซึ่งเหมาะกับการนำไปประยุกต์ใช้งานในส่วนของการเชื่อมต่อกับวงจรคีย์บอร์ดแบบ Matrix ซึ่งสามารถจะใช้ได้กับคีย์บอร์ดแบบ Matrix ขนาด 4x3 หรือ 4x4 ก็ได้

สำหรับในกรณีที่ไม่มีความจำเป็นต้องใช้งานคีย์บอร์ดแล้ว ขั้วต่อ KBI/IO นี้ก็ยังสามารถนำไปต่อใช้งานเป็น Input หรือ Output ทั่วๆไปได้อีกด้วย

รูปแสดงลักษณะของการจัดเรียงสัญญาณของขั้ว KBI/IO

การใช้งานขั้วต่อ ADC/IO(CPU)

พอร์ต ADC/IO(CPU) นี้จะถูกเชื่อมต่อออกมารอไว้ยังขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อนี้ สามารถนำไปต่อใช้งานเป็นอินพุตของสัญญาณอนาลอกทั้ง 8 เชนแนล (AN0-AN7) ซึ่งหากไม่ใช้งานในส่วนนี้สามารถใช้งานเป็นอินพุตเอาต์พุตพอร์ตได้ตามปกติ ซึ่งขาสัญญาณต่างๆ ที่นำมาต่อจะเป็นดังรูปด้านล่าง

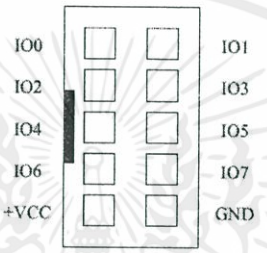


รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว ADC/IO(CPU)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานขั้วต่อ I²C IN/OUT

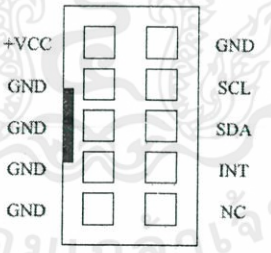
พอร์ต I²C IN/OUT นี้จะถูกเชื่อมต่อออกมารอไว้ยังขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อ I²C IN/OUT นี้ จะเชื่อมต่อมาจากขาสัญญาณ I/O Port ของ PCF8574/A ซึ่งสามารถโปรแกรมหน้าที่การใช้งาน ให้เป็น Input หรือ Output ก็ได้ตามต้องการจากโปรแกรม แต่ต้องกำหนดหน้าที่ให้เป็น Input หรือ Output อย่างใดอย่างหนึ่งเท่านั้น ไม่สามารถใช้งานทั้งสองหน้าที่พร้อมๆกันได้ และในการกำหนดหน้าที่ให้เป็น Input หรือ Output ก็ต้องกำหนดให้เหมือนกันทั้ง 8 บิต ด้วย โดยลักษณะของขาสัญญาณที่จัดเรียงไว้ที่ขั้ว I²C IN/OUT จะเป็นดังรูป



รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว I²C IN/OUT

การใช้งานขั้วต่อ I²C BUS EXPAND

พอร์ต I²C BUS EXPAND นี้จะถูกเชื่อมต่อออกมารอไว้ยังขั้ว Connector ขนาด 10 PIN แบบ IDE ซึ่งขั้วต่อนี้จะมีอยู่เฉพาะบอร์ดในรุ่น CP-PIC V4.0(ICD2) เท่านั้น โดยขั้วต่อ I²C BUS EXPAND นี้ จะใช้สำหรับทำการขยายหรือเพิ่มเติมจำนวนอุปกรณ์ที่ใช้การติดต่อสื่อสารแบบ I²C ให้กับบอร์ด



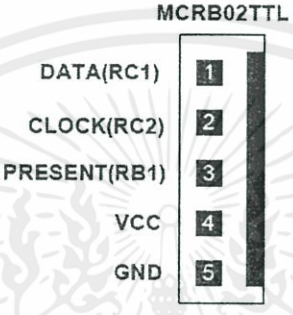
รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้ว I²C BUS EXPAND

การใช้งานเครื่องอ่านบัตรแถบแม่เหล็ก (MAGNETIC-CARD READER)

สำหรับบอร์ด CP-PIC V4.0(ICD2) นั้น จะออกแบบให้สามารถเชื่อมต่อกับเครื่องอ่านบัตรแถบแม่เหล็กรุ่น "MCR-B02TTL" ได้โดยตรง โดยไม่ต้องดัดแปลงวงจรใดๆทั้งสิ้น โดยในบอร์ดจะจัดเตรียมขั้วแบบ CPA ขนาด 5 PIN ไว้รองรับอยู่แล้ว ผู้ใช้สามารถนำขั้วต่อของเครื่องอ่านบัตรแถบแม่เหล็ก รุ่น "MCR-B02TTL" ของบริษัท อีทีที ต่อเข้าไปได้ทันที สำหรับในการเขียนโปรแกรมเพื่อเชื่อมต่อระหว่างบอร์ด CP-PIC V4.0(ICD2) กับเครื่องอ่านบัตรแถบแม่เหล็กนั้น จะสามารถทำได้ 2 แบบ คือ ใช้วิธีการวนรอบตรวจสอบสัญญาณจากเครื่องอ่านบัตรแถบแม่เหล็กเอง ซึ่งวิธีการนี้จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณจาก CPU เพียง 2 เส้นสัญญาณคือ RC1 ทำหน้าที่เป็น DATA และ RC2 ทำหน้าที่เป็น CLOCK โดยต้องกำหนดคุณสมบัติของสัญญาณทั้ง 2 เส้นให้มีทิศทางเป็น Input ส่วนอีกวิธีหนึ่งคือการใช้วิธีการ Interrupt ซึ่งสามารถทำได้โดยการ SHORT JUMPER "INT1" ที่อยู่ใกล้ๆกับขั้วต่อเครื่องอ่านบัตรแถบแม่เหล็ก ซึ่งการ SHORT JUMPER จะเป็นการเชื่อมต่อสัญญาณอินเตอร์รัพท์ของ CPU เข้ากับ สัญญาณ PRESENT ของเครื่องอ่านบัตรแถบแม่เหล็ก ดังนั้นเมื่อมีการนำบัตรแถบแม่เหล็กไปรูดผ่านเครื่องอ่านบัตรแถบแม่เหล็กก็จะมีการส่งสัญญาณ Interrupt ออกมายัง CPU ซึ่งผู้ใช้ก็เพียงแต่เขียนโปรแกรมสำหรับบริการการ Interrupt ของ INT1 ไว้ก็สามารถใช้งานได้แล้ว



รูปแสดง ลักษณะของการจัดเรียงสัญญาณของขั้วต่อเครื่องอ่านบัตรแถบแม่เหล็ก MCR-B02TTL

<u>หมายเหตุ</u> เนื่องจากการเลือกใช้ Interrupt จากสัญญาณ INT1 ของ CPU นั้น ภายในบอร์ด รุ่น CP-PIC V4.0(ICD2) สัญญาณ INT1 จะถูกออกแบบให้สามารถเลือกกำหนดใช้งานร่วมกับอุปกรณ์หลายๆตัว เช่น การ Interrupt จาก PORT I/O เบอร์ PCF8574/A รวมทั้งการ Interrupt จาก เครื่องอ่านบัตรแถบแม่เหล็ก MCR-B02TTL นี้ด้วย ดังนั้น ในกรณีที่ต้องการเลือกใช้วิธีการ Interrupt จากเครื่องอ่านบัตรแม่เหล็กนั้น จะต้องทำการ OPEN JUMPER สำหรับเลือกการ Interrupt จากอุปกรณ์อื่นๆที่ไม่เกี่ยวข้องออกเสียก่อน ให้เหลือการเชื่อมต่อสัญญาณ INT1 จาก CPU กับอุปกรณ์เพียงตัวใดตัวหนึ่งเท่านั้น เพื่อให้แน่ใจว่าสัญญาณ Interrupt ที่เกิดขึ้นจะถูกส่งมาออกมาจากเครื่องอ่านบัตรแถบแม่เหล็กเท่านั้น ไม่เช่นนั้นแล้วอาจเกิดความผิดพลาดขึ้นได้

การใช้งาน OUTPUT RELAY

ภายในบอร์ด CP-PIC V4.0(ICD2) นั้น จะออกแบบวงจรควบคุม RELAY ไว้ให้ผู้ใช้งานสามารถนำไปประยุกต์ใช้งานทั่วไปได้ด้วยจำนวน1ชุดโดยวงจรRELAYดังกล่าวสามารถใช้งานได้ ทั้งหน้าสัมผัสแบบปกติเปิด (Normal Open : NO) และแบบหน้าสัมผัสปกติปิด (Normal Close : NC)

สำหรับสัญญาณ Output ในการควบคุมการทำงานของ REALY นั้น จะมาจากขา RB3 ของ CPU ซึ่งเมื่อต้องการใช้งาน RELAY จะต้องทำการ SHORT JUMPER RB3(RELAY) ไว้ด้วยเพื่อเชื่อมต่อสัญญาณRB3 มาทำการควบคุมการทำงานของ RELAY และต้องแน่ใจว่าไม่ได้ต่อสัญญาณ RB3 จากขั้วต่ออื่นๆออกไปใช้งานกับอุปกรณ์ใดๆนอกเหนือจาก RELAY เนื่องจากสัญญาณ RB3 นั้น นอกจากจะต่อมาใช้ควบคุมการทำงานของ RELAY แล้วยังต่อไปยังขั้วต่อ 34PIN อีกด้วย โดยการทำงานของ RELAY จะถูกควบคุมการทำงานจากขาสัญญาณ RB3 โดยผู้ใช้ต้องกำหนดคุณสมบัติของสัญญาณ RB3 ให้ทำหน้าที่เป็น OUTPUT ไว้ด้วย ซึ่งเมื่อขาสัญญาณ RB3 มีสภาวะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTPUT และให้สถานะเป็น "1" จะทำให้ RELAY ทำงาน แต่ถ้าสถานะของสัญญาณ RB3 มีค่าเป็น "0" จะทำให้ RELAY หยุดทำงาน

รูปแสดง ลักษณะขั้วต่อสัญญาณจากหน้าสัมผัสของ RELAY

หมายเหตุ เนื่องจากสัญญาณ RB3 ที่นำมาใช้ควบคุมการทำงานของ RELAY จะเป็นสัญญาณเส้นเดียวกับ RB3 ที่ต่อไว้ยังขั้ว 34PIN ดังนั้น เมื่อต้องการใช้งาน RELAY โดยการควบคุมจาก RB3 แล้ว ต้องแน่ใจว่าไม่ได้ต่อใช้งานสัญญาณ RB3 ในจุดอื่นๆอีก แต่ถ้าหากมีความจำเป็นต้องใช้งาน RB3 พร้อมกับการใช้งาน RELAY ด้วยในเวลาเดียวกัน ก็อาจดัดแปลงวงจรได้โดยการ OPEN JUMPER RB3(RELAY) ออก แล้วใช้วิธีการเชื่อมสายสัญญาณเส้นอื่นๆจาก PORT I/Oของ CPU ที่ไม่ได้ใช้งานมาเข้ากับวงจรควบคุม RELAY แทนก็ได้เช่นเดียวกัน โดยให้เชื่อมต่อสายสัญญาณที่ต้องการไปยัง Jumper RB3(RELAY) ด้านที่ต่อกับตัวต้านทานค่า 1KOhm แต่การดัดแปลงวิธีนี้ควรถอดตัว JUMPER RB3(RELAY) ออกจากบอร์ดเสียก่อน เพื่อจะได้ไม่หลงลืมทำการ SHORT JUMPER นี้ซ้ำอีกในภายหลัง เนื่องจากจะเป็นการ SHORT สัญญาณ RB3 เข้ากับสัญญาณเส้นใหม่ที่บัดกรีมายังวงจร RELAY นี้อีก

การใช้งานลำโพงขนาดเล็ก หรือ BUZZER

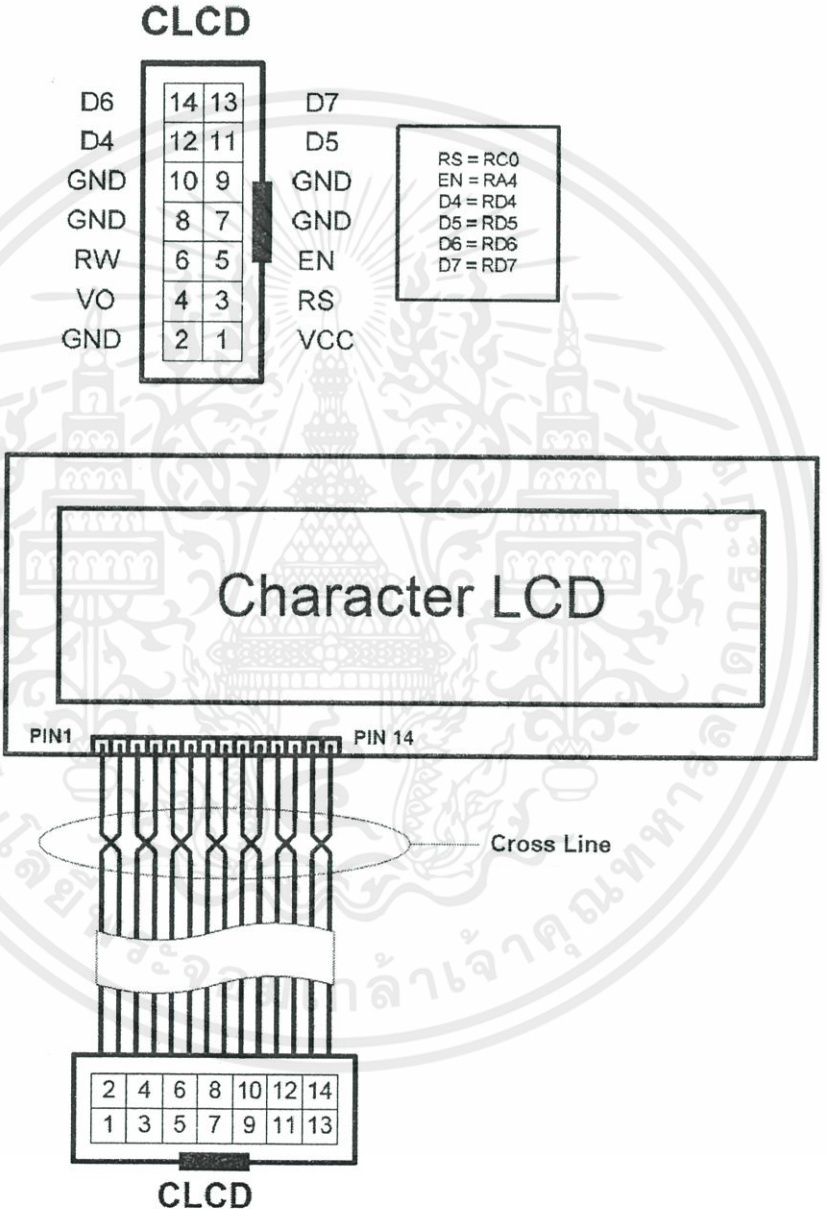
ภายในบอร์ด CP-PIC V4.0(ICD2) จะมีวงจรกำเนิดเสียงรวมอยู่ด้วย 1 จุด ซึ่งในตำแหน่งนี้สามารถเลือกใส่อุปกรณ์กำเนิดเสียงแบบลำโพงขนาดเล็ก หรือ จะเลือกใส่ BUZZER แทนก็ได้เช่นเดียวกัน โดยในกรณีที่เลือกใช้ลำโพงจะมีข้อดีคือ สามารถสร้างความถี่เสียงได้หลากหลายความถี่ตามต้องการแต่การเขียนโปรแกรมจะยุ่งยากกว่า BUZZER เนื่องจากต้องสร้างเป็นสัญญาณความถี่จึงจะสามารถทำให้ลำโพงกำเนิดเสียงให้ได้ ส่วนในกรณีที่เลือกใช้ BUZZER นั้น จะมีข้อดีคือ เขียนโปรแกรมควบคุมการกำเนิดเสียงได้ง่ายกว่าลำโพง เนื่องจากใช้วิธีการ ON หรือ OFF เท่านั้น โดยการ ON บิตควบคุม BUZZER ให้มีสภาวะเป็น "1" เท่านั้น BUZZER ก็จะกำเนิดเสียงให้แล้ว แต่ความถี่เสียงของ BUZZER จะไม่สามารถเลือกได้ เหมือนกับลำโพง

สำหรับสัญญาณ Output ในการควบคุมการทำงานของ ลำโพง หรือ BUZZER นั้น จะมาจากขา RB2 ของ CPU ซึ่งการควบคุมการทำงานของลำโพง หรือ BUZZER ผู้ใช้ต้องทำการกำหนดคุณสมบัติของสัญญาณ RB2 ให้ทำหน้าที่เป็น OUTPUT ไว้ด้วย ซึ่งเมื่อขาสัญญาณ RB2 มีสภาวะเป็น OUTPUT และให้สถานะเป็น "1" จะทำให้ ลำโพง หรือ BUZZER ทำงาน แต่ถ้าสถานะของสัญญาณ RB2 มีค่าเป็น "0" จะทำให้ ลำโพง หรือ BUZZER หยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานจอแสดงผลแบบ LCD (Dot-Matrix Character LCD)

บอร์ด CP-PIC V4.0(ICD2) สามารถใช้เชื่อมต่อกับจอแสดงผล LCD แบบ Dot-Matrix โดยเชื่อมต่อผ่านทาง Connector ขนาด 14 PIN และใช้ตัวต้านทานปรับค่าได้แบบเกือกม้าขนาด 10K สำหรับปรับระดับความสว่างของหน้าจอ LCD โดยวงจรในการเชื่อมต่อ LCD ของบอร์ดนี้จะออกแบบวงจรให้ใช้วิธีการควบคุมการทำงานแบบ "DATA 4-BIT"



รูปแสดง ขาสัญญาณของขั้วต่อ CLCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับวิธีการเชื่อมต่อสัญญาณจากขั้วต่อ LCD ของบอร์ดไปเข้ากับตัว LCD นั้น เพื่อความสะดวกขอแนะนำให้ใช้สายแพร์ ขนาด 14 เส้น เป็นตัวเชื่อมต่อสัญญาณระหว่างบอร์ดและตัว LCD จะสะดวกมากที่สุด ซึ่งในปัจจุบันพบว่า ลักษณะขั้วสัญญาณที่อยู่ทางด้านของจอแสดงผล LCD เองนั้น ที่พบเห็นได้ทั่วๆไป จะมีอยู่ ด้วยกัน 2 แบบ คือ

- แบบที่เป็นขั้วต่อแบบแถวเดี่ยว ขนาด 14PIN (HEADER 14X1) โดยการต่อสายของ LCD แบบนี้ จะใช้สายแพร์ขนาด 14 เส้น แบบเข้าหัว CONNECTOR ไว้ด้านเดียว สำหรับเสียบกับขั้วต่อ CLCD ภายในบอร์ด CP-PIC V4.0 ส่วนปลายสายอีกด้านหนึ่งของสายแพร์ทั้ง 14 เส้น จะต้องนำไปบัดกรีเข้ากับขั้วต่อของตัว LCD ให้ครบทั้ง 14 เส้น โดยในการบัดกรีจะต้องสลับปลายสายเป็นคู่ๆเรียงลำดับกันไป คือ ขา 2 สลับกับ 1,ขา4 สลับกับ 3...ขา14 สลับกับ 13 ตามลำดับ กล่าวคือ สายเส้นที่1 ต่อกับ PIN2 ของ LCD ส่วนสายเส้นที่2 จะต้องต่อกับ PIN1 ของ LCD และในทำนองเดียวกันสายเส้นที่3 ก็จะต้องต่อกับ PIN4 ของ LCD อย่างนี้เรื่อยไปจนครบทั้ง 14 เส้น
- แบบที่เป็นขั้วต่อแบบแถวคู่ 14PIN (HEADER ขนาด 7X2) โดยการต่อสายของ LCD แบบนี้ จะใช้สายแพร์ขนาด 14 เส้น แบบเข้าหัว CONNECTOR ไว้ทั้งสองด้าน โดยในการเชื่อมต่อนั้น ให้ต่อสายแต่ละด้านเข้ากับขั้วต่อ โดยให้ตำแหน่งของ PIN1 ของขั้วต่อแต่ละด้านอยู่ในตำแหน่งที่ตรงกันก็สามารถใช้งานได้แล้ว

หมายเหตุ ใน CLCD บางรุ่นนั้น อาจมีขั้วต่อสัญญาณเพิ่มเป็น ขนาด 16 PIN ซึ่งในกรณีนี้ ก็จะยังคงใช้วิธีการเชื่อมต่อแบบเดิม คือจะใช้สัญญาณในการเชื่อมต่อระหว่าง CLCD กับบอร์ด เพียงแค่ 14 PIN เท่านั้น ส่วนสัญญาณขา 15 และ 16 ที่เพิ่มเข้ามานั้นจะเป็นขาไฟเลี้ยงของวงจร LED Back-Light (A และ K) ซึ่งถ้า LCD ที่ซื้อมาใช้งานมี LED Back-Light รวมอยู่ด้วย ขอแนะนำให้แยกต่อไฟเลี้ยง LED Back-light เข้ากับแหล่งจ่ายไฟ +5V โดยตรงต่างหากก็ได้ หรือถ้าต้องการให้ LED Back-Light ทำงานตลอดเวลาก็อาจทำการต่อขาสัญญาณ (A) หรือขา 15 เข้ากับขา 2 ของ LCD ส่วนขา (K) ก็ให้ต่อเข้ากับขา 1 ของ LCD ก็ได้เช่นกัน

การเชื่อมต่อกับอุปกรณ์ I^2C BUS

สำหรับอุปกรณ์แบบ I^2C Bus ที่ใช้ในบอร์ด CP-PIC V4.0 นั้น จะออกแบบให้สามารถติดตั้งใช้งานอุปกรณ์ I^2C ได้พร้อมกันในบอร์ดทั้งหมดด้วยกัน 3 ตัว คือ

- I^2C RTC เบอร์ PCF8583 ของ PHILIPS
- EEPROM ในตระกูล 24XX ซึ่งสามารถเลือกใช้ได้หลายเบอร์หลายผู้ผลิต ขึ้นอยู่กับขนาดความจุของหน่วยความจำที่ต้องการจะใช้ ซึ่งในบอร์ด CP-PIC V4.0(ICD2) นั้น สามารถติดตั้งใช้งานหน่วยความจำ EEPROM แบบ I^2C นี้ได้ตั้งแต่ เบอร์ 24XX32,24XX64,24XX128,24XX256 หรือ 24XX512 เป็นต้น
- I^2C I/O เบอร์ PCF8574 หรือ PCF8574A ของ Phillips

โดยอุปกรณ์ I^2C ทั้ง 3 ตัวนี้จะต่อร่วมกันอยู่ภายในบัสเดียวกัน และใช้สัญญาณ RC3 เป็นขาสัญญาณ SCL และใช้สัญญาณ RC4 เป็นสัญญาณ SDA ในการควบคุมบัส ซึ่ง CPU จะทำหน้าที่เป็นตัวแม่ในการควบคุมบัส นอกจากนี้แล้วยังสามารถขยายอุปกรณ์จำพวก I^2C นี้ได้อีก แต่ต้องเป็นอุปกรณ์ที่มีรหัสควบคุม Control Word ไม่ซ้ำกันกับอุปกรณ์ที่มีอยู่แล้วภายในบอร์ดด้วยโดยอาจเชื่อมต่อผ่านทางขั้วต่อ "I^2C EXPANSION" ที่บอร์ดจัดเตรียมไว้ให้แล้วก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน Interrupt ของอุปกรณ์ I²C

สำหรับสัญญาณ Interrupt จากอุปกรณ์ I²C นั้น เนื่องจาก CPU มีสัญญาณ Input ที่ใช้สำหรับรับการ Interrupt จากภายนอก 2 สัญญาณ คือ INT0 และ INT1 ดังนั้น เราจึงได้ออกแบบการต่อใช้งานสัญญาณ Interrupt ดังนี้ คือ

- ต่อสัญญาณการ Interrupt INT1ให้กับเครื่องอ่านบัตรแถบแม่เหล็ก
- ต่อสัญญาณการ Interrupt INT0ให้กับ RTC PCF8583
- ต่อสัญญาณการ Interrupt INT1ให้กับ PCF8574/A

จะเห็นได้ว่าสัญญาณ Interrupt INT1 จะถูกต่อใช้งานถึงสองส่วนคือ เชื่อมต่อกับสัญญาณ Interrupt จากเครื่องอ่านแถบแม่เหล็ก และ ต่อกับสัญญาณ Interrupt ของ PCF8574/A ดังนั้นในการใช้งานจะต้องเลือกใช้งานอย่างใดอย่างหนึ่งเท่านั้น ซึ่งสามารถทำได้โดยการกำหนดการทำงานให้กับ จัมเปอร์ INT1 เช่น หากใช้งาน Interrupt ของ PCF8574/A ก็ให้ Short Jumper INT1 ของ PCF8574/A และ Open Jumper INT1 ของเครื่องอ่านบัตรแม่เหล็ก MCRB02TTL

แอดเดรสของอุปกรณ์ I²C

เนื่องจากคุณสมบัติของ BUS แบบ I²C นั้น สามารถเชื่อมต่ออุปกรณ์ต่างๆที่ใช้วิธีการสื่อสารแบบ I²C ได้มากมากหลายตัวภายในบัสเดียวกันได้ เพียงแต่มีข้อแม้ว่า อุปกรณ์ที่จะนำมาต่อร่วมกันภายในบัสเดียวกันนั้น จะต้องมีรหัสตำแหน่งในการติดต่อสื่อสาร (Control Byte) ที่ไม่ซ้ำกัน ซึ่งอุปกรณ์บางตัวนั้น ผู้ผลิตได้มีการออกแบบให้สามารถกำหนดค่ารหัสตำแหน่ง Control Byte ได้มากกว่า 1 ค่าเพื่อให้สามารถเชื่อมต่ออุปกรณ์ประเภทเดียวกันร่วมกันภายในบัสเดียวกันได้มากกว่า 1 ตัว โดยใช้วิธีการกำหนดค่าโลจิกให้กับขาสัญญาณสำหรับใช้ระบุตำแหน่ง (Address) ของอุปกรณ์เบอร์นั้นๆได้เอง เช่น I/O Port เบอร์ PCF8574 นั้น สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 8 ตัว และยังสามารถเชื่อมต่ออุปกรณ์ I/O Port ที่มีคุณสมบัติเหมือนกันแต่มีรหัสตำแหน่งที่แตกต่างกันคือ PCF8574A เพิ่มเติมได้อีก 8 ตัว ซึ่งจะเห็นได้ว่าอุปกรณ์ I/O Port นั้นสามารถทำการเพิ่มเติมเข้าไปให้ระบบบัสเดียวกันได้มากถึง 16 ตัว และในทำนองเดียวกัน หน่วยความจำ E²PROM เบอร์ 24LC256 ก็สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 8 ตัว จากตัวอย่างข้างต้นจะเห็นได้ว่าภายในบัสเดียวกันของ I²C นั้น อุปกรณ์เพียง 2 ประเภท คือ I/O และ E²PROM สามารถต่อร่วมกันภายในบัสเดียวกันได้มากถึง 24 ตัว คือ I/O PCF8574 8 ตัว ,PCF8574A 8ตัว และ 24LC256 อีก 8 ตัว และยังสามารถนำอุปกรณ์ I²C อื่นๆที่มีรหัสตำแหน่งของ Control Byte ไม่ซ้ำกันมาต่อเพิ่มเติมได้อีก แต่อย่างไรก็ตามถึงแม้ว่าอุปกรณ์แบบ I²C นี้ยอมให้มีการเชื่อมต่อร่วมกันภายในบัสเดียวกันได้หลายตัวภายในระบบบัสเดียวกันก็ตาม แต่ในทางปฏิบัติแล้วอาจเกิดข้อจำกัดในเรื่องของโหลด (FAN-IN/FAN-OUT) เนื่องจากคุณสมบัติของ Port I/O ของ CPU เอง ก็มีข้อจำกัดในการขับกระแสให้กับโหลดได้ประมาณ 25mA เท่านั้น ซึ่งคงไม่สามารถต่ออุปกรณ์ร่วมกันในบัสได้โดยไม่จำกัดจำนวนเหมือนในทฤษฎีบอกไว้ ซึ่งในความเป็นจริงอาจต้องพิจารณาตามความเหมาะสมและความจำเป็นในการใช้งานจริงๆด้วยว่าในระบบบัสหนึ่งๆของ I²C นั้นควรต่ออุปกรณ์ในบัสจำนวนเท่าใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่และเบอร์ของอุปกรณ์ I²C	รหัสตำแหน่งมาตรฐานในการอ่าน/เขียน	รหัสตำแหน่งของบอร์ด CP-PIC V4.0	
		รหัสตำแหน่งในการอ่าน	รหัสตำแหน่งในการเขียน
RTC : PCF8583	[1][0][1][0][0][0][X][?]	[1][0][1][0][0][0][1][1]	[1][0][1][0][0][0][1][0]
E²PROM:24XX	[1][0][1][0][X][X][X][?]	[1][0][1][0][1][0][0][1]	[1][0][1][0][1][0][0][0]
I/O : PCF8574	[0][1][0][0][X][X][X][?]	[0][1][0][0][0][0][0][1]	[0][1][0][0][0][0][0][0]
I/O : PCF8574A	[0][1][1][1][X][X][X][?]	[0][1][1][1][0][0][0][1]	[0][1][1][1][0][0][0][0]

ตารางแสดง รหัสตำแหน่งของอุปกรณ์ I2C ภายในบอร์ด CP-PIC V4.0

หมายเหตุ 1. ค่า X หมายถึงค่าโลจิกของขาสัญญาณ Address ของอุปกรณ์ที่กำหนดในวงจร

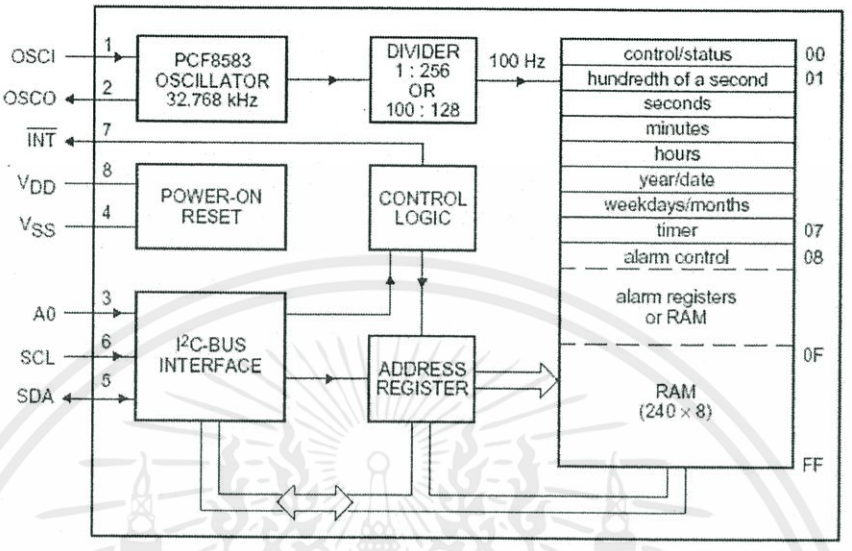
2. ค่า ? หมายถึงบิตสำหรับกำหนดว่าต้องการเขียน หรือ อ่าน ข้อมูลกับอุปกรณ์

3. เนื่องจาก RTC เบอร์ PCF8583 และ E²PROM เบอร์ในกลุ่ม 24XX นั้นมีรหัสตำแหน่ง 4บิตแรกซ้ำกัน หรือ อยู่ในกลุ่มเดียวกัน ซึ่งในบอร์ด CP-PIC V4.0 นั้นออกแบบให้ RTC เบอร์ PCF8583 มีรหัสตำแหน่งของ Control Byte คงที่เป็น 1010001X ไว้ ส่วน E²PROM ก็กำหนดรหัสตำแหน่ง Control Byte ไว้ที่ 1010100X ดังนั้นถ้าต้องการเพิ่มเติมอุปกรณ์ใดๆเข้าไปอีกต้องกำหนดให้ค่า Control Byte ของอุปกรณ์ที่จะต่อเพิ่มเข้าไปไม่ซ้ำกับค่า Control Byte ทั้งสองดังกล่าวนี้ด้วย

การใช้งาน I²C RTC เบอร์ PCF8583

สำหรับวงจรฐานเวลา RTC นั้น ในบอร์ด CP-PIC V4.0 นั้น จะเลือกใช้ Chips Support ของ PHILIPS เบอร์ PCF8583 ซึ่งเป็นชิพฐานเวลาแบบ I²C-Bus และมีฐานเวลาให้ใช้งานอย่างครบถ้วน ตั้งแต่ วินาที/นาที/ชั่วโมง/วันที่/เดือน/วันในสัปดาห์ และปีคศ. นอกจากนี้ยังมีความอ่อนตัวในการใช้งานค่อนข้างดีเกี่ยวกับระบบเวลา เช่น ค่าของชั่วโมงสามารถกำหนดได้จากโปรแกรมว่าจะให้เป็นระบบ 12 ชั่วโมง หรือ 24 ชั่วโมง และในส่วนของวันที่และวันในสัปดาห์ก็สามารถปรับเปลี่ยนได้เองว่า เดือนใดมี 28/29/30 หรือ 31 วันอย่างอัตโนมัติ ซึ่งนอกจากจะใช้งานเป็นฐานเวลา RTC แล้ว PCF8583 นี้ยังมีฟังก์ชั่นพิเศษในการตั้งเวลาสำหรับเปิดปิดการทำงานของอุปกรณ์ต่างๆ (ALARM FUNCTION) ได้อีกด้วย นอกจากนี้แล้วในตัวของ RTC เองยังมีหน่วยความจำ RAM ขนาด 8บิต จำนวน 240ไบท์ สำหรับให้ผู้ใช้นำไปใช้งานเก็บข้อมูลได้อย่างอิสระ เช่น อาจนำไปใช้ในการเก็บค่าการตั้ง เวลา เพื่อใช้ ตั้งเวลาเปิด-ปิดอุปกรณ์ไฟฟ้า เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงโครงสร้างภายในของ RTC เบอร์ PCF8583

จะเห็นได้ว่า PCF8583 ประกอบขึ้นจากวงจรหลายๆส่วน เช่น วงจร Power-on Reset วงจรเชื่อมต่อแบบ I^2C วงจรถอดรหัสตำแหน่งแอดเดรส วงจรหารความถี่ และวงจรกำเนิดความถี่ขนาด 32.768KHz โดยต้องต่อคริสตัลจากภายนอกให้กับขา OSCI และ OSCO ด้วย สำหรับหน่วยความจำนั้น PCF8583 จะมีโครงสร้างของหน่วยความจำขนาด 8บิต จำนวน 256 ไบท์ โดยจัดสรรสำหรับแบ่งออกเป็นรีจิสเตอร์ของส่วนที่เป็นฐานเวลาจำนวน 16ไบท์(00H-0FH) และใช้งานเป็น หน่วยความจำ RAM ทั่วไปได้อีก 240ไบท์(10H-FFH) ซึ่งในการประยุกต์ใช้งานนั้น ตามปรกติแล้วจะสามารถใช้งานในหน้าที่ของ RTC(Clock Mode) หรือใช้งานเป็นตัวนับ Counter (Event Counter) สำหรับนับความถี่จากขาสัญญาณ OSCI ก็ได้ แต่สำหรับวงจรของ PCF8583 ภายในบอร์ด CP-PIC V4.0 นั้นจะออกแบบให้ใช้งาน PCF8583 ในโหมด RTC หรือ Clock Mode เท่านั้น

การติดต่อสื่อสารกับ RTC เบอร์ PCF8583

ในการเขียนโปรแกรมติดต่อกับ RTC นั้น จะใช้วิธีการเชื่อมต่อแบบมาตรฐาน I^2C Bus โดยใน RTC เบอร์ PCF8583 นี้จะมีตำแหน่งแอดเดรสในการติดต่อภายในบัส หรือ Control Byte เป็น "1010001X" ดังนั้นในการติดต่อกับ RTC ไม่ว่าจะเป็นการเขียนข้อมูลหรืออ่านข้อมูลจากตัว RTC ก็ตามที หลังจากสร้างสภาวะเริ่มต้น (Start Condition) แล้วจะต้องส่งค่า Control Byte ของตัว RTC ในบัส ด้วยค่า "1010001X" เพื่อบอกให้ RTC รับรู้ว่า CPU ต้องการจะอ่านหรือเขียนข้อมูลให้กับ RTC จากนั้นจึงส่งรหัส ไบท์แอดเดรส เพื่อระบุตำแหน่งแอดเดรสเริ่มต้นภายในตัว RTC ที่ต้องการจะอ่านหรือเขียนข้อมูลให้กับ RTC เป็นลำดับต่อไป โดยถ้าเป็นตำแหน่งแอดเดรสของฐานเวลาภายในตัว RTC จะมีค่าตำแหน่งแอดเดรสอยู่ระหว่าง 00H-0FH แต่ ถ้าเป็นตำแหน่งแอดเดรสของ RAM ภายในตัว RTC จะมีค่าอยู่ระหว่าง 10H-FFH ตามลำดับ โดยรหัส Control Byte ของ RTC นั้นมีลักษณะโครงสร้างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
1	0	1	0	0	0	A0	R/$\overline{W}$

รูปแสดง โครงสร้างของ Control Byte ของ PCF8583

หมายเหตุ บิต0 (R/W) นั้นจะใช้สำหรับกำหนดว่าจะอ่านหรือเขียนข้อมูลจากอุปกรณ์

ซึ่งจะเห็นว่าตามสภาวะปรกติแล้ว Control Byte ของ PCF8583 นั้น สามารถเลือกได้ 2 ค่า โดยการกำหนดโลจิกให้กับขาสัญญาณ A0 ของ PCF8583 เอง ดังนั้นในระบบบัสเดียวกัน จึงสามารถทำการติดตั้ง RTC เบอร์ PCF8583 นี้ได้ 2 ตัว โดยต้องกำหนดให้ขาสัญญาณ A0 ของแต่ละตัวมีสภาวะเป็น "0" และ "1" ซึ่ง ตัวที่ขาสัญญาณ A0 มีสภาวะเป็น "0" ก็จะมีรหัส Control Byte เป็น "1010000X" ส่วนตัวที่ขาสัญญาณ A0 ได้รับสภาวะลอจิกเป็น "1" ก็จะมีรหัส Control Byte เป็น "1010001X" แทน

แต่สำหรับบอร์ด CP-PIC V4.0 นั้น จะกำหนดให้ขาสัญญาณ A0 ของ PCF8583 มีสภาวะทางโลจิกเป็น "1" คงที่ไว้เลย ดังนั้น RTC เบอร์ PCF8583 ในบอร์ด CP-PIC V4.0 นั้นจึงมีรหัส Control Byte คงที่เป็น "1010001X" เสมอ

หมายเหตุ ค่า X หรือ บิต0 ใน Control Byte เป็นบิตสำหรับกำหนดคุณสมบัติในการอ่านหรือเขียนข้อมูลกับอุปกรณ์ I^2C โดยถ้าหากว่าบิต0 มีค่าเป็น "0" จะหมายถึง CPU ต้องการเขียนค่าไปยังอุปกรณ์ แต่ถ้าค่าในบิต 0 มีค่าเป็น "1" จะหมายถึง CPU ต้องการอ่านค่าจากอุปกรณ์ เช่นรหัส Control Byte ของ RTC เบอร์ PCF8583 มีค่า "1010001X" ถ้าต้องการเขียนค่าไปยัง RTC จะต้องส่งรหัส Control Byte เป็น "10100010" แต่ถ้าต้องการอ่านค่าจาก RTC ก็จะต้องส่งรหัส Control Byte ด้วยค่า "10100011" แทน เป็นต้น

การใช้งานหน่วยความจำ E^2PROM (24XX)

หน่วยความจำ Serial EEPROM ที่ใช้ในบอร์ดจะใช้การเชื่อมต่อแบบ I^2C-Bus ในตระกูล 24XX ซึ่งหน่วยความจำแบบนี้มีคุณสมบัติที่น่าสนใจหลายประการคือ มีตัวถังขนาดเล็ก ใช้สัญญาณในการเชื่อมต่อน้อยเส้น และสามารถเก็บรักษาข้อมูลไว้ได้นานกว่า 200 ปี นอกจากนี้ยังสามารถลบและเขียนซ้ำได้ถึง 1 ล้านครั้ง (อ้างอิงจาก Microchip) จึงสามารถนำไปประยุกต์ใช้งาน ในด้านที่เกี่ยวข้องกับการเก็บรักษาข้อมูลสำหรับงานต่างๆได้ดี

โดยผู้ใช้เองสามารถเลือกติดตั้งหน่วยความจำเพื่อใช้งาน กับบอร์ดได้มากมายหลายเบอร์ ขึ้นอยู่กับจุดประสงค์และขนาดของหน่วยความจำที่ต้องการ โดยให้เลือกใช้ E^2PROMในตระกูล 24XX (I^2C Bus) ในกลุ่มที่สามารถกำหนดตำแหน่งรหัส Control Byte ของหน่วยความจำจากฮาร์ดแวร์ (ขาสัญญาณ A2,A1 และ A0) ได้ เช่น เบอร์ 24XX32,64,128 และ 24XX256 ของ Microchip เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
1	0	1	0	A2	A1	A0	R/$\overline{W}$

รูปแสดงรหัส Control Byte ของ 24XX32/64/128/256 ของ Microchip

สำหรับหน่วยความจำเบอร์ 24XX32,24XX64,24XX128 และ 24XX256 ของ Microchip นั้น จะเห็นได้ว่ารหัส Control Byte ในตำแหน่ง 4บิตบน (บิต7,6,5 และ 4) จะมีค่าเป็น "1010" ส่วน บิต3 บิต2 และ บิต1 นั้นจะขึ้นอยู่กับสภาวะทางลอจิกของขาสัญญาณ A2,A1 และ A0 ในวงจร ซึ่งจากคุณสมบัติดังกล่าวจะทำให้สามารถทำการต่อหน่วยความจำดังกล่าวได้มากถึง 8 ตัวภายในระบบบัสเดียวกัน โดยกำหนดสภาวะของขา สัญญาณ ลอจิก แอดเดรสที่แตกต่างกันออกไป โดยสามารถสรุปให้เห็นได้ดังตารางต่อไปนี้

เบอร์(ความจุ)	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
24XX32 (4Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX64 (8Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX128 (16Kx8)	1	0	1	0	A2	A1	A0	R/W
24XX256 (32Kx8)	1	0	1	0	A2	A1	A0	R/W

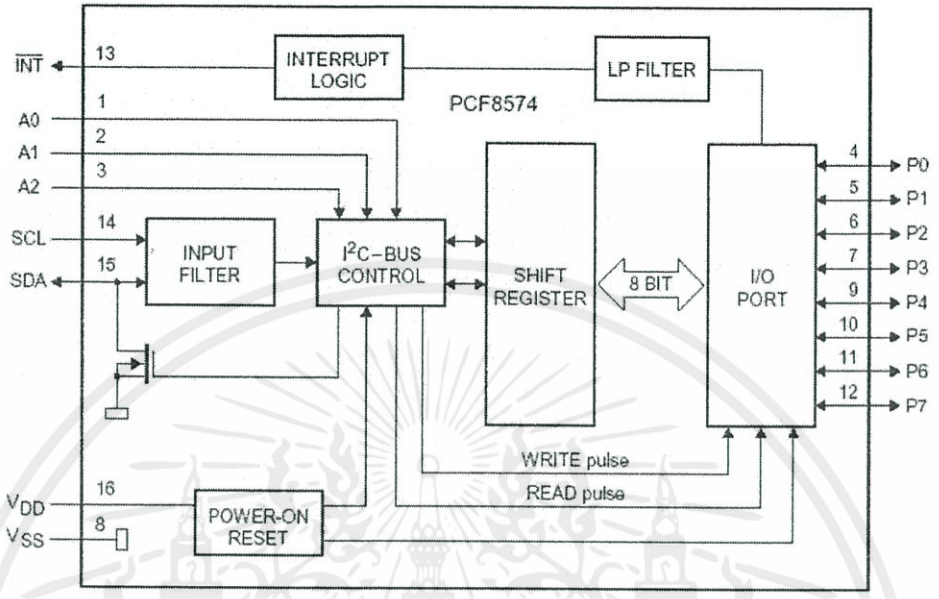
ตารางแสดงรหัส Control Byte ของหน่วยความจำแบบ I^2C Bus ของ Microchip

จากตารางจะเห็นได้ว่า หน่วยความจำ E^2PROM แบบ I^2C-BUS นั้น 24XX32/64/128/256 ของ Microchip นั้นจะมีรหัส Control Code ที่เหมือนกันทุกเบอร์ แต่จะมีความแตกต่างกันที่ A0-A2 ดังนั้นเมื่อทำการติดตั้งใช้งานหน่วยความจำเบอร์เหล่านี้กับบอร์ด CP-PIC V4.0 แล้วจะมีรหัส Control Byte เป็น "1010100X" คงที่ตลอด แต่ถ้ามีการต่อหน่วยความจำเหล่านี้เพิ่มเติมจากภายนอกบอร์ดแล้วรหัส Control Byte ก็จะขึ้นอยู่กับการกำหนดสภาวะทางลอจิกให้กับขาสัญญาณ A2,A1 และ A0 ของหน่วยความจำที่ต่อไว้

การใช้งาน I/O Port แบบ I^2C (PCF8574/A)

ตามปรกติแล้ว CPU เบอร์ PIC 16F877,18F442 และ 18F458 นั้นจะมีพอร์ต I/O สำหรับให้ผู้ใช้สามารถนำไปใช้งานได้มากถึง 5 พอร์ต อยู่แล้ว ซึ่งในส่วนของบอร์ด CP-PIC V3.0 นั้น พอร์ต I/O ทั้งหมดของ CPU จะปล่อยว่างไว้ให้ผู้ใช้เลือกใช้งานอย่างอิสระตามต้องการ แต่สำหรับบอร์ดรุ่น CP-PIC V4.0 นั้น พอร์ต I/O ต่างๆของ CPU จะถูกจัดสรรออกไปใช้งานในวงจรต่างๆ ดังได้กล่าวอธิบายมาแล้วในข้างต้น แต่ถ้าหากว่าผู้ใช้งานมีความจำเป็นต้องใช้งานพอร์ต I/O จำนวนมาก และจำนวนพอร์ต I/O ของ CPU ที่มีอยู่ไม่เพียงพอต่อการใช้งานแล้ว ผู้ใช้ก็สามารถทำการเพิ่มเติมพอร์ต I/O ได้อีก โดยในบอร์ด CP-PIC V4.0 นั้นจะออกแบบให้ผู้ใช้สามารถทำการเพิ่มเติม พอร์ต I/O แบบ I^2C ซึ่งมีขนาด I/O จำนวน 8 บิต I/O โดยใช้ไอซี สำหรับทำหน้าที่เป็นพอร์ต I/O ของ Phillips เบอร์ PCF8574 หรือ PCF8574A โดย PCF8574/A มีโครงสร้างดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง Block Diagram ของ PCF8574/A

นอกจากนี้แล้วผู้ใช้งานยังสามารถทำการขยาย จำนวนพอร์ต I/O ของ PCF8574/A นี้ได้อีกมากถึง 15 ตัว (120 บิต I/O) ทางขั้วต่อ "I²C EXPAND" ของบอร์ด เนื่องจาก PCF8574 หรือ PCF8574A นั้น สามารถต่อร่วมกันภายในระบบบัสเดียวกันได้มากถึงอย่างละ 8 ตัว กล่าวคือ ในระบบบัสของ I²C นั้น จะสามารถต่อใช้งาน PCF8574 ได้มากถึง 8 ตัว และยังสามารถต่อพอร์ต I/O เบอร์ PCF8574A ได้อีก 8 ตัว รวมเป็น 16 ตัวภายในบัสเดียวกัน โดยการกำหนดตำแหน่งแอดเดรสของอุปกรณ์แต่ละตัวให้มีความแตกต่างกัน ซึ่งตามปรกติแล้ว PCF8574 หรือ PCF8574A นั้น จะมีขาสัญญาณแอดเดรสจำนวน 3 เส้น คือ A0,A1 และ A2 โดยการกำหนดสภาวะทางโลจิกให้กับขาสัญญาณแอดเดรสทั้ง 3 ให้มีค่าไม่ซ้ำกัน โดย PCF8574 และ PCF8574A นั้น จะมีคุณสมบัติและวิธีการใช้งานที่เหมือนกันทุกประการ แตกต่างกันเพียงรหัส Control Byte เท่านั้น โดยโครงสร้างของรหัส Control Byte ของ PCF8574 และ PCF8574A สามารถแสดงให้เห็นได้ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	1	0	0	A2	A1	A0	R/$\overline{W}$

รูปแสดง รหัส Control Byte ของ PCF8574

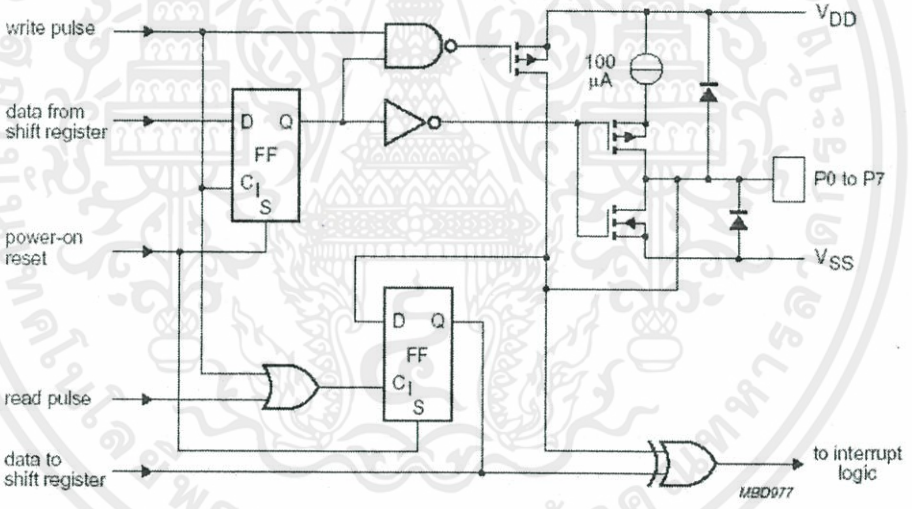
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	1	1	1	A2	A1	A0	R/$\overline{W}$

รูปแสดง รหัส Control Byte ของ PCF8574A

สำหรับรหัส Control Byte ของพอร์ต I/O เบอร์ PCF8574 หรือ PCF8574A ของบอร์ดนั้น จะถูกกำหนดไว้ตายตัว โดยขาสัญญาณแอดเดรส A0,A1 และ A2 จะถูกกำหนดสภาวะโลจิกเป็น "0" ไว้ทั้งหมด ซึ่งในกรณีที่ผู้ใช้ทำการติดตั้งพอร์ต I/O เบอร์ PCF8574 จะมีรหัส Control Byte เป็น "0100000X" แต่สำหรับกรณีที่ผู้ใช้ทำการติดตั้งพอร์ต I/O เบอร์ PCF8574A จะมีรหัส Control Byte เป็น "0111000X" แทน

โดยที่ พอร์ต I/O เบอร์ PCF8574/A นั้น ตามปรกติแล้วจะสามารถใช้งานเป็น Input หรือ Output ก็ได้ตามต้องการ แต่จำเป็นต้องเลือกกำหนดหน้าที่เพียงหน้าที่เดียวเท่านั้น ไม่สามารถใช้งานเป็นทั้ง Input และ Output ในเวลาเดียวกันได้ โดยลักษณะโครงสร้างภายในของขาสัญญาณ I/O ของ PCF8574 เป็นดังนี้



รูปแสดง ลักษณะโครงสร้างของขาสัญญาณ I/O แต่ละขาของ PCF8574/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานพอร์ตสื่อสารอนุกรม RS232/RS422/RS485

ภายในตัว CPU เบอร์ PIC 16F877,18F442 และ 18F458 ที่ใช้กับบอร์ด CP-PIC V3.0&V4.0 นั้น จะมีวงจรสื่อสารแบบอนุกรม (Universal Synchronous Asynchronous Receiver Transmitter : USART) บรรจุรวมไว้ด้วยแล้ว ซึ่งวงจรส่วนนี้ผู้ใช้งานสามารถทำการเขียนโปรแกรมควบคุมการสื่อสารข้อมูลของ CPU กับอุปกรณ์อื่นๆได้ตามต้องการ โดยในส่วนของโปรแกรมนั้น ผู้ใช้สามารถกำหนดรูปแบบของการสื่อสารข้อมูลได้เองจากโปรแกรมที่เขียนขึ้น ไม่ว่าจะเป็นความเร็วในการสื่อสาร (Baudrate) จำนวนบิตข้อมูลในการรับส่ง (Data Bit) การกำหนดบิตตรวจสอบความถูกต้องข้อมูล (Parity) และคุณสมบัติอื่นๆ ซึ่งในรายละเอียดส่วนนี้จะไม่ขอกล่าวถึงขอให้ผู้ใช้ศึกษาจากคู่มือสถาปัตยกรรมทางฮาร์ดแวร์หรือ Data Sheet ของ CPU เบอร์ ต่างๆ เหล่านี้เอง

ซึ่งปกติแล้วขาสัญญาณสำหรับ รับ-ส่ง ข้อมูลของ CPU นั้น สามารถนำไปเชื่อมต่อกับขาสัญญาณ รับ-ส่ง ของอุปกรณ์อื่นๆได้แล้ว โดยขาส่ง (TX) ของ CPU ต้องนำไปต่อกับขารับ (RX) ของอุปกรณ์ที่จะนำมาสื่อสารกัน ส่วนขารับข้อมูล (RX) ของ CPU ก็ต้องต่อกับขาส่งข้อมูล (TX) จากอุปกรณ์ที่จะนำมาสื่อสารกัน แต่เนื่องจากขาสัญญาณ RX และ TX ของ CPU นั้น จะสามารถเชื่อมต่อกับสัญญาณที่มีคุณสมบัติเป็นแบบ ระดับลอจิก TTL เท่านั้น ซึ่งถ้าใช้วิธีการเชื่อมต่อสัญญาณรับส่งของ CPU กับอุปกรณ์โดยตรงนั้น จะสามารถสื่อสารกันได้เพียงระยะทางใกล้ๆหรือภายในแผงวงจรเดียวกันเท่านั้น ไม่สามารถสื่อสารกันด้วยระยะทางไกลๆได้ ดังนั้นบอร์ด CP-PIC V3.0&V4.0 จึงได้ออกแบบวงจร Line Driver สำหรับทำหน้าที่เป็น Buffer เพื่อเปลี่ยนแปลงระดับสัญญาณทางไฟฟ้าของขาสัญญาณ รับ-ส่งข้อมูลของ CPU ที่เป็น แบบ TTL ให้สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลมากขึ้น (สามารถอ่านรายละเอียดเพิ่มเติมได้จากหัวข้อ "ความรู้ทั่วไปเกี่ยวกับการสื่อสารอนุกรม" ในภาคผนวกท้ายเล่มของคู่มือนี้) โดยบอร์ด CP-PIC V4.0 นั้น จะสามารถเลือกกำหนดรูปแบบของวงจร Line Driver สำหรับการสื่อสารอนุกรมได้ 3 แบบด้วยกัน คือ

การสื่อสารอนุกรมแบบ RS232

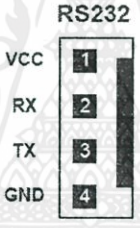
ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เพื่อเปลี่ยนระดับสัญญาณทางไฟฟ้าของขาสัญญาณสำหรับรับ-ส่ง ข้อมูลแบบ TTL ของ CPU (RX และ TX) ให้เป็นระดับสัญญาณทางไฟฟ้าแบบ RS232 (±12V) โดยการติดตั้งไอซีเบอร์ MAX232 เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณ TTL จากขาสัญญาณส่งข้อมูล (TX) ของ CPU ให้เป็นระดับสัญญาณ ±12V สำหรับส่งไปยังขารับสัญญาณ (RX) ของอุปกรณ์ภายนอก และในทางกลับกัน ก็จะทำหน้าที่เปลี่ยนระดับสัญญาณส่ง (TX) แบบ RS232 (±12V) จากอุปกรณ์ภายนอก ให้กลับมาเป็นระดับ TTL เพื่อส่งให้กับขารับข้อมูล (RX) ของ CPU ด้วย โดยเมื่อเปลี่ยนระดับสัญญาณในการรับส่งข้อมูลจาก TTL มาเป็นแบบ RS232 นี้แล้วจะทำให้สามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอกที่ใช้ระดับสัญญาณทางไฟฟ้าในการ รับ-ส่ง แบบเดียวกัน (RS232) ได้ไกลขึ้น ประมาณ 50ฟุต หรือ ประมาณ 15 เมตร โดยสามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เท่านั้น

สำหรับสายสัญญาณที่จะนำมาใช้สำหรับทำการสื่อสารแบบ RS232 นั้น จะใช้สัญญาณเพียง 2-3 เส้น เท่านั้น ทั้งนี้ขึ้นอยู่กับความต้องการในการสื่อสารว่าต้องการสื่อสารแบบทิศทางเดียวหรือสองทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การสื่อสาร RS232 แบบสองทิศทาง** ซึ่งจะมีทั้งการรับข้อมูลและส่งข้อมูลไปมา ระหว่างด้านรับและด้านส่ง โดยในกรณีนี้จะต้องใช้สายสัญญาณจำนวน 3 เส้น สัญญาณรับข้อมูล (RXD) สัญญาณส่งข้อมูล (TXD) และสัญญาณอ้างอิง (GND) โดยในการเชื่อมต่อสายนั้นจะต้องทำการสลับสัญญาณกับอุปกรณ์ปลายทางด้วย คือ สัญญาณส่ง (TXD) จากบอร์ด CP-PIC V3.0&V4.0 จะต้องต่อเข้ากับสัญญาณรับ (RXD) ของอุปกรณ์ และสัญญาณส่ง (TXD) จากอุปกรณ์ก็ต้องต่อกับสัญญาณรับ (RXD) ของบอร์ด ส่วนสัญญาณอ้างอิง (GND) จะต้องต่อตรงถึงกัน จึงจะสามารถทำการ รับ-ส่ง ข้อมูลกันได้
- **การสื่อสาร RS232 แบบทิศทางเดียว** ซึ่งอาจเป็นการรอรับข้อมูลจากด้านส่งเพียงอย่างเดียว หรืออาจเป็นการส่งข้อมูลออกไปยังปลายทางเพียงอย่างเดียว โดยไม่มีการโต้ตอบข้อมูลซึ่งกันและกัน ซึ่งวิธีนี้จะใช้สายสัญญาณเพียง 2 เส้น เท่านั้น โดยถ้าเป็นทางด้านส่ง ก็จะต่อเพียงสัญญาณส่ง (TXD) และสัญญาณอ้างอิง (GND) แต่ถ้าเป็นทางด้านรับ ก็จะต่อเพียงสัญญาณรับ (RXD) และ สัญญาณอ้างอิง (GND) เท่านั้น

โดยขั้วต่อของสัญญาณ RS232 ของบอร์ด CP-PIC V3.0&V4.0 นั้น จะเป็นจุดเชื่อมต่อของสัญญาณ รับ-ส่งข้อมูล ที่เปลี่ยนระดับสัญญาณเป็นแบบ RS232 แล้ว ซึ่งจะมีลักษณะเป็นแบบขั้ว CPA ขนาด 4 PIN สำหรับใช้เป็นจุดเชื่อมต่อสัญญาณ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอก โดยมีลักษณะการจัดเรียงสัญญาณดังนี้



แสดงขั้วต่อสัญญาณ RS232 ของบอร์ด CP-PIC V3.0 & V4.0

ซึ่งจะเห็นได้ว่าขั้วต่อสัญญาณ RS232 ของบอร์ดนั้น จะมีทั้งหมด 4 เส้น แต่ในการ รับ-ส่ง ข้อมูลแบบปรกตินั้น จะใช้สัญญาณเพียงแค่ 3 เส้น คือ RXD,TXD และ GND เท่านั้น ส่วน +VCC ซึ่งเป็นไฟเลี้ยงวงจร +5V นั้น จะไม่จำเป็นต้องนำมาใช้ในการสื่อสารกันแต่อย่างใด โดย +VCC หรือ +5V นี้ จะออกแบบเผื่อไว้ในกรณีที่อุปกรณ์ปลายทางเป็นวงจรขนาดเล็กและไม่สะดวกที่จะหาแหล่งจ่ายไฟให้กับอุปกรณ์ปลายทางด้วย ก็อาจต่อไฟเลี้ยงวงจร +VCC นี้ออกไปให้กับอุปกรณ์ปลายทางด้วยก็ได้เช่นกัน

******หมายเหตุ***** สำหรับไอซี Line Drive แบบ RS232 นั้น จะจัดเป็นอุปกรณ์มาตรฐานของบอร์ดในตระกูล CP-PIC V3.0&V4.0 ซึ่งจะมีติดตั้งให้ไปกับบอร์ดอยู่แล้ว ผู้ใช้ไม่ต้องจัดหาเพิ่มเติม แต่พึงระลึกไว้เสมอว่า จะต้องทำการติดตั้งไอซี Line Driver สำหรับเลือกชนิดสัญญาณทางไฟฟ้าของการสื่อสารอนุกรม ได้เพียงอย่างใดอย่างหนึ่งเท่านั้น เช่น เมื่อเลือกติดตั้งไอซี Line Driver เป็นแบบ RS232 โดยการติดตั้ง MAX232 ในบอร์ดแล้ว จะต้องไม่ติดตั้งไอซี Line Driver แบบอื่น เช่น RS422 หรือ RS485 เข้าไปด้วย เพราะจะทำให้ไม่สามารถรับส่งข้อมูลกันได้อย่างถูกต้อง ผู้ใช้ต้องเลือกติดตั้งไอซี Line Driver อย่างใดอย่างหนึ่งเท่านั้น

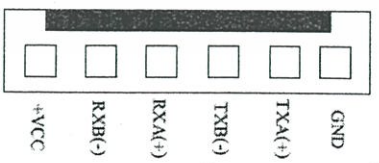
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารอนุกรมแบบ RS422

ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 จำนวน 1-2 ตัว เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณการไฟฟ้าในการ รับ-ส่ง แบบ TTL จาก CPU ให้เป็นระดับสัญญาณแบบ Balance Line เพื่อ รับ-ส่งสัญญาณกับอุปกรณ์ที่มีระดับสัญญาณทางไฟฟ้าเป็นแบบ Balance Line เหมือนกัน โดยถ้าต้องการใช้การสื่อสารแบบ 2 ทิศทาง ก็จะต้องติดตั้งไอซี Line Driver จำนวน 2 ตัว โดยแบ่งเป็นตัวแปลงสัญญาณทางด้านรับ 1 ตัว และตัวแปลงสัญญาณด้านส่งอีก 1 ตัว แต่ถ้าต้องการสื่อสารแบบทิศทางเดียวก็อาจทำการติดตั้งไอซี Line Driver เพียงตัวเดียว โดยถ้าต้องการให้เป็นฝ่ายรับข้อมูลเพียงอย่างเดียวก็ให้ติดตั้งไอซี Line Driver เฉพาะในตำแหน่งของ "RXD/422" เพียงตัวเดียว แต่ถ้าต้องการให้เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียวก็ให้ทำการติดตั้งไอซี Line Driver เฉพาะในตำแหน่ง "TXD/485" เพียงตัวเดียวเท่านั้น

ซึ่งการสื่อสารแบบ RS422 นี้ สามารถนำไปทดแทนการสื่อสารแบบ RS232 ได้ทันที โดยไม่ต้องดัดแปลงหรือแก้ไขโปรแกรมเลย ซึ่งการสื่อสารโดยใช้ระดับสัญญาณในการ รับ-ส่ง แบบ RS422 นี้จะมีข้อดี คือสามารถทำการสื่อสารกันได้ในระยะทางที่ไกลขึ้นกว่าแบบ RS232 มาก กล่าวคือ สามารถจะทำการ รับ-ส่ง ข้อมูลกันได้ในระยะทางประมาณ 4000 ฟุต หรือ 1200 เมตร หรือ 1.2 กิโลเมตรเลยทีเดียว เพียงแต่ต้องใช้สายสัญญาณที่ออกแบบมาสำหรับรองรับการใช้งานในด้านการสื่อสารแบบนี้โดยเฉพาะ ซึ่งได้แก่ สายสัญญาณแบบ UTP (Un-Shiled Twist Pair) หรือ STP (Shiled Twist Pair) โดยการสื่อสารด้วยระดับสัญญาณทางไฟฟ้าแบบ RS422 นี้ ถ้าเป็นการสื่อสารแบบ 2 ทิศทาง คือ ทั้งรับข้อมูลและส่งข้อมูล จะสามารถทำการรับส่งข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เหมือนกับ RS232 ทุกประการ แต่ในกรณีที่เป็นการสื่อสารแบบทิศทางเดียวนั้น สามารถจะทำการต่อขนานสัญญาณทางด้านรับ จำนวนหลายๆจุด เข้ากับสัญญาณส่งเพียงจุดเดียวได้ โดยถ้าเลือกใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานจำนวนอุปกรณ์สำหรับด้านรับข้อมูลได้ประมาณ 32จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 นั้น จะสามารถต่อขนานจำนวนอุปกรณ์ทางด้านรับข้อมูลได้มากถึง 256 จุด เลยทีเดียว แต่ถ้าเป็นอุปกรณ์ทางด้านส่งนั้น จะไม่สามารถนำมาต่อขนานสัญญาณส่งข้อมูลเข้าด้วยกันมากกว่า 1 จุด เหมือนทางด้านฝ่ายรับได้ ซึ่งวงจร Line Driver แบบ RS422 นี้จะมีอยู่เฉพาะในบอร์ดรุ่น CP-PIC V4.0 เท่านั้น

สำหรับลักษณะของขั้วต่อของสัญญาณ RS422 นั้น จะเป็นแบบ CPA ขนาด 6 PIN ดังรูป โดยในการสื่อสารกันนั้น จะใช้สายสัญญาณในการ รับ-ส่ง ข้อมูลกัน จำนวน 4 เส้น สัญญาณ คือ สัญญาณในการรับข้อมูล จำนวน 2 เส้น คือ RXA (RX+) และ RXB (RX-) และสัญญาณในการส่งข้อมูลอีก 2 เส้น คือ TXA (TX+) และ TXB (TX-) ซึ่งในการต่อสัญญาณนั้น จะต้องทำการต่อสัญญาณในลักษณะของการสลับกัน คือ สัญญาณส่งจะต้องต่อเข้ากับสัญญาณรับ นั่นก็คือ สัญญาณ RXA (RX+) จะต้องต่อกับ TXA (TX+) ส่วน RXB (RX-) ก็จะต้องต่อกับ TXB (TX-) ด้วยเช่นกัน โดยลักษณะของขั้วต่อสัญญาณ RS422 เป็นดังรูป



แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-PIC V4.0 เมื่อเลือกเป็น RS422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

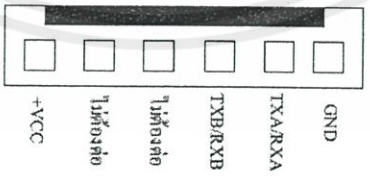
การสื่อสารอนุกรมแบบ RS485

ในการสื่อสารแบบ RS485 นี้จะมีคุณสมบัติของสัญญาณทางไฟฟ้าเหมือนกับ RS422 ทุกประการ เพียงแต่ว่าในการสื่อสารแบบ RS485 นี้จะใช้สายสัญญาณในการรับส่งข้อมูลกันเพียง 2 เส้น เท่านั้น แต่จะมีความพิเศษกว่าแบบ RS422 ตรงที่ ทิศทางของสัญญาณจะสามารถปรับเปลี่ยนได้จากโปรแกรม กล่าวคือ สัญญาณทั้ง 2 เส้น นี้สามารถจะสลับหน้าที่เป็นด้านส่ง และ เป็นด้านรับได้ ตามต้องการ โดยการควบคุมจาก CPU โดยจากบอร์ด CP-PIC V4.0 นั้น จะกำหนดให้สัญญาณ RC5 ทำหน้าที่สำหรับควบคุมทิศทางของข้อมูลว่าจะให้เป็นรับหรือส่ง โดยถ้าควบคุมให้ RC5 มีสภาวะเป็น "1" จะเป็นการกำหนดทิศทางให้เป็นฝ่ายส่งข้อมูล แต่ถ้าสภาวะของ RC5 เป็น "0" จะเป็นการกำหนดทิศทางให้เป็นฝ่ายรับข้อมูล ซึ่งจากคุณสมบัติข้อนี้จะทำให้การสื่อสารแบบ RS485 สามารถทำการต่อขนานอุปกรณ์ร่วมกันในสายส่งเดียวกันได้จำนวนหลายๆจุด โดยถ้าใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานอุปกรณ์กันได้ จำนวน 32 จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 แล้วจะสามารถต่อขนานอุปกรณ์ในสายคู่เดียวกันได้มากถึง 256 จุด เลยทีเดียว แต่มีข้อแม้ว่า เมื่อมีการต่ออุปกรณ์ขนานกันในสายสัญญาณคู่เดียวกันมากกว่า 2 จุดแล้วจะต้องเขียนโปรแกรมควบคุมให้มีการส่งข้อมูลออกมาในสายครั้งละ 1 จุดเท่านั้น เพราะถ้ามีการกำหนดทิศทางของข้อมูลให้เป็นส่งในเวลาเดียวกันมากกว่า 1 จุดแล้วจะทำให้เกิดการชนกันของข้อมูลและไม่สามารถสื่อสารกันได้อย่างถูกต้อง

โดยเมื่อต้องการใช้วิธีการสื่อสารแบบ RS485 นี้ จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 ในตำแหน่งของ "TXD/485" เพียงตัวเดียว พร้อมกับเลือกกำหนดเป็นแบบ RS485 ดังนี้

- ทำการเลือก Jumper สำหรับเลือก "422/485" ไว้ทางด้าน 485 (RS485)
- ทำการเลือก Jumper "F/H" ไว้ทางด้าน H (Half Duplex)
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ "TL" ไว้
- ทำการ Short Jumper สำหรับต่อตัวต้านทาน Fail Safe Resister คือ "TH" ไว้
- สายสัญญาณที่ใช้จะต่อจาก TXB(TX-) และ TXA(TX+) เพียง 2 เส้น ออกไปใช้งาน

ซึ่งในการสื่อสารข้อมูลแบบ RS485 นี้ จะต้องเขียนโปรแกรมขึ้นมารองรับการสื่อสารโดยเฉพาะ เนื่องจากทิศทางของข้อมูลสามารถจะกำหนดจากโปรแกรมได้โดยตรง ซึ่งการสื่อสารวิธีนี้จะมีข้อดีคือ ใช้สายสัญญาณในการรับส่งน้อยเส้น แต่จะเสียเวลาในการสื่อสารมากกว่าวิธีอื่นๆ เนื่องจากว่า การสื่อสารแบบนี้จะไม่สามารถทำการรับและส่งข้อมูลในเวลาเดียวกันได้ แต่จะต้องใช้วิธีการ ผลัดกันรับ ผลัดกันส่ง แทน ซึ่งในความเป็นจริงแล้วในปัจจุบันนี้ ราคาของสายสัญญาณแบบ 2 เส้น และ 4 เส้น แทบจะไม่มีความแตกต่างกันเลย ดังนั้นเพื่อลดความยุ่งยากในการเขียนโปรแกรมสำหรับควบคุมการรับส่งข้อมูลของ CPU ขอแนะนำให้เลือกใช้วิธีการสื่อสารแบบ RS422 จะง่ายและสะดวกรวดเร็วกว่ากันมาก



แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-PIC V4.0 เมื่อเลือกเป็น RS485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนด Jumper สำหรับการสื่อสารแบบ RS422/485

เนื่องจากวงจร Line Driver ของพอร์ตสื่อสารอนุกรมของบอร์ดนั้น ออกแบบให้ผู้ใช้สามารถเลือกกำหนดได้หลายแบบ ดังนั้น จึงต้องมีการใช้ Jumper สำหรับเป็นตัวเลือกรูปแบบการสื่อสารร่วมด้วย โดยจะมี Jumper ที่เกี่ยวข้องกับการใช้งานการสื่อสารแบบ RS422 และ RS485 ดังต่อไปนี้ คือ

- **Jumper 422/485** เป็น Jumper สำหรับเลือกกำหนดการทำงานของไอซี Line Driver ในตำแหน่ง TXD/485 ให้ทำงานเป็นแบบ RS422 หรือ RS485 โดยถ้าต้องการให้เป็นแบบ RS422 จะต้องกำหนด Jumper ไว้ทางด้าน "422" ซึ่งจะทำให้ไอซี Line Driver ตำแหน่ง "TXD/485" ทำหน้าที่เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียว แต่ถ้าต้องการใช้งานแบบ RS485 จะต้องกำหนด Jumper ไว้ทางด้าน "485" เพื่อกำหนดให้ไอซี Line Driver ในตำแหน่ง "TXD/485" ทำหน้าที่เป็นทั้งฝ่ายรับและฝ่ายส่ง ตามการควบคุมของสัญญาณ RC5
- **Jumper F/H** (Full/Half) เป็น Jumper ใช้สำหรับเลือกกำหนดรูปแบบการสื่อสารให้เป็นแบบ Full Duplex (F) หรือ Half Duplex (H) โดยถ้าต้องการใช้งานแบบ RS422 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้านด้าน F(Full Duplex) แต่ถ้าต้องการใช้งานเป็นแบบ RS485 จะต้องเลือกกำหนด Jumper นี้ไว้ทางด้าน H(Half Duplex) แทน
- **Jumper RL** เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXB (RX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXB (RX-) มีสภาวะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- **Jumper RH** เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ RXA (RX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ RXA (RX+) มีสภาวะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- **Jumper RZ** เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน RZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านรับ ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากๆจะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทานจากภายนอกไปชดเชยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง RZ นี้ไว้ก็จะเป็นการต่อตัวต้านทาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คร่อมระหว่าง RXA (RX+) และ RXB (RX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้

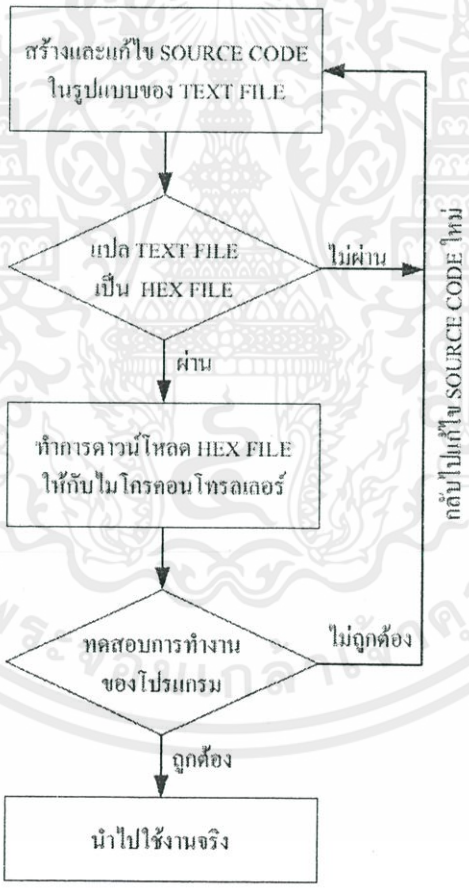
- Jumper TL เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXB (TX-) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXB (TX-) มีสภาวะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper TH เป็น Jumper ใช้สำหรับเลือกกำหนดการเชื่อมต่อ ตัวต้านทานสำหรับทำหน้าที่คงสถานะของสัญญาณ TXA (TX+) หรือ Fail Safe Resister เพื่อให้สัญญาณ TXA (TX+) มีสภาวะแน่นอนเมื่อไม่มีการส่งสัญญาณใดๆออกมาในสายเลย ซึ่งถ้าหากว่ามีการต่อสายสัญญาณระยะทางไกลๆหรือมีการต่อสายระยะทางไกล้ๆแต่ไม่ได้มีการส่งข้อมูลออกมาในสายตลอดเวลาแล้ว ควรที่จะทำการ Short Jumper นี้ไว้ด้วยเสมอ โดยเฉพาะอย่างยิ่งเมื่อใช้งานเป็นแบบ RS485 หรือใช้งานเป็นตัวอุปกรณ์ที่อยู่ในตำแหน่งต้นสายและปลายสายควรทำการ Short Jumper นี้ไว้เสมอ ส่วนอุปกรณ์ที่อยู่ในตำแหน่งอื่นๆที่มีระยะไม่ไกลจากจุดต้นสายและปลายสายมากนักก็อาจ Open Jumper นี้ออกก็ได้ แต่อย่างน้อยที่สุด ควรมีการ Short Jumper นี้ให้กับอุปกรณ์ที่ต่อร่วมอยู่ในสายสัญญาณจำนวน 1 จุดเสมอ
- Jumper TZ เป็น Jumper สำหรับเลือกกำหนดการต่อตัวต้านทาน TZ เพื่อชดเชย ค่าความต้านทานของสายสัญญาณ (Impedance) ทางด้านส่ง ซึ่งถ้าหากว่ามีการต่อสายสัญญาณในการรับส่งเป็นระยะทางไกลๆแล้วก็ควรทำการ Short Jumper นี้ไว้ด้วยเนื่องจากเมื่อสายมีความยาวมากๆจะเกิดค่าความต้านทานในสายขึ้น ดังนั้นจึงต้องทำการต่อค่าความต้านทานจากภายนอกไปชดเชยค่าความต้านทานของสายสัญญาณด้วย โดยเมื่อทำการ Short Jumper ตำแหน่ง TZ นี้ไว้ก็จะเป็นการต่อตัวต้านทานคร่อมระหว่าง TXA (TX+) และ TXB (TX-) ไว้ แต่ถ้าหากว่าต่อสายสัญญาณในระยะทางที่ไม่ไกลมากนัก ก็ให้ทำการ Open Jumper นี้ออกก็ได้

ข้อสังเกตุ จะเห็นได้ว่าวงจร Line Driver ทั้งแบบ RS422 และ RS485 นั้นจะมีความใกล้เคียง กันมาก แต่มีข้อแตกต่างอย่างหนึ่งที่เห็นได้ชัดเจนที่สุด คือ ถ้าเป็นแบบ RS422 จะไม่สามารถสั่งเปลี่ยน ทิศทางการรับส่งข้อมูลด้วยโปรแกรมได้ ซึ่งทิศทางการรับส่งจะกำหนดตายตัวจากวงจร แต่ถ้าเป็นแบบ RS485 นั้น จะสามารถสั่งควบคุมทิศทางการรับส่งจากโปรแกรมได้ว่าจะให้ทำหน้าที่เป็นฝ่ายรับ หรือฝ่ายส่ง อย่างใดอย่างหนึ่งได้ตามต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมของบอร์ด CP-PIC V3.0&V4.0(ICD2)

สำหรับการพัฒนาโปรแกรมนั้น ผู้ใช้งานสามารถเลือกที่จะใช้ภาษาใดก็ได้ในการพัฒนา เช่น ภาษาAssembly ,Basic หรือ ภาษา C ทั้งนี้ขึ้นอยู่กับความถนัดของผู้ใช้งาน แต่สุดท้ายแล้วจะต้องได้ไฟล์ที่จะโปรแกรมให้กับ CPU นั่นก็คือ HEX FILE ดังนั้นในการพัฒนาโปรแกรมจึงต้องมี Compiler สำหรับแปลภาษาที่เราเขียน (TEXT FILE) ให้เป็นภาษาที่ไมโครคอนโทรลเลอร์เข้าใจ (HEX FILE) สำหรับในที่นี้จะขอกล่าวถึงเฉพาะวิธีการ Download Hex File ให้กับบอร์ดเท่านั้น ส่วนวิธีการเขียนโปรแกรมและการสั่งแปลคำสั่งให้ได้เป็น Hex File นั้น ขอให้ผู้ใช้ศึกษาจากข้อกำหนดของโปรแกรมแปลภาษาที่จะนำมาใช้ในการเขียนโปรแกรมเอง ซึ่งบอร์ด CP-PIC V3.0&V4.0(ICD2) จะออกแบบพอร์ตสัญญาณสำหรับโปรแกรม (ICD2 Port) ให้สามารถทำการโปรแกรมข้อมูลลง CPU ได้โดยการเชื่อมต่อกับเครื่องโปรแกรมจากภายนอก (เช่น เครื่องโปรแกรม ET-PGMPIC USB) โดยในขั้นตอนการพัฒนาโปรแกรมนั้นสามารถสรุปเป็น โฟลว์ชาร์ต ดังนี้

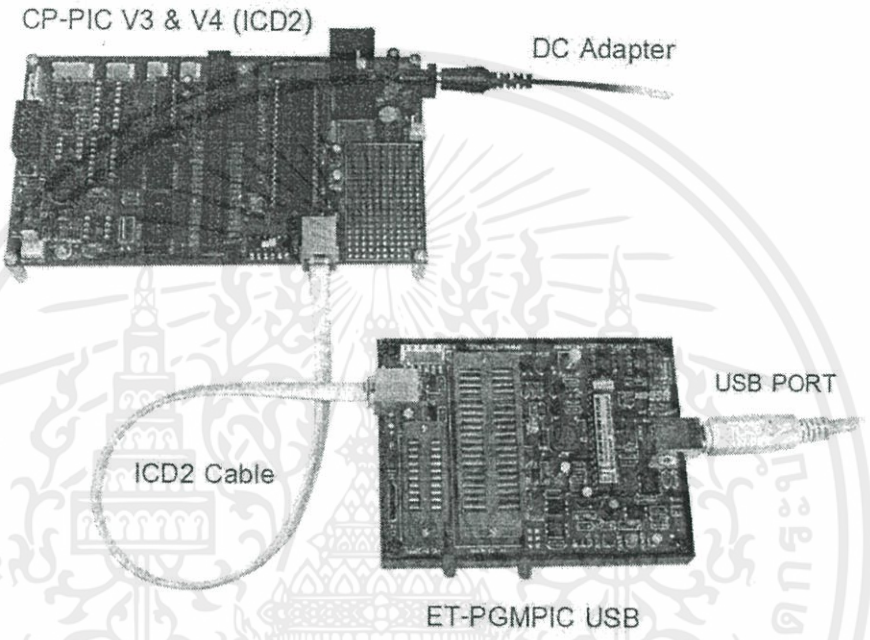


แผนผัง แสดงขั้นตอนในการพัฒนาโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการดาว์นโหลดโปรแกรมด้วยเครื่องโปรแกรม ET-PGMPIC USB

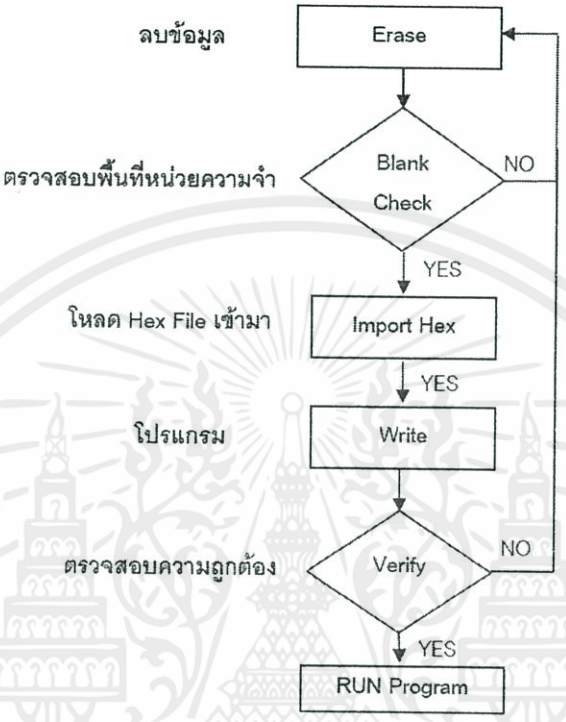
1. เชื่อมต่อ ET-PGMPIC USB เข้ากับพอร์ต USB ของคอมพิวเตอร์
2. เชื่อมต่อสายสัญญาณโปรแกรมจากพอร์ต ICD2 ของเครื่องโปรแกรม ET-PGMPIC USB มาเข้าที่พอร์ต ICD2 ของ บอร์ด CP-PIC V3 & V4(ICD2) และ ต่อแหล่งจ่ายไฟเข้ากับบอร์ดดังรูปต่อไปนี้



3. เลื่อนตำแหน่งสวิตช์ของบอร์ด CP-PIC V3 & V4 มาที่ตำแหน่ง PROG

4. เปิดโปรแกรม PicKit 2 ซึ่งในขั้นตอนนี้เราสามารถทำการ Erase , Blank Check ,Write , Read และ Verify ได้ตามต้องการ (รายละเอียดให้ศึกษาในคู่มือการใช้งานเครื่องโปรแกรม ET-PGMPIC USB) โดยทั่วไปขั้นตอนการโปรแกรมแบบคร่าวๆ จะเป็นดังโฟว์ชาร์ตต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



5. เมื่อต้องการกลับสู่โหมดการ RUN โปรแกรมให้เลื่อนสวิตช์ RUN/PROG มาที่ตำแหน่ง RUN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่อาจเกิดขึ้น และแนวทางการแก้ไข

CPU ไม่ทำงานหรือทำงานผิดพลาด อาจเกิดขึ้นจากหลายสาเหตุดังนี้

- สวิตช์ PROG/RUN อยู่ในตำแหน่ง PROG ทำให้ CPU อยู่ในโหมดโปรแกรมจึงไม่สามารถทำงานตามโปรแกรมได้ การแก้ไขให้เลือกสวิตช์มาที่โหมด RUN เมื่อต้องการให้ CPU ทำงานในโหมด Run Program
- การตั้งค่า Configuration โดยเฉพาะใน โหมดของ OSC มีผลมาก เพราะ PIC 18F458 สามารถทำงานได้ 2 ความถี่ คือ ความถี่จาก OSC ภายนอก(HS :ความถี่ 10 MHz) และ โหมด PLL (H4 : OSC ภายนอก x 4 ก็คือ 40MHz) ให้ตรวจสอบให้ถูกต้องว่าใช้งานในโหมดไหน
- Watchdog Timer หากเลือกการทำงานนี้ ใน Configuration แล้วไม่เขียนโปรแกรมไป Clear ค่า Watchdog จะทำให้เกิดการรีเซต ตลอดเวลาตามค่า Watchdog Timer ดังนั้นจึงควรตรวจสอบก่อนว่ามีการใช้งาน Watchdog Timer หรือไม่ ถ้าไม่ก็ไม่ควรเลือกการทำงานในส่วนนี้
- การ Set Jumper ต่างๆไม่ถูกต้อง ให้ตรวจสอบให้ถูกต้อง ซึ่งสามารถดูรายละเอียดได้จากหัวข้อรายละเอียดการใช้งาน Jumper
- ในกรณีที่เขียนด้วยภาษา BASIC (ใช้ PIC Basic Pro) จะต้องมีการกำหนดค่าความถี่ที่ใช้งานโดยใช้คำสั่ง DEFINE OSC __ ซึ่ง หากไม่มีการ กำหนดความถี่ให้กับ โปรแกรม มันจะตีว่าเป็นความถี่ 4 MHz ซึ่งจะทำให้โปรแกรมทำงานผิดพลาดได้ดังนั้นเราจึงควรกำหนดให้ถูกต้อง

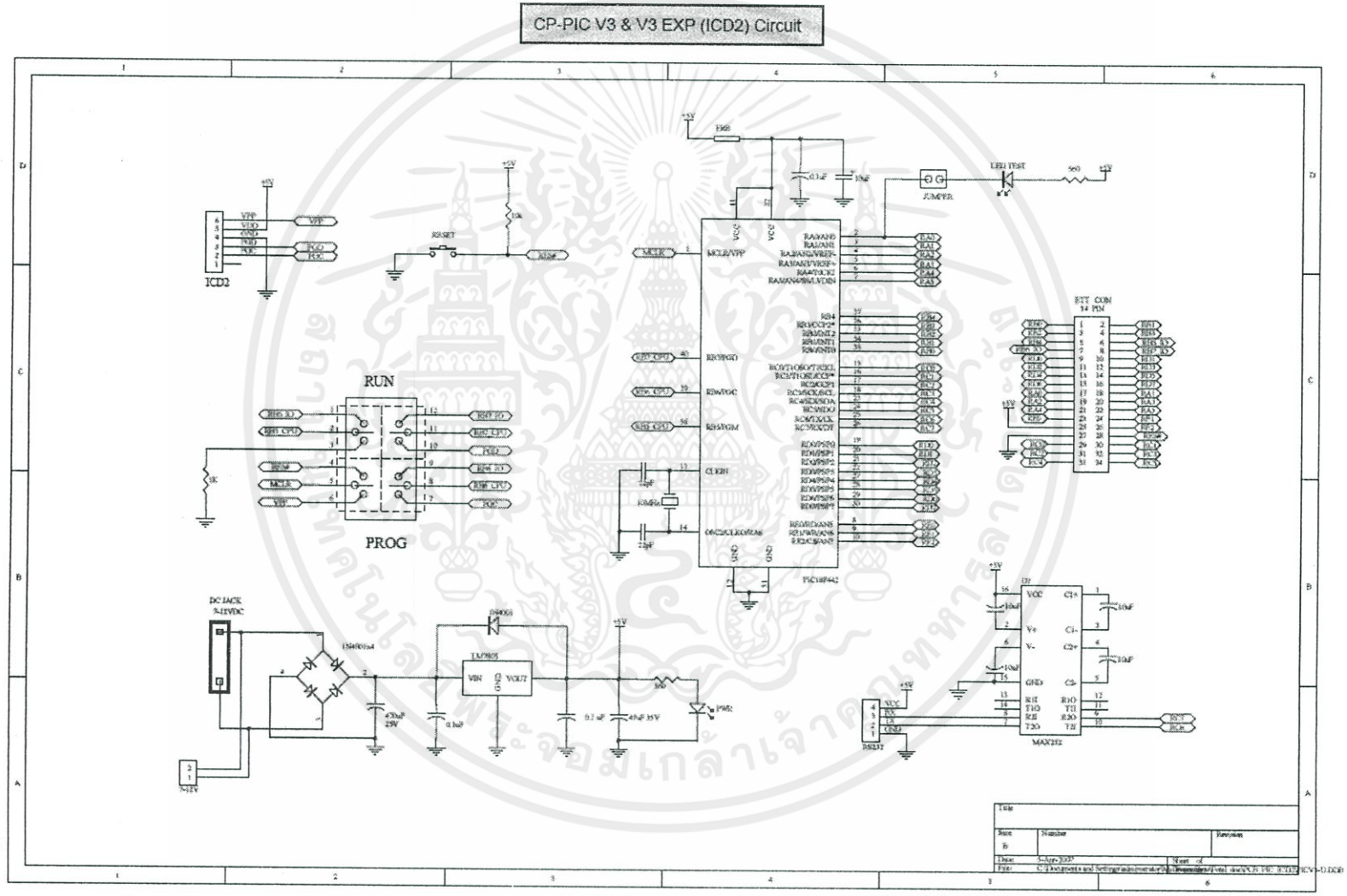
ตัวอย่าง

```
DEFINE   OSC   10   ' การกำหนดความถี่ 10 MHz
DEFINE   OSC   40   ' การกำหนดความถี่ 40 MHz
```

- การเปลี่ยนโหมดความถี่ เนื่องจาก PIC 18F458 ในบอร์ด CP-PIC V3.0&V4.0 สามารถทำงานได้ทั้ง 10MHz(HS) และ 40 MHz(H4) ฉะนั้นในการเปลี่ยนโหมดการทำงานจาก 10 MHz ไปเป็น 40MHz หรือจาก 40MHz เป็น 10MHz เมื่อทำการโปรแกรมเปลี่ยนโหมดเสร็จ CPU จะยังคงทำงานในโหมดความถี่เดิมอยู่ จนกว่าจะมีการปลดไฟเลี้ยง แล้วจ่ายไฟเข้ามาใหม่ CPU จึงจะเปลี่ยนมาทำงานในโหมดความถี่ใหม่ที่โปรแกรมให้ครั้งสุดท้าย

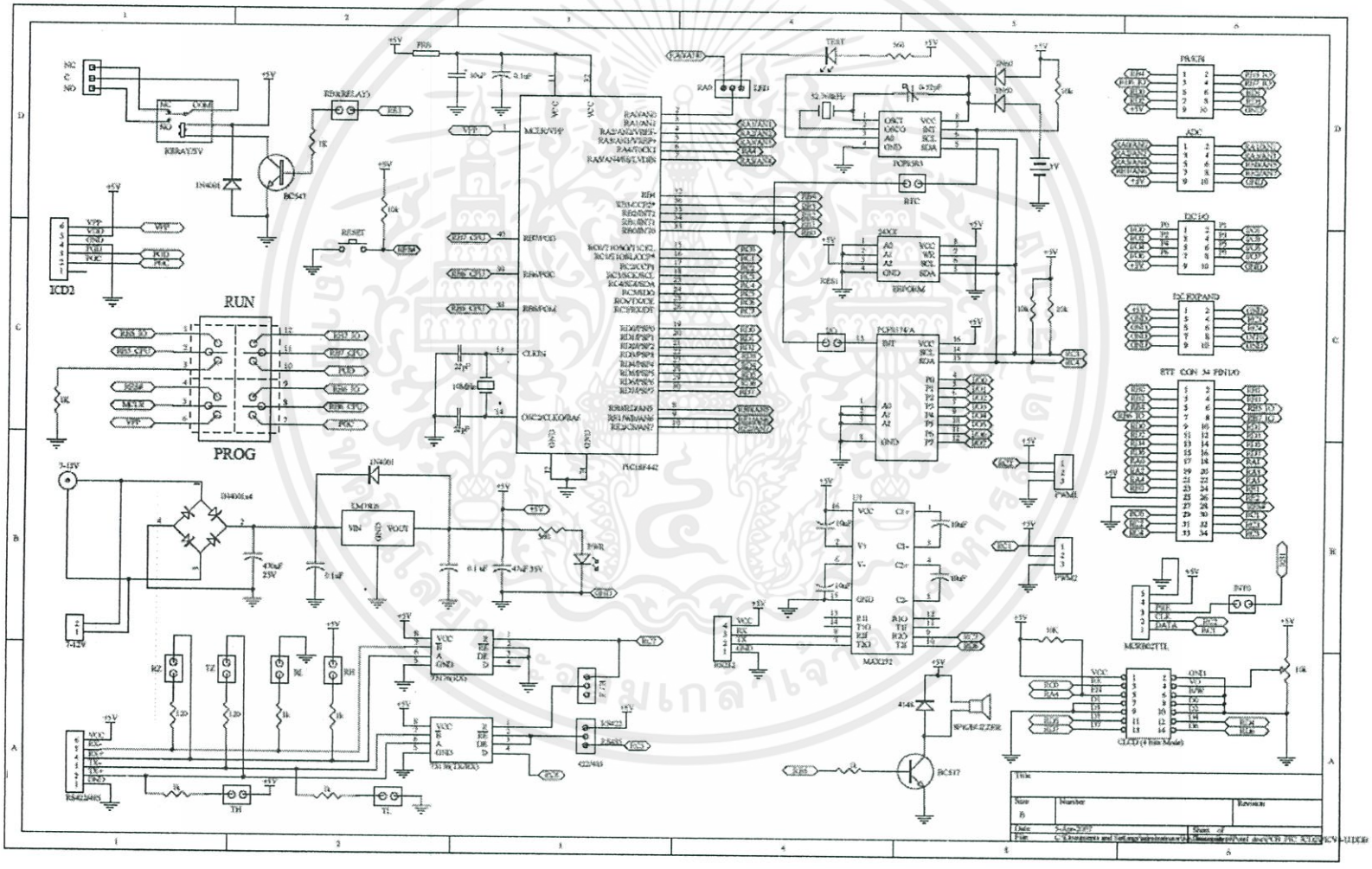
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP-PIC V3 & V3 EXP (ICD2) Circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

CP-PIC V4 (ICD2) Circuit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ข.3 เอกสารคู่มือการใช้งาน ET-USB/RS232 MINI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

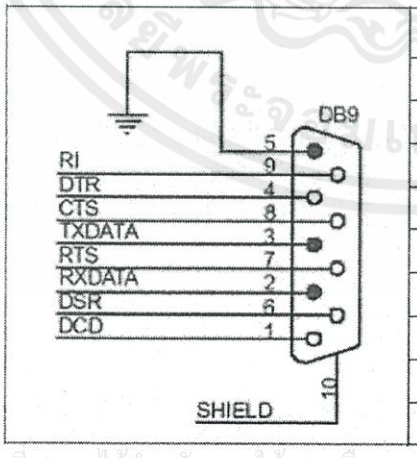
ET-USB/RS232 MINI

คุณสมบัติของ ET-USB/RS232 MINI

1. รองรับมาตรฐาน USB 1.1 และ USB 2.0
2. อัตราการรับส่งข้อมูล 300 bps ถึง 128 kbps
3. ใช้ไฟเลี้ยงจาก USB Port โดยตรงไม่ต้องต่อเพิ่มภายนอก
4. USB Connector แบบ TYPE A ,RS232 Connector แบบ DP9 ตัวผู้
5. แสดงสถานการณ์ทำงานด้วย LED 3 สี คือ การรับ (RX) สีเขียว,การส่ง (TX) สีเหลือง และ Power (PWR) สีแดง
6. Driver รองรับ Windows 98SE/ME/2000/XP
7. ขนาด กว้าง 3 cm x ยาว 6.8 cm

การต่อใช้งาน ET-USB/RS232 MINI

1. เสียบ ET-USB/RS232 MINI เข้ากับ Port USB ของคอมพิวเตอร์ ขณะนี้จะสังเกตเห็นว่าไฟแสดงสถานะ PWR จะยังไม่ติดสว่าง
2. ติดตั้ง Driver สำหรับ ET-USB/RS232 MINI โดยจากคู่มือการติดตั้ง Driver เมื่อติดตั้งเรียบร้อยแล้ว ไฟแสดงสถานะ PWR จะติดสว่างแสดงว่าพร้อมที่จะใช้งานแล้ว
3. ต่อสายพอร์ตอนุกรม เพื่อใช้งานโดยตำแหน่งขาใช้งาน จะเป็นดังรูป



Pin	Name	Dir	Description
1	CD	←	Carrier Detect
2	RXD	←	Receive Data
3	TXD	→	Transmit Data
4	DTR	→	Data Terminal Ready
5	GND	—	Signal Ground
6	DSR	←	Data Set Ready
7	RTS	→	Request to Send
8	CTS	←	Clear to Send
9	RI	←	Ring Indicator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

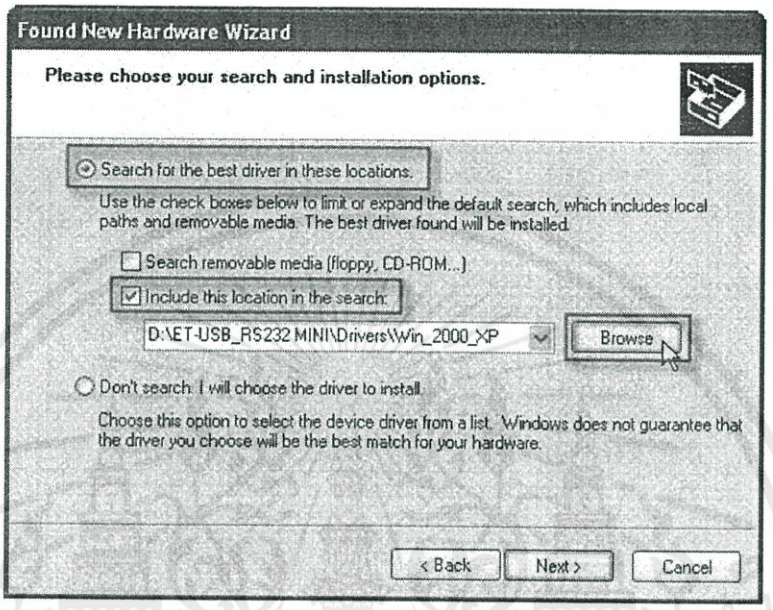
การติดตั้ง Driver ของ ET- USB/RS232 MINI

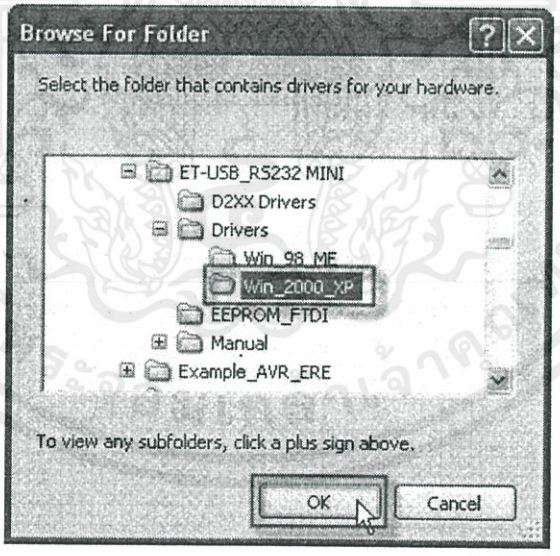
1. ทำการเชื่อมต่อ ET-USB/RS232 MINI เข้ากับคอมพิวเตอร์ทางพอร์ต USB จากนั้นวินโดวส์จะตรวจพบฮาร์ดแวร์ใหม่ดังรูป ตอนนี้จะสังเกตว่าไฟแสดงสถานะ PWR จะยังไม่ติดสว่าง

2. จากนั้นจะปรากฏหน้าต่าง Found New Hardware Wizard ให้เลือกที่ Install from a list or specific location (Advanced) และคลิก Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

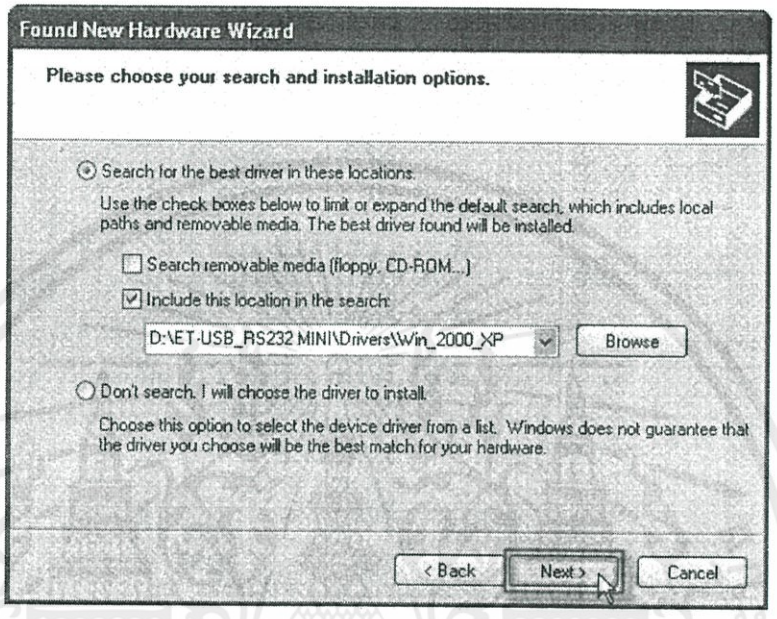
3. ทำการเลือกดังรูป และคลิกปุ่ม Browse เพื่อระบุตำแหน่งที่เก็บ Driver จากนั้นคลิก OK

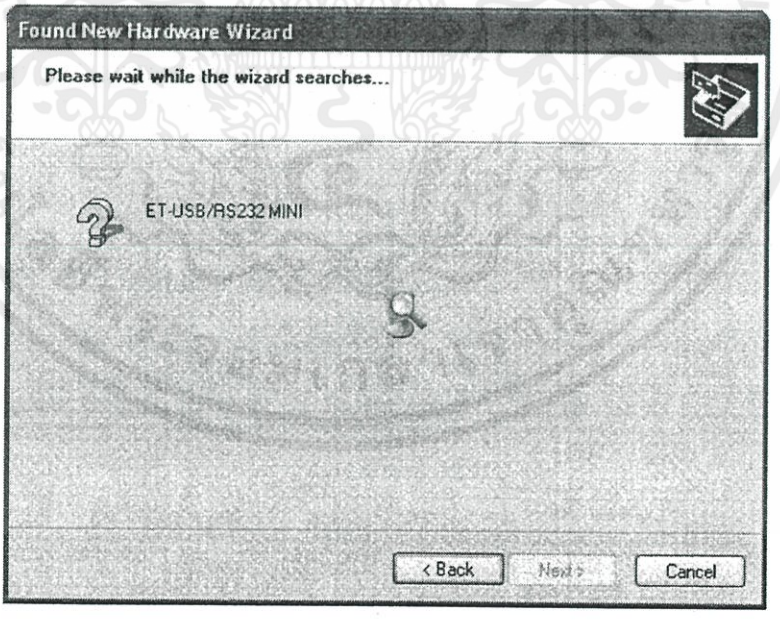




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

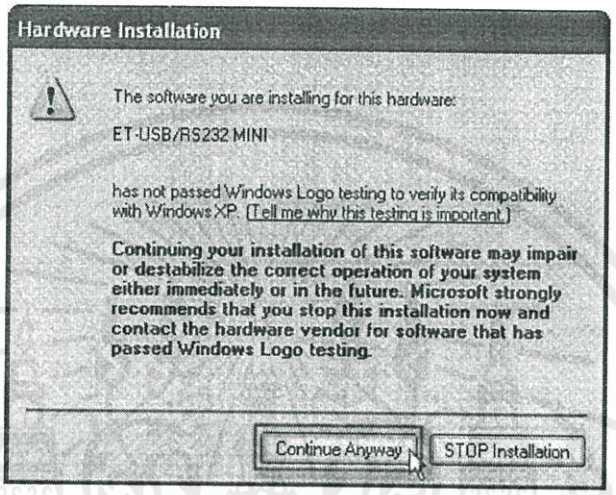
4. เมื่อทำการเลือกเรียบร้อยแล้วให้คลิกปุ่ม Next จากนั้นวินโดวส์จะทำการหาฮาร์ดแวร์เพื่อที่จะทำการติดตั้ง Driver

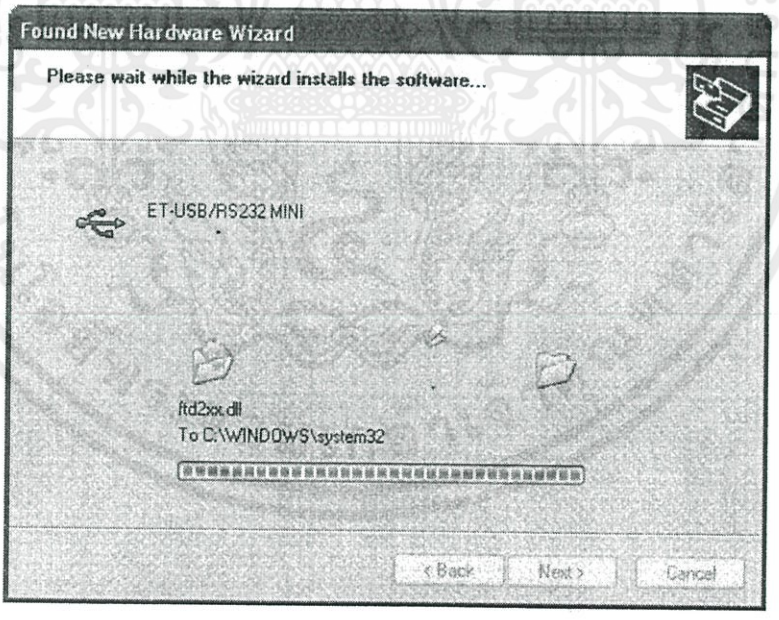




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

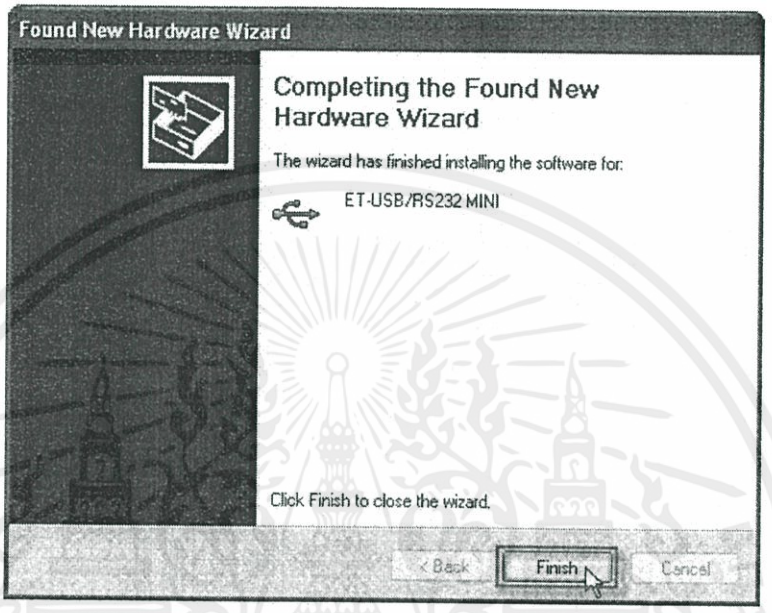
5. จากนั้นจะปรากฏหน้าต่าง Hardware Installation บอกว่าฮาร์ดแวร์ไม่ได้ผ่านการทดสอบของวินโดวส์ ให้ทำการคลิกที่ Continue Anyway ซึ่งวินโดวส์จะทำการติดตั้ง Driver ของ ET- USB/RS232 MINI



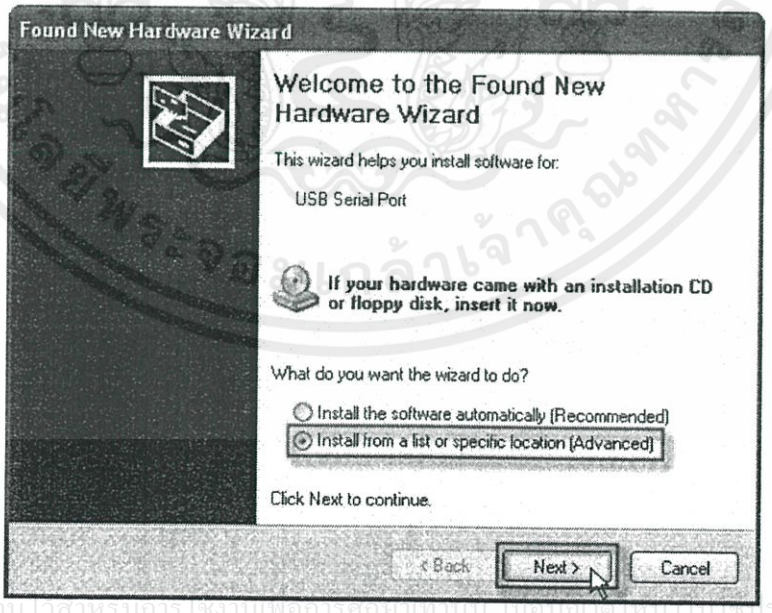


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อปรากฏหน้าต่างว่าได้ทำการติดตั้งเรียบร้อยแล้วให้คลิก Finish จะสังเกตเห็นว่าไฟแสดงสถานะ PWR ติดสว่าง

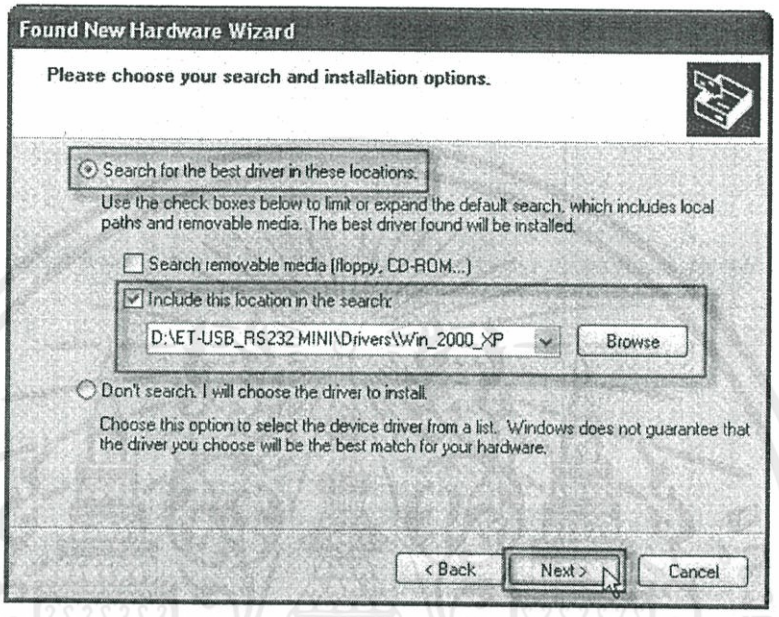


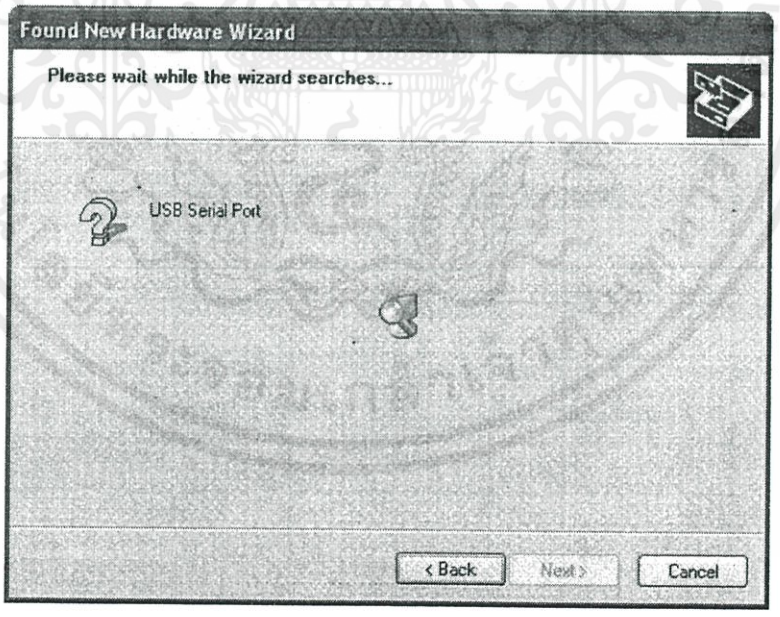
7. จากนั้นไม่นานตัว Driver จะมีการสร้างพอร์ตอนุกรมเสมือนขึ้นมาและมีหน้าต่างให้ติดตั้ง Driver ของ USB Serial Port ดังรูป ให้ทำการเลือกเหมือนข้อที่ผ่านมาจากนั้นคลิก Next



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

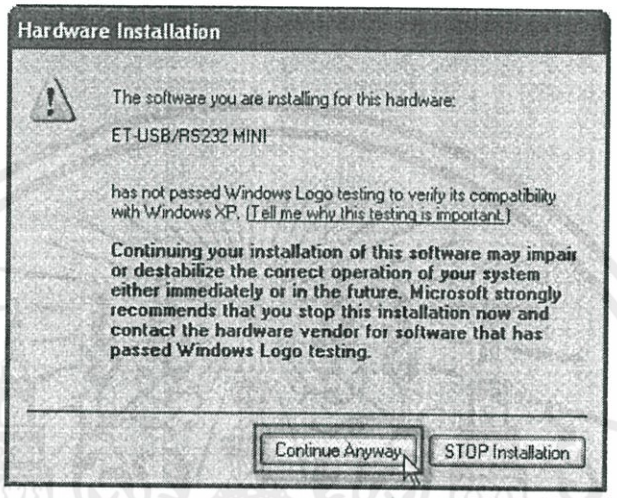
8. เลือกตำแหน่งที่ตั้งของ Driver ซึ่งปกติจะจำค่าเดิมไว้อยู่แล้วให้คลิก Next ได้เลย จากนั้นวินโดวส์จะทำการหาฮาร์ดแวร์เพื่อที่จะทำการติดตั้ง Driver

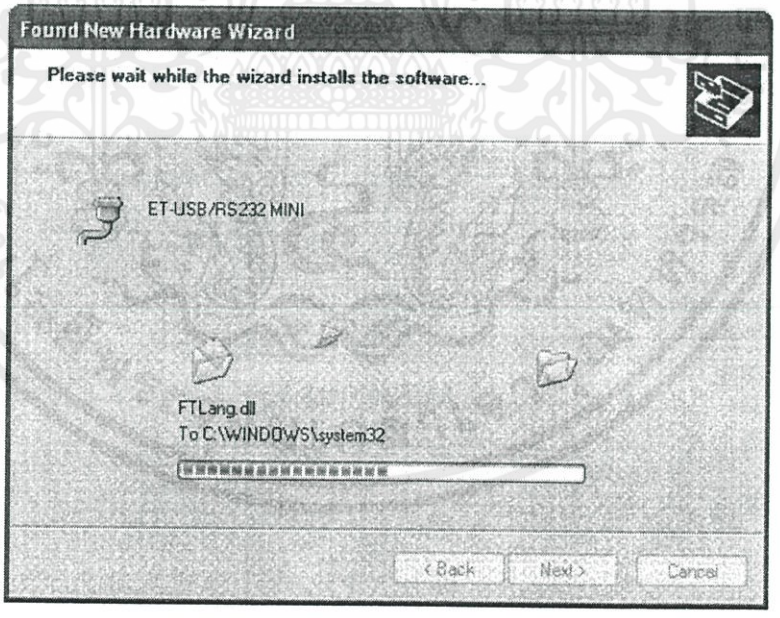




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

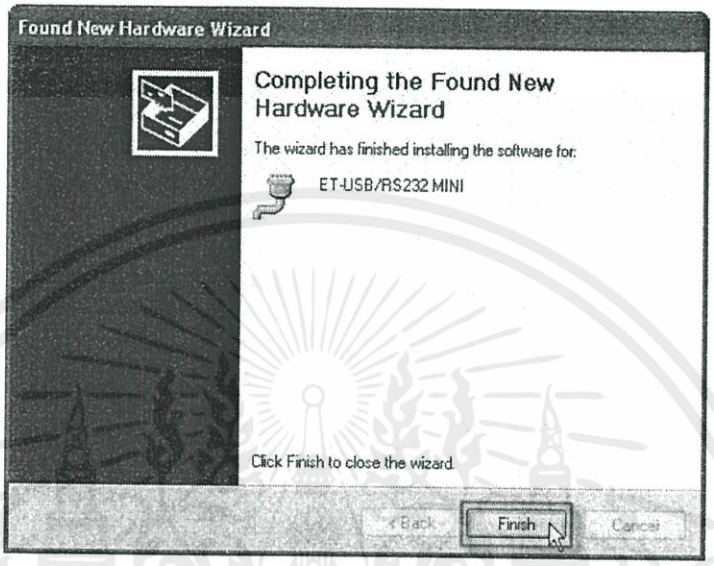
9. จากนั้นจะปรากฏหน้าต่าง Hardware Installation บอกว่าฮาร์ดแวร์ไม่ได้ผ่านการทดสอบของวินโดวส์ให้ทำการคลิกที่ Continue Anyway ซึ่งวินโดวส์จะทำการติดตั้ง Driver ของ ET-USB/RS232 MINI



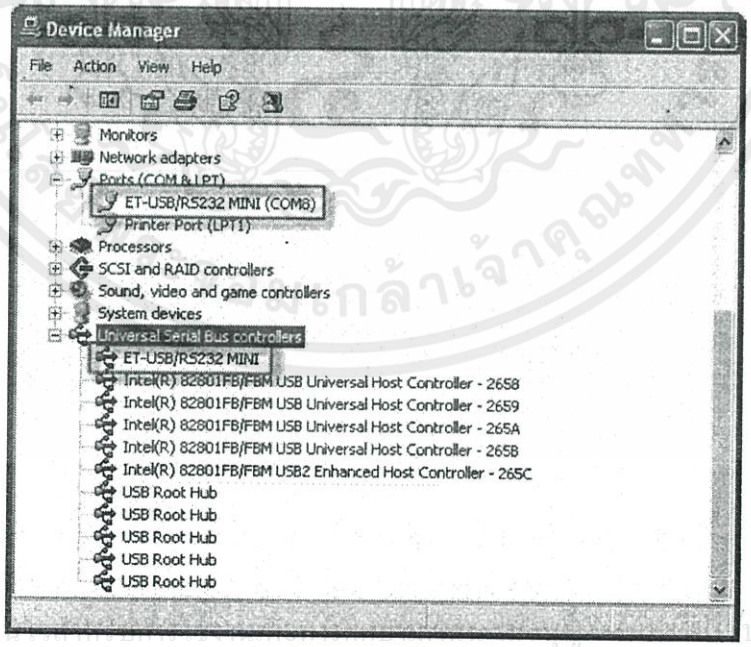


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อปรากฏหน้าต่างว่าได้ทำการติดตั้งเรียบร้อยแล้วให้คลิก Finish

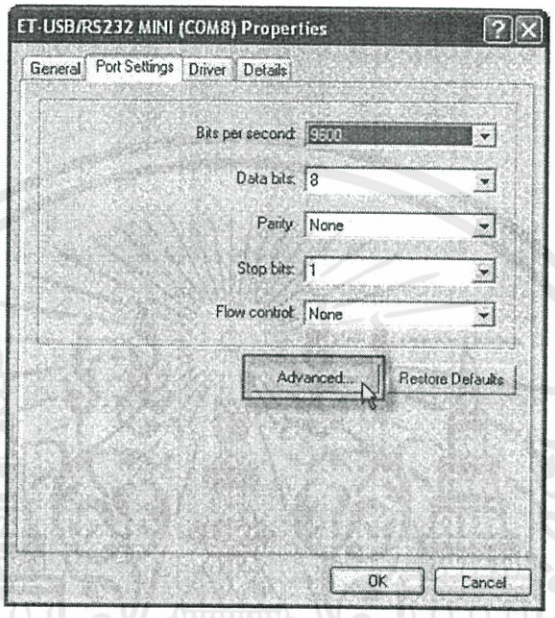


11. เราสามารถที่จะตรวจสอบว่าได้ติดตั้ง Driver ของ ET-USB/RS232 MINI เสร็จสมบูรณ์หรือไม่โดยดูที่ Control Panel → System เลือกแท็ป Hardware และเลือกที่ Device Manager ซึ่งจะเห็นรายการฮาร์ดแวร์ ET-USB/RS232 MINI ดังรูป

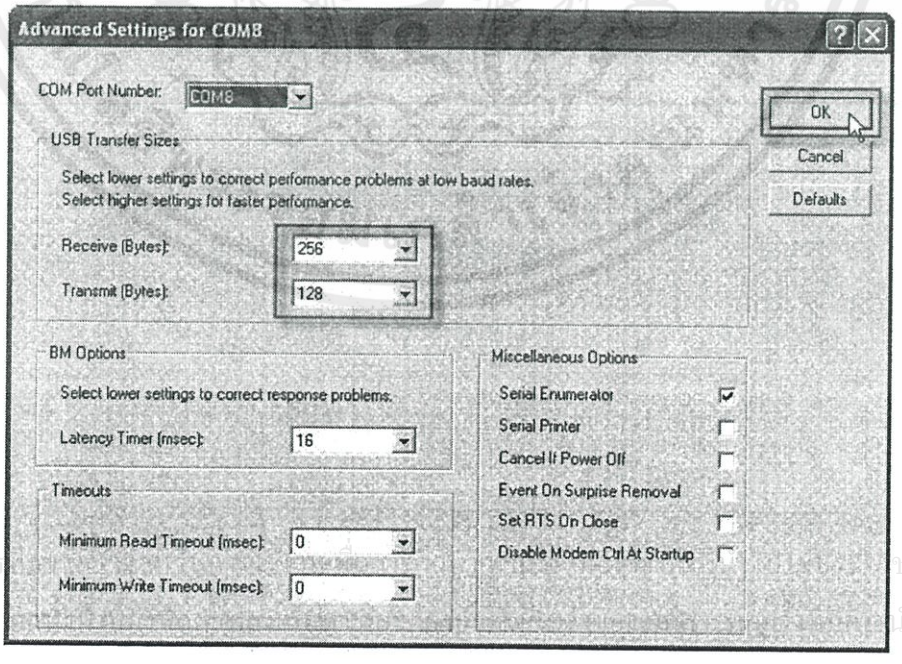


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. ดับเบิ้ลคลิกที่ ET- USB/RS232 MINI ตรงส่วนของ Ports(COM&LPT) จากนั้นจะปรากฏหน้าต่าง Properties ดังรูป ให้เลือกมาที่ Port Setting และทำการคลิกที่ปุ่ม Advanced...



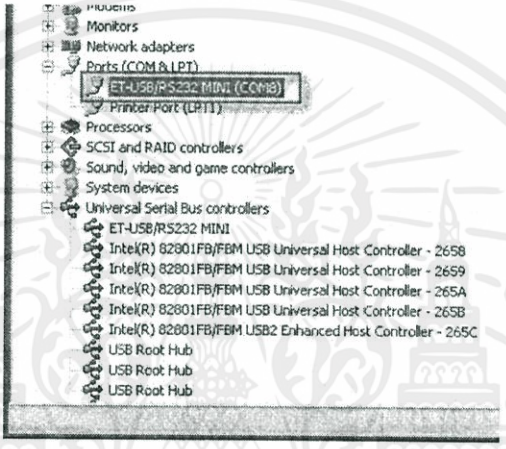
13. กำหนดค่า Receive(Bytes) และ Transmit(Bytes) ดังรูป และคลิกที่ OK เพื่อยืนยันการเปลี่ยนแปลง จากนั้นให้ทำการรีสตาร์ทเครื่องคอมพิวเตอร์หรือสแกนหาฮาร์ดแวร์ใหม่



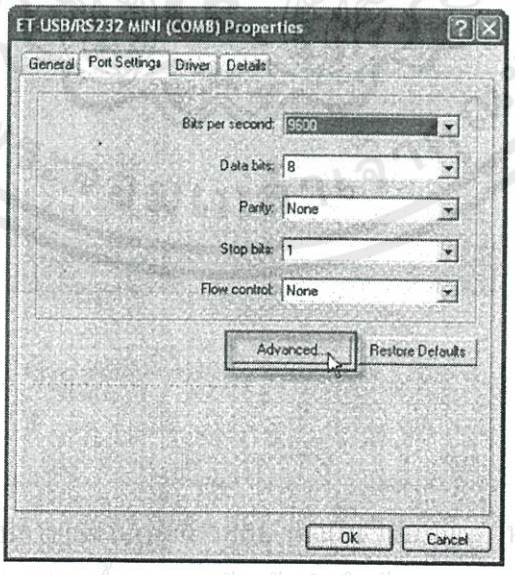
เอกสารนี้เป็นเอกสาร... ...ชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น... ...นำไปใช้

ขั้นตอนเปลี่ยนแปลงตำแหน่งหมายเลขของ COM Port

1. ไปที่ Control Panel → System เลือกแท็ป Hardware และเลือกที่ Device Manager ซึ่งจะเห็นรายการฮาร์ดแวร์ ET-USB/RS232 MINI ซึ่งจากรูปจะเห็นว่าอยู่ที่ตำแหน่ง COM8 (หมายเลข COM Port อาจมีการเปลี่ยนแปลงขึ้นอยู่กับคอมพิวเตอร์แต่ละเครื่อง)

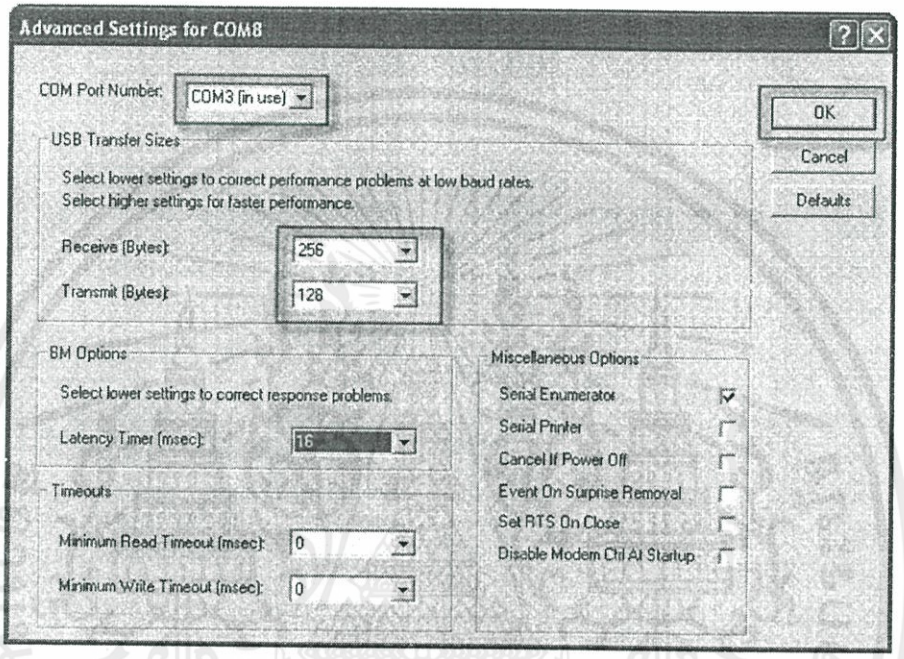


2. ซึ่งจากรูปสมมุติว่า ET- USB/RS232 MINI อยู่ที่ตำแหน่ง COM8 ซึ่งถ้าต้องการเปลี่ยนให้เป็น COM3 ก็สามารถทำได้โดยการดับเบิลคลิกที่ ET- USB/RS232 MINI จากนั้นจะปรากฏหน้าต่าง Properties ดังรูป ให้เลือกมาที่ Port Setting และทำการคลิกที่ปุ่ม Advanced...



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ [illegible] ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

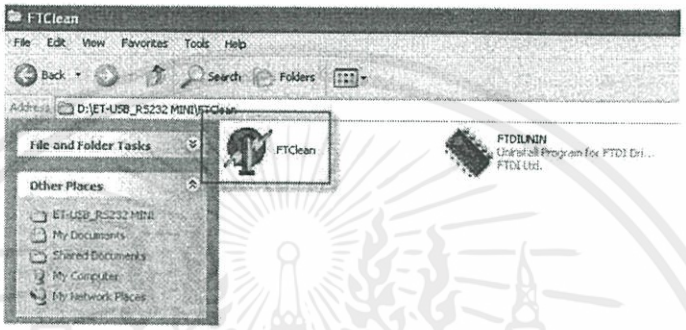
3. ทำการเปลี่ยน COM Port Number เป็น COM3 กำหนดค่า Receive(Bytes) และ Transmit(Bytes) ดังรูป และคลิกที่ OK เพื่อยืนยันการเปลี่ยนแปลง จากนั้นให้ทำการรีสตาร์ทเครื่องคอมพิวเตอร์หรือสแกนหาฮาร์ดแวร์ใหม่

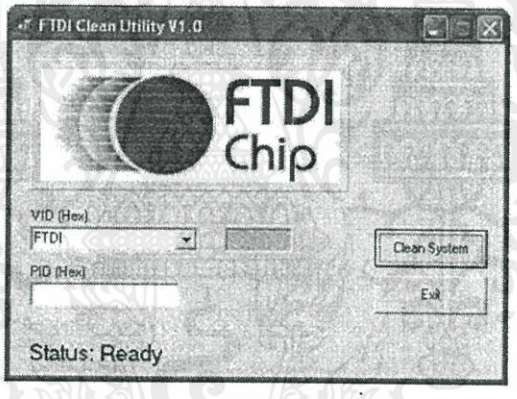


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

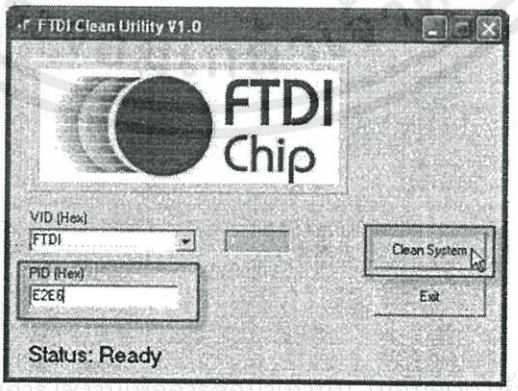
วิธีการยกเลิกการติดตั้ง Driver

1. แนะนำให้ใช้โปรแกรม FTDI Clean Utility ของทาง FTDI โดยการเปิดโปรแกรม FTDI Clean Utility ดังรูป



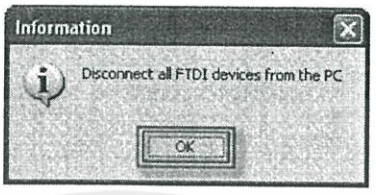


2. ใส่ค่า PID ซึ่ง ET-USB/RS232 MINI จะใช้ค่า E2E6 จากนั้นคลิกที่ Clean System

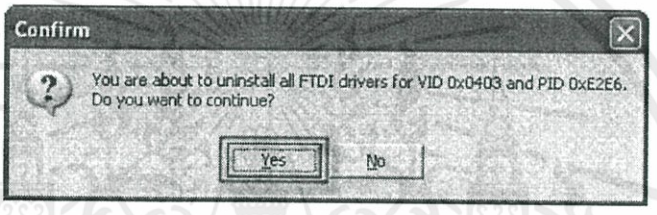


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

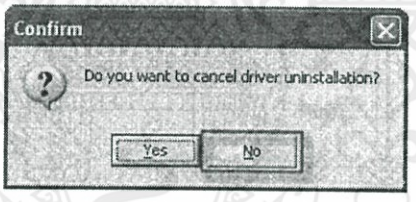
3. จากนั้นจะมีหน้าต่างแนะนำให้ถอดอุปกรณ์ (ET-USB/RS232 MINI) ออกจากคอมพิวเตอร์



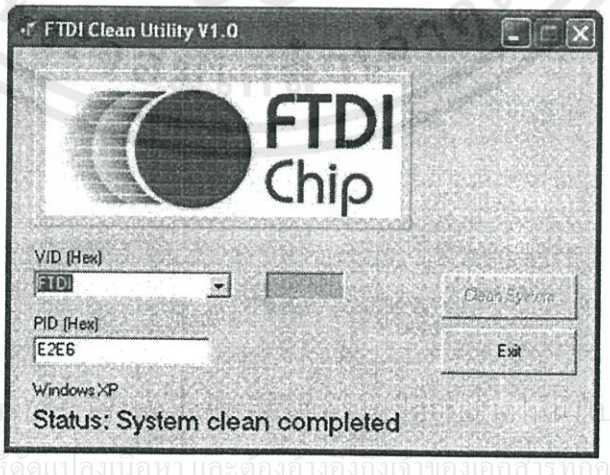
4. จากนั้นจะมีหน้าต่างยืนยันยกเลิกการติดตั้งให้เลือก Yes เพื่อยืนยัน



5. จากนั้นโปรแกรมจะถามว่าต้องการจะยกเลิกการถอน Driver หรือเปล่าให้ตอบ No



6. เมื่อทุกอย่างเสร็จสมบูรณ์จะปรากฏข้อความ System clean completed ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] พีรพร หมุนสนิท. ใช้งาน SQL แบบมืออาชีพ. กรุงเทพฯ : จัดพิมพ์โดย บ.เคทีพี คอมพ์-แอนด์ คอนซัลท์ 2551

[2] สัจจะ จรัสรุ่งคู่มือ. Visual C# 2005 ฉบับสมบูรณ์. กรุงเทพฯ : จัดพิมพ์โดย บ.ไอดีซี อินโฟ-ดิสทริบิวเตอร์ เซ็นเตอร์ 2550

[3] ศุภชัย สมพานิช. สร้างระบบงานฐานข้อมูลด้วย Visual Basic.NET ฉบับโปรแกรมเมอร์. กรุงเทพฯ : จัดพิมพ์โดย บ.ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ 2546

[4] ผศ. ดร.วรพงศ์ ตั้งศรีรัตน์. เซนเซอร์และทรานสดิวเซอร์ ทฤษฎีและการประยุกต์ใช้ในระบบการวัดและระบบควบคุม. พิมพ์ครั้งที่ 7. กรุงเทพฯ : จัดพิมพ์โดย สำนักพิมพ์ส.ส.ท. 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้