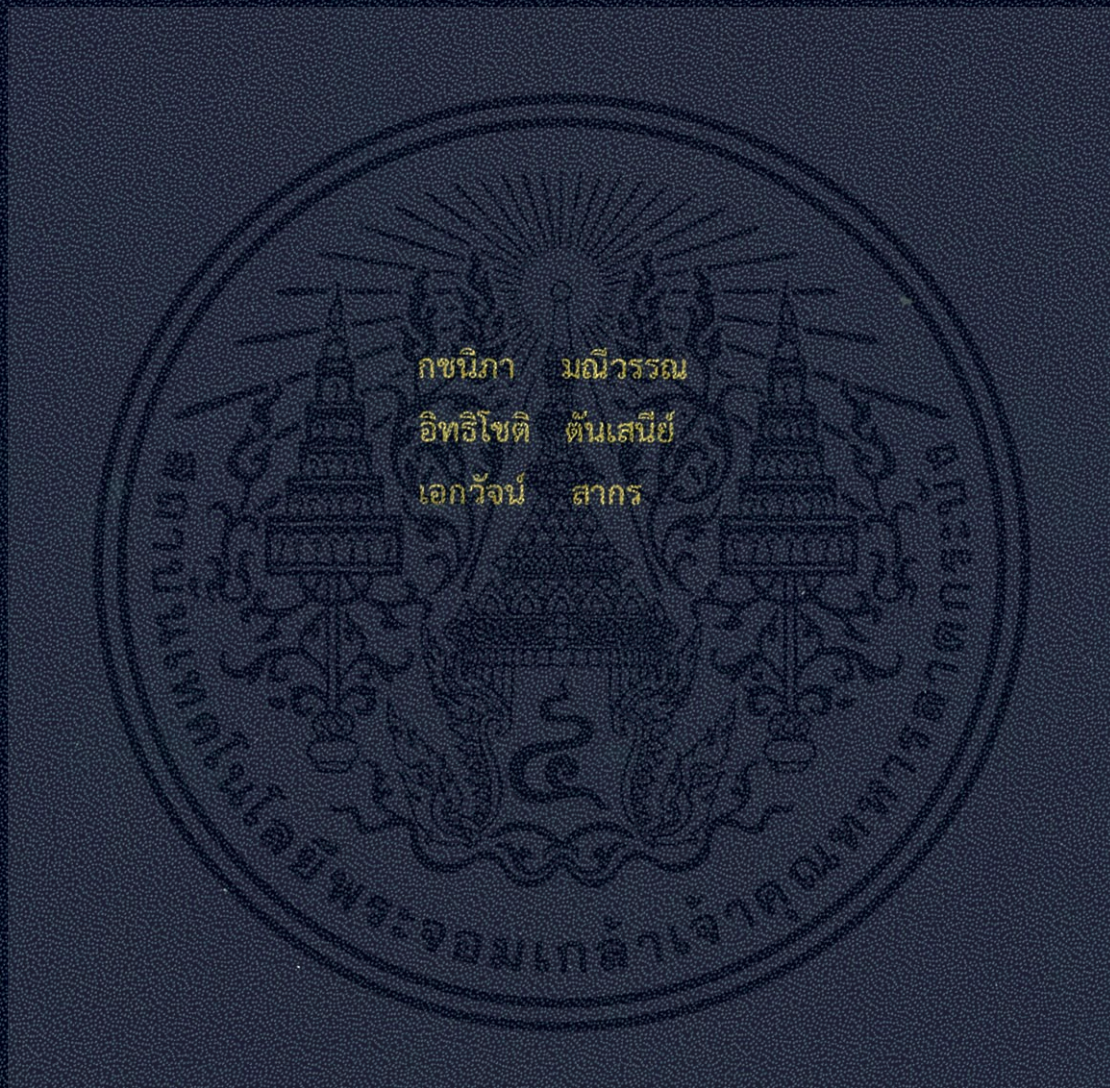


ระบบตรวจวัดอุณหภูมิ
Temperature Monitoring System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ระบบตรวจวัดอุณหภูมิ

Temperature Monitoring System



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temperature Monitoring System



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบตรวจวัดอุณหภูมิ
TEMPERATURE MONITORING SYSTEM
นักศึกษาผู้จัดทำ นางสาวกชนิภา มณีวรรณ รหัสนักศึกษา 53011759
นายอิทธิโชติ ตันเสนีย์ รหัสนักศึกษา 53011939
นายเอกวิจน์ สากร รหัสนักศึกษา 53011962
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
อาจารย์สุธรรม สัทธรรมสกุล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบตรวจวัดอุณหภูมิ	
	TEMPERATURE MONITORING SYSTEM	
นักศึกษาผู้จัดทำ	นางสาวกชนิภา มณีวรรณ	รหัสนักศึกษา 53011759
	นายอิทธิโชติ ต้นเสนีย์	รหัสนักศึกษา 53011939
	นายเอกวัจน์ สากร	รหัสนักศึกษา 53011962
อาจารย์ที่ปรึกษา	อาจารย์สุธรรม สัทธรรมสกุล	
ปีการศึกษา	2556	

บทคัดย่อ

โครงการนี้เป็นการนำเสนอการวัดอุณหภูมิด้วยไมโครคอนโทรลเลอร์โดยใช้ PIC Microcontroller เป็นตัวรับค่าอุณหภูมิจากเซนเซอร์และส่งค่าอุณหภูมินั้นผ่าน CAN BUS ไปยัง SNAP ระบบตรวจวัดสามารถแสดงผลอุณหภูมิจริงและอุณหภูมีย้อนหลังผ่านเว็บเบราว์เซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title TEMPERATURE MONITORING SYSTEM
Authors Miss Kodchanipha Maneewan
 Mr. Itthichote Tansenee
 Mr. Ekawat Sakorn
Thesis Advisor Mr. Sutham Satthamsakul
Year 2013

ABSTRACT

This project presents the measurement of temperature sensor modules by using PIC Microcontroller to receive and send temperature value to SNAP by CAN BUS. The monitoring system can be displayed the real-time and logger report on web browser.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งไม่อาจจะนำมา
กล่าวทั้งหมดได้ ผู้มีพระคุณท่านแรกที่คณะผู้จัดทำใคร่ขอกราบขอบพระคุณคือ อาจารย์ สุธรรม
สัทธรรมสกุล อาจารย์ที่ปรึกษาปริญญาานิพนธ์ อาจารย์ผู้สอนที่ได้ให้ความรู้ คำแนะนำตรวจทาน และ
แก้ไขข้อบกพร่องต่างๆด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้การทำปริญญาานิพนธ์ฉบับนี้สมบูรณ์ที่สุด อีก
ทั้งยังช่วยแนะนำเทคนิคการนำเสนอปริญญาานิพนธ์หน้าชั้นเรียน คณะผู้จัดทำใคร่ขอกราบขอบพระคุณ
เป็นอย่างสูงไว้ ณ โอกาสนี้ นอกจากนี้คณะผู้จัดทำขอขอบพระคุณ คุณพ่อ คุณแม่ ที่อยู่เบื้องหลังใน
ความสำเร็จที่ได้ให้ความช่วยเหลือสนับสนุนและให้กำลังใจมาโดยตลอด

ขอขอบคุณท่านอาจารย์ภาควิชาวิศวกรรมวัดคุมทุกท่านที่คอยให้คำแนะนำปรึกษาและให้ความรู้
แก่คณะผู้จัดทำตลอดมา

ขอขอบคุณพี่ๆและเพื่อนๆ ที่ช่วยในการสืบค้นข้อมูลแลกเปลี่ยนความรู้ความคิดเห็น และให้
กำลังใจในการจัดทำปริญญาานิพนธ์ฉบับนี้

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC	3
2.1.1 ชนิดของ PIC	3
2.1.1.1 OTP (one time programmable).....	3
2.1.1.2 EPROM (erasable programmable ROM).....	3
2.1.1.3 EEPROM/Flash(electronically erasable programmable ROM).....	4
2.1.2 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล PIC ในแต่ละกลุ่ม.....	4
2.1.2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC16C5XX.....	4
2.1.2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC12CXXX และ PIC12FXXX.....	4
2.1.2.3 ไมโครคอนโทรลเลอร์ตระกูล PIC17CXX.....	4
2.1.2.4 ไมโครคอนโทรลเลอร์ตระกูล PIC16FXXX.....	5
2.1.2.5 ไมโครคอนโทรลเลอร์ตระกูล PIC18CXXX และ PIC18FXXX.....	5
2.2 ตัวตรวจวัดอุณหภูมิ.....	5
2.2.1 คุณสมบัติเด่นของ DS 1820.....	6

สารบัญ (ต่อ)

	หน้า
2.2.2 บล็อกไดอะแกรมภายในของ DS 1820.....	7
2.2.3 การวัดอุณหภูมิ.....	8
2.2.4 การสื่อสารข้อมูลแบบอนุกรม 1 สาย (1-Wire)	9
2.3 SNAP (Simple Network Application Platform).....	12
2.3.1 Base Board.....	13
2.3.2 คุณสมบัติของ SNAP.....	14
2.3.3 การประยุกต์ใช้งานของ SNAP	14
2.4 Control Area Network (CAN).....	15
2.4.1 ประวัติของ Control Area Network (CAN).....	15
2.4.2 โครงสร้างของ Control Area Network (CAN).....	16
2.4.3 รูปแบบของ Control Area Network (CAN).....	17
2.4.3.1 คำอธิบาย CAN2.0A.....	17
2.4.3.2 คำอธิบาย CAN2.0B.....	18
2.4.4 ข้อแตกต่างระหว่าง CAN2.0A และ CAN2.0B.....	18
2.4.5 หลักการทำงานของ Control Area Network (CAN).....	18
2.4.5.1 การเข้ารหัสบิต.....	19
2.4.5.2 ความมีเสถียรภาพในการทำงาน.....	19
2.4.5.3 ความยืดหยุ่นของระบบเครือข่าย.....	19
2.4.5.4 การจัดลำดับข้อความ.....	20
2.4.5.5 การหยุดพักหรือการเริ่มต้นทำงานใหม่.....	20
2.4.5.6 รูปแบบของการส่งข้อความ.....	20
2.4.6 ข้อผิดพลาดที่เกิดขึ้น.....	21
2.4.6.1 การตรวจจับข้อผิดพลาด.....	21
2.4.6.2 ขอบเขตข้อผิดพลาดที่เกิดขึ้น.....	22
2.4.6.3 การจัดการเมื่อตรวจพบข้อผิดพลาด.....	22
2.4.7 ข้อดีของ Control Area Network (CAN).....	22

สารบัญ (ต่อ)

	หน้า
2.5 J2ME (Java 2 Platform , Micro Edition).....	23
2.5.1 โครงสร้างสถาปัตยกรรมของ J2ME.....	23
2.5.1.1 การตั้งค่า.....	24
2.5.1.2 Profiles.....	25
2.5.2 ประโยชน์ของ J2ME ในการพัฒนาแอปพลิเคชันบนอุปกรณ์ไร้สาย.....	27
2.6 Adobe Flash CS3.....	28
2.6.1 ส่วนประกอบของโปรแกรม Adobe Flash CS3.....	29
2.7 การสร้างเว็บเพจด้วยภาษา HTML.....	35
2.7.1 ขั้นตอนในการสร้างเว็บเพจ.....	35
2.7.2 โครงสร้างภาษา HTML.....	36
2.7.3 การกำหนดโครงสร้างหลัก.....	36
2.8 การติดต่อกับฐานข้อมูล MySQL ด้วยคำสั่ง PHP.....	37
2.9 ฐานข้อมูล.....	40
2.9.1 องค์ประกอบของฐานข้อมูล.....	40
2.9.2 ชุดคำสั่งที่ใช้จัดการกับข้อมูล.....	41
2.9.3 โครงสร้างของภาษา My SQL.....	41
2.9.3.1 คำสั่งเบื้องต้นสำหรับ My SQL.....	42
บทที่ 3 วิธีการดำเนินงาน.....	43
3.1 การออกแบบ.....	43
3.1.1 โครงสร้างของตัววัดคุณทรมุมิ.....	43
3.1.1.1 คำสั่งที่ใช้ในการควบคุมการทำงาน.....	44
3.1.1.2 การควบคุมการทำงาน.....	44
3.1.1.3 การควบคุมการทำงานของสัญญาณเตือน.....	45
3.1.1.4 บิตเลเซอร์รวม.....	46
3.1.2 ไมโครคอนโทรลเลอร์.....	47

สารบัญ (ต่อ)

	หน้า
3.2 การทำงานของระบบตรวจวัดอุณหภูมิ.....	48
3.3 วิธีการดำเนินงาน.....	49
3.3.1 ขั้นตอนการดำเนินงาน.....	49
3.3.1.1 ขั้นตอนการติดตั้งจาวาและการตั้งค่า.....	49
3.3.1.2 การติดตั้งโปรแกรม Snap Developer.....	53
3.3.1.3 การเปิด FTP Server บน Snap Board.....	58
3.3.1.4 การติดตั้ง FTP Client.....	59
3.3.1.5 การทดลองรันโปรแกรม.....	61
3.3.1.6 การควบคุม SNAP ผ่าน Telnet	62
3.3.2 โปรแกรมของ SNAP Board.....	63
3.3.2.1 TempReader.java.....	64
3.3.2.2 ThreadCan.java.....	65
3.3.2.3 ShowTempCqi.java.....	65
3.3.2.4 ShowTempOnWeb.java.....	65
3.3.3 การสร้างหน้าจอแสดงผลค่าอุณหภูมิ.....	66
3.3.3.1 การออกแบบฐานข้อมูลใน MySQL Server.....	66
3.3.3.2 ไฟล์โปรแกรมของระบบ.....	68
3.3.3.3 วิธีการสร้างไฟล์โปรแกรมของระบบ.....	69
3.4 การเขียนภาษา JAVA บน SNAP.....	69
3.4.1 การออกแบบระบบและการพัฒนาระบบ.....	70
3.4.1.1 การให้ SNAP ส่งข้อมูลไปยัง MySQL Server.....	70
3.5 SERVER.....	71
3.5.1 ส่วนประกอบหลักของ server.....	71
3.5.1.1 Main Program.....	71
3.5.1.2 Function Program.....	71
3.5.1.3 Database.....	72

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน.....	73
4.1 บทนำ.....	73
4.2 การทดลองและผลการทดลอง.....	73
4.2.1 การทดลองแสดงค่าอุณหภูมิบน LCD	73
4.2.2 การทดลอง SNAP.....	73
4.2.3 การทดลอง server.....	74
4.2.3.1 การทดลองรัน servermain บน server.....	74
4.2.3.2 ผลการทดลองรัน servermain บน server.....	75
4.2.3.3 ผลการทดลองการสร้างฐานข้อมูล.....	75
4.2.3.4 ผลการทดลองการสร้าง web browser.....	76
4.2.3.5 ผลการเรียกดูอุณหภูมิก่อนหน้า.....	76
บทที่ 5 สรุปผล และ ปัญหา.....	77
5.1 สรุปผล.....	77
5.2 ปัญหา.....	77
5.2.1 ปัญหาและแนวทางแก้ไข.....	77
บรรณานุกรม.....	78

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดง CAN Message Frame.....	20
2.2 ตารางเปรียบเทียบ CDC และ CLDC.....	24
2.3 ตารางแสดง Profile และ Configuration.....	26



สารบัญรูป

รูปที่	หน้า
2.1 แสดงลักษณะโครงสร้างและการจัดขาของ DS1820.....	6
2.2 แสดงบล็อกไดอะแกรมภายในของ DS1820.....	7
2.3 แสดงบล็อกไดอะแกรมการทำงานของ DS1820.....	8
2.4 แสดงการจัดรูปแบบข้อมูล 9 บิต.....	9
2.5 แสดงแผนผังของระบบบัสแบบ 1- Wire Bus.....	9
2.6 แสดงจังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส.....	10
2.7 แสดงจังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล '0' หรือ '1' ไปยังอุปกรณ์บนบัส.....	11
2.8 แสดงรูปของ SNAP (Simple Network Application Platform).....	12
2.9 แสดงโครงสร้างของ CAN Nodes.....	16
2.10 แสดงรูปร่างคำสั่ง (Message Frame) ของ CAN (Control Area Network).....	17
2.11 แสดงความแตกต่างระหว่าง CAN 2.0A และ 2.0B.....	18
2.12 แสดงการสื่อสารของแต่ละโหนดบนระบบบัส.....	22
2.13 แสดงส่วนประกอบของโปรแกรม Adobe Flash CS3.....	29
2.14 แสดง Time Line ส่วนควบคุมโปรแกรมและกำหนดการนำเสนองาน.....	30
2.15 แสดงส่วนของ Layer	31
2.16 แสดงแถบเครื่องมือ Tool Bar.....	32
2.17 แสดงแถบเครื่องมือ Panel.....	33
2.18 แสดงแถบเครื่องมือ Stage.....	34
2.19 แสดงแถบเครื่องมือ Properties.....	35
2.20 แสดงการจัดวางคำสั่งแบบมาตรฐานของ HTML.....	36
3.1 แสดงภาพรวมการทำงานของโครงงาน.....	43
3.2 แสดงวงจรวัดอุณหภูมิแสดงผลบนจอ LCD.....	44
3.3 แสดงตารางความสัมพันธ์ของค่าอุณหภูมิกับข้อมูลดิจิทัลเอาต์พุต.....	45
3.4 แสดงการแบ่งส่วนในหน่วยความจำรอมขนาด 64 บิต.....	46
3.5 แสดงไดอะแกรมของ PIC18F258.....	47
3.6 แสดงการทำงานของระบบตรวจวัดอุณหภูมิ.....	48

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7 แสดงการเปิดโปรแกรม J2DK Version.....	49
3.8 แสดงรายละเอียดต่างๆของโปรแกรม.....	50
3.9 แสดงการเลือก Directory.....	50
3.10 แสดงการลงโปรแกรม.....	51
3.11 แสดงการลงโปรแกรมเสร็จสมบูรณ์.....	51
3.12 แสดงการ Configuration Program J2DSK.....	52
3.13 แสดงการกำหนด Path.....	52
3.14 แสดงการทดสอบ JAVA.....	53
3.15 แสดงตำแหน่ง directory ของ Snap Developer 1.0.8.....	53
3.16 แสดงรายละเอียดในไฟล์ build.bat.....	54
3.17 แสดงรายละเอียดของแฟ้ม Built.bat.....	54
3.18 แสดงโปรแกรม SNAP Dev.....	55
3.19 แสดงการกำหนด Configuration.....	55
3.20 แสดงการกำหนดค่า Baudrate.....	56
3.21 แสดงโปรแกรม SnapDev เมื่อติดต่อกับ Snap.....	56
3.22 แสดงการใช้คำสั่ง help เพื่อโชว์ program ใน root directory.....	57
3.23 แสดงสถานะของ SNAP.....	57
3.24 แสดงหน้าจอแสดงผลลัพธ์ของคำสั่ง ipconfig.....	58
3.25 แสดงการ Active Server ทั้ง 3 Server คือ Telnet, Serial, FTP.....	58
3.26 แสดงหน้าต่างโปรแกรม FileZilla.....	59
3.27 แสดงหน้าต่างการกำหนดคุณสมบัติของ Fields.....	60
3.28 แสดงการ Upload File.....	60
3.29 แสดงการเปลี่ยน Sub Directory ไปที่ classes.....	61
3.30 แสดงแสดงผลลัพธ์ของการ RUN โปรแกรม.....	61
3.31 แสดงหน้าจอของโปรแกรม Telnet.....	62
3.32 แสดงหน้าจอหลังจากผ่านการ login เข้ามาแล้ว.....	62

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.33 แสดงผลของโปรแกรม HelloWorld ผ่าน Telnet.....	63
3.34 แสดงการกำหนด CAN Controller บน SNAP.....	64
3.35 แสดงการตรวจสอบข้อมูลบนสายสัญญาณ.....	64
3.36 แสดงการสร้างให้ TempReader ทำงานในสถานะของ Thread	65
3.37 แสดงการสร้างหน้าจอแสดงผลของการอ่านค่าอุณหภูมิบน Browser.....	65
3.38 แสดงการสร้างหน้าจอแสดงผลของการอ่านค่าอุณหภูมิบน Browser	66
3.39 แสดง Context Diagram.....	66
3.40 แสดงการอธิบายหน้าแรกของ PhpMyAdmin.....	67
3.41 แสดงการสร้างชื่อฐานข้อมูล.....	67
3.42 แสดงการสร้างตารางโดยกำหนดชื่อและจำนวน fields.....	67
3.43 แสดงหัวข้อเพื่อกำหนด fields และ คุณสมบัติต่างๆ.....	68
3.44 แสดงตารางการประเมิน.....	68
3.45 แสดงตารางค่าอุณหภูมิ.....	68
3.46 แสดงการทำงานของระบบโดยรวม.....	69
3.47 แสดง Diagram ของระบบ.....	71
4.1 แสดงเครื่องตรวจวัดอุณหภูมิ.....	73
4.2 แสดงค่าของอุณหภูมิที่รับมาจาก CAN BUS ไปยัง Server.....	74
4.3 แสดงโปรแกรม Servermain.....	75
4.4 แสดงผลการรันโปรแกรม Servermain.....	75
4.5 แสดงผลการรับข้อมูลลง Database.....	76
4.6 แสดงผลค่าของอุณหภูมิบน Web browser โดยใช้ Adobe FLASH CS3.....	76

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันทางสถาบันมีภาระรายจ่ายในเรื่องของค่าพลังงานไฟฟ้าอยู่ในระดับที่สูงมาก ซึ่งทางคณะเองก็มีนโยบายและรณรงค์ให้ทุกคนช่วยกันลดการใช้พลังงานไฟฟ้าในส่วนที่ไม่จำเป็น โดยเฉพาะเครื่องปรับอากาศที่เป็นเครื่องใช้ไฟฟ้าที่ใช้พลังงานไฟฟ้าสูงมาก ประกอบกับปัญหาสภาวะโลกร้อนที่ทำให้โลกมีอุณหภูมิที่สูงขึ้นส่งผลทำให้เครื่องปรับอากาศต้องทำงานหนักขึ้น นั้นยิ่งส่งผลถึงพลังงานไฟฟ้าที่ต้องใช้มากขึ้นไปอีก ดังนั้นทางผู้จัดทำพิจารณาแล้วเห็นว่าหากมีระบบในการตรวจสอบอุณหภูมิภายในห้องต่างๆ ภายในอาคารพร้อมทั้งมีการจัดเก็บข้อมูลของอุณหภูมินั้นลงในฐานข้อมูลตลอดเวลาเพื่อประโยชน์ในการที่จะนำข้อมูลดังกล่าวกลับมาวิเคราะห์และใช้งานในภายหลัง น่าจะเป็นประโยชน์อย่างยิ่งในการวางแผนเพื่อหาวิธีการที่จะลดปริมาณการใช้พลังงานไฟฟ้าลงอย่างยั่งยืน จึงเป็นที่มาของโครงการระบบตรวจวัดอุณหภูมิ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของตัวตรวจวัดอุณหภูมิ
2. เพื่อศึกษาการเขียนโปรแกรมควบคุมบนไมโครคอนโทรลเลอร์เพื่อให้ทำงานร่วมกับตัวตรวจวัดอุณหภูมิ
3. เพื่อศึกษาการเขียนโปรแกรมสื่อสารระหว่าง SNAP กับ ไมโครคอนโทรลเลอร์ ผ่าน CAN BUS
4. เพื่อศึกษาการเขียนโปรแกรมสื่อสารระหว่าง SNAP กับเครื่องแม่ข่าย เพื่อบันทึกข้อมูลลงฐานข้อมูล
5. เพื่อศึกษาการเขียนเว็บไซต์เพื่อแสดงข้อมูลอุณหภูมิจากฐานข้อมูล

1.3 ขอบเขต

1. สามารถสร้างอุปกรณ์สำหรับตรวจวัดค่าอุณหภูมิที่สามารถส่งค่าอุณหภูมิไปยัง SNAP ผ่าน CAN BUS ได้
2. สามารถนำข้อมูลของอุณหภูมิบันทึกลงฐานข้อมูลได้
3. สามารถตรวจสอบอุณหภูมิผ่านทางเว็บเบราว์เซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการเขียนโปรแกรมภาษา C
2. ศึกษาการทำงานของ SNAP
3. ศึกษาการเขียนโปรแกรมเพื่อส่งข้อมูลผ่านระบบ CAN BUS
4. ศึกษาการเขียนโปรแกรมควบคุม บน SNAP ด้วย JAVA
5. ศึกษาการเขียนโปรแกรมติดต่อระหว่าง PIC และเซนเซอร์
6. ศึกษาการเขียนโปรแกรมด้วยภาษา HTML
7. ศึกษาการเขียนโปรแกรม Adobe Flash CS3



บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC

PIC คือ ไมโครคอนโทรลเลอร์อีกตระกูลหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่งเป็นไมโครคอนโทรลเลอร์ที่พยายาม รวมเอาทุกอย่างเอาไว้ในตัวไม่ว่าจะเป็น PROGRAM MEMORY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้มันเหมือนกับ CPU

2.1.1 ชนิดของ PIC

MCU ในตระกูล PIC ถ้าแบ่งออกตามชนิดของ PROGRAM MEMORY แบ่งได้เป็น 3 แบบคือ

1. OTP (one time programmable)
2. EPROM (erasable programmable ROM)
3. EEPROM / Flash (electronically erasable programmable ROM)

2.1.1.1 OTP (one time programmable)

OTP เป็นชิปที่มีราคาถูกที่สุดในสามประเภท สาเหตุก็มาจากว่าชิปแบบ OTP จะสามารถทำการโปรแกรมได้แค่ครั้งเดียวเท่านั้น หลังจากชิปได้ถูกโปรแกรมไปแล้วจะไม่สามารถโปรแกรมเข้าไปใหม่ได้อีก ดังนั้นชิปประเภทนี้จะนิยมใช้หลังจากได้พัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆในโปรแกรมไม่มีอีกแล้วเพราะจะมีต้นทุนต่ำเมื่อเทียบกับที่ใช้หน่วยความจำ ประเภทอื่น จะมีตัวอักษร C แสดงบนตัวชิป เช่น 16C84, 16C74

2.1.1.2 EPROM (erasable programmable ROM)

EPROM เป็นชิปที่มี program memory ที่เมื่อเขียนโปรแกรมเข้าไปแล้วสามารถโปรแกรมใหม่ด้วยการลบโปรแกรมเดิมโดยให้แสง UV (Ultra Violet) ส่องผ่านเข้าไปยังชิปประมาณ 5-10 นาที ดังนั้นที่ด้านบนของชิปจะมีกรอบกระจกเพื่อให้เห็นแสง UV สามารถส่องผ่านเข้าไปในตัวชิปได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรมเช่นกัน เมื่อลบโปรแกรมด้วยแสง UV มากๆ เข้าก็จะเป็นการด้านคือโปรแกรมไม่เข้านั่นเอง จะมีตัวอักษร JW หรือว่าดูเอาว่ามีกรอบกระจกอยู่บน ชิปหรือไม่

2.1.1.3 EEPROM/Flash(electronically erasable programmable ROM)

EEPROM/Flash เป็นชิปที่ออกมาไม่กี่ปีนี้เอง ส่วนของหน่วยความจำของโปรแกรม สามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ใช้เวลาในการ ลบข้อมูลไม่กวีนาที และสามารถลบ และเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดใน 3 ประเภท มีตัวอักษร F เป็นตัวบอก เช่น 16F84, 16F877

2.1.2 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล PIC ในแต่ละกลุ่ม

การแบ่งกลุ่มของ PIC จะแบ่งตามตัวเลขที่ขึ้นต้น เช่นขึ้นต้นด้วย PIC12, PIC16, และ PIC17 โดยแต่ละกลุ่มก็จะมีเบอร์ต่างๆ แยกออกไปอีกหลายเบอร์ ซึ่งสามารถแบ่งออกได้ 5 กลุ่ม คือ

2.1.2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC16C5XX

เป็นชิปที่ผลิตออกมาเป็นยุคแรกๆ เหมาะกับงานที่ไม่ยุ่งยากและซับซ้อน โดยมีคำสั่งใช้งานที่เป็นภาษาแอสเซมบลี 33 คำสั่งมี I/O, TIMER และ WatchDogแต่ไม่มี I2C หรือ Serial ซึ่งต้องเขียนโปรแกรมขึ้นมาเอง ชิปในกลุ่มนี้มีหลายเบอร์ แต่ละขาจะมีจำนวนขาและขนาดของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลต่างกัน

2.1.2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC12CXXX และ PIC12FXXX

เป็นชิปขนาดเล็กที่มีเพียงแค่ 8 ขาเท่านั้นซึ่งเหมาะกับงานที่มีขนาดเล็กมีคำสั่งภาษาแอสเซมบลี 33 และ 35 คำสั่ง สำหรับจุดเด่นของ PIC กลุ่มนี้คือมีสัญญาณนาฬิกาหรือออสซิลเลเตอร์ (Oscillator) ขนาด 4 MHz อยู่ภายในชิป ทำให้ไม่ต้องต่อออสซิลเลเตอร์ภายนอกและมีหน่วยความจำข้อมูลภายในเป็นแบบ EEROM แต่ส่วนของหน่วยความจำโปรแกรมยังเป็นแบบ OTP และ EPROM หลังจากได้ผลิต PIC16FXXX แล้วทางบริษัทไมโครชิปก็ได้ผลิต PIC12FXXX ขึ้นมา ซึ่งเป็นแบบแฟลชทำให้สามารถเขียนและลบโปรแกรมได้หลายครั้ง

2.1.2.3 ไมโครคอนโทรลเลอร์ตระกูล PIC17CXXX

เป็นชิปที่ออกมาพร้อมกับ PIC16CXXX แต่แตกต่างกันที่ PIC17CXXX จะมีความสามารถสูงกว่า มีจำนวนขามากกว่า มีคำสั่งแอสเซมบลี 58 คำสั่ง มีคำสั่งการคูณหาร รวมทั้งขนาดของหน่วยความจำโปรแกรม หน่วยความจำข้อมูลที่มีขนาดใหญ่ขึ้นและยังสามารถต่อกับหน่วยความจำภายนอกได้ด้วย

2.1.2.4 ไมโครคอนโทรลเลอร์ตระกูล PIC16FXXX

เป็นชิปที่ได้รับความนิยมมาก เพราะเป็นชิปรุ่นแรกที่หน่วยความจำโปรแกรมเป็นแบบแฟลชและมีหน่วยความจำข้อมูลแบบ EEPROM ทำให้สามารถพัฒนาโปรแกรมได้ง่าย ซึ่ง PIC เบอร์ PIC16FXXX สนับสนุนการทำงานแบบอินเซอร์กิตดีบักเกอร์ (In Circuit Debugger) ทำให้ไม่ต้องใช้อีพรอมอีมูเลเตอร์ (EPROM Emulator) ซึ่งมีราคาแพงมาก มีคำสั่งภาษาแอสเซมบลี 35 คำสั่ง และมีวงจรแปลงสัญญาณแอนาล็อกเป็นสัญญาณดิจิทัล (A/D) ขนาด 10 บิตอยู่ภายในด้วย

2.1.2.5 ไมโครคอนโทรลเลอร์ตระกูล PIC18CXXX และ PIC18FXXX

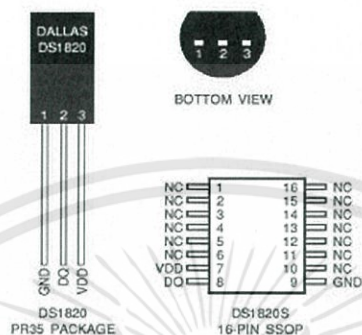
เป็นอีกกลุ่มที่ได้รับความนิยม เนื่องจาก PIC18CXXX และ PIC18FXXX มีความสามารถมากและใหญ่กว่าเบอร์ PICFXXX ไม่ว่าจะเป็คำสั่งภาษาแอสเซมบลีที่มีถึง 77 คำสั่งหรือหน่วยความจำโปรแกรมที่มีขนาดใหญ่ขึ้น ทำให้สามารถรองรับการเขียนโปรแกรมภาษาซีได้

2.2 ตัวตรวจวัดอุณหภูมิ

DS1820 เป็นไอซีที่มีระบบการสื่อสารข้อมูลอนุกรมแบบหนึ่งสายซึ่งถือได้ว่าเป็นระบบที่มีความชาญฉลาดและใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกาควบคุมจังหวะการถ่ายทอดข้อมูลเหมือนกับระบบสื่อสารข้อมูลอนุกรมในแบบอื่น สายข้อมูลนี้จะทำหน้าที่เสมือนเป็นสายสัญญาณนาฬิกา สำหรับในส่วนค่าของข้อมูลจะพิจารณาจากลักษณะรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่องของเวลา ซึ่งจะเรียกว่าไทม์สล็อต (Time Slot) โดยคาบเวลาดำสุดและสูงสุดของสถานะต่างๆในการสื่อสารข้อมูลของแต่ละไทม์สล็อต จะมีการกำหนดขอบเขตไว้อย่างชัดเจน การถ่ายทอดข้อมูลที่เกิดขึ้นในแต่ละไทม์สล็อตนั้น รูปแบบการถ่ายทอดข้อมูลจะเป็นแบบอะซิงโครนัสในระดับบิตไม่มีการกำหนดความยาวของข้อมูลเป็นระดับไบต์ ระบบสื่อสารแบบนี้เหมาะที่จะใช้ในการสื่อสารข้อมูลระหว่างไอซีแผงวงจรเดียวกัน

DS1820 จะให้สัญญาณเอาต์พุตออกมาเป็นแบบดิจิทัล มีระดับความผิดพลาดที่ 0.5 องศาเซลเซียส สามารถวัดอุณหภูมิในช่วง -55 องศาเซลเซียส ถึง 125 องศาเซลเซียส ที่ความละเอียด 9-12 บิต และยังสามารถที่จะทำการโปรแกรมเข้าไปยังหน่วยความจำและควบคุมฟังก์ชันภายในตัวอุปกรณ์นี้ได้ ซึ่งมีหน่วยความจำรวมภายในขนาด 64 บิตแบบเลเซอร์รอม ดังนั้นจึงสามารถที่จะทำการอ่านและเขียนข้อมูลต่างๆ ที่เกี่ยวกับหน้าที่การทำงานในการตรวจวัดอุณหภูมิได้อย่างมากมายตามการประมวลผลของไมโครโปรเซสเซอร์ นอกจากนั้นแล้วยังสามารถติดตั้ง DS1820 เพื่อการตรวจวัดอุณหภูมิได้ในหลายลักษณะและหลายสถานที่ ไม่ว่าจะเป็นการติดตั้งภายในอาคาร , อุปกรณ์เครื่องใช้ต่างๆ หรือ ภายใน

เครื่องจักร และที่สำคัญยังสามารถเชื่อมต่อ DS1820 บนสายสัญญาณเพียงเส้นเดียวได้หลายๆชุดโดยไม่สับสนข้อมูลซึ่งกันและกัน ในรูปที่ 2.1 แสดงรูปร่างลักษณะและการจัดขาใช้งานของ DS1820



รูปที่ 2.1 ลักษณะและการจัดขาใช้งานของ DS1820

2.2.1 คุณสมบัติเด่นของ DS1820

1. อินเทอร์เฟซสัญญาณผ่านขาเอาต์พุตเพียงพอร์ตเดียวแบบ 1 สายข้อมูล
2. ขยายจุดตรวจจับอุณหภูมิได้หลายๆ จุดบนสายข้อมูลเพียง 1 สายข้อมูล
3. ไม่ต้องใช้อุปกรณ์ภายนอกมาต่อร่วม
4. สามารถควบคุมการทำงานเพาเวอร์ออนได้ผ่านทางสายข้อมูล
5. เพาเวอร์ขณะสแตนด์บายเป็นศูนย์
6. ย่านการวัดอุณหภูมิตั้งแต่ -55 องศาเซลเซียสถึง +125 องศาเซลเซียสที่ 0.5 องศาเซลเซียสต่อสแต็ป หรือ ตั้งแต่ย่าน -67 องศาฟาเรนไฮต์ ถึง +257 องศาฟาเรนไฮต์ที่ 0.9 องศาฟาเรนไฮต์ต่อสแต็ป
7. อุณหภูมิจะถูกอ่านออกมาเป็นค่าทางดิจิตอล 9 บิต
8. อัตราความเร็วในการแปลงจากอุณหภูมิมาเป็นค่าตัวเลขทางดิจิตอล เท่ากับ 200 มิลลิวินาที
9. ผู้ใช้งานสามารถกำหนดการเซตค่าเตือนย่านอุณหภูมิได้ในแบบ non - volatile
10. การเตือนย่านอุณหภูมินั้นสามารถกำหนดรหัสผ่านการสั่งการและแอดแตรสของอุปกรณ์ได้จากภายนอกพื้นที่ตรวจวัดอุณหภูมิผ่านทางโปรแกรมภายนอก
11. เหมาะกับการประยุกต์ใช้งานตรวจวัดอุณหภูมิ และสามารถติดตั้งไว้ในอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

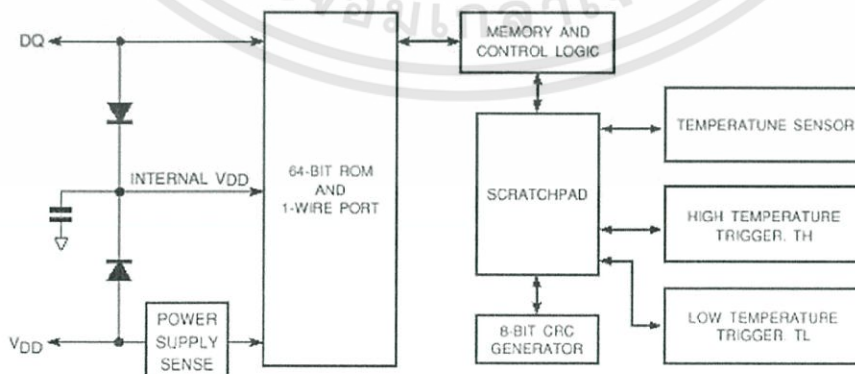
ควบคุมเทอร์โมสแตติก, ระบบโรงงานอุตสาหกรรม , ผลิตภัณฑ์, เทอร์โมมิเตอร์ หรือระบบอื่นๆที่มีส่วนตรวจจับอุณหภูมิทำงานร่วมอยู่

2.2.2 บล็อกไดอะแกรมภายในของ DS1820

ในรูปที่ 2.2 แสดงบล็อกไดอะแกรมส่วนประกอบภายในของการทำงานต่างๆ ภายในตัว DS1820 มีส่วนประกอบหลักๆ 3 ส่วนด้วยกันคือ หน่วยความจำเลเซอร์รอมขนาด 64 บิต, ส่วนเซ็นเซอร์อุณหภูมิและส่วนกระตุ้นเตือนอุณหภูมิแบบ non – volatile (TH และ TL) โดยอุปกรณ์ตรวจวัดอุณหภูมินี้จะถูกควบคุมสภาวะการเพาเวอร์ออนและเพาเวอร์ออฟจากสายข้อมูลเพียง 1 สาย ซึ่งข้อมูลจะถูกเก็บรักษากำลังงานสำรองไว้ในตัวเก็บประจุภายใน ในช่วงระหว่างคาบเวลาเมื่อสัญญาณภายในสายข้อมูลมีสถานะเป็น high และจะทำงานต่อเนื่องไปเรื่อยๆ การหยุดการทำงานจะเกิดขึ้นก็ต่อเมื่อมีการหยุดจ่ายแหล่งจ่ายไฟในช่วงระหว่างคาบเวลานั้นมีสถานะเป็น low การทำงานจะหยุดอยู่เช่นนั้นจนกว่าสายข้อมูลจะกลับมาเป็น high อีกครั้ง และแหล่งจ่ายไฟหลักนี้ก็จะได้จากแหล่งจ่ายไฟ +5 โวลต์ภายนอก

การติดต่อข้อมูลกับ DS1820 นี้จะติดต่อผ่านพอร์ตเพียงพอร์ตเดียวคือ 1 – Wire port ภายในพอร์ต 1 – Wire นี้เป็นส่วนของหน่วยความจำและควบคุมฟังก์ชันโปรโตคอลของรอมจะถูกทำการเสตค่าเสียก่อน ในส่วนสำคัญของการทำงานฟังก์ชันอันดับแรกซึ่งเป็นหนึ่งในห้าลำดับของการสั่งการฟังก์ชันในรอมก็คือ

- การอ่านหน่วยความจำรอม
- ทำการแมตซ์รอม
- ค้นหารอม
- กระโดดข้ามรอม
- เตือนการค้นหา



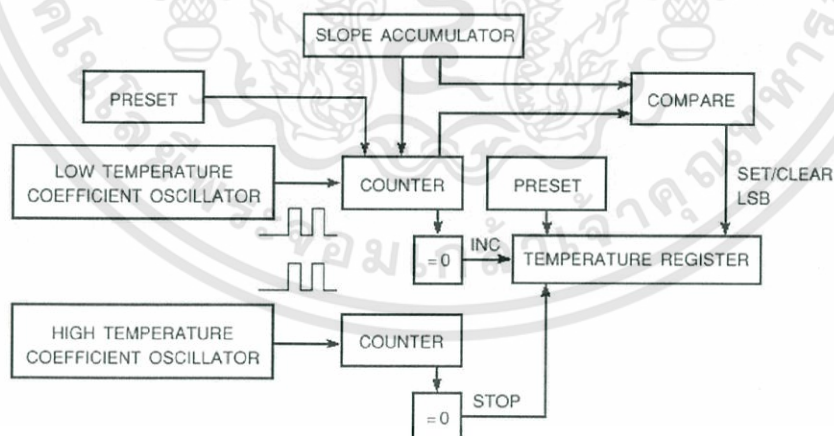
รูปที่ 2.2 บล็อกไดอะแกรมภายในของ DS1820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 การวัดอุณหภูมิ

DS1820 นี้จะทำการวัดค่าของอุณหภูมิ โดยอาศัยเทคนิคหลักการวัดแบบอนบอร์ดพิเศษซึ่งเป็นเทคนิคการวัดโดยเฉพาะของอุปกรณ์ชนิดนี้ ในรูปที่ 2.3 แสดงบล็อกไดอะแกรมของวัดค่าอุณหภูมิของ DS1820 ซึ่งจะอาศัยการวัดอุณหภูมิโดยการนับจำนวนวงรอบของสัญญาณนาฬิกาที่ออสซิลเลเตอร์ผลิตขึ้นมา ช่วงเวลาของสัญญาณนาฬิกาที่ออสซิลเลเตอร์ผลิตขึ้นมาจะเป็นการกำหนดได้จากช่วงคาบเวลาที่ค่าสัมประสิทธิ์อุณหภูมิต่ำสุด ไปจนถึงค่าสัมประสิทธิ์อุณหภูมิสูง ซึ่งจะมีค่าความถี่สัญญาณนาฬิกาที่ไม่เท่ากัน โดยที่ค่าการนับตัวเลขจะเริ่มนับที่ค่าอุณหภูมิต่ำสุดพื้นฐาน คือ -55 องศาเซลเซียส ถ้าการนับสัญญาณนาฬิกาไปถึงค่าศูนย์ รีจิสเตอร์อุณหภูมิก็จะแสดงผลที่ค่า -55 องศาเซลเซียส ถ้าหากค่าอุณหภูมิเพิ่มขึ้น การแสดงผลของอุณหภูมิขณะนั้นก็จะสูงกว่า -55 องศาเซลเซียส

ในทำนองเดียวกันนี้ การตั้งค่าของการนับจะกำหนดได้จาก การเพิ่มความลาดลงของวงจรรนับ ซึ่งวงจรรนับต้องการการชดเชยสำหรับการแสดงคุณสมบัติของส่วนโค้งของออสซิลเลเตอร์ที่อุณหภูมิมีค่าเกินมา วงจรรนับก็จะนับสัญญาณนาฬิกาอีกครั้งจนกว่ามันจะได้ค่าเป็นศูนย์ ถ้าคาบเวลาอยู่ในสถานะสงบนิ่งไม่มีการปรับแต่งก็จะเกิดการประมวลผลใหม่อีกครั้งหนึ่ง การคำนวณค่าภายใน DS1820 จะให้ความละเอียด 0.5 องศาเซลเซียสต่อสเต็ปของการเปลี่ยนแปลงอุณหภูมิ การอ่านค่าอุณหภูมิจะถูกกำหนดไว้ภายใน 16 บิต ข้อมูลจะถูกส่งออกมาเป็นแบบอนุกรมบนการอินเตอร์เฟสกับสายข้อมูล 1 - wire ซึ่ง DS1820 สามารถทำการวัดค่าอุณหภูมิได้เกินย่านตั้งแต่ -55 องศาเซลเซียสถึง +125 องศาเซลเซียส



รูปที่ 2.3 บล็อกไดอะแกรมการวัดค่าอุณหภูมิ

ที่ 0.5 องศาต่อสแต็ป ค่าอุณหภูมิที่ถูกทำการปรับตั้งไว้ใน DS1820 ในเทอมของ 1/2 องศาเซลเซียส LSB ซึ่งจะเป็นไปตามรูปแบบของข้อมูล 9 บิตที่ MSB บิตเป็นคู่เปรียบเทียบกับทุกบิตใน MSB สูงสุดของรีจิสเตอร์อุณหภูมิขนาด 2 ไบต์ ในหน่วยความจำซึ่งการอ่านค่าอุณหภูมิแบบ 16 บิต ในลักษณะสำคัญต่างๆ แสดงไว้ในรูปที่ 2.4

2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	LSB
MSb (unit = °C) LSb								
S	S	S	S	S	S	S	S	MSB

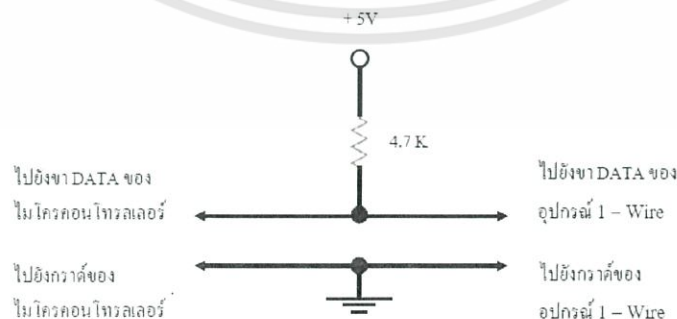
TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+85°C	0000 0101 0101 0000	0550h*
+125°C	0000 0000 1111 1010	00FAh
+25.0°C	0000 0000 0011 0010	0032h
+0.5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1111	FFFFh
-25.0°C	1111 1111 1100 1110	FFCEh
-55°C	1111 1111 1001 0010	FF92h

*The power on reset register value is +85°C

รูปที่ 2.4 การจัดรูปแบบข้อมูล 9 บิต

2.2.4 การสื่อสารข้อมูลแบบอนุกรม 1 สาย (1-wire)

อุปกรณ์ที่สนับสนุนระบบบัสเพียงเส้นเดียว หรือ การสื่อสารข้อมูลอนุกรมแบบหนึ่งสาย (1 – Wire Bus) จะมีสายสัญญาณเพียง 2 เส้นเท่านั้น คือ สายกราวด์และสายสัญญาณ ซึ่งเรียกอีกอย่างว่าสาย DATA สายนี้จะจัดการเกี่ยวกับทั้งสัญญาณข้อมูลและสัญญาณนาฬิกาที่ใช้ในการแลกเปลี่ยนข้อมูล ซึ่งสาย DATA จะเป็นชนิด Open Drain ดังนั้นในการออกแบบวงจรจะต้องออกแบบ วงจรให้มีตัวต้านทานมาต่อในลักษณะพูลอัพที่สาย DATA นี้ด้วย ดังรูปที่ 2.5



รูปที่ 2.5 แผนผังของระบบบัสแบบ 1- Wire Bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแลกเปลี่ยนข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ใช้ 1 – Wire Bus นี้ ไม่ได้ง่าย ๆ เหมือนกับการส่งข้อมูลผ่านทางบัสแบบ SPI เพราะในระบบ 1 – Wire Bus นั้น สาย DATA จะต้องจัดการเกี่ยวกับจังหวะเวลา (Timing) ระดับสัญญาณ (Level) และทิศทาง (Direction) ของข้อมูล ทั้งหมด ทำให้การเขียนซอฟต์แวร์ของไมโครคอนโทรลเลอร์ที่ติดต่อกับอุปกรณ์พวกนี้ต้องมีความ ซับซ้อนมากขึ้นในสภาวะพัก (Quiescent State) อุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus จะทำให้สาย DATA อยู่ในสภาวะลอย (float) ทำให้ขานี้มีแรงดันเท่ากับแรงดันพูลอัพ (Vcc) ซึ่งปกติก็คือ 5 โวลต์นั่นเอง ส่วนไมโครคอนโทรลเลอร์ก็จะปล่อยให้ขาเอาต์พุตที่ใช้ติดต่อกับขา DATA นี้ อยู่ในสภาวะ ความต้านทานสูง (High-Impedance State) เช่นกัน เมื่ออุปกรณ์ทั้งสองชนิดนี้ปล่อยให้ขา DATA อยู่ในสภาวะลอยแล้ว ความต้านทานพูลอัพก็จะช่วยรักษาระดับแรงดันไฟเลี้ยงที่จ่ายให้กับอุปกรณ์ที่ใช้ บัสแบบ 1 – Wire Bus นี้ได้อย่างสม่ำเสมอ เพราะอุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus นี้จะใช้พลังงานในการทำงานน้อยมาก ในการแลกเปลี่ยนข้อมูลกันไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus จะต้องดำเนินการอย่างระมัดระวังตามลำดับขั้นตอนในการทำให้สาย DATA มีลอจิกเป็น 'low' ปล่อยให้สาย DATA กลับมามีลอจิกเป็น 'high' และตรวจจับการตอบรับกับอุปกรณ์อีกด้านหนึ่ง ช่วงจังหวะเวลาที่ใช้ในกระบวนการนี้จะถูกกำหนดโดยข้อกำหนดเฉพาะของระบบ 1 – Wire Bus นี้ ซึ่งจะต้องใช้ไมโครคอนโทรลเลอร์ที่มีการตอบสนองได้อย่างรวดเร็วด้วย ดังนั้นเราจะต้องตรวจสอบความสามารถของระบบให้ดีเสียก่อนที่จะใช้ระบบบัสแบบ 1 – Wire Bus นี้ จากสภาวะพักข้างต้นการแลกเปลี่ยนข้อมูลจะเริ่มขึ้นด้วยการที่ไมโครคอนโทรลเลอร์กระทำการรีเซ็ต (Reset Sequence) ซึ่งทำได้โดยการทำให้สาย DATA มีลอจิกเป็น "low" เป็นเวลาอย่างน้อย 480 ไมโครวินาที แล้วจึงปล่อยให้กลับมาอยู่ในสภาวะ "high" อีกครั้งหนึ่ง ดังรูปที่ 2.6



รูปที่ 2.6 จังหวะเวลาในการทำการกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส

เมื่ออุปกรณ์ที่ใช้บัสแบบ 1 – Wire Bus นี้ตรวจพบสถานะ RESET แล้ว มันก็จะตอบสนองด้วยการส่งพัลส์กลับไปเพื่อบอกให้ไมโครคอนโทรลเลอร์รู้ว่าบนสายบัสนี้มีอุปกรณ์แบบ 1 – Wire Device กำลังทำงานอยู่ โดยอุปกรณ์แบบ 1 – Wire Device จะปล่อยให้สาย DATA อยู่ในลอจิก ‘high’ อย่างน้อย 15 ไมโครวินาที จากนั้นมันก็จะให้สาย DATA ลงมา มีลอจิก เป็น ‘low’ ในช่วงเวลาประมาณ 60-240 ไมโครวินาที แล้วจึงปล่อยให้กลับไปมีลอจิก ‘high’ เช่นเดิม ช่วงเวลานี้เองมีชื่อเรียกกันหลากหลายชื่อ เช่น ช่วงเวลาเริ่มติดต่อกัน (Initialization) หลังจากอุปกรณ์ 1 – Wire Device ปล่อยให้สาย DATA กลับมาอยู่ในลอจิก ‘high’ แล้ว ไมโครคอนโทรลเลอร์จะต้องปล่อยให้สาย DATA อยู่ในลอจิกนี้นานอย่างน้อย 240 ไมโครวินาที จากนั้นไมโครคอนโทรลเลอร์ก็จะส่ง คำสั่ง เริ่มต้น (Initial Command) ขนาด 1 ไบต์ไปยังอุปกรณ์ 1 – Wire Bus ซึ่งคำสั่งนี้อาจจะเป็นคำสั่งอะไรก็ได้ ไมโครคอนโทรลเลอร์จะส่งบิตของ คำสั่งนั้นออกไป โดยการเปลี่ยนสถานะของสาย DATA กลับไปกลับมาโดยตอนแรกจะให้เป็นลอจิก ‘low’ แล้วจึงกลับมาเป็น ‘high’ ตามช่วงจังหวะเวลาที่เหมาะสม ในช่วงเวลานานที่ ไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก ‘low’ จะเป็นตัวแยกแยะว่าบิตไหนที่มี ลอจิกเป็น ‘1’ บิตไหนมีลอจิกเป็น ‘0’ ให้ดูแผนภูมิเวลาในการเขียนบิต ‘1’ หรือ ‘0’ ดังรูปที่ 2.7

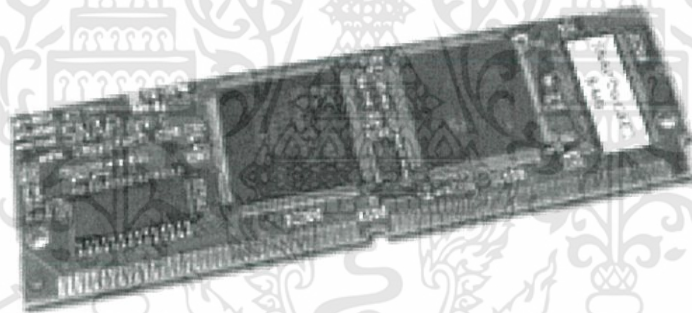


รูปที่ 2.7 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนข้อมูล ‘0’ หรือ ‘1’ ไปยังอุปกรณ์บนบัส

ช่วงเวลาระหว่าง 15-120 ไมโครวินาที หลังจากที่ไม่โครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก ‘low’ ในตอนนี้จะเรียกว่าช่วงเวลาการอ่านสถานะบิตข้อมูล (Sampling Window) ซึ่งในการอ่านบิตข้อมูลจากบัสแบบ 1-Wire Bus อันดับแรกไมโครคอนโทรลเลอร์จะต้องทำให้สาย DATA มีลอจิก ‘low’ เป็นเวลานานไม่เกิน 15 ไมโครวินาทีแล้วจึงปล่อยให้สาย DATA กลับมา มีลอจิก ‘high’ เช่นเดิม จากนั้นอุปกรณ์ 1 – Wire Device ก็จะเข้าควบคุมสาย DATA แทนโดยจะส่งบิต ‘0’ โดยการทำให้สาย

DATA มีลอจิกเป็น 'low' และส่งบิต '1' โดยการปล่อยให้สาย DATA กลับมามีลอจิก 'high' ตามเดิม เมื่อส่งข้อมูลออกไปยังไมโครคอนโทรลเลอร์เรียบร้อยแล้วจะมีการพักอยู่ชั่วขณะหนึ่ง จากนั้นอุปกรณ์ 1 – Wire Device จะปล่อยการควบคุมจากสาย DATA ให้เป็นอิสระและรอรับคำสั่งการอ่านของข้อมูลครั้งต่อไป ถ้าอุปกรณ์ 1 – Wire Device ส่งบิต '0' ออกไปไมโครคอนโทรลเลอร์ก็จะสามารถตรวจสอบจุดสิ้นสุดของลอจิก '0' นี้ง่ายขึ้น เพราะสาย DATA จะกลับมาอยู่ที่ลอจิก 'high' ตามเดิม แต่ถ้าตรวจสอบจุดสิ้นสุดของการส่งบิต '1' ของอุปกรณ์ 1 – Wire Device จะต้องใช้เทคนิคมากกว่านี้ เพราะสาย DATA จะอยู่ที่ลอจิก 'high' อยู่แล้ว นี่ก็เป็นเหตุผลว่าทำไมจึงหวัะเวลาในการอ่านเขียนข้อมูลจึงเป็นเรื่องที่สำคัญมากเมื่อจะอ่านบิต '1' จากอุปกรณ์ 1 – Wire Device ไมโครคอนโทรลเลอร์จะต้องทำตามช่วงเวลา que แสดงในแผนภูมิเวลาอย่างเคร่งครัด และต้องไม่ทำการอ่าน ลอจิกของบิตถัดมาจนกว่าจะผ่านเวลาไปแล้วอย่างน้อย 60 ไมโครวินาที

2.3 SNAP (Simple Network Application Platform)



รูปที่ 2.8 SNAP (Simple Network Application Platform)

Simple	: ใช้งานง่าย ใช้ภาษา JAVA เป็นมาตรฐาน
Network	: สนับสนุนความหลากหลายของอินเทอร์เน็ตเฟสและโปรโตคอล
Application	: สามารถใช้กับโปรแกรมที่มีประสิทธิภาพสูง
Platform	: complete reference on board

Simple Network Application Platform (SNAP) คือ Embedded Board ที่ได้รับการพัฒนาให้ทำงานร่วมกับภาษา JAVA(J2ME-CLDC) ซึ่งได้รับการรับรองโดยบริษัท SunMicrosystem พร้อมกับการพัฒนา Class ต่างๆไว้อย่างหลากหลาย เพื่อให้ง่ายต่อการติดต่อสื่อสารในระบบเครือข่าย (Network) และการควบคุมอุปกรณ์ผ่านอินพุตเอาต์พุตพอร์ตต่างๆบนตัว SNAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SNAP จะมีการติดตั้งระบบปฏิบัติการ (OS: Operating System) ที่อยู่ในกลุ่มของ Unix ซึ่ง เป็น Multi Processing อย่างแท้จริง และเมื่อรวมกับความสามารถของ JAVA ทำให้ SNAP มีรูปแบบการทำงานแบบ Multi Processing อย่างแท้จริง สามารถเปิด FTP Server, Telnet Server, Serial Server และ HTTP Server ได้พร้อมๆ กัน

2.3.1 Baseboard

เนื่องจาก SNAP ได้จัดรูปแบบการอินเตอร์เฟซเป็นขาแบบ Standard 72 pins ทำให้ไม่ค่อยสะดวกต่อการนำเอา SNAP ไปใช้งาน ทางผู้ผลิตจึงได้ผลิต Baseboard ขึ้นมาเพื่อ อำนวยความสะดวกต่อการนำเอา SNAP ไปใช้งาน หรือ เรียนรู้ได้รวดเร็วขึ้น เนื่องจาก SNAP มีการออกแบบอินเตอร์เฟซต่างๆ ไว้หลายประเภทดังนี้

1. Standard RS-232 Baseboard ได้ทำการออกแบบให้มี Serial Port 1 Port. (DB-9(F)) สำหรับใช้ ติดต่อสื่อสารระหว่างตัว SNAP กับ อุปกรณ์ต่างๆ ได้โดยตรง ซึ่งอัตราความเร็วสูงสุดอยู่ที่ 115200 Bit/s พร้อมทั้งสัญญาณควบคุมต่างๆ

2. CAN Controller (Controller Area Network) เป็นมาตรฐานการส่งข้อมูลพร้อมๆ กันบน Serial bus ได้รับการออกแบบมาเพื่อใช้สำหรับ ECU (Electronic Control Units)ซึ่งสามารถทนต่อสัญญาณรบกวนสูงได้ สูงกว่าการทำงานของ RS-485 ซึ่งเป็นการทำงานแบบ Balanced Line ซึ่งเหมาะสำหรับการใช้งานบนรถยนต์ขนส่ง และ ในโรงงานที่มีสัญญาณรบกวนสูง อัตราการส่ง อยู่ที่ 1Mbits/S จะได้ระยะทางประมาณ 40M และ ที่ 125Kbits/s จะได้ระยะทาง 500M

3. I2C port ถูกออกแบบโดยบริษัท Philips สำหรับติดต่อสื่อสารกับอุปกรณ์ต่างๆ ผ่านทางสายสัญญาณสองเส้นคือSCL, SDAมีการนำไปใช้งานกันอย่างกว้างขวาง เช่น Masterboard, Embedded System, Mobile Phone เป็นต้น โดยที่ Maximum Voltage = 5 โวลต์และต่ำสุดที่ 3.3 Volts โดยการสื่อสารจะเป็นแบบสองทางบน SDA ความเร็วสูงสุดที่ทำได้อยู่ที่ 400Mbits/s

4. Expansion port ถูกออกแบบมาสำหรับกรณีที่ต้องการออกแบบส่วนต่อขยาย เช่น ใช้งาน กับเครื่องคอมพิวเตอร์พื้นฐาน เป็นต้น

5. MMC portSNAP ได้ออกแบบให้สามารถใช้งาน MMC (Multi Media Card) หรือ SD (Secure Digital Cards) ได้ โดยกำหนดโครงสร้างของการเก็บข้อมูลอยู่ในรูปแบบ FAT16 ซึ่งสามารถนำไปใช้เก็บ Application Program หรือ Data ได้ตามความต้องการ

6. Local Area Network ใช้สำหรับการติดต่อสื่อสารบนระบบโครงข่ายและเป็นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10/100 Base-T Ethernet รวมถึงใช้ในการส่ง Application Program ไปยัง Memory ของ SNAP ผ่านช่องทาง FTP ซึ่ง SNAP สามารถรองรับการทำงานของ FTP Server, HTTP Server, Telnet Server, Serial Server ได้ในเวลาเดียวกัน

7. Terminal Port พอร์ตใช้ในการควบคุมการทำงานของ SNAP เช่น การสั่ง execute Application Program หรือการใช้ Command ต่างของ OS เช่น การใช้คำสั่ง ipconfig เพื่อกำหนดค่าของ IP Address, Subnet Mask, Gateway หรือคำสั่ง mkdir เพื่อสร้าง Subdirectory

2.3.2 คุณสมบัติของ SNAP (Simple Network Application Platform)

คุณสมบัติทางด้าน Hardware

- รองรับแหล่งจ่ายไฟมาตรฐาน 5V
- บอร์ดถูกออกแบบให้ใช้พลังงานต่ำ 3.3V
- 72 pin SIMM board, 31.8 mm.x102.9mm.
- 10/100 Base T Ethernet
- 1x RS232, 115.2 kbit/s
- I2C, High speed 3.4 Mbit/s
- 2 Mbyte flash memory
- 8 Mbyte DRAM
- Real time clock and calendar
- RoHs compliant

คุณสมบัติทางด้าน Software

- J2ME CLDC รับรองของ SunMicro Systems
- java.net
- javax.comm
- TCP / IP stack

2.3.3 การประยุกต์ใช้งานของ SNAP (Simple Network Application Platform)

- การควบคุมเครื่องหยอดเหรียญ
- ตรวจสอบติดตามควบคุม GPRS และ GPS
- การวินิจฉัยทางการแพทย์ (Medical Diagnostics)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Physical Access Control Units
- Weather Stations

2.4 Control Area Network (CAN)

2.4.1 ประวัติของ Control Area Network (CAN)

CAN (Control Area Network) หรือ CAN Protocol ถูกพัฒนาโดย Bosch และ ได้รับมาตรฐาน ISO 11898 ในปี 1994 ระบบ Control Area Network (CAN) ถูกนำมาใช้ในการควบคุมและส่งข้อมูลระหว่างอุปกรณ์ควบคุมอิเล็กทรอนิกส์ในรถยนต์ นอกจากนี้ CAN (Control Area Network) สามารถนำมาประยุกต์ใช้กับไมโครคอนโทรลเลอร์ช่วยควบคุมระบบเครือข่ายโรงงานอุตสาหกรรมเช่นการควบคุมระบบบัส (System Bus) ที่อยู่ในเครื่องจักรและอาจมีการควบคุมการทำงานของเซ็นเซอร์ (Sensor) หรืออุปกรณ์ต่างๆได้อย่างมีประสิทธิภาพ ในระบบ CAN (Control Area Network) จะมีข้อความ (Message) ที่ส่งออกมาที่มีความชัดเจนในการส่งข้อมูลมาตรฐานของ CAN (Control Area Network) จะมีการส่งข้อความ (Message) ออกมาครั้งละ 11 บิต ซึ่งจะทำให้เกิดข้อความ (Message) หรือคำสั่งที่แตกต่างกันได้ถึง 2048 ข้อความ (Message) และนอกจากนี้ได้มีการขยายมาตรฐานของ CAN (Control Area Network) ออกมาอีกโดยมีการกำหนดข้อความ (Message) ครั้งละ 29 บิต ซึ่งจะทำให้เกิดข้อความ (Message) หรือคำสั่งที่ต่างกันได้สูงสุดถึง 536 ล้านข้อความ (Message)

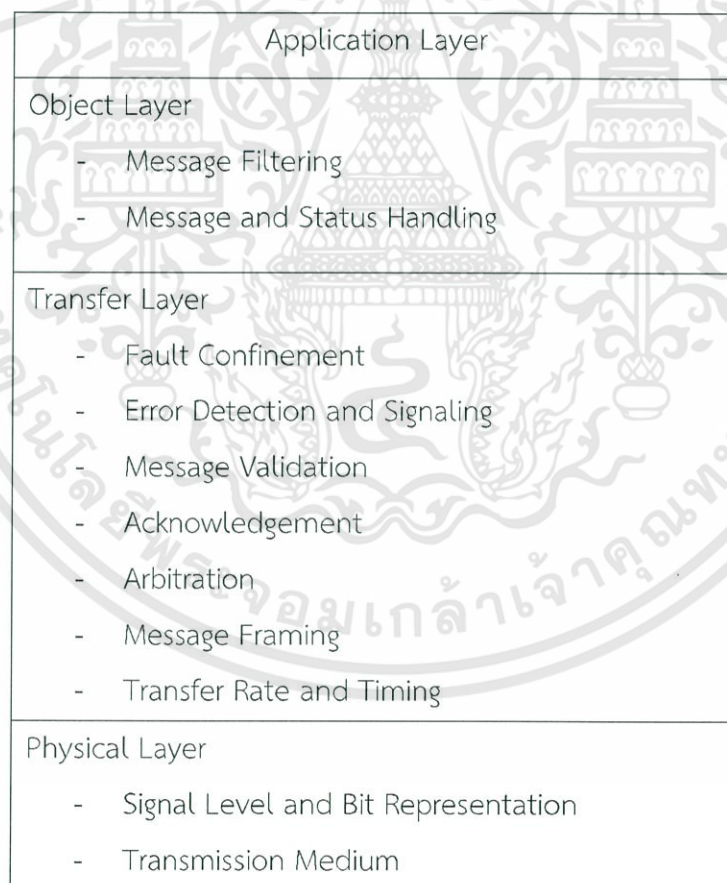
CAN (Control Area Network) สามารถมีตัวควบคุมหลัก (Master) ได้มากกว่าหนึ่งตัว ในกรณีที่มีตัวควบคุมหลักมากกว่าหนึ่งตัว แต่ละตัวจะมีสิทธิเท่าเทียมกัน และจะมีหน้าที่การทำงานเหมือนกัน ที่สำคัญของการใช้ CAN (Control Area Network) แบบตัวควบคุมหลักหลายตัว (Multi-Master) คือตัวควบคุมหลัก (Master) แต่ละตัวจะสามารถทำงาน และควบคุมได้ด้วยตัวเองและสามารถส่งข้อความ (Message) หรือการร้องขอ (Request) ได้แล้วแต่กรณี การที่ใช้ CAN (Control Area Network) แบบ Multi-Master จะมีข้อดีต่อระบบเครือข่ายคือในกรณีที่มีการเสียหรือเกิดข้อผิดพลาด (Error) ที่ตัวควบคุมหลัก (Master) จะไม่ส่งผลกระทบต่อระบบเครือข่ายและจะไม่ทำให้ระบบบัส (System Bus) ล้มเหลว หรือทำงานไม่ได้ CAN (Control Area Network) ถูกพัฒนาโดยบริษัทในประเทศเยอรมันของ Robert Bosch สำหรับใช้ในอุตสาหกรรมประกอบรถยนต์เพื่อที่จะช่วยลดปัญหาด้านทุนระบบบัส (System Bus) และการเลือกใช้ระบบ CAN (Control Area Network) เป็นทางเลือกที่เหมาะสมและประหยัดที่สุด

อุตสาหกรรมรถยนต์ใช้ CAN (Control Area Network) เพื่อช่วยเพิ่มความสามารถในการประยุกต์ใช้งานและเนื่องจาก CAN (Control Area Network) สามารถแสดงให้เห็นถึงประสิทธิภาพและความสามารถ ในปัจจุบัน CAN (Control Area Network) ถูกนำมาใช้ในอุตสาหกรรมหลายประเภท

เพื่อใช้ควบคุมและเนื่องจากจุดเด่นของ CAN (Control Area Network) คือความเร็ว ประสิทธิภาพและความปลอดภัย CAN (Control Area Network) ได้รับมาตรฐาน ISO 11898 (สำหรับ High-Speed Application) และ ISO 11519 (สำหรับ Lower-Speed Application)

2.4.2 โครงสร้างของ Control Area Network (CAN)

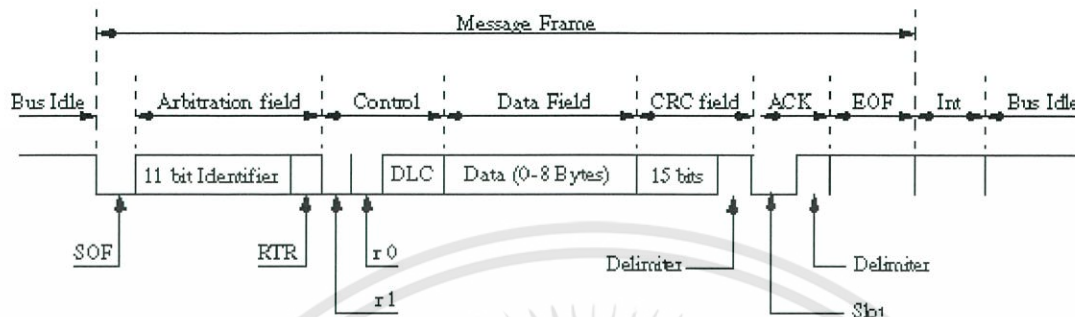
CAN (Control Area Network) คือระบบเครือข่ายการสื่อสารข้อมูลที่มีประสิทธิภาพสูงทำงานด้วยความรวดเร็วแบบทันทีทันใด (Real Time) และมีการรักษาความปลอดภัยของข้อมูลอยู่ในระดับสูง เป็นระบบเครือข่ายที่มีความเร็วสูง และมีค่าใช้จ่ายในการติดตั้งหรือดูแลรักษาต่ำ จึงได้มีการนำ CAN (Control Area Network) มาใช้ในรถยนต์ในส่วนของเครื่องยนต์เพื่อช่วยควบคุมการทำงานของเซ็นเซอร์ (Sensor) เช่น การควบคุมกระจกไฟฟ้า การควบคุมระบบไฟฟ้าในรถยนต์ เป็นต้น โครงสร้างของ CAN Nodes สามารถจัดแบ่งได้ 3 ส่วนหลัก แสดงดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของ CAN Nodes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 รูปแบบของ CAN (Control Area Network)



รูปที่ 2.10 รูปร่างคำสั่ง (Message Frame) ของ CAN (Control Area Network)

2.4.3.1 คำอธิบาย CAN2.0A

1. มาตรฐานของ CAN 2.0A ในส่วนของรูปร่างคำสั่ง (Message Frame) จะประกอบด้วย 7 bit field ที่แตกต่างกัน
2. Start Of Frame (SOF) Field คือ ส่วนที่เป็นจุดเริ่มต้นของรูปร่างคำสั่ง (Message Frame) ซึ่งจะประกอบด้วย โดมิแนนท์บิต (Dominant bit) เพียงตัวเดียว
3. Arbitration Field ประกอบด้วย ข้อความตัวระบุ (Message Identifier) และ Remote Transmission Request (RTR) บิต จะประกอบด้วยตัวระบุ (Identifier) ความยาว 11 บิต โดยจะเริ่มจาก ID10 ถึง ID 0
4. Remote Transmission Request BIT (RTR) ในชุดข้อมูลจะมี RTR บิต คือ โดมิแนนท์บิต (Dominant bit) ส่วนใน Remote Frame จะมี RTR บิต คือ รีเซสซีฟ บิต (Recessive Bit)
5. ส่วนของ Control Field จะประกอบด้วย 6 บิต ซึ่งจะแบ่งออกเป็นบิตสำรอง (Reserve Bit) คือ r0 กับ r1 ในส่วนบิตที่เหลือเรียกว่า Data Length Code (DLC) ซึ่ง Data Length Code (DLC) จะเป็นตัวที่บ่งบอกจำนวนไบต์ที่เกิดขึ้นในส่วนของ Data Field
6. Data Field จะเริ่มจาก 0 ถึง 8 ไบต์
7. Cyclic Redundancy Check Field (CRC Field) จะประกอบด้วย 15 บิต จะทำหน้าที่เช็คค่าข้อมูลที่ส่งไป เมื่อถอดรหัสแล้วจะเท่ากับค่าที่ได้ใน Cyclic Redundancy Check (CRC) ถ้าได้เท่ากันนั้นคือไม่มีข้อผิดพลาด
8. Acknowledge Field (ACK Field) จะประกอบด้วยบิตจำนวน 2 บิต คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Slot บิตและ Delimiter บิตซึ่งจะคอยเช็คดูว่าบิตที่ส่งมามีค่าของบิตมีการเปลี่ยนแปลงหรือไม่ ถ้ามีการเปลี่ยนแปลงแสดงว่ามีข้อผิดพลาดในการส่ง

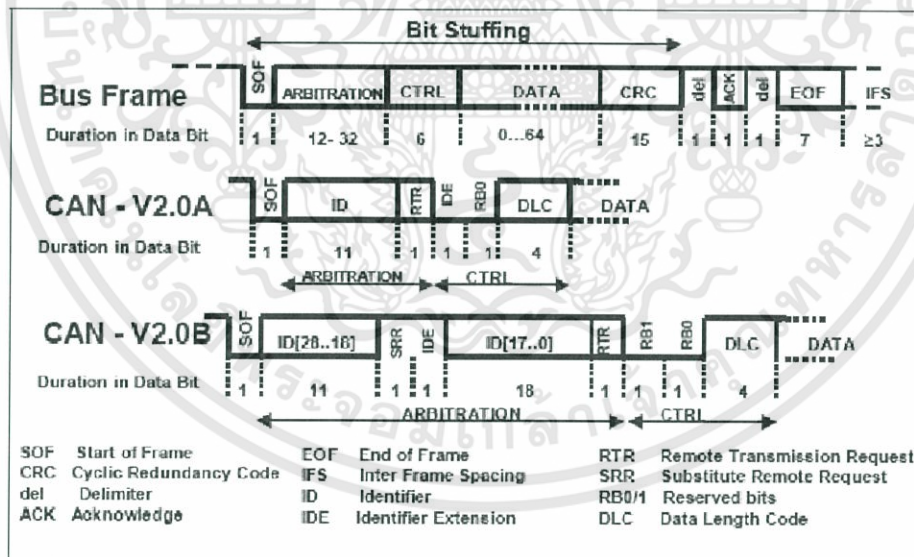
9. End Of Frame Field (EOF Field) ซึ่งจะประกอบไปด้วย รีเซสซีฟบิต (Recessive Bit) จำนวน 7 บิต

2.4.3.2 คำอธิบาย CAN 2.0B

CAN 2.0B จะมีตัวระบุ (Identifier) จำนวน 29 บิต และยังสามารถ จัดการกับการติดต่อสื่อสารระหว่าง CAN 2.0A ได้อย่างไม่มีปัญหา ซึ่ง CAN 2.0B ได้ขยายหรือพัฒนาต่อจาก CAN 2.0A

2.4.4 ข้อแตกต่างระหว่าง CAN 2.0A และ CAN 2.0B

ใน CAN 2.0B ส่วนของ Arbitration Field ประกอบด้วย 2 บิต ซึ่งบิตแรกคือ base ID มีขนาด 11 บิต สามารถส่งข้อมูลเข้ากับรูปแบบของ CAN 2.0A ได้ ส่วนบิตสอง คือ ID Extension มีขนาด 18 บิต ซึ่งจะทำให้ CAN 2.0B มีขนาด 29 บิต (11+18) ความแตกต่างระหว่าง CAN 2.0B เกิดจากในส่วนของ IDE (Identifier Extension Bit) แสดงในรูปที่ 2.11



รูปที่ 2.11 ความแตกต่างระหว่าง CAN 2.0A และ CAN 2.0B

2.4.5 หลักการทำงานของ CAN

Data Message จะส่งออกมาจากทุกโหนด (node) บน CAN Bus (Control Area Network System Bus) และจะไม่มี address ทั้งโหนด (nodes) ส่ง และโหนด (nodes) รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดข้อความ (Message) ถูกกำหนดโดยตัวระบุ (Identifier) ที่มีลักษณะเฉพาะตัวในเครือข่าย ทุกๆ โหนด (nodes) บนเครือข่ายจะรับข้อความ (Message) และคำสั่งการทำงานโดยดูจากตัวระบุ (Identifier) และแต่ละโหนด (nodes) จะตัดสินใจรับข้อความ (Message) หรือไม่โดยดูจากตัวระบุถ้า ข้อความ (Message) นั้นมาประมวลผล ถ้าไม่ใช่ก็จะหลีกเลี่ยงโดยไม่รับข้อความ (Message) และถ้ามี หลายๆ คำสั่งเข้ามาพร้อมๆ กันก็จะตัดสินใจในการรับข้อความ (Message) ที่เข้ามาโดยการดูว่ามาก่อน หรือหลัง ในบางสถานการณ์ที่มีโหนด (nodes) มากกว่าสองโหนด (nodes) ที่พยายามจะส่งข้อมูลใน เวลาเดียวกันซึ่ง CAN (Control Area Network) จะมีเทคนิคในการจัดข้อมูล โดย CAN (Control Area Network) มีการให้ลำดับก่อน-หลัง อย่างชัดเจน ซึ่งรับประกันได้ว่าไม่มีข้อมูลใดสูญหายขณะที่ส่งพร้อม กันในเวลาเดียวกัน

2.4.5.1 การเข้ารหัสบิต (Bit Encoding)

CAN (Control Area Network) ใช้ NRZ (Non Return to Zero) เพื่อเข้า รหัสสำหรับข้อมูลที่สื่อสารบนสายการสื่อสาร 2 สายที่แตกต่างกันโดยใช้ NRZ (Non Return to Zero) เข้ารหัสเพื่อให้แน่ใจว่าข้อความ (Message) กะทัดรัด กระชับและมีขนาดเล็ก ยืดหยุ่น ทนต่อการถูกรบกวนจากภายนอก

2.4.5.2 ความมีเสถียรภาพในการทำงาน

CAN (Control Area Network) สามารถที่จะจัดการกับข้อมูล (Data) ที่ ส่งหรือรับในสภาวะแวดล้อมที่ไม่เป็นปกติและมีการขยายการตรวจจับข้อผิดพลาด (Error Checking) เพื่อให้แน่ใจว่าการสื่อสารข้อมูลนั้นๆ ค่าความผิดพลาด (Error) จะถูกตรวจพบ ตามมาตรฐาน ISO 11898 ได้กล่าวสนับสนุนว่า Control Area Network (CAN) ยังคงทำงานอยู่ได้เมื่อเกิดเหตุการณ์เหล่านี้คือ สาย สื่อสารระหว่างกันมีปัญหาเกิดการเสียหาย สายสื่อสารมีการช๊อตกับไฟชั๊วบวก หรือสายสื่อสารมีการช๊อต กันกับขั้วไฟชั๊วลบ

2.4.5.3 ความยืดหยุ่นของระบบเครือข่าย(Network Flexibility and Expansion)

โดยธรรมชาติของ CAN Message จะมีแบบแผนและความยืดหยุ่นสูงสำหรับ สภาวะแวดล้อมที่แตกต่างกัน โหนด (nodes) ใหม่ๆ สามารถที่จะรับหรือส่งข้อมูลและสามารถเพิ่มเข้ามา ในระบบเครือข่ายโดยไม่ต้องมีการเปลี่ยนแปลงทั้งฮาร์ดแวร์ หรือ ซอฟต์แวร์

2.4.5.4 การจัดลำดับข้อความ

ในระบบต่างๆ ค่าพารามิเตอร์อาจเปลี่ยนแปลงรวดเร็ว เช่น การเปลี่ยนแปลงค่าของเครื่องยนต์ในรถยนต์หรือการเปลี่ยนแปลงชั้นของลิฟท์ (Lift) แต่ค่าพารามิเตอร์บางค่าอาจเปลี่ยนแปลงอย่างช้า เช่น ค่าอุณหภูมิในเครื่องทำความเย็นของรถยนต์ ดังนั้นเมื่อมีการเปลี่ยนแปลงค่าพารามิเตอร์อย่างรวดเร็ว จึงต้องมีการส่งค่าพารามิเตอร์บ่อยขึ้น จึงต้องมีการจัดลำดับก่อน-หลังเป็นลำดับที่ดีเพื่อลดความซ้ำซ้อน และความสับสนของข้อมูล

การจัดลำดับก่อน-หลังใน CAN Message นั้นใช้หลักในการตัดสินใจโดยใช้ตัวระบุ (Identifier) ซึ่งตัวระบุ (Identifier) จะมีการกำหนดค่าในระหว่างช่วงเริ่มต้น (Initial Phase) ของตัวระบุ (Identifier) จะมีค่าไม่มากแต่จะมีการกำหนดสิทธิ์ลำดับการมาก่อน-หลังของข้อความ(Message) ที่ดีมากที่สุด

2.4.5.5 การหยุดพักหรือการเริ่มต้นทำงานใหม่ (Sleep Mode / Wake-up)

เพื่อช่วยในการลดการสิ้นเปลืองพลังงานในการทำงาน ในกรณีที่โหนด (nodes) ไม่ได้ทำการติดต่อกับระบบ ณ เวลานั้นๆ หรือไม่ได้มีการทำงานภายในโหนด (nodes) นั้นๆ โหนด (nodes) นั้นๆ จะเข้าสู่สภาวะหยุดพัก (Sleep Mode) และการหยุดการติดต่อกับระบบบัส (Bus) สภาวะการหยุดพักจะตื่นตัว (Wake – up) เมื่อมีการกระตุ้นจากระบบบัส (Bus) หรือเงื่อนไขอื่นๆในโหนด (nodes) ที่สั่งขึ้นเอง

2.4.5.6 รูปแบบของการส่งข้อความของ CAN (Control Area Network)

ในระบบ CAN ข้อมูลการสื่อสาร และการรับส่ง ใช้รูปร่างคำสั่ง (Message Frame) รูปร่างคำสั่ง (Message Frame) ดังแสดงในรูปที่ 2.13 จะนำเอาข้อมูลจากโหนด (nodes) ไปสู่โหนด (nodes) อื่นๆ

มาตรฐานของ CAN 2.0A ทำให้ทราบว่า Base Frame Format ซึ่งสนับสนุนการทำงานของข้อความ (Message) ขนาด 11 บิต CAN ที่ขยายออกจาก CAN 2.0A คือ CAN 2.0B ซึ่งข้อความ (Message) ขนาด 29 บิต ทำให้ทราบว่า Extended Frame Format สนับสนุนการทำงานของข้อความ (Message) ทั้งสองแบบคือแบบ CAN2.0A และ CAN 2.0B

SOF	Identifier	RTR	Control	Data	CRC	ACK	EOF
-----	------------	-----	---------	------	-----	-----	-----

ตารางที่ 2.1 CAN Message Frame

2.4.6 ข้อผิดพลาดที่เกิดขึ้น

CAN 2.0B สามารถที่จะส่งข้อมูลผ่านไปยัง CAN 2.0A เพราะมีความเข้ากันได้ของระบบ แต่อย่างไรก็ตาม การส่งสัญญาณหรือการสื่อสารระหว่าง CAN 2.0B ไปยัง CAN2.0A แบ่งเป็น 2 กรณี คือ

- กรณีที่ 1 เมื่อมีการส่งข้อมูล CAN 2.0B ไปยัง CAN 2.0A จะเกิดความผิดพลาด
 กรณีที่ 2 เมื่อมีการส่งข้อมูลผ่านระหว่าง CAN2.0B ไปยัง CAN 2.0A เกิดความผิดพลาด (Error) แต่จะไม่มีผลต่อการทำงานเนื่องจากในส่วนของ CAN 2.0A จะไม่สนใจในความผิดพลาด (Error) ที่เกิดขึ้นในกรณีนี้

2.4.6.1 การตรวจจับข้อผิดพลาด (Error Detection)

CAN จะแบ่งการตรวจจับข้อผิดพลาดออกเป็น 5 อย่าง ซึ่งจะจัดอยู่ใน 2 หมวดคือ

1. การตรวจสอบข้อความ (Message)

1.1 CRC (Cyclic Redundancy Check) ซึ่งทุกๆ ข้อความที่ถูกส่งไป จะประกอบด้วย CRC (Cyclic Redundancy Check) ซึ่งจะเกิดจากการเข้ารหัสข้อความ เพื่อเป็นการตรวจเช็คเมื่อถึงผู้รับจะมีการเข้ารหัสข้อความ เพื่อเช็คกับค่าของ CRC (Cyclic Redundancy Check) เพื่อตรวจสอบว่าตรงกันหรือไม่ ถ้าไม่ตรงกันแสดงว่าเกิดการผิดพลาดของข้อมูล

1.2 Frame Check คือ มีการเช็คที่บิตของข้อมูลทั้งเฟรม (Frame) ว่าถูกต้องเหมือนตอนที่ส่งมาหรือไม่ถ้าไม่ถูกต้อง แสดงว่าเกิดผิดพลาดหรือข้อมูลเสียหายกับรูปร่างคำสั่งนั้นๆ

1.3 ACK (Acknowledgement Error Check) จะมีการตรวจสอบที่ผู้รับจะเช็คบิตในจุด ACK (Acknowledgement) ว่ามีการเปลี่ยนแปลงหรือไม่ ถ้ามีการเปลี่ยนแปลงได้ค่าไม่ตรงแสดงว่าเกิดข้อผิดพลาด

2. หมวดการตรวจสอบบิต (bit)

2.1 Bit Monitoring จะทำการตรวจสอบทุกๆบิตว่าบิตที่ส่งออกไปถูกต้องหรือไม่

2.2 Bit Stuffing จะมีการแทรกบิตเข้าไป และจะมีการเช็คที่ปลายทางว่าบิตที่แทรกเข้าไปว่าตรงกับที่ใส่ไปหรือไม่ ถ้าไม่แสดงว่าเกิดข้อผิดพลาด (Error)

2.4.6.2 ขอบเขตข้อผิดพลาดที่เกิดขึ้น (Error Confinement)

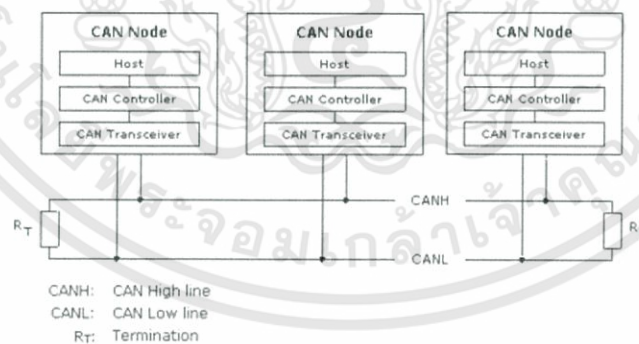
1. Error Counts จะมีการนับเรื่อย ๆ เมื่อมีข้อผิดพลาด (Error) และจะกลับคืนค่าเป็นศูนย์เมื่อหาย
2. Error Active Mode เมื่อมีการนับข้อผิดพลาดได้ไม่เกิน 127
3. Error Passive Mode เมื่อมีการนับข้อผิดพลาดได้เกิน 127 ขึ้นไป
4. Bus Off Mode เมื่อมีการนับข้อผิดพลาดได้เกิน 255 จะทำการปิดการทำงานของระบบบัส (System Bus) ข้อมูล

2.4.6.3 การจัดการเมื่อตรวจพบข้อผิดพลาด (Performance of Error Detection)

เมื่อตรวจพบข้อผิดพลาด (Error) ระบบจะทำการหยุดการรับข้อความ (Message) และระบบจะทำการส่งข้อความ (Message) นั้นใหม่อีกครั้งในทันทีและก็จะตรวจสอบข้อผิดพลาดถ้าไม่มีระบบก็จะดำเนินต่อไป แต่ถ้าเกิดข้อผิดพลาดระบบก็จะทำการส่งใหม่อีกครั้ง

2.4.7 ข้อดีของCAN (Control Area Network)

CAN (Control Area Network) หรือ CAN bus คือการสื่อสารแบบต่อเนื่องสำหรับควบคุมอุปกรณ์ต่างๆ แบบทันทีทันใด (Real Time) ซึ่ง CAN (Control Area Network) สามารถจัดการกับอัตราการขนถ่ายข้อมูลได้สูงสุดถึง 1 เมกกะบิตต่อวินาที มีความสามารถในการตรวจจับข้อผิดพลาด (Error Detection) ได้อย่างมีประสิทธิภาพและช่วยในการประหยัดค่าใช้จ่าย ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 การสื่อสารของแต่ละโหนดบนระบบบัส

2.5 J2ME (Java 2 Platform, Micro Edition)

J2ME (Java 2 Platform, Micro Edition) เป็นเทคโนโลยีที่ทำให้โปรแกรมเมอร์สามารถใช้จาวาและเครื่องมือที่เกี่ยวข้องในการพัฒนาโปรแกรมสำหรับอุปกรณ์ไร้สาย (Virtual Machine) ขนาดเล็กที่เรียกว่า KVirtual Machine ทำให้โปรแกรมที่พัฒนาขึ้นโดยJ2MEสามารถรันบนอุปกรณ์ไร้สายได้ สามารถแบ่งส่วนการทำงานเป็น 2 ส่วนคือ CLDC (Connected, Limited Device Configuration) และ MIDP (Mobile Information Device Profile) โดยจะมีการแบ่งหน้าที่กันทำงานโดย CLDC ดูแลเกี่ยวกับ API และคุณสมบัติของ Virtual Machine ที่จำเป็นต้องใช้ในการรันโปรแกรมบนอุปกรณ์ไร้สาย และสำหรับ MIDP จะเป็นส่วนที่เสริม CLDC ดูแลเรื่องส่วนติดต่อผู้ใช้(UI) และ เน็ตเวิร์ค และคำสั่งที่ติดต่อกับ อุปกรณ์ไร้สาย MIDP ยังครอบคลุมถึงแนวคิดของ MIDlet หรือโปรแกรมจาวาขนาดเล็กที่คล้ายกับ applet แต่ใช้คุณสมบัติของ CLDC และ MIDP และคิดค้นขึ้นเพื่อจุดประสงค์ของงานอุปกรณ์ไร้สาย เท่านั้น

อุปกรณ์ที่สามารถใช้เทคโนโลยี J2ME ได้แบ่งออกเป็น 2 กลุ่มดังนี้ คือ Personal, mobile, connect, information device เช่น โทรศัพท์มือถือ และเพจเจอร์ เป็นต้น ซึ่งเป็นอุปกรณ์ที่มีการอินเตอร์เฟซแบบง่าย ๆ มีหน่วยความจำประมาณ 128-512 กิโลไบต์และมีแบนด์วิดท์ที่ต่ำ ส่วนการเชื่อมต่อของอุปกรณ์ไม่ได้ใช้โปรโตคอลTCP/IP และ Shared, fixed, connected, information device เช่น อุปกรณ์เซตท็อป (set-top boxes) อินเทอร์เน็ตทีวี ความบันเทิงในรถยนต์และระบบการเดินทาง เป็นต้น อุปกรณ์จำพวกนี้มียูสเซอร์อินเตอร์เฟซให้ใช้หลายแบบมีหน่วยความจำประมาณ 2-16 เมกกะไบต์ ส่วนการเชื่อมต่อกับระบบเครือข่ายนั้น โดยมากแล้วใช้โปรโตคอล TCP/IP ที่มีแบนด์วิดท์ค่อนข้างสูง

จุดมุ่งหมายที่สำคัญของการออกแบบภาษาจาวาคือ โปรแกรมต้องทำงานบนเครื่องต่างระบบกัน ได้ โดยเรียกคุณสมบัตินี้ว่า “ไม่ขึ้นกับระบบ” ซึ่งสามารถทำงานได้ทุกระบบปฏิบัติการ

2.5.1 โครงสร้างสถาปัตยกรรมของ J2ME

การออกแบบโครงสร้างเทคโนโลยีจาวา ตระกูล J2ME ได้มีการจัดแบ่งหน้าที่การทำงาน ออกเป็น 4 ระดับชั้นด้วยกัน เพื่อรองรับกับการใช้งานกับอุปกรณ์ต่างๆ คือ Operating System Java Virtual Machine Layer Configuration และ Layer Profile Layer

ระดับชั้นที่หนึ่ง Host Operating System จะเป็นส่วนของระบบปฏิบัติการ เช่นเดียวกับบนเครื่อง Desktop หรือ Laptop ที่มี Window เป็นระบบปฏิบัติการ บนโทรศัพท์มือถือก็มี ด้วยเช่นกัน อาทิ Nokia 7650, 3650 จะมี Symbian OS เป็นระบบปฏิบัติการ, เครื่อง Plamจะมี Plam OS เป็นระบบปฏิบัติการ

ระดับชั้นที่สอง Java Virtual Machine(JVM) จะเป็นส่วนของระบบจัดการ ที่ควบคุม และทำงาน ให้สามารถทำงานร่วมกันได้ ระหว่าง Java กับ Host Operating System โดยมากจะเป็น การแปลงจาก code Java ไปเป็นคำสั่ง ที่ Host Operating System เข้าใจ และทำงานร่วมกันได้

ระดับชั้นที่สาม Configuration เป็นกลุ่มของ Class Library (คลัง Class) ที่ครอบคลุม ถึงอุปกรณ์ต่างๆที่อยู่ในกลุ่ม

ระดับชั้นที่สี่ Profile เป็นกลุ่มของคำสั่ง, API (Application Programming Interface) ที่ใช้สำหรับอุปกรณ์ แต่ละประเภทโดยเฉพาะ

หัวใจหลักของโครงสร้าง J2ME สามารถแบ่งออกเป็น 2 ส่วน คือ Configurations และ Profile ซึ่งมีลักษณะเป็นโมดูลที่มีความยืดหยุ่นพร้อมกับ สามารถนำไปใช้งานกับอุปกรณ์ที่ต่างกันได้ โดยสามารถปรับแต่งได้ตามความต้องการของผู้ใช้งานและนอกเหนือจากนี้ผู้ผลิต หรือ ผู้พัฒนาซอฟต์แวร์ บนอุปกรณ์ ยังสามารถขยายหรือเพิ่มความสามารถในเวอร์ชวลแมชีนได้ ดังนั้นจึงไม่ได้เป็นแพลตฟอร์ม เฉพาะกับอุปกรณ์ชนิดใดชนิดหนึ่ง

2.5.1.1 การตั้งค่า (Configurations)

สำหรับการตั้งค่าของ J2ME ที่ใช้งานกับอุปกรณ์ใดๆ ต้องสามารถทำงานร่วมกับมาตรฐานที่กำหนดไว้ในกลุ่ม JCP และต้องสามารถใช้งาน Runtime Classes ตามข้อ กำหนดได้ นอกจากนี้การตั้งค่ายังเป็นตัวกำหนดฟีเจอร์หรือไลบรารีมาตรฐาน ซึ่งจะมีเหมือนกันในทุกอุปกรณ์ที่จัด อยู่ในประเภทเดียวกัน โดยการจัดแบ่งกลุ่มคอนฟิกูเรชันปัจจุบันมี 2 ประเภท คือ Connected Device Configuration (CDC) และ Connected Limited Device Configuration (CLDC) โดยแต่ละตัวจะใช้ VM (Virtual Machine) ที่ต่างกันด้วย โดยคุณสมบัติพื้นฐานทั้งหมดนี้จะใช้ในการจัดแบ่งกลุ่มของอุปกรณ์ ซึ่งมีความคล้ายกันในเรื่องของหน่วยความจำ หน้าจอ เครือข่ายในการเชื่อมต่อและพลังงาน การ เปรียบเทียบความแตกต่างระหว่าง CDC และ CLDC สามารถดูได้จากตารางที่ 2.2 ด้านล่าง

ตารางที่ 2.2 ตารางเปรียบเทียบระหว่าง CDC และ CLDC

CDC	CLDC
ทุกๆส่วนของ features และ APIs ของ Java	บางส่วนของ features และ APIs ของ Java
สำหรับอุปกรณ์ ในกลุ่ม High-end devices	สำหรับอุปกรณ์ในกลุ่ม Low-end devices
หน่วยประมวลผล 32 bit	หน่วยประมวลผล 16 และ 32 bit
หน่วยความจำเครื่องอย่างน้อย 2 MB .	หน่วยความจำเครื่อง 160-512 KB
ใช้ CVM (Compact Virtual Machine)	ใช้ KVM (Kilo Virtual Machine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 Profiles

เป็นหัวใจหลักอันสำคัญของเทคโนโลยี J2ME เนื่องจากลักษณะของการใช้งานในแต่ละงานนั้นแตกต่างกัน โดยการสร้างไว้ยู่เหนือระดับการตั้งค่าจึงเป็นตัวกำหนดฟังก์ชันในการทำงานพร้อมทั้งเป็นตัวกลางระหว่างแอปพลิเคชันกับสภาวะแวดล้อมของ J2ME ดังนั้น Profile จึงเกี่ยวข้องกับคุณลักษณะทางด้านฮาร์ดแวร์ของอุปกรณ์แต่ละตัว เช่น Profile ของอุปกรณ์ประเภทมือถือ (Mobile Information Device Profile) สำหรับชุดของ API (Application Programming Interface) นี้ใช้สำหรับงานในแต่ละ vertical market ส่วนผู้ใช้งานคอมพิวเตอร์ โน้ตบุ๊ก คือ Input/Output, Even handling, Persistent storage, Networking และ Timers

Profile เป็นส่วนของ API และ Class ที่ใช้งานได้บนตัวอุปกรณ์แต่ละประเภท เป็นการขยายความสามารถของ Connected Device Configuration หรือ Connected, Limited Device Configuration มากขึ้น และมีส่วนของการทำงานที่เป็นลักษณะเฉพาะของอุปกรณ์นั้นๆ (สามารถดูการจับคู่ระหว่าง Profile และ Configuration ได้ตามตารางที่ 2.3 ด้านล่าง) ตัวอย่างของ Profile ที่ใช้งานการพัฒนา เช่น MIDP (Mobile Information Device Profile) เป็นประเภทของ Device ที่มีคุณสมบัติต่อไปนี้ small display (min. 96 x 54 pixels), มี touch screen หรือ keypad, สามารถ connect mobile network ด้วย bandwidth ที่จำกัด MIDP ประกอบด้วย APIs ที่ทำหน้าที่ดังต่อไปนี้

- User Interface จัดการเกี่ยวกับการแสดงผล
- Persistent storage จัดการเกี่ยวกับการเก็บข้อมูลและฐานข้อมูล
- Network จัดการเกี่ยวกับการเชื่อมต่อเน็ตเวิร์ค
- Application life-cycle จัดการเกี่ยวกับลำดับขั้นตอนการทำงาน
- Event handling จัดการเกี่ยวกับอีเวนต์ต่างๆ
- Mobile Information Device Next Generation(MIDP_NG)เป็นGeneration ที่จะพัฒนาเป็นตัวถัดไปของ MIDP ซึ่งจะเพิ่มคุณสมบัติต่างๆดังนี้
- Domain security module
- HTTPS Support
- Socket, datagrams support
- มี OTA Provisioning
- Push architecture โดยข้อความหรือเหตุการณ์ภายนอกถูกส่งไป MIDlet
- เพิ่มความสามารถในการ Control large screen
- มี XML parser

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มี Sound API
- Personal Digital Assistant Profile (PDA Profile) สำหรับอุปกรณ์
- ประเภท Organizer
- Foundation Profile สำหรับอุปกรณ์ในกลุ่มของ High-end device, เป็นส่วนขยายเพิ่มเติมเฉพาะด้านให้กับ CDC ซึ่งจะประกอบด้วย API และ ฟังก์ชันพื้นฐาน เป็น Profile ที่พัฒนามาบน CDC เหมาะสำหรับ Device ที่มีคุณสมบัติลักษณะต่อไปนี้
- 1024k minimum ROM
- 512k minimum RAM
- Connectivity กับระบบ เน็ตเวิร์คได้
- ไม่มี GUI ยกเว้นจะใช้ additional profile อื่นๆ เพิ่มเติมเข้ามาเพื่อทำ GUI
- Personal Profile สำหรับอุปกรณ์ในกลุ่มของ High-end device, เป็นส่วนขยายเพิ่มเติมเฉพาะด้านให้กับ Foundation Profile จะประกอบด้วยการจัดการด้าน GUI
- RMI Profile สำหรับอุปกรณ์ในกลุ่มของ High-end device เป็นส่วนขยายเพิ่มเติมเฉพาะด้านให้กับ Foundation Profile จะประกอบด้วย การจัดการด้าน RMI (Remote Method Invocation)

ตารางที่ 2.3 ตารางแสดง Profile และ Configuration

Profile	Configuration
MID	CLDC
Foundation	CDC
Personal	CDC
PDA	CLDC
Bluetooth	CLDC
RMI	CDC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 ประโยชน์ของ J2ME ในการพัฒนาแอปพลิเคชันบนอุปกรณ์ไร้สาย

1. เป็นแพลตฟอร์มอิสระ เนื่องจากอุปกรณ์ไร้สายแต่ละชนิดนั้นมีขนาดและหน้าที่การทำงานที่แตกต่างกัน แอปพลิเคชันใช้งานที่ติดตั้งพร้อมกับอุปกรณ์สื่อสารไร้สาย ส่วนใหญ่ได้รับการพัฒนาโดยใช้ไลบรารีเฉพาะ โดยระบบที่ผู้ผลิตจัดเตรียมไว้ให้ภาษาที่ใช้ในการเขียนแอปพลิเคชันมีหลากหลาย ภาษา ตั้งแต่ C++ , Virtual Basic ไปจนถึงภาษาที่มีในสคริปต์เฉพาะตัวแอปพลิเคชันที่เขียนขึ้นเพื่อใช้กับอุปกรณ์ชนิดใดชนิดหนึ่งไม่สามารถนำไปใช้กับอุปกรณ์อื่นๆได้ ในขณะที่ J2ME นั้นนำปรัชญาเก่าแก่ในการออกแบบโปรแกรม โดยเขียนเพียงครั้งเดียวแล้วนำไปใช้ที่ใดก็ได้ (WriteOnce, Run Anywhere) มาสู่โลกอุปกรณ์ไร้สาย แอปพลิเคชันบนอุปกรณ์ไร้สายที่พัฒนาด้วยจาวาสามารถทำงานบนอุปกรณ์ที่ผลิตโดยผู้ผลิตรายอื่นได้ ช่วยให้เคลื่อน ย้ายไปใช้งานอื่นได้สะดวกยิ่งขึ้น

2. สามารถเขียนแอปพลิเคชันด้วยภาษาที่เข้าใจง่าย โดยเทคโนโลยีจาวาช่วยประหยัดเวลาและต้นทุนในการพัฒนาแอปพลิเคชัน ทำให้ผลงานที่ได้มีประสิทธิภาพมากขึ้นในตลาดที่มีการแข่งขันสูง และก้าวหน้าอย่างรวดเร็วประกอบด้วยฟังก์ชันด้านเครือข่ายจำนวนมาก แอปพลิเคชันบนอุปกรณ์ไร้สายจะทำงานเกี่ยวข้องกับระบบเครือข่าย เพื่อให้ผู้ใช้งานสามารถติดต่อสื่อสารกับโลกภายนอกได้จากทุกๆที่ในทันที จาวานั้นได้รับการออกแบบให้มีความสามารถด้านเครือข่ายในตัวเอง โดยมีไลบรารีระบบเครือข่ายมากมายทำให้การเขียนแอปพลิเคชันด้านเครือข่ายทำได้ง่ายกว่าเดิม

3. มีระบบรักษาความปลอดภัย ภาษาจาวามีระบบรักษาความปลอดภัยหลายระดับ ตั้งแต่ class loader byte code verifier ไปจนถึงโปรแกรมรักษาความปลอดภัย เช่น Security Manager ซึ่งสามารถป้องกัน client จากแอปพลิเคชันที่ไม่น่าไว้วางใจ นอกจากนี้ยังมี APIs รักษาความปลอดภัยการรับส่งข้อมูลผ่านเว็บอีกด้วย วิธีนี้ช่วยให้การทำธุรกรรมการค้าและการเงินผ่านโทรศัพท์เคลื่อนที่มีความปลอดภัยมากยิ่งขึ้น

4. สามารถติดตั้งแอปพลิเคชันได้เองโดยอุปกรณ์ไร้สายส่วนใหญ่จะติดตั้งแอปพลิเคชันมาพร้อม และมีคุณสมบัติการใช้ที่มีจำกัด ยากที่จะอัปเดตและลงแอปพลิเคชันใหม่ๆ โดยปราศจากความช่วยเหลือจากบริษัทผู้ผลิต J2ME มีระบบกลไกติดตั้ง ซึ่งยินยอมให้ดาวน์โหลดแอปพลิเคชันผ่านระบบสื่อสารไร้สายแล้วนำมาติดตั้งบนอุปกรณ์คล้ายกับการเรียกโปรแกรมจาวาแอปเพล็ต กลไกการติดตั้งนี้ไม่เพียงช่วยให้นักผลิตและนักพัฒนาแอปพลิเคชันประหยัดต้นทุน ในการแจกซอฟต์แวร์ แต่ยังช่วยให้ผู้ใช้งานสามารถดาวน์โหลดและปรับแต่งแอปพลิเคชันบนอุปกรณ์ได้ตามต้องการ

5. เป็นแอปพลิเคชันที่แพร่กระจายไปทั่วทุกพื้นที่ จาวาได้รับความนิยมอย่างสูงเมื่ออินเทอร์เน็ตกลายเป็นแพลตฟอร์มหลักสำหรับแอปพลิเคชัน จาวาถือเป็นตัวเลือกในการพัฒนาแอปพลิเคชัน นอกจากนี้แอปพลิเคชันที่พัฒนาด้วย J2ME ยังสามารถประสานเข้ากับแอปพลิเคชันที่

พัฒนาด้วย J2EE ได้ง่ายช่วยสนับสนุนการออกแอปพลิเคชันบนอุปกรณ์ไร้สายสำหรับองค์กร โดยแอปพลิเคชันบนอุปกรณ์ไร้สายมักเป็น Client ขนาดเล็ก การที่ XML ได้รับการสนับสนุนในจาวาทำให้แอปพลิเคชันแบบ client/server หรือแอปพลิเคชันติดต่อสื่อสารระหว่างกันเกิดขึ้นได้ในอุปกรณ์ไร้สาย

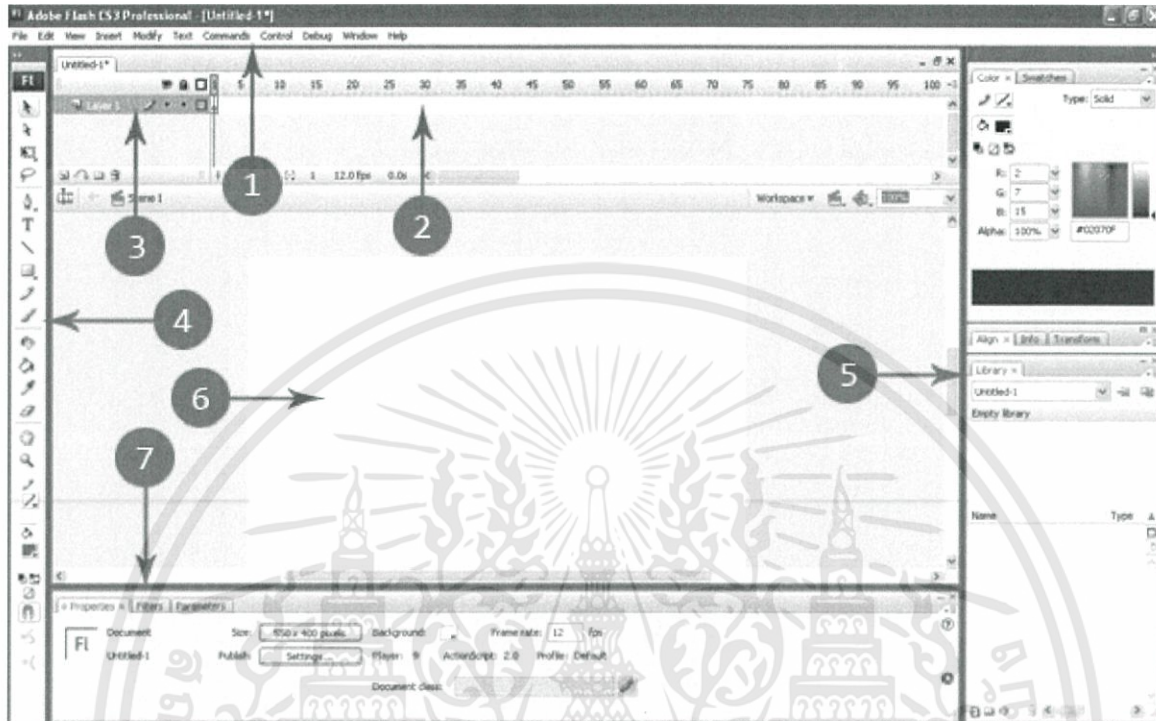
6. มีส่วนติดต่อกับผู้ใช้แบบกราฟฟิก (GUI : Graphic User Interface) ใน J2ME มีลักษณะเดียวกับใน J2SE คือ มีส่วนติดต่อกับผู้ใช้ที่หลากหลายและมีคลาสไลบรารีรองรับอีเวนต์ต่างๆ โดยใช้พื้นที่การแสดงผลหน้าจอบนอุปกรณ์ไร้สายให้เกิดประโยชน์สูงสุด โดยส่วนติดต่อกับผู้ใช้นี้ทำให้อุปกรณ์ไร้สายสามารถสนับสนุนการทำงานของวิดีโอเกมส์ขั้นสูง และแอปพลิเคชันเพื่อความบันเทิงที่ซับซ้อนได้

2.6 Adobe Flash CS3

โปรแกรมนี้เป็นซอฟต์แวร์ที่ช่วยในการสร้างสื่อมัลติมีเดีย, กราฟิกสำหรับงานเว็บ ผลงานที่พัฒนาด้วย Flash มีทั้งสื่อภาพนิ่ง ภาพเคลื่อนไหว สื่อมัลติมีเดีย ตลอดจนสื่อที่มีระบบโต้ตอบกับผู้ใช้ (Interactive Multimedia) เช่น การสอนการกดปุ่มสั่งงาน หรือการป้อนข้อมูลของผู้เข้าชมเว็บไซต์ได้อีกด้วย จุดเด่นของโปรแกรมนอกจากจะสร้างภาพที่มีความคมชัดสวยงามแล้ว ยังใช้สร้างภาพกราฟิกแบบเวกเตอร์ซึ่งเป็นภาพลายเส้น ซึ่งแตกต่างจากการสร้างภาพแบบบิตแมพตรงที่ไฟล์ภาพที่ได้จะมีขนาดขนาดเล็กมาก และเมื่อขยายขนาดของภาพให้ใหญ่ขึ้น ภาพก็ยังคงความคมชัดเหมือนเดิมทุกประการ ด้วยคุณสมบัติเช่นนี้จึงนิยมนำไฟล์ที่ได้ไปใช้บนเว็บ เนื่องจากสามารถโหลดผ่านเว็บเบราว์เซอร์ได้รวดเร็ว มีความคมชัดสูงแม้ว่าจะถูกขยายขนาด ทั้งนี้สามารถนำเสนอได้ทั้งบนเว็บ หรือผ่านโปรแกรม Flash Player หรือสร้างเป็น exe file เพื่อเรียกใช้งานได้ทันที นอกจากนี้ยังสามารถแปลงไฟล์ไปอยู่ในฟอร์แมตอื่นได้ด้วย เช่น Animation Gif, AVI, QuickTime

Adobe Flash CS3 เป็นโปรแกรมกราฟิกและมัลติมีเดีย ที่มีการพัฒนามาเป็นเวลานาน ตั้งแต่ยังเป็น Macromedia Flash ในเวอร์ชัน 3, 4, MX, MX 2004 และ Flash 8 ของบริษัท Macromedia แต่ปัจจุบันถูกซื้อกิจการมารวมอยู่ในแพ็คเกจ Adobe Design Premium CS3 ของบริษัท Adobe ซึ่งได้นำมาพัฒนาเพิ่มเติมให้มีความสามารถสูงขึ้น และรองรับการใช้งานได้ทั้งระบบปฏิบัติการ Window XP และ Window Vista

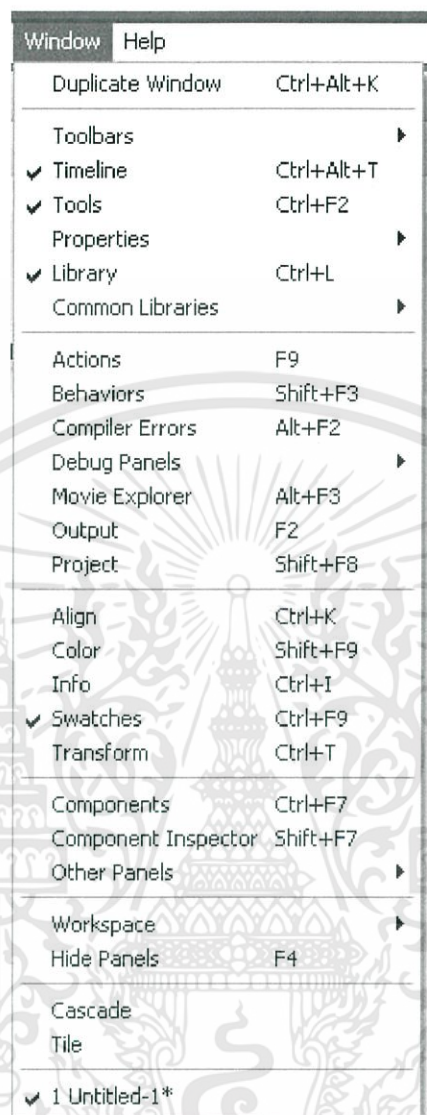
2.6.1 ส่วนประกอบของโปรแกรม Adobe Flash CS3



รูปที่ 2.13 แสดงส่วนประกอบของโปรแกรม Adobe Flash CS3

1. Menu Bar (เมนูบาร์) ซึ่งประกอบด้วยเมนูหลายอย่างที่จำเป็นในการทำงาน เช่น เมนู Window มีสำหรับแสดง และ ซ่อน เครื่องมือทุกชนิด หน้าต่างเครื่องมือที่หายไปสามารถมาสั่งเรียกเปิดที่นี่

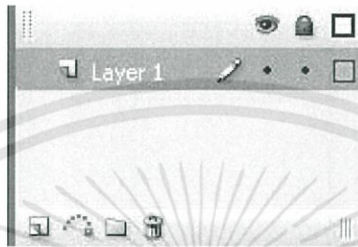
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 Time Line ส่วนควบคุมและกำหนดการนำเสนอผลงาน

2. Time Line (ทามไลน์) มีไว้สำหรับควบคุม และ กำหนดการนำเสนอผลงาน การเคลื่อนไหวต่างๆ โดยจะประกอบ ด้วยเฟรม (frame) ซึ่งจะสามารถบรรจุสิ่งต่างๆไม่ว่าจะเป็น ภาพ เสียง และอื่นๆ ลงไปเพื่อให้ในช่วง เวลานั้นประกอบด้วยอะไรบางอย่างที่ต้องการแสดงผล เวลาสร้างงานจะทำการตั้งค่างานที่ตรงนี้

3. Layer (เลเยอร์) คือชั้นต่างๆ ที่จะทำให้สามารถทำงานเป็นระบบมากขึ้น การแยกวัตถุต่างๆ ออกจากกันให้เป็นอิสระก็สามารถทำได้ ซึ่งทำให้สะดวกและง่ายต่อการทำงาน โดยที่เลเยอร์วัตถุที่อยู่ ด้านบนนั้นจะทับวัตถุเลเยอร์ด้านล่าง หากต้องการสลับบน-ล่าง ก็เพียงให้คลิกค้างไว้ที่แถบเลเยอร์ แล้วลากขึ้นหรือลง



รูปที่ 2.15 แสดงส่วนของ Layer

นอกจากนั้นแล้ว ในการทำงานกับเลเยอร์ที่ดีควรจะต้องทำการตั้งชื่อของเลเยอร์นั้นๆไว้ เพื่อให้สามารถกลับมาแก้ไขสิ่งต่างๆ ในเลเยอร์นั้นได้ง่ายและสะดวก ไม่เช่นนั้นแล้วหากทำงานที่ต้องมีเลเยอร์มากๆ เราจะต้องหาจนปวดหัว และการตั้งชื่อเลเยอร์ ก็ควรจะสื่อกับวัตถุหรืองานต่างๆที่อยู่ในเลเยอร์นั้น เช่น เลเยอร์นั้นเป็นส่วนของพื้นหลัง เราก็ควรจะตั้งชื่อว่า background หรือ bg เป็นต้น

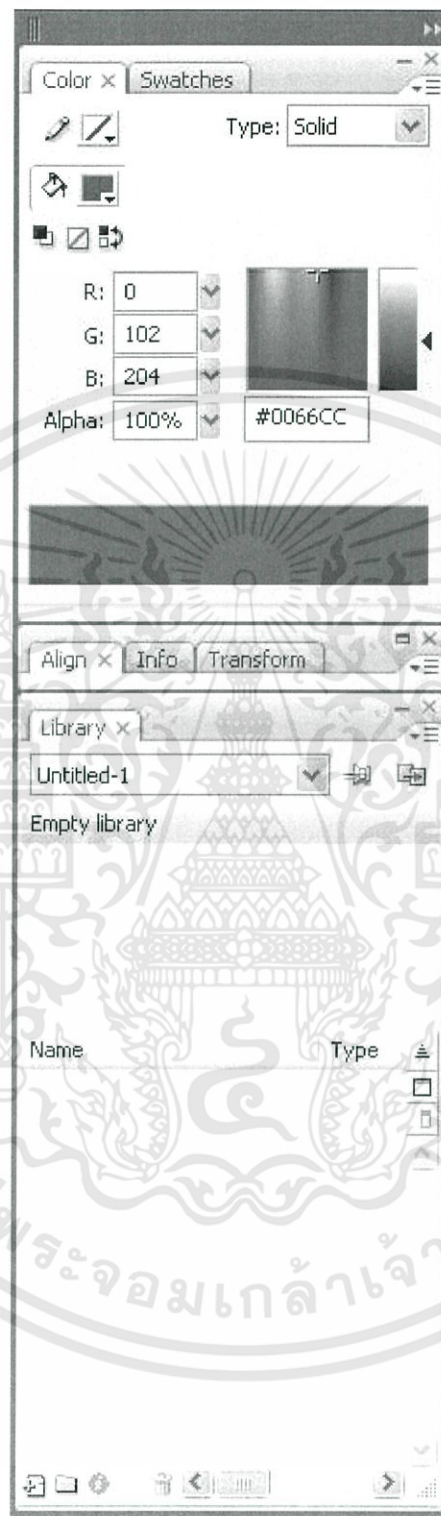
4. Tool Bar หรือแถบเครื่องมือ กลุ่มเครื่องมือสร้างงานและจัดการวัตถุ ประกอบด้วย ปุ่มเครื่องมือย่อยต่างๆ โดยแบ่งเครื่องมือเป็นหมวดๆ ได้ 5 กลุ่ม

- เครื่องมือเลือกวัตถุ (Selection)
- เครื่องมือวาดภาพ (Drawing)
- เครื่องมือจัดแต่งวัตถุ (Modify)
- เครื่องมือควบคุมมุมมอง (View)
- เครื่องมือควบคุมสี (Color)



รูปที่ 2.16 แสดงแถบเครื่องมือ Tool Bar

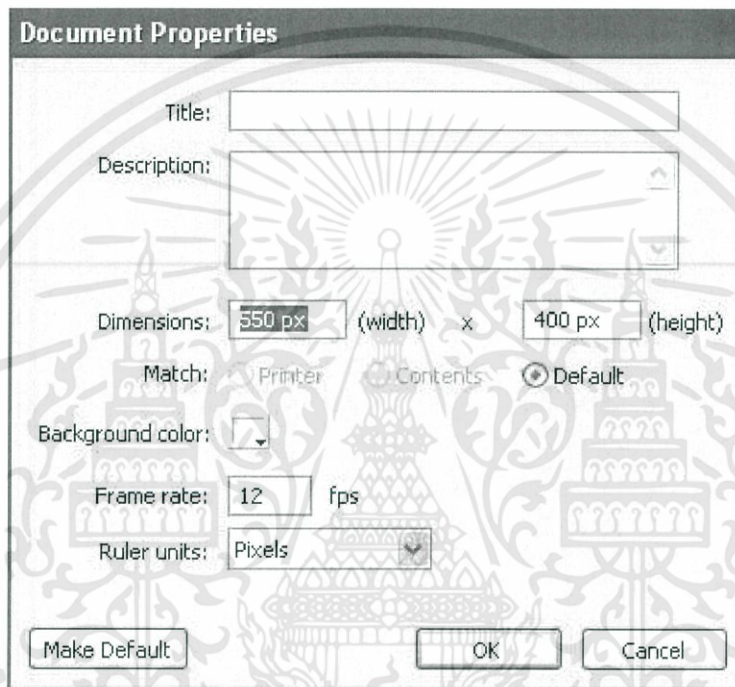
5. Panel (พาเนล) หน้าต่างที่ทำหน้าที่แสดง พาเนลย่อย ของโปรแกรม โดยในแต่ละพาเนลย่อย ก็จะมีประกอบด้วยรายละเอียดของการควบคุมการแสดงผลการปรับแต่งไว้ในตัว เช่น swatches ก็จะมีสีต่างๆ มากมายให้เลือกใช้ในการปรับแต่งสี



รูปที่ 2.17 แสดงแถบ Panel

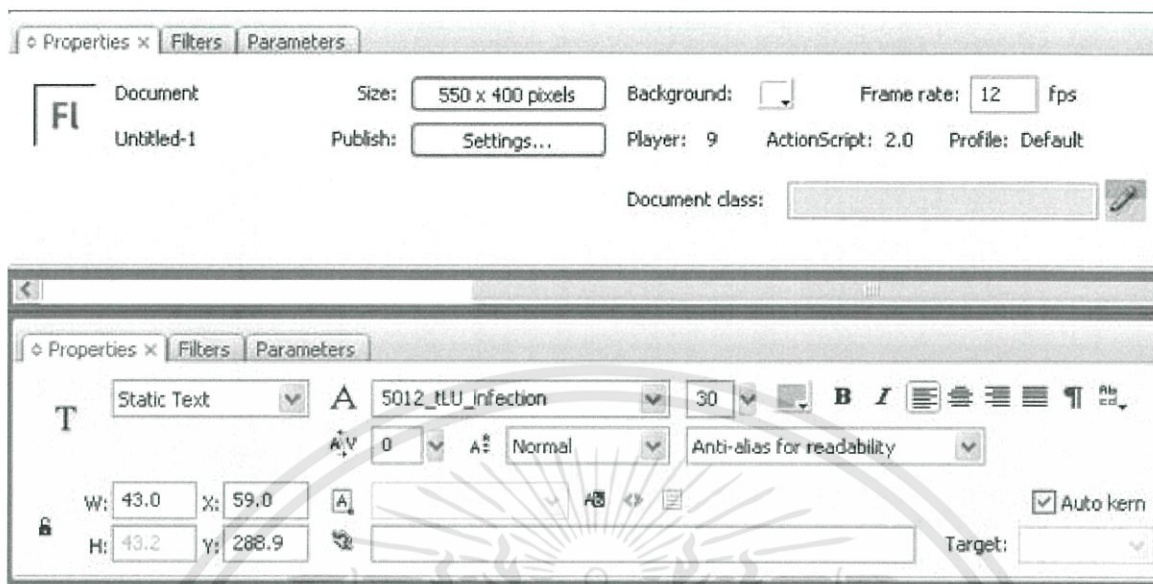
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Stage (สแตจ) คือพื้นที่ที่เรากำหนดขอบเขตขนาดของการทำงาน ซึ่งสามารถตั้งค่าหรือปรับได้ที่ Properties โดยกดที่ปุ่มที่มีค่า 550x400 จะได้หน้าต่างขึ้นมาในหน้าต่างนี้สามารถตั้งชื่อของงาน ระบุเนื้อหารายละเอียด ความกว้างและสูง สีพื้นหลัง frame rate ได้ตามความเหมาะสมกับงาน โดยก่อนเริ่มทำงานควรวางแผนในการกำหนดขนาดไว้ก่อนจะดี เพราะหากมาแก้ไขขนาดภายหลังจะทำให้ยุ่งยากในการปรับตำแหน่งของวัตถุต่าง



รูปที่ 2.18 แสดงแถบ Stage

7. Properties ไว้สำหรับกำหนดคุณสมบัติให้กับพื้นที่การทำงานและสิ่งต่างๆที่จะใช้งาน หากว่าต้องการปรับปรุง เปลี่ยนแปลง แก้ไขวัตถุไหนก็นำเมาส์ไปคลิกที่วัตถุนั้น ค่าที่ส่วน Properties ก็ จะเปลี่ยนแปลงไป ตามวัตถุนั้น เช่น คลิกที่ตัวอักษร ก็จะสามารถเปลี่ยนเรื่องฟอนท์ , สี , ขนาด และอื่นๆ



รูปที่ 2.19 แสดงแถบ Properties

2.7 การสร้างเว็บเพจด้วยภาษา HTML

HTML ย่อมาจากคำว่า "HyperText Markup Language" เป็นภาษาที่ใช้ในการเขียน โปรแกรม ภาษาหนึ่งของคอมพิวเตอร์ ที่แสดงผลในลักษณะของเว็บเพจ ซึ่งสามารถแสดงผลได้ใน รูปแบบต่างๆ ไม่ว่าจะเป็นภาพกราฟิก ภาพนิ่ง ภาพเคลื่อนไหว เสียง หรือการเชื่อมโยงไปยังเว็บไซต์ อื่นๆ ภาษา HTML เป็น ภาษาที่มีลักษณะของโปรแกรม กล่าวคือ จะเป็นไฟล์ที่เก็บข้อมูลที่เป็นตัวอักษรใน มาตรฐานของรหัสแอสกี (ASCII Code) โดยเขียนอยู่ในรูปแบบของเอกสารข้อความ จึงสามารถ กำหนดรูปแบบ และ โครงสร้างได้ง่าย

2.7.1 ขั้นตอนในการสร้างเว็บเพจ

- เปิดโปรแกรม TextEditor แล้วทำการพิมพ์คำสั่ง HTML แล้วบันทึกเป็นไฟล์นามสกุล .htm หรือ .html
- เปิดโปรแกรม WebBrowser เพื่อใช้ในการดูผลลัพธ์ที่ได้จากการเขียนภาษา HTML จาก ที่ได้เขียนจาก TextEditor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 โครงสร้างภาษา HTML

การเขียนภาษา HTML นั้นมีส่วนประกอบหลักอยู่ 2 ส่วน คือ ส่วนที่เป็นเนื้อหา และส่วนที่เป็นคำสั่ง ส่วนที่เป็นคำสั่งนั้นจะอยู่ในรูปของ Tag ซึ่งจะเขียนอยู่ในเครื่องหมาย มากกว่า และน้อยกว่า <> แต่ละ Tag มีหน้าที่ที่แตกต่างกันออกไป Tag แบ่งออกเป็น 2 ประเภท คือ

1. แท็กเดี่ยว คือ คำสั่งที่มีคำสั่งเพียงอย่างเดียว ซึ่งสามารถใช้และสิ้นสุดคำสั่งได้ด้วยตัวของมันเองเช่น

```
ข้อความ..<br>
<hr>
<! - ข้อความ - >
```

2. แท็กคู่ คือ คำสั่งที่ต้องมีส่วนเริ่มต้นและส่วนจุดจบของคำสั่งนั้นๆ โดยแท็กที่เป็นส่วนจุดจบจะมีเครื่องหมาย slash / ติดไว้ เช่น

```
<html> ส่วนของเนื้อหา </html>
<center> ข้อความ </center>
<p> ข้อความ.... </p>
```

ถ้าหากมีการใช้แท็กคู่หลายๆ คำสั่ง เช่น คำสั่งตัวขีดเส้นใต้ <U> </U> และตามด้วย คำสั่ง ตัว เอียง<l>....</l>จะต้องปิดคำสั่งตัวเอียงก่อนแล้วจึงจะมาปิดคำสั่งตัวหนา***

```
<l><U> ข้อความ.... </U></l>
```

2.7.3 การกำหนดโครงสร้างหลัก

การจัดวางที่เห็นเป็นการจัดวางแบบมาตรฐาน ซึ่งประกอบไปด้วย 4 ส่วนเวลาเริ่มเขียนภาษา HTML ควรเริ่มจากตรงนี้ก่อนทุกครั้ง



รูปที่ 2.20 รูปการจัดวางแบบมาตรฐานของ HTML ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำสั่งหลัก <HTML>.... </HTML> เป็นคำสั่งที่ไว้กำหนดจุดเริ่มต้นและจุดจบของเอกสาร
2. คำสั่งหลัก <HEAD>.... </HEAD> เป็นคำสั่งที่ทำหน้าที่กำหนดส่วนหัวเรื่อง ภายในคำสั่งนี้จะประกอบไปด้วย
 - คำสั่งหลัก <TITLE>.... </TITLE> เป็นคำสั่งที่ใช้กำหนดข้อความที่ต้องการให้ขึ้นอยู่กับส่วนของ Title Bar โดยสามารถพิมพ์ได้ยาว 64 ตัวอักษร
 - คำสั่งหลัก <BODY>.... </BODY>เป็นคำสั่งที่ใช้ในการกำหนดรูปแบบของ เอกสารทั้งหมด ว่าจะให้มีลักษณะอย่างไร

2.8 การติดต่อกับฐานข้อมูล MySQL ด้วยคำสั่ง PHP

โปรแกรมการจัดการฐานข้อมูล MySQL เป็นลักษณะฐานข้อมูลที่สามารถทำงานได้อย่างมีประสิทธิภาพ ทั้งบนระบบปฏิบัติการ Windows และ Linux ซึ่งได้รับความนิยมเป็นอย่างมาก เนื่องจากใช้งานได้โดยไม่มีค่าใช้จ่าย และมีคุณสมบัติการใช้งานมากมาย การติดต่อฐานข้อมูล MySQL และนำเสนอข้อมูลแสดงผลทางเว็บเบราว์เซอร์มีขั้นตอนดังนี้

1. เริ่มติดต่อฐานข้อมูลในการติดต่อฐานข้อมูล MySQL จะใช้ฟังก์ชัน `mysql_connect ()` มีรูปแบบดังนี้

```
mysql_connect ($hostname,$user,$password) or die ("message");
```

โดยที่ \$hostname คือ ข้อความที่เป็นชื่อโฮสต์ (Host) \$user คือ ชื่อ login\$password คือ รหัสผ่านสำหรับเข้าใช้ฐานข้อมูลMySQL or die ("message") คือ ข้อความเพื่อแจ้งว่าติดต่อโฮสต์ไม่ได้

2. เลือกฐานข้อมูลที่ต้องการ เมื่อติดต่อฐานข้อมูล MySQL ได้แล้วขั้นตอนต่อไปเป็นการเลือกฐานข้อมูลที่ต้องใช้โดยใช้ฟังก์ชัน `mysql_select_db()` ซึ่งมีรูปแบบดังนี้

```
mysql_select_db ($dbname) or die ("message");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ \$dbname คือ ชื่อฐานข้อมูลที่ต้องการใช้ Or die ("message") คือ ข้อความเพื่อแจ้งว่าติดต่อกับฐานข้อมูลไม่ได้

3. กำหนดคำสั่ง SQL และสั่งให้ใช้งาน เมื่อเลือกฐานข้อมูลได้แล้วขั้นต่อไป คือ การเขียนคำสั่ง SQL แล้วสั่งให้ทำงาน โดยใช้ฟังก์ชัน mysql_query() ซึ่งมีรูปแบบดังนี้

```
mysql_query($sql)
```

โดยที่ \$sql คือ คำสั่ง SQL ที่ต้องการ จาก 2 คำสั่งที่ผ่านมา ซึ่งสามารถกำหนดคำสั่งได้ด้วยบรรทัดเดียวคือ mysql_db_query(\$dbname,\$sql);

4. การเก็บข้อมูลลงตัวแปรอาร์เรย์ หลังจากคำสั่งที่ให้ SQL ทำงานแล้ว จะได้ผลลัพธ์ออกมาซึ่งต้องนำผลลัพธ์เก็บลงในอาร์เรย์ด้วยฟังก์ชัน mysql_fetch_array() ซึ่งมีรูปแบบ คำสั่งดังนี้

```
$dbquery= mysql_db_query($dbname,$sql);
$result = mysql_fetch_array($dbquery);
```

เนื่องจากตัวแปร \$result เป็นอาร์เรย์ ดังนั้น \$result[0] คือ การอ้างอิงสมาชิกตัวแรก \$result[ID] คือ การอ้างอิงชื่อฟิลด์ ID

5. นับจำนวนเรคคอร์ด ในการเก็บผลลัพธ์ลงในตัวแปรอาร์เรย์ จะต้องเก็บได้มากมาย ตามจำนวนเรคคอร์ดที่มีอยู่ ดังนั้นก่อนการแสดงผลทางเว็บเบราว์เซอร์ ขั้นต่อไปจะเป็นการวนรอบ ตามจำนวนเรคคอร์ดที่มีนั้น ดังนั้นจำเป็นต้องทราบว่าในตารางที่ใช้งานมีกี่เรคคอร์ด ซึ่งสามารถทราบได้ด้วยฟังก์ชัน mysql_numrows() ซึ่งมีรูปแบบการใช้งานดังนี้

```
$dbquery= mysql_db_query($dbname,$sql);
$num_rows= mysql_num_rows($dbquery);
```

6. การแสดงผลทางเว็บเบราว์เซอร์เมื่อได้ผลลัพธ์ตามต้องการแล้ว ขั้นตอนต่อไปที่สำคัญคือการนำข้อมูลมาแสดงผลทางเว็บเบราว์เซอร์ ซึ่งนิยมแสดงในตารางโดยใช้การวนรอบด้วยคำสั่ง while แล้วฟ้าผลลัพธ์ไปใส่ในแท็ก<Td>ของ html แสดงตัวอย่างตามคำสั่งดังนี้

```

$1 = 0;

While ($1 <$num_rows)

{

$result = mysql_fetch_array($dbquery);

echo"<Tr><Td>$result["ID"]</Td><Td>$result["Name"]</Td>

<Td>$result["Surname"]</Td><Td>$result["Salary"]</Td></Tr>"

$1++;

}

```

7. ปิดการติดต่อกับฐานข้อมูล MySQL หลังจากทำงานเสร็จเรียบร้อยแล้วจะเป็นการปิดการติดต่อกับฐานข้อมูล MySQL ด้วยฟังก์ชัน mysql_close() ซึ่งจริงๆอาจไม่ต้องทำการปิดการติดต่อกับฐานข้อมูล MySQL เนื่องจากฐานข้อมูลจะปิดการติดต่อเองเมื่อรันสคริปต์จบ ซึ่งคำสั่งของการปิดการติดต่อกับฐานข้อมูล MySQL คือ

```
mysql_close();
```

2.9 ฐานข้อมูล

ฐานข้อมูล (Database) คือ วิธีการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างมีระเบียบ ซึ่งจะทำให้ ง่ายต่อการใช้งาน และค้นหาข้อมูล ซึ่งฐานข้อมูลที่คนส่วนใหญ่คุ้นเคยคือ ฐานข้อมูลเชิงสัมพันธ์ เป็นรูปแบบการจัดเก็บข้อมูลที่สัมพันธ์กันโดยมองข้อมูลในลักษณะของตารางที่มีความสัมพันธ์กัน

2.9.1 องค์ประกอบของฐานข้อมูล

การใช้งานฐานข้อมูลจะต้องประกอบด้วยองค์ประกอบต่อไปนี้

1. แอปพลิเคชันฐานข้อมูล (Database Application) เป็นแอปพลิเคชันที่สร้างไว้ให้ผู้ใช้สามารถติดต่อกับฐานข้อมูลได้อย่างสะดวก ซึ่งมีการติดต่อกับฐานข้อมูลแบบเมนู
2. ระบบจัดการฐานข้อมูล (DBMS) เป็นซอฟต์แวร์ที่ทำหน้าที่ในการจัดการข้อมูลในฐานข้อมูล เช่น การจัดเก็บ, การแสดงผล, การค้นหา, การสำรองข้อมูล เป็นต้น โดยจะเป็นเครื่องมือในการทำงานของผู้บริหารฐานข้อมูล และเป็นตัวกลางที่เชื่อมผ่านระหว่างแอปพลิเคชันฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล
3. ดาต้าเบสเซิร์ฟเวอร์ (Database Server) เป็นคอมพิวเตอร์ที่ให้บริการในการจัดการฐานข้อมูลเป็นคอมพิวเตอร์ที่มีระบบจัดการฐานข้อมูลทำงาน เพราะฉะนั้นจึงมักจะเป็นคอมพิวเตอร์ ที่มีประสิทธิภาพการทำงานสูงกว่าคอมพิวเตอร์ที่ใช้งานทั่วไป
4. ข้อมูล (Data) คือ ตัวเนื้อหาของข้อมูลที่ใช้งาน ซึ่งจะถูกเก็บไว้ในหน่วยความจำของดาต้าเบสเซิร์ฟเวอร์โดยจะถูกเรียกมาใช้จากระบบจัดการฐานข้อมูล
5. ผู้บริหารฐานข้อมูล (Database Administrator) ผู้บริหารฐานข้อมูลเป็นผู้ทำหน้าที่ดูแลข้อมูลในฐานข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งควบคุมให้การทำงานเป็นไปอย่างราบรื่น และ ทำหน้าที่กำหนดผู้ที่จะมีสิทธิ์ใช้งานฐานข้อมูล กำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแลดาต้าเบสเซิร์ฟเวอร์ให้ทำงานได้อย่างปกติ

ฐานข้อมูลมีหลักการพื้นฐานสำคัญมาจากคณิตศาสตร์เรื่องทฤษฎีบทของเซต (Theory of Set) โดยการจัดเก็บหรือแสดงข้อมูลให้ผู้ใช้เห็นจะเป็นแบบตารางข้อมูลต่างๆ จะแสดงในรูปของตาราง โดยในแต่ละตารางจะเป็นการจัดรวบรวมข้อมูลประเภทเดียวกันไว้ด้วยกัน โดยแต่ละแถวที่ประกอบขึ้นเป็นตารางนั้นก็คือ เรคคอร์ด ซึ่งจะเป็นที่เก็บข้อมูลแต่ละชุดของตารางนั้น และในแต่ละแถวก็จะประกอบด้วยคอลัมน์ซึ่งเป็นหน่วยย่อยที่แสดงคุณสมบัติของข้อมูล แต่ละแถวในแต่ละตารางมักจะมี

บางคอลัมน์หรืออาจจะหลายคอลัมน์ประกอบกันสามารถบอกถึงความแตกต่าง ของฐานข้อมูลในแต่ละแถวได้ คอลัมน์ดังกล่าวเรียกว่า Primary Key เช่น คอลัมน์เลขประจำตัว พนักงานที่แตกต่างกันทุกคน และในการเชื่อมโยงตารางฐานข้อมูลที่มีหลายตารางนั้นจะทำได้ ก็ต่อเมื่อแต่ละตารางมีคอลัมน์เกี่ยวข้องกัน ซึ่งคอลัมน์ที่เกี่ยวข้องนี้ จะเรียกว่า Foreign Key

2.9.2 ชุดคำสั่งที่ใช้จัดการกับข้อมูล

แบ่งออกเป็น 2 ประเภทหลักๆ คือ

1. การจัดการกับโครงสร้างฐานข้อมูล เป็นการสร้าง หรือแก้ไขข้อมูลและตาราง เช่น สร้างตารางฐานข้อมูลพนักงานขึ้นมา ชุดคำสั่งที่เกี่ยวข้องกับการสร้างหรือเปลี่ยนแปลงโครงสร้าง ของฐานข้อมูลมีศัพท์เรียกว่า Data Definition Language (DDL)
2. การจัดการกับฐานข้อมูลที่มีอยู่ในตารางในฐานข้อมูลจะเป็นการทำงานของการทำงานของการเพิ่ม ลบ และ แก้ไขข้อมูลในตาราง เช่น การเพิ่มข้อมูลพนักงานใหม่เข้าไปในตารางข้อมูลพนักงาน คำสั่งประเภท นี้เรียกว่า Data Management Language (DML)

2.9.3 โครงสร้างของภาษาMySQL

MySQL เป็นโปรแกรมบริหารจัดการด้านฐานข้อมูลระดับเซิร์ฟเวอร์ หรือเรียกว่า Database Management System (DBMS) ซึ่งจะคอยทำงานตามคำสั่ง SQL ที่ส่งมาจากโปรแกรมไคลเอนต์ เช่น mysql หรือ mysqlfront มีคุณสมบัติดังนี้

1. ทำงานแบบ multi-thread เป็นการแบ่งการทำงานเป็นส่วนย่อย ทำให้สามารถทำงานได้ เร็ว และทำงานมีอิสระไม่ขึ้นต่อกัน สามารถใช้ได้กับเครื่องที่มี CPU มากกว่า 1ตัว
2. รองรับชนิดข้อมูลที่หลากหลาย เช่น signed/unsigned INTEGER ขนาด 1,2,3,4 และ 8 ไบต์, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET และ ENUM
3. ใช้ได้กับหลากหลายภาษา อาทิ C, C++, JAVA, Perl, PHP, VB, Delphi เป็นต้น
4. ทำงานกับฐานข้อมูลขนาดใหญ่ได้
5. รองรับภาษา SQL มาตรฐาน ที่เรียกว่า ANSI SQL92 หรือ SQL92 และมีคำสั่งที่ขยายต่อ จาก SQL92 เพื่อเพิ่มศักยภาพในการทำงานมากยิ่งขึ้น เช่น คำสั่ง LIMIT
6. ใช้ได้กับระบบปฏิบัติการที่หลากหลาย เช่น Linux, Solaris, Mac OS X Server , Windows 95/98/2K/XP และระบบตระกูล Unix อื่นๆมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. รองรับ ODBC 2.5 (Open Database Connectivity) ได้หมดทุกฟังก์ชัน

2.9.3.1 คำสั่งเบื้องต้นสำหรับMySQL

คำสั่งเพื่อการใช้งาน MySQL บน Windows เหมือนกับเวอร์ชันบน Linux โดยมี คำสั่งพื้นฐานในการใช้งานดังนี้

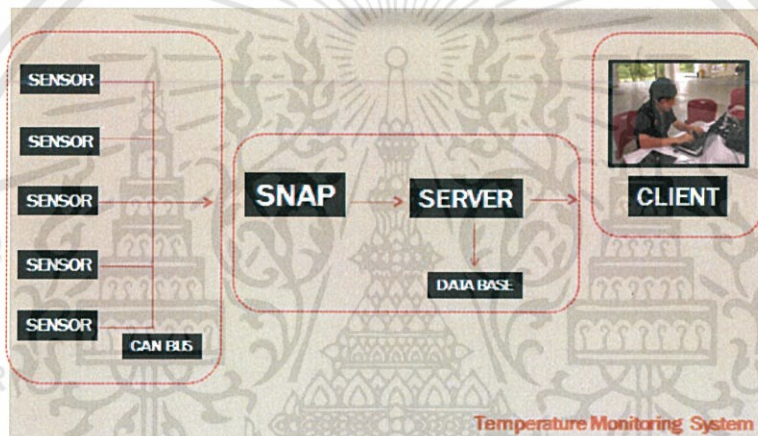
- Help ดูระบบช่วยเหลือ
- Status แสดงสถานะของ MySQL
- Exit ออกจาก MySQL
- Quit ออกจาก MySQL
- Use ใช้งานฐานข้อมูล
- Create database สร้างฐานข้อมูลใหม่
- Create table สร้างตารางใหม่
- Show database แสดงฐานข้อมูลที่มีอยู่ใน MySQL
- Show table แสดงตารางฐานข้อมูลที่มีอยู่ในฐานข้อมูลปัจจุบันที่ใช้
- Select เลือกฟิลด์ที่จะแสดงผลข้อมูล
- Insert into เพิ่มข้อมูลเข้าสู่ตารางที่กำหนด
- Delete from ลบข้อมูลออกจากตารางตามเงื่อนไข
- Load data local infile โหลดข้อมูลจากเท็กซ์ไฟล์เข้าสู่ตาราง

บทที่ 3

วิธีการดำเนินงาน

3.1 การออกแบบ

โครงการนี้เป็นการวัดค่าอุณหภูมิจากเซนเซอร์วัดอุณหภูมิแล้วส่งค่าอุณหภูมิเหล่านั้น ผ่าน CAN Bus มายัง SNAP หลังจากนั้นข้อมูลจะถูกนำไปเก็บไว้ในฐานข้อมูลของเครื่องแม่ข่ายและแสดงผลบนหน้าเว็บเบราว์เซอร์ โดยมีการออกแบบดังต่อไปนี้



รูปที่ 3.1 แผนภาพแสดงโครงการ

ขั้นตอนการทำงานของระบบ

1. การรับค่าอุณหภูมิจากตัวเซนเซอร์ส่งผ่าน CAN Bus มายัง SNAP
2. การนำค่าอุณหภูมิจาก SNAP มาเก็บลงในฐานข้อมูลบนเครื่องแม่ข่าย
3. แสดงค่าอุณหภูมิตน เว็บเบราว์เซอร์

3.1.1 โครงสร้างของตัววัดอุณหภูมิ

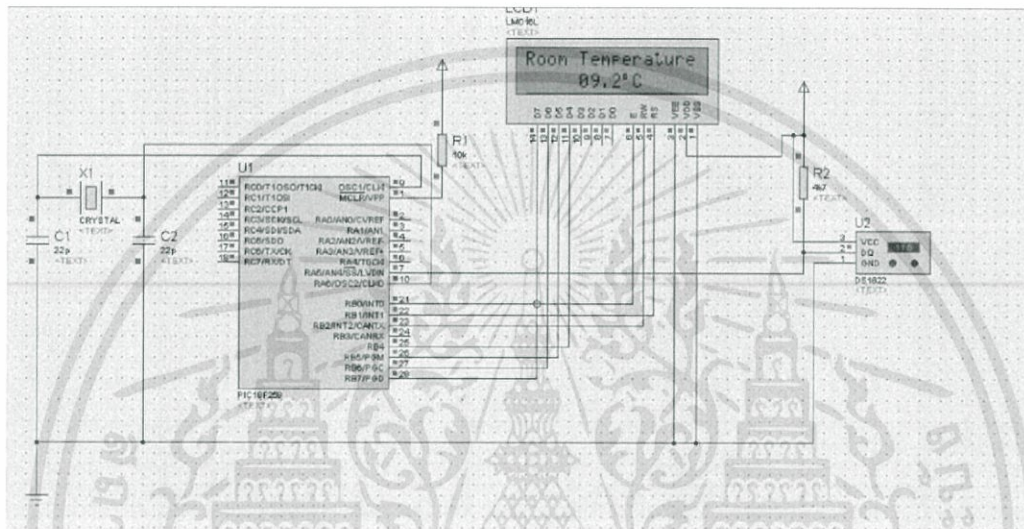
การออกแบบวงจรเพื่อใช้สำหรับวัดอุณหภูมินั้นมีอุปกรณ์ที่ใช้งานดังต่อไปนี้

- | | |
|--------------------------------|-------|
| - IC วัดอุณหภูมิ DS1820 | 1 ตัว |
| - ไมโครคอนโทรลเลอร์ PIC 18F258 | 1 ตัว |
| - Crystal 8 MHz | 1 ตัว |
| - ตัวเก็บประจุ 22 pF | 2 ตัว |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวต้านทาน 4.7 กิโลโอห์ม 1 ตัว
- ตัวต้านทาน 10 กิโลโอห์ม 1 ตัว
- ตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม 1 ตัว
- จอแสดงผล LCD 1 อัน

การต่อวงจรวัดอุณหภูมิเพื่อแสดงค่าอุณหภูมิออกจอ LCD สามารถต่อได้ตามรูปที่ 3.2



รูปที่ 3.2 วงจรวัดอุณหภูมิแสดงผลบนจอ LCD

3.1.1.1 คำสั่งควบคุมการทำงานของ DS1820

คำสั่งที่ใช้มากที่สุดมี 3 คำสั่งคือ

1. คำสั่งไม่ติดต่อกับหน่วยความจำรวม (Skip Rom:0CCH) ถ้ามี DS1820 ต่ออยู่บนสายสัญญาณเพียงตัวเดียวก็ไม่จำเป็น ต้องมีการอ่านแอดเดรสของอุปกรณ์
2. คำสั่งแปลงอุณหภูมิ (Convert T:044H) ถ้าส่งคำสั่งนี้ให้ DS1820 แล้วต้องรอน้อย 120 ไมโครวินาทีเพื่อให้ DS1820 ใช้ในการแปลงค่าอุณหภูมิมาเก็บที่ Scratchpad
3. คำสั่งอ่านข้อมูลจาก Scratchpad (Read Scratchpad:0BEH) เมื่อส่งคำสั่งนี้แล้ว DS1820 จะทยอยส่งข้อมูลค่าอุณหภูมิจาก Scratchpad ทั้ง 9 ไบต์ออกมา

3.1.1.2 การควบคุมการทำงานของ DS1820

ซึ่งการทำงานของระบบการสั่งการนี้จะทำงานบน พื้นที่หน่วยความจำเลเซอร์รวมขนาด 64 บิต ผ่านพอร์ตไอซีแต่ละตัวและสามารถให้เอาต์พุตเดี่ยวเพื่อกำหนดคุณสมบัติของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ตรวจจับอุณหภูมินี้หลายๆ ตัวก็ทำได้โดยส่งการผ่านสายข้อมูล 1- wire นี้ หลังจากฟังก์ชันในรอม ถูกลำดับการทำงานแล้ว ก็พร้อมที่จะถูกใช้งาน และสามารถที่จะเข้าถึงการทำงานภายในตัวไอซีได้ทั้งหมด หน่วยความจำและส่วนควบคุมฟังก์ชันก็จะถูกเข้าถึงการทำงานได้ และส่วนจัดเก็บค่าที่เซตไว้สามารถหรืออาจจะถูกเก็บไว้ในพื้นที่ 1 ส่วนจากทั้งหมด 6 ส่วนของหน่วยความจำและส่วนควบคุมฟังก์ชันการสั่งการ ส่วนควบคุมฟังก์ชันการสั่งการหนึ่งส่วนจะถูกกำหนดคุณสมบัติของ DS1820 ให้อยู่ในรูปแบบของการวัดค่าของอุณหภูมิซึ่งผลของการวัดนี้จะถูกบันทึกไว้ใน DS1820 ในส่วนของหน่วยความจำส่วนหนึ่ง (scratchpad) และบางครั้งก็จะอ่านออกมาได้จากตารางสารบัญของหน่วยความจำฟังก์ชันการสั่งการ ซึ่งเป็นการอ่านออกมาเฉพาะหัวข้อที่ถูกบันทึกไว้ในหน่วยความจำ scratchpad สัญญาณกระตุ้นเตือนค่าอุณหภูมิสูงเกินและต่ำเกิน (TH และ TL) จะประกอบด้วย 1 ไบต์ EEPROM ถ้าสัญญาณการเตือนการค้นหาไม่ถูกจ่ายเข้าไปยัง DS1820 รีจิสเตอร์เหล่านี้บางครั้งจะถูกรู้ใช้ได้อย่างทั่วๆ ไปจากหน่วยความจำที่ผู้อ่านกำหนดได้ และการเขียนเข้าไปในส่วนของการเตือน TH และ TL จะไม่ใช่หน่วยความจำฟังก์ชัน การสั่งการและการเข้าไปถึงรีจิสเตอร์นี้จะอ่านผ่านหน่วยความจำ scratchpad และข้อมูลอื่นๆ ที่ต้องการอ่านและเขียนจะกระทำได้ในบิตแรกของ LSB

3.1.1.3 การทำงานของสัญญาณเตือน

หลังจากที่ DS1820 มีการตรวจวัดอุณหภูมิเกิดขึ้นแล้ว ค่าของอุณหภูมิก็นำมาทำการเปรียบเทียบเพื่อทำเป็นสัญญาณกระตุ้น การเปรียบเทียบค่าของอุณหภูมิก็นำมาเปรียบเทียบกับค่าที่ถูกบันทึกหรือกำหนดไว้ของค่าอุณหภูมิสูงสุด (TH) และค่าอุณหภูมิต่ำสุด (TL) ตลอดย่านอุณหภูมิที่วัดได้ โดยจะใช้พื้นที่รีจิสเตอร์ 8 บิต สำหรับการทำงานนี้ใน MSB ของ TH หรือ TL ที่ตรงกันก็จะถูกส่งไปยัง SB ของรีจิสเตอร์อุณหภูมิขนาด 16 บิต ถ้าผลของการวัดอุณหภูมิก็นำค่าสูงเกินกว่า TH หรือต่ำกว่า TL ลำดับสัญญาณเตือนภายในอุปกรณ์ก็จะถูกเซต ซึ่งลำดับของสัญญาณเตือนนี้จะถูกอัปเดตทุกครั้งที่มีการวัดค่าอุณหภูมิ เมื่อลำดับสัญญาณเตือนถูกเซต DS1820 จะมีการตอบสนองนำไปสู่การค้นหาสัญญาณเตือนการสั่งการและจะยอมให้ทำการต่อ DS1820 ในลักษณะขนานกันหลายตัวได้ เพื่อทำการจำลองการวัดค่าอุณหภูมิแล้วนำมาเฉลี่ยค่าของการวัดในครั้งนั้นอีกขั้นตอนหนึ่ง

8-BIT CRC CODE		48-BIT SERIAL NUMBER		8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB

รูปที่ 3.3 ตารางแสดงความสัมพันธ์ของค่าอุณหภูมิกับข้อมูลดิจิทัลเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.4 บิตเลเซอร์รวม

ค่าอุณหภูมิ	ดิจิทัลเฮกซ์ทูด (Binary)	ดิจิทัลเฮกซ์ทูด (Hex)
+125 °C	00000000 11111010	00FA
+25 °C	00000000 00110010	0032h
+1/2 °C	00000000 00000001	0001h
+0 °C	00000000 00000000	0000h
-1/2 °C	11111111 11111111	FFFFh
-25 °C	11111111 11001110	FFCEh
-55 °C	11111111 10010010	FF92h

รูปที่ 3.4 แสดงการแบ่งส่วนในหน่วยความจำรวมขนาด 64 บิต

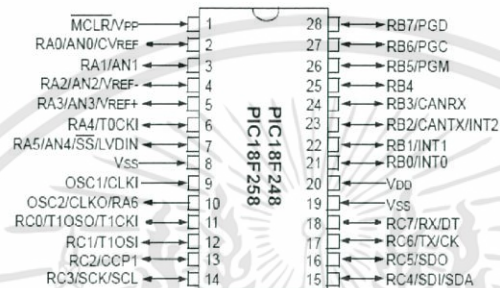
ใน DS1820 นั้นจะประกอบด้วยส่วนของรหัสหน่วยความจำรวม ที่มีความยาวถึง 64 บิตโดยใน 8 บิตแรกจะเป็นรหัสระบุตระกูล (family code) 1 – Wire ของ DS1820 (DS1820 มีรหัสเป็น 10h) และอีก 48 บิตต่อมาเป็นส่วนระบุอนุกรมตัวเลข (serial number) และอีก 8 บิตสุดท้ายคือส่วนบันทึก CRC ของ 56 บิตแรก ดังแสดงการแบ่งส่วนไว้ในรูปที่ 4 หน่วยความจำรวมขนาด 64 บิต และส่วนควบคุมฟังก์ชันรอมนี้จะยอมให้ DS1820 สามารถทำงานเป็นอุปกรณ์อินเตอร์เฟสแบบ 1 – Wire ได้ และมีรายละเอียดตามโปรโตคอลของระบบบัส 1 – Wire ซึ่งฟังก์ชันและส่วนควบคุมต่างๆใน DS1820 จะยังไม่สามารถทำงานหรือมีการเข้าถึงได้ จนกว่าจะมีการเซตอัฟโปรโตคอลฟังก์ชันในหน่วยความจำรวมเสียก่อน ในการอินเตอร์เฟสในส่วนหลักของฟังก์ชันการสั่งการในหน่วยความจำรวมจะต้องมีลำดับฟังก์ชันดังนี้คือ

1. การอ่านหน่วยความจำรวม
2. ทำการแมตซ์รอม
3. ค้นหารอม
4. กระโดดข้ามรอม
5. เตือนการค้นหา

หลังจากที่มีการเซตลำดับฟังก์ชันรอมดังข้างต้นเสร็จแล้ว ฟังก์ชันต่างๆ ของ DS1820 ก็จะสามารถเข้าถึงได้

3.1.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

โครงการนี้ใช้ภาษาซี ในการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ กับอุปกรณ์ต่างๆ และใช้โปรแกรมคอมไพเลอร์ของ MikroC Pro for Pic เป็นตัวสร้างไฟล์ HEX เพื่อทำการเบิร์นลงสู่ไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์ที่ใช้ในโครงการนี้เป็นไมโครคอนโทรลเลอร์ชนิด PIC เบอร์ 18F258 ซึ่งแสดงรูปขาของ PIC18F258 ดังรูปที่ 3.5



รูปที่ 3.5 ไดอะแกรมของขา PIC18F258

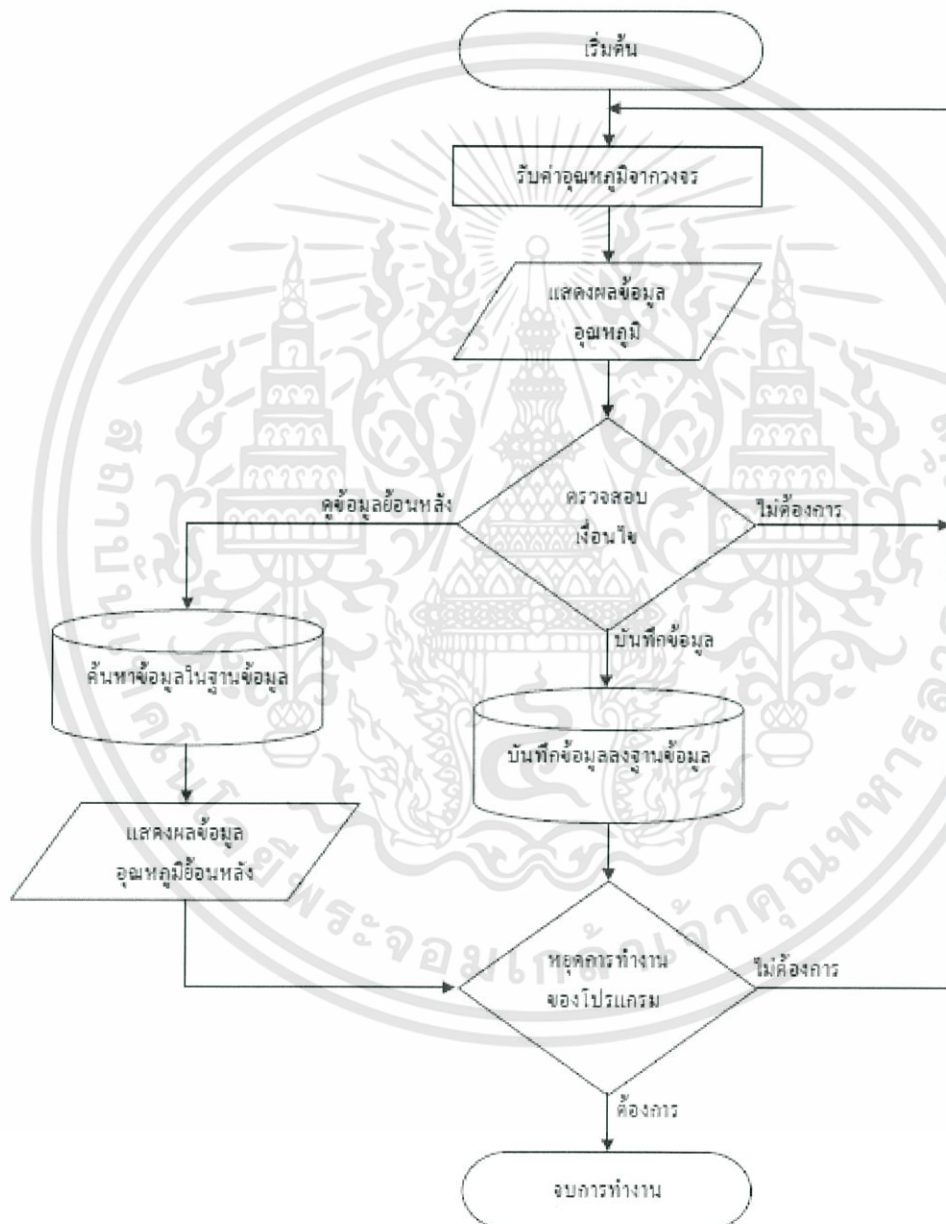
ไมโครคอนโทรลเลอร์ PIC 18F258 มีคุณสมบัติที่สำคัญดังนี้

- ทำงานได้สูงสุดที่ 40 MHz
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 32 กิโลไบต์
- มีหน่วยความจำ RAM ขนาด 1536 ไบต์
- มี EEPROM ขนาด 256 ไบต์
- มีระบบ Code Protection
- มี CAN คอนโทรลเลอร์ในตัว
- สามารถโปรแกรมด้วยไฟ +5 V DC ได้
- ทำงานที่ไฟเลี้ยงตั้งแต่ 2V DC ถึง 5.5V DC ได้
- มีระบบ UART สำหรับเชื่อมต่อแบบ RS232
- มี I/O พอร์ตทั้งหมด 3 พอร์ต

จากคุณสมบัติที่สำคัญที่กล่าวไว้ข้างต้น ทำให้ไมโครคอนโทรลเลอร์ PIC 18F258 มีคุณสมบัติที่เหมาะสมกับโครงการนี้ เนื่องจากโครงการนี้ต้องการส่งข้อมูลโดยใช้ CAN Bus จึงต้องการ

ไมโครคอนโทรลเลอร์ที่มี CAN คอนโทรลเลอร์ในตัว และการรับค่าอุณหภูมิจากตัวเซนเซอร์แล้วนำค่าออกไปแสดงผลบนจอ LCD ทำให้ไมโครคอนโทรลเลอร์ถูกใช้งานทั้งหมด 3 พอร์ต เป็นต้น

3.2 การทำงานของระบบตรวจวัดอุณหภูมิ



รูปที่ 3.6 การทำงานของระบบตรวจวัดอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วิธีการดำเนินงาน

ในโครงการระบบตรวจวัดอุณหภูมินี้ได้แบ่งเครื่องมือที่ใช้พัฒนาระบบออกเป็น 2 ส่วนด้วยกันคือ

(ก) ส่วนของฮาร์ดแวร์ (Hardware) ที่ใช้พัฒนาประกอบด้วย คอมพิวเตอร์สำหรับจำลองเป็นเครื่องแม่ข่าย และใช้เขียนโปรแกรมควบคุมซึ่งใช้ระบบปฏิบัติการ Windows อุปกรณ์ภาคจ่ายไฟ SNAP Base Board สายแลนเชื่อมต่อแบบอนุกรม RS-232

(ข) ส่วนของซอฟต์แวร์ (Software) ที่ใช้พัฒนาประกอบด้วยโปรแกรมต่างๆดังต่อไปนี้

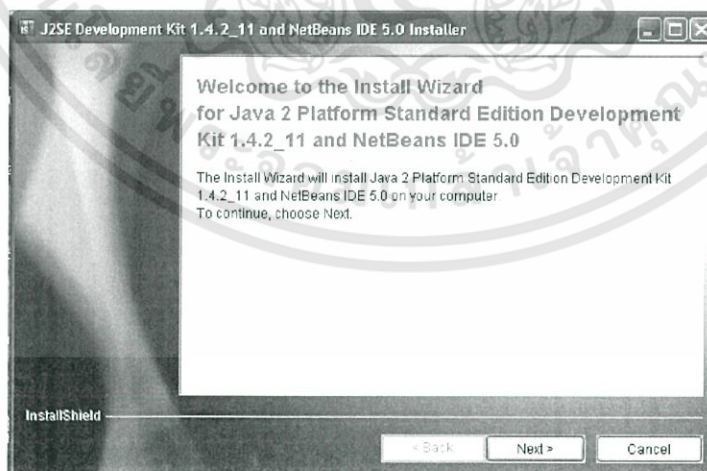
- โปรแกรม EditPlus
- โปรแกรม FTP Commander
- โปรแกรม Java 2 SDK-SE v1.4.2_14
- โปรแกรม jdk-6u3-window-i586-p.exe
- โปรแกรม SnapDev (SNAP Developer)

3.3.1 ขั้นตอนการดำเนินงาน

สามารถแบ่งวิธีการดำเนินออกเป็น การตั้งค่า , การใช้งาน และการเขียนโปรแกรมควบคุม SNAP ด้วยภาษา JAVA

3.3.1.1 ขั้นตอนการติดตั้งจาวาและการตั้งค่า

1. เปิดโปรแกรม J2DK version 1.4.1-11แสดงดังรูปที่ 3.7 กดปุ่ม next เพื่อทำการเข้าสู่ขั้นตอนต่อไป

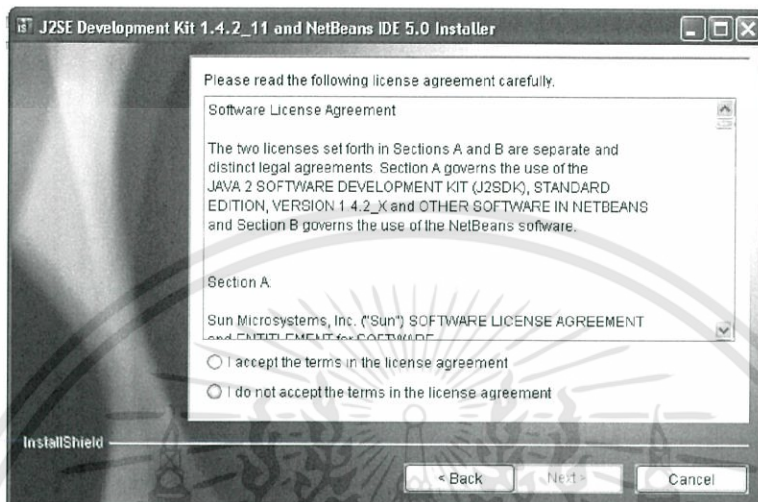


รูปที่ 3.7 เปิดโปรแกรม J2DK version

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

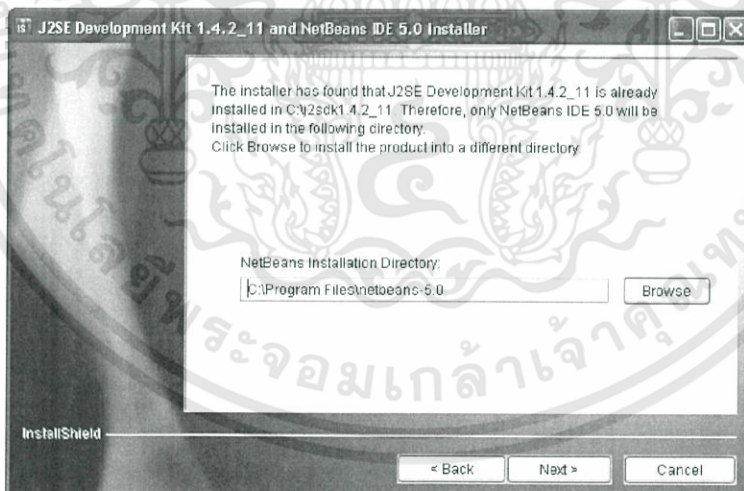
2. คลิกเลือก ยอมรับเงื่อนไขในรายละเอียดต่างๆ ของโปรแกรม แสดง

ตามรูปที่ 3.8



รูปที่ 3.8 รายละเอียดต่างๆ ของโปรแกรม

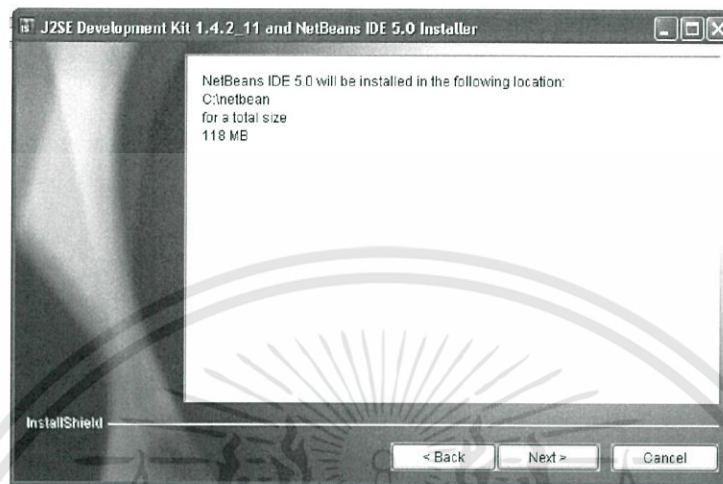
3. เลือก directory ไปยังที่ต้องการจะเก็บไว้ แสดงดังรูปที่ 3.9



รูปที่ 3.9 การเลือก directory

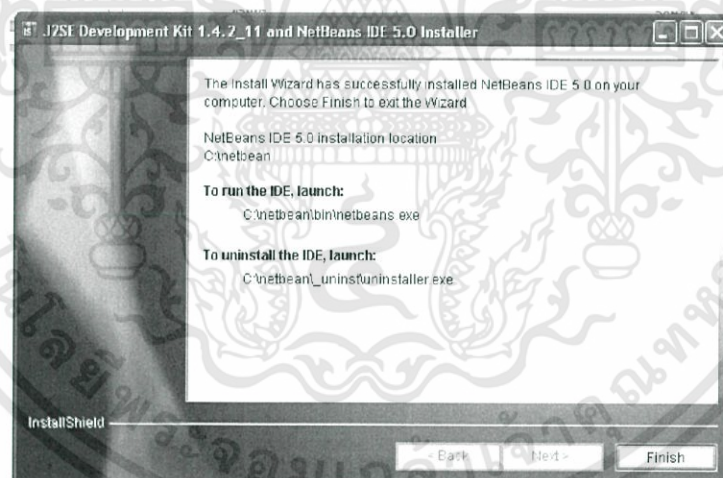
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. รอจนกว่าการลงโปรแกรมจะเสร็จสมบูรณ์ แสดงตามรูปที่ 3.10



รูปที่ 3.10 การลงโปรแกรม

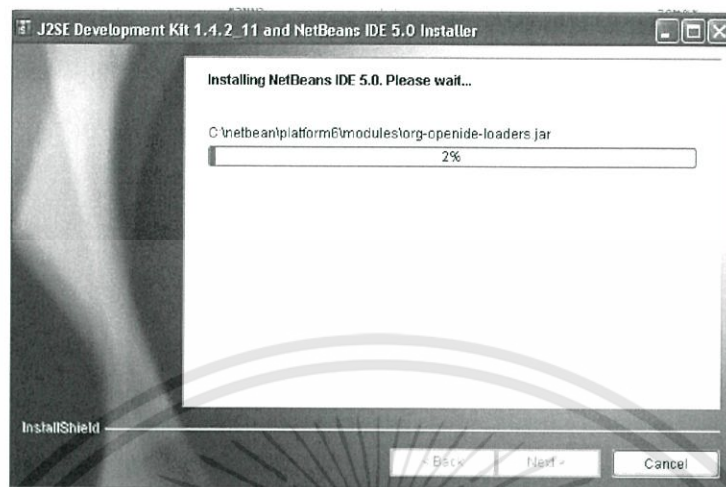
5. การลงโปรแกรมเสร็จสมบูรณ์ แสดงตามรูปที่ 3.11



รูปที่ 3.11 แสดงการลงโปรแกรมเสร็จสมบูรณ์

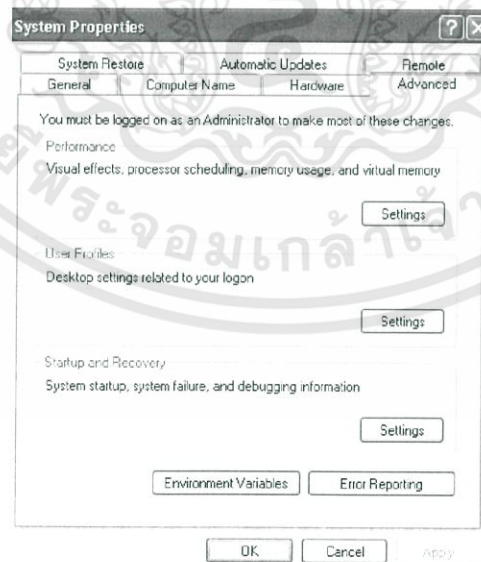
1. ทำการ Configuration โปรแกรม J2SDK โดยไปที่ Control panel และ Double Click ที่ icon System แสดงดังรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 Configuration โปรแกรม J2SDK

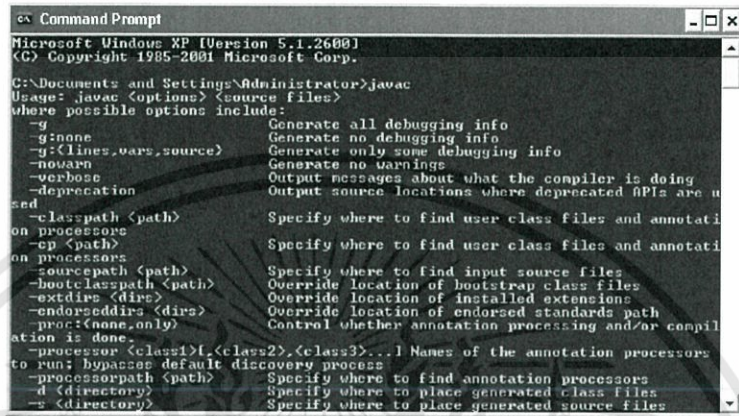
2. เปิดหน้าต่างโปรแกรม System ให้เลือก Advance Tab จากนั้นให้เลือก Environment Variable แล้วทำการกดปุ่ม New ที่กล่องของ User variable for Administrator โดยกำหนด Variable name = javapath และกำหนด Variable value = path ที่ลงโปรแกรม JAVA จำกัด Variable value = c:\j2sdk1.4.2_11\bin เป็นต้น จากนั้นให้เลือก path กดปุ่มแก้ไขในช่อง Variable path โดยให้เลื่อนเคอร์เซอร์ไปที่ท้ายสุด พิมพ์ข้อความนี้ ;%javapath% แล้วกดปุ่ม OK ทุกหน้าจอนจนหมด ดังแสดงในรูปที่ 3.13



รูปที่ 3.13 เลือก Environment Variable เพื่อทำการกำหนด path

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทดสอบการทำงานของโปรแกรม Java ให้เปิด Window command โดย Start เลือก พิมพ์ Command หรือ cmd ใน open box กดปุ่ม OK จะปรากฏจอ Window command พิมพ์ javac บน prompt ในหน้าต่างจะต้องได้ตามรูปที่ 3.14



```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

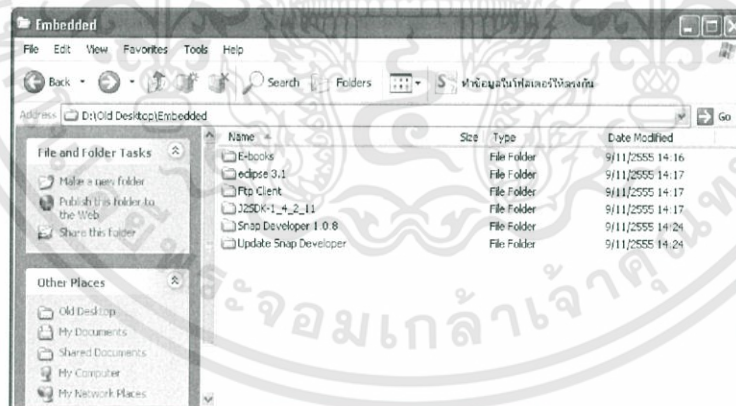
C:\Documents and Settings\Administrator>javac
Usage: javac <options> <source files>
where possible options include:
  -g             Generate all debugging info
  -g:none       Generate no debugging info
  -g:(lines,vars,source) Generate only some debugging info
  -nowarn       Generate no warnings
  -verbose      Output messages about what the compiler is doing
  -deprecation  Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotation processors
  -cp <path>      Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>  Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:(none,only) Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>[,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>  Specify where to place generated class files
  -s <directory>  Specify where to place generated source files

```

รูปที่ 3.14 ทดสอบ Java

3.3.1.2 การติดตั้งโปรแกรม Snap Developer

1. เริ่มจากการ copy subdirectory “Snap Developer 1.0.8” ไปยัง Drive ที่ต้องการจะได้ sub directory ดังแสดงในรูปที่ 3.15



รูปที่ 3.15 ตำแหน่ง directory ของ Snap Developer 1.0.8

2. ทดลอง Compile โปรแกรม HelloWorld โดยเปลี่ยน sub directory ไปที่:./SnapDeveloper 1.0.5/samples/HelloWorld จะเห็นเพิ่ม build.bat Double click ที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

build.batจากนั้นจะปรากฏ เพิ่ม HelloWorld.class ขึ้นมา เป็นอันเสร็จสิ้น การ compile โปรแกรม HelloWorldดังรูปที่ 3.16



รูปที่ 3.16 รายละเอียดในไฟล์ build.bat

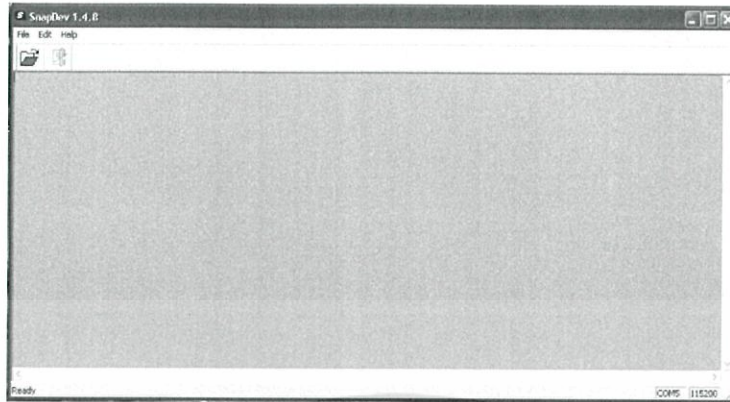
รายละเอียดโปรแกรมภายในไฟล์แสดงดังรูปที่ 3.17



รูปที่ 3.17 รายละเอียดของแฟ้ม Built.bat

3. การติดต่อ SNAP Board กับ Computer ผ่าน SNAP Devให้ประกอบ SNAP เข้ากับ Baseboard จากนั้นให้ ต่อสาย Terminal console โดยให้ใช้ หัว RJ-11 ต่อเข้ากับ Connector RJ-11 ของ Baseboard ด้านหัว DB-9 ให้ต่อเข้ากับ Com1 ของ Computer จากนั้นให้เปิด โปรแกรม SnapDev (Snap Developer1.0.5\bin\snapdev.exe) ดังแสดงตามรูปที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



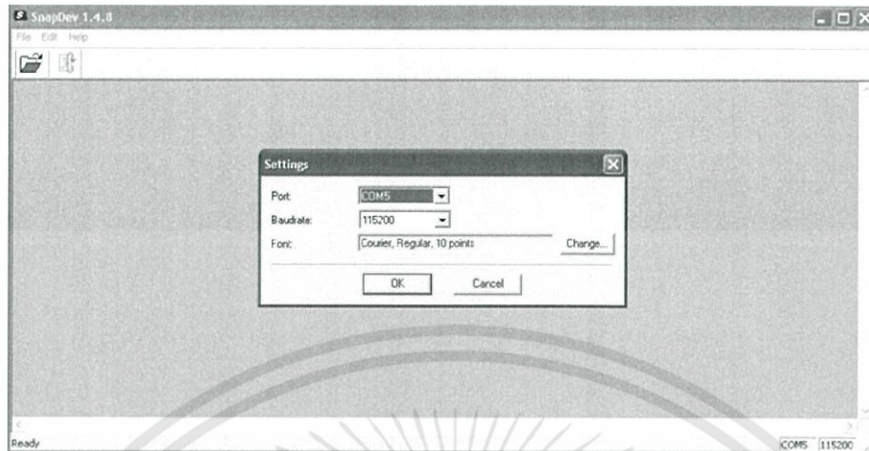
รูปที่ 3.18 โปรแกรม SNAP Dev

4. กำหนด Configuration ให้กับโปรแกรม Snap Dev โดยเลือก File แล้วเลือกไปที่ Settings ดังรูปที่ 3.19



รูปที่ 3.19 กำหนด Configuration

5. เลือก Comport ที่ต้องการใช้ติดต่อกับ SNAP เลือก File >> Setting เพื่อกำหนดค่า Baudrate = 115200 และ com port ที่จะใช้ติดต่อกับ Snap board เมื่อกำหนดเสร็จเรียบร้อยแล้วให้กดปุ่ม OK เป็นอันเสร็จสิ้น การ Config Comport ดังแสดงในรูปที่ 3.20



รูปที่ 3.20 กำหนดค่า Baudrate

6. เลือกหัวข้อ File แล้วไปที่ Open Port เพื่อให้โปรแกรม SnapDev พร้อมจะรับ ข้อมูลจากSNAP board จากนั้นให้จ่าย Power ให้กับ SNAP บอร์ด หรือไม่ก็ให้กดปุ่ม Hardware Reset ในกรณีที่จ่ายไฟให้กับ SNAP แล้วก่อนจะเปิด โปรแกรม SnapDev จะได้น้ำจอบน โปรแกรม SnapDev ดังรูปที่ 3.21



รูปที่ 3.21 โปรแกรม SnapDev เมื่อติดต่อกับ SNAP

7. ทดสอบคำสั่งพิมพ์ help เพื่อแสดงคำสั่งที่เก็บใน Sub Directory Root แต่ถ้าเป็น Sub Directory อื่นให้พิมพ์คำสั่ง ls แทน ตามรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SnapDev 1.4.0
File Edit Help

FTP server [OK]
Telnet server [OK]
Setting timezone to GMT
Reading /system/startup.ini [OK]

localhost localdomain ()
Login root
Password
/root> help

about alias cat cd
chgrp chacd date clear
cp cp echo exit find
flash ftp hd heap
help history hostname hview
id in ipconfig java
kill in ls mkdir
skniod mount nmount more
nv netstat nlookup od
out passwd ping ps
pwd reboot recv rx
radir route setenv send
source startserver stats stopserver
su sync touch unalias
umount update useradd userdel
who write vail

Ready
COMS 115200

```

รูปที่ 3.22 แสดงการใช้คำสั่ง help เพื่อโชว์ program ใน root directory

8. ใช้คำสั่ง stats สำหรับ แสดงผลสถานะของ SNAP board เช่น Version Snap, RAM usage, Disk usage, Uptime, Java heap usage, Server ตัวไหน Active บ้าง เป็นต้น ดังรูปที่ 3.23

```

SnapDev 1.4.0
File Edit Help

RAM usage 1905683/8389608 Bytes
Java heap usage 55528/524208 Bytes
Telnet server alive
Serial server alive
FTP server alive
/root> ipconfig
interface "enet0"
ip address 192.168.1.2
Netmask 255.255.255.0
Gateway 192.168.1.1
MAC 00:0b:b9:00:01:bc

Primary DNS N/A
Secondary DNS N/A
DNS timeout 10
Mailhost N/A
/root> stats
[No version information available]
SNAP Version May 15 2007 15:44:31
Build date Sat Aug 16 01:01:01 1975
Uptime 0 days 00:04:27
Disk usage 13075% 1638408 bytes
RAM usage 1905683/8389608 bytes
Java heap usage 55528/524208 bytes
Telnet server alive
Serial server alive
FTP server alive
/root> |

Ready
COMS 115200

```

รูปที่ 3.23 แสดงสถานะของ SNAP

9. คำสั่ง ipconfig เป็นคำสั่งที่แสดง สถานะของ Network ต่างๆ ไม่ว่าจะเป็น IP, Netmask, Gateway, Mac address, Primary DNS, Secondary DNS แสดงดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SnapDev 1.4.8
File Edit Help
su          sync          touch         unalias
umount     update       useradd      userdel
who
/root> status
No such command
/root> stats
[No version information available]
SNAP Version   May 15 2007 15:44:36
Build date     Sat Aug 16 01:01:1975
Reboot date    0 days, 00:03:23
Uptime        1207296/1638400 bytes
Disk usage    1905080/8388608 bytes
RAM usage     55528/524288 bytes
Java heap usage
Telnet server alive
Serial server alive
FTP server    alive
/root> ipconfig
Interface "enet0"
Ip address    192.168.1.2
Netmask      255.255.255.0
Gateway      192.168.1.1
MAC          00:0b:b9:00:01:be
Primary DNS   N/A
Secondary DNS N/A
DNS timeout  10
Mailhost     N/A
/root> |
Ready
COMS 115200

```

รูปที่ 3.24 หน้าจอแสดงผลลัพธ์ของคำสั่ง ipconfig

3.3.1.3 เปิด FTP Server บน SNAP Board เตรียมพร้อมสำหรับการ Upload Application file

ให้ต่อสาย Terminal console กับ SNAP board เข้ากับ Serial Port จากนั้น เปิดโปรแกรม SnapDev จากนั้นให้ Login เข้าสู่ Admin mode ตรวจสอบว่าได้มีการ เปิด FTP Server แล้วหรือยังด้วย การพิมพ์ คำสั่ง stats ที่ root prompt (/root> stats <enter>) จะได้ผลลัพธ์ดังรูปที่ 3.25

```

SnapDev 1.4.8
File Edit Help
df          echo          exit          find
flush      ftp           hd            heap
help       history      hctname      hrev
id         in           ipconfig     java
kill      ln           ls            aldir
mkdir     mkdir       acct         more
mv        netstat     nslookup     od
out       passwd      ping         ps
pwd       reboot     recv         ra
rmdir    route      setenv       send
source   startserver state        stopserver
unalias  unalias    unalias     unalias
umount   update     usredd      userdel
su       sync       touch       wall
who      write

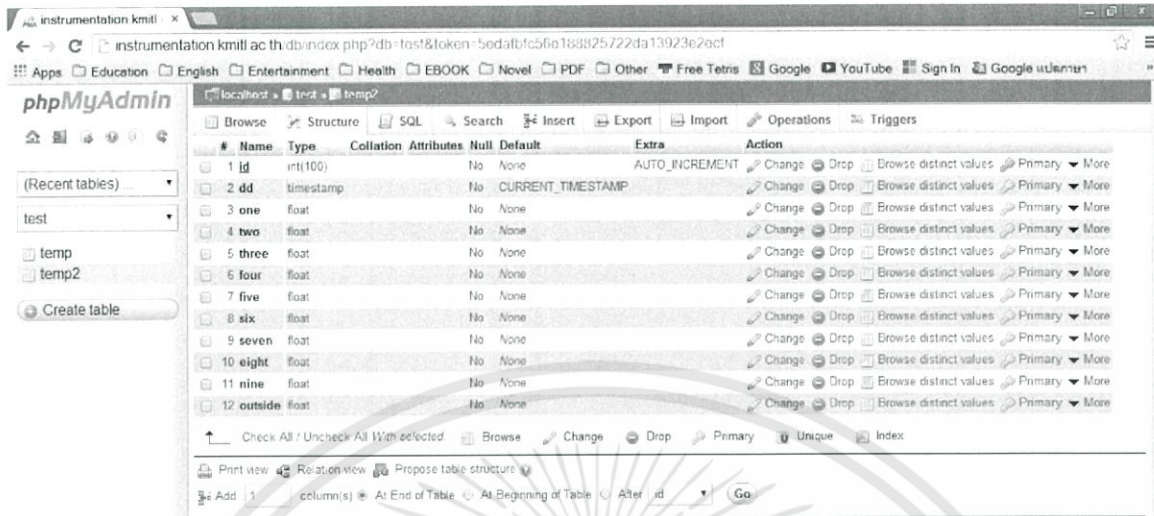
/root> status
No such command
/root> stats
[No version information available]
SNAP Version   May 15 2007 15:44:36
Build date     Sat Aug 16 01:01:1975
Reboot date    0 days, 00:03:23
Uptime        1207296/1638400 bytes
Disk usage    1905080/8388608 bytes
RAM usage     55528/524288 bytes
Java heap usage
Telnet server alive
Serial server alive
FTP server    alive
/root> |
Ready
COMS 115200

```

รูปที่ 3.25 แสดงว่ามีการ Active Server ทั้ง 3 Server คือ Telnet, Serial, FTP

จากรูปจะเห็นได้ว่ามีการ Active FTP server ไว้ให้เรียบร้อยแล้ว แต่ถ้า FTP server แสดงค่าเป็น dead หมายความว่า FTP server ยังไม่ได้ start ให้ทำการ start ด้วย คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 แสดงหน้าต่างการกำหนดคุณสมบัติของ Fields

ในกรณีที่ไม่สามารถติดต่อกับ SNAP Board ได้ ให้ตรวจสอบ IP ของ SNAP Board โดยให้ต่อ SNAP Board ผ่านโปรแกรม SnapDev จากนั้นใช้คำสั่ง ipconfig เพื่อตรวจสอบ IP Address, Netmask, Gateway ว่าถูกต้องหรือเปล่า ถ้าต้องการเปลี่ยนแปลงให้ใช้ คำสั่ง ipconfig -a ipaddress และค่าต่างๆ ตามที่ต้องการ เสร็จแล้วให้ใช้ คำสั่ง ipconfig -c เพื่อ commit ข้อมูลที่ได้เปลี่ยนแปลง ลงไปเก็บที่ Flash ไม่เช่นนั้นเมื่อที่ การ reboot หรือ ปิดเครื่องแล้วเปิดใหม่ จะได้ค่าเดิม ก่อนที่จะเปลี่ยนแปลง

ให้ทดลองส่ง HelloWorld.class ที่ Compile ไว้แล้วขึ้นไปยัง SNAP Board โดยการ Upload ผ่าน FileZilla โดยให้เปลี่ยน Upload Folder ของ SNAP ไปที่ "classes" จากนั้นให้เปลี่ยน Location Drive ไปที่ C:\Snap Developer 1.0.8\Samples\HelloWorld\ ซึ่งเป็นที่เก็บ class ไฟล์ของ Application ดังแสดงในรูปที่ 3.28



รูปที่ 3.28 การ upload file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ click ขวาที่ HelloWorld.classแล้ว เลือกหัวข้อ Upload หลังจากนั้นโปรแกรมจะ Upload เพิ่ม HelloWorld.classไปที่ SNAP Board เป็นเสร็จสิ้นการส่ง Application โปรแกรม ไปที่ SNAP

3.3.1.5 ทำการทดลอง RUN โปรแกรม HelloWorld

ต่อ SNAP Board เข้ากับ เครื่องคอมพิวเตอร์ผ่าน สาย Terminal Console เปิดโปรแกรม SnapDevจากนั้น login เข้าสู่ user root เปลี่ยน Sub Directory ไปที่ classes โดยใช้คำสั่ง cd ../classesจะได้ผลลัพธ์ดังรูปที่ 3.29

```

SnapDev 1.4.8
File Edit Help
TCP/IP startup [OK]
FTP server [OK]
Telnet server [OK]
Setting timezone to GMT
Reading /system/startup ini [OK]

localhost localdomain ()
Login
SNAP rev A/B/C/D restarting @ Sat Aug 16 00 10 29 1975
Reading /system/ish ini [OK]
Reading system.snap ini [OK]
Serial server [OK]
JVM startup [OK]
Setting host name [OK]
TCP/IP startup [OK]
FTP server [OK]
Telnet server [OK]
Setting timezone to GMT
Reading /system/startup ini [OK]

localhost localdomain ()
Login root
Password
/root> cd /classes
/classes>
Ready
COMS 115200

```

รูปที่ 3.29 แสดงการเปลี่ยน Sub Directory ไปที่ classes

ใช้ คำสั่ง ls เพื่อแสดงชื่อแฟ้มที่เก็บอยู่ใน Sub Directory classe จากนั้นให้ใช้คำสั่ง java -r HelloWorld เพื่อสั่งให้โปรแกรม HelloWorld ทำงานหลังจากกด <enter> ก็จะได้ผลลัพธ์ดังรูป 3.30

```

SnapDev 1.4.8
File Edit Help
Setting host name [OK]
TCP/IP startup [OK]
FTP server [OK]
Telnet server [OK]
Setting timezone to GMT
Reading /system/startup ini [OK]

localhost localdomain ()
Login root
Password
/root> cd /classes
/classes> ls
test/
ServerSnap class
FonyBeer class
Thanaman class
Main class
TempReader class
Provider class
ThreadCan class
ShowTempCai class
ShowTempOnWeb class
HelloWorld class
Connect class
/classes> java -r HelloWorld
Hello world
/classes>
Ready
COMS 115200

```

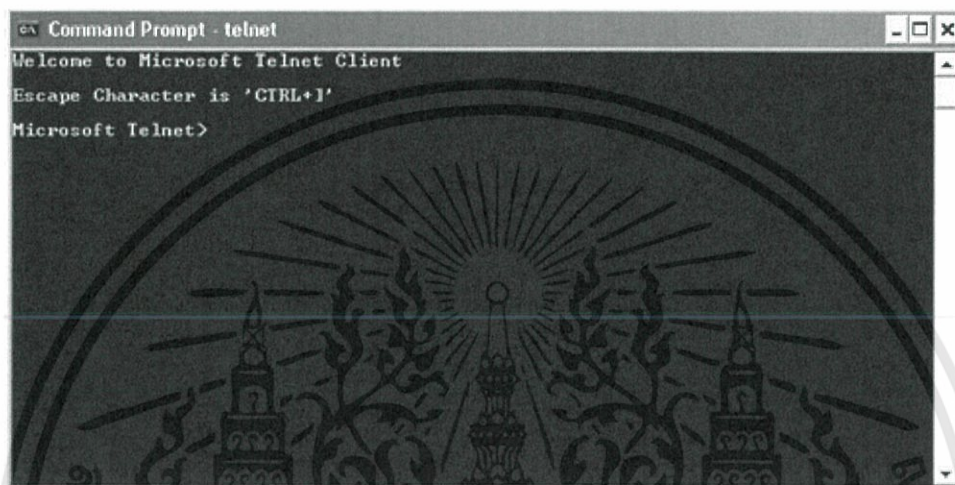
รูป 3.30 แสดงผลลัพธ์ของการ RUN โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.6 การควบคุม SNAP Board ผ่านเครือข่าย (network) ด้วยโปรแกรม

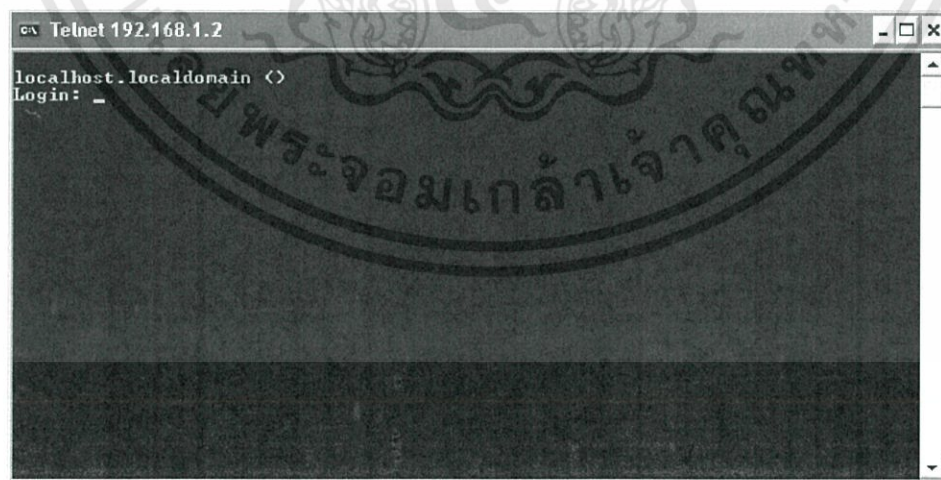
Telnet

เปิดโปรแกรม Telnet จากนั้นใช้ คำสั่ง open ตามด้วย ipaddress ของ SNAP Board จะได้ผลลัพธ์แสดงดังรูปที่ 3.31



รูปที่ 3.31 หน้าจอของโปรแกรม Telnet

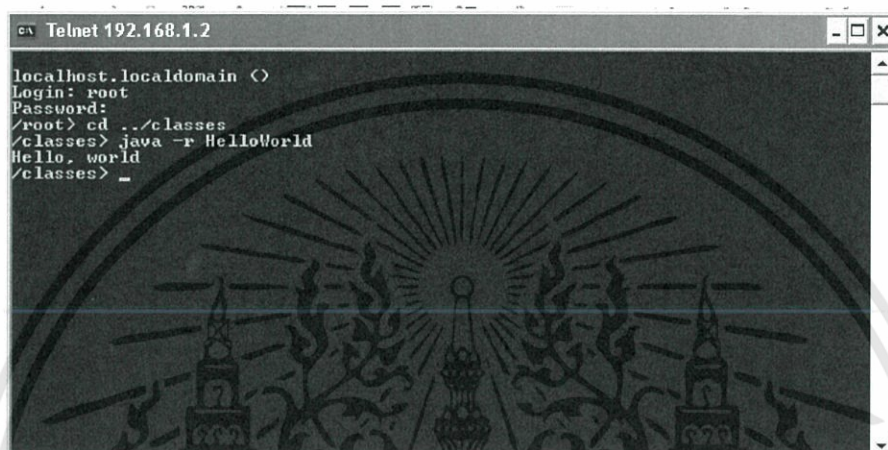
ใช้คำสั่ง open ตามด้วย ip address ของ SNAP Board จะได้หน้าจอสำหรับ login ให้กรอก login และ password (default login = root password = snap) ดังรูปที่ 3.32



รูปที่ 3.32 แสดงหน้าจอหลังจากผ่านการ login เข้ามาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการสั่ง RUN Application หรือ Configuration SNAP ผ่านโครงข่าย Networkทดสอบการ RUN โปรแกรม HelloWorld ให้เปลี่ยนSub Directory ไปที่ classes จากนั้นให้ใช้ คำสั่ง java -r HelloWorldให้ระวังเรื่องตัวอักษรตัวเล็กตัวใหญ่ด้วยครับ หลังจาก Run จะได้ดังรูปที่ 3.33



```

c:\ Telnet 192.168.1.2
localhost.localdomain <
Login: root
Password:
/root> cd ../classes
/classes> java -r HelloWorld
Hello, world
/classes> _

```

รูปที่ 3.33 แสดงผลของโปรแกรม HelloWorld ผ่าน Telnet

3.3.2 โปรแกรมของ SNAP board

ซึ่งมีอยู่ 5 ไฟล์ด้วยกัน ดังต่อไปนี้

- TempReader.java
- ThreadCan.java
- ShowTempCgi.java
- ShowTempOnWeb.java
- Index.htm.

3.3.2.1 TempReader.java

มีหน้าที่รับข้อมูลอุณหภูมิที่ส่งมาจาก Temperatrue Reader ผ่านทาง Port Can bus ซึ่งจะถูกส่งมาให้ทุกๆ หนึ่งวินาที โดยเริ่มการจาก

- การ Implement TempReader class จาก CANEventListener

“ public class TempReader implements CANEventListener “

- กำหนดคุณสมบัติของ CAN Controller บน SNAP.board เช่น อัตราความเร็ว รูปแบบการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public boolean init() {
    boolean canOpen = false;
    boolean canEnabled = false;
    try {
        cp= new CAN();
        canOpen = true;
        cf= new CANFrame();//create a CAN fream
        cp.setSampleRate(3);//single sample point at 85%, 20 clocks per bit
        cp.setTSEG1(4);
        cp.setTSEG2(3);
        cp.setSynchronizationJumpWidth(1);
        cp.setBaudRatePrescaler(BRP);

        cp.set29BitGlobalIDMask(COM_TO_CAN_ID ^ CAN_SUBMASK);
        cp.set29BitGlobalArbitrationID(COM_TO_CAN_ID);
        cp.addEventListener(this);

        cp.writeRegister(cp.REG_IER, (byte) (cp.readRegister(cp.REG_IER) | CAN.IER_DOIE));
        cp.enableController();
        canEnabled= true;
    }catch (CANException e) {
        .....
    }
    .....
}

```

รูปที่ 3.34 แสดงการกำหนด CAN Controller บน SNAP

- ตรวจสอบข้อมูลบน สายสัญญาณ ว่าเป็นข้อมูลที่ส่งมาจาก Reader 0x550 หรือ ว่า Reader 0x551 ว่าถูกต้องตามข้อกำหนดหรือเปล่า และ เป็นข้อมูลที่ต้องการส่งมาให้ SNAP (0x350) หรือเปล่า ถ้าใช่ก็ให้ Unpack ข้อมูลและนำไปเก็บไว้ในตัวแปรสำหรับเก็บข้อมูลของแต่ละ Reader ต่อไป

```

public void CANEvent(int event) {
    // TODO Auto-generated method stub
    try {
        if ((event & CAN.IR_RI) != 0){
            while (cp.receivePoll(cf)){
                if (!cf.extendedID && (cf.ID == COM_TO_CAN_ID)){
                    cf.ID = CAN_TO_COM_ID2;
                    cp.sendFrame(cf);
                }else{//if (!cf.extendedID && (cf.ID == COM_TO_CAN_ID));
                    //cf.ID = 0x550
                    if (cf.extendedID && (cf.ID == CAN_TO_COM_ID1)){// Reader 0x350
                        String datacan = new String(cf.data);
                        datacan.trim();
                        Temperature1= datacan.substring(2);
                    }//if (cf.extendedID && (cf.ID == COM_TO_CAN_ID));
                    if (cf.extendedID && (cf.ID == CAN_TO_COM_ID2)){//Reader 0x351
                        String datacan = new String(cf.data);
                        datacan.trim();
                        Temperature2= datacan.substring(2);
                    }//if (cf.extendedID && (cf.ID == CAN_TO_COM_ID2));
                    }//if (!cf.extendedID && (cf.ID == COM_TO_CAN_ID));

                }//while(cp.receivePoll(cf));
            }//if ((event & CAN.IR_RI) != 0);
        }catch (CANException e){
            .....
        }
    }
}

```

รูปที่ 3.35 แสดงการตรวจสอบข้อมูลบนสายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 ThreadCan.java

จะต้องการมีการรับข้อมูลจาก Reader ตลอดเวลา ดังนั้นจึงจะสร้างให้ TempReader ทำงานในสถานะของ Thread จะสะดวกและควบคุมการทำงานได้ดีกว่า

```
class ThreadCan implements Runnable {
    TempReader app = new TempReader();
    public void run() {
        // TODO Auto-generated method stub
        if (app.init()){
            app.readTempViaCAN();
        }//if (app.init());
    }
}
```

รูปที่ 3.36 แสดงการสร้างให้ TempReader ทำงานในสถานะของ Thread

3.3.2.3 ShowTempCgi.java

สร้างหน้าจอแสดงผลของการอ่านค่าอุณหภูมิของ Reader ทั้งสอง ให้แสดงข้อมูลบน Browser

```
public class ShowTempCgi implements HttpCgi {
    public ShowTempCgi(){
    }
    public String getName(){
        return("show.cgi");
    }
    public void execute(Hashtable variables, OutputStream os) throws IOException, HttpException{
        String s;
        s = "Content-Type: text/html\r\n\r\n\r\n";
        s += "<HTML><TITLE>Result for Temperature.</TITLE>\n";
        s += "<BODY>\n";
        s += "<H1>Temperature Reader (0x350) = "+TempReader.Temperature1+"</H1><br>\n";
        s += "<H1>Temperature Reader (0x351) = "+TempReader.Temperature2 + "</H1><br>\n";
        s += "</BODY></HTML>\n";
        os.write(s.getBytes());
    }
}
```

รูปที่ 3.37 แสดงการสร้างหน้าจอแสดงผลของการอ่านค่าอุณหภูมิตน Browser

3.3.2.4 ShowTempOnWeb.java

เป็น Main โปรแกรม 2 Thread ด้วยกัน คือ CAN Thread และ HTTPServer Thread ซึ่ง CAN Thread ใช้สำหรับการรับข้อมูลจาก Temperature Reader ซึ่งจะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งมาทุกๆ 1 วินาที ส่วน HTTPServer Thread จะให้บริการเกี่ยวกับการแสดงผลของข้อมูล อุณหภูมิทั้ง 2 Reader เมื่อมีการร้องขอมาจาก Browser เมื่อมีการร้องขอเข้ามา

```
public static void main(String[] args) {
    ThreadCan can = new ThreadCan();
    Thread c = new Thread(can);
    c.start();

    HttpServer hs = new HttpServer();
    Thread t = new Thread(hs);
    hs.addCgi(new ShowTempCgi());
    t.start();
}
```

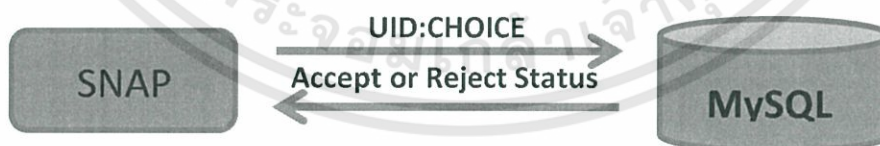
รูปที่ 3.38 แสดงการสร้างหน้าจอแสดงผลของการอ่านค่าอุณหภูมิตั้งบน Browser

3.3.3 การสร้างหน้าจอแสดงผลค่าอุณหภูมิตั้งบน Browser

ความต้องการของระบบ

- PHP Engine version 5 หรือสูงกว่า
- MySQL Database Server version 5 หรือสูงกว่า
- Web Server เช่น Apache
- การทำงานของระบบ (flowchart)

3.3.3.1 การออกแบบฐานข้อมูลใน MySQL Server

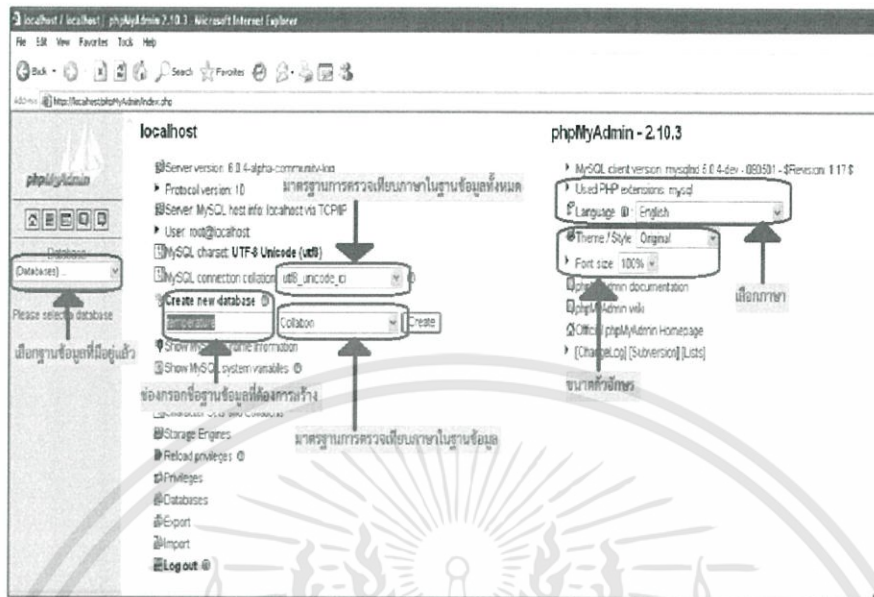


รูปที่ 3.39 Context Diagram

การสร้างฐานข้อมูลในส่วนนี้ สร้างขึ้นเพื่อใช้ในการสร้างฐานข้อมูลของค่า อุณหภูมิที่วัดได้ โดยใช้ MySQL ในการสร้างฐานข้อมูล ผ่าน โปรแกรม PhpMyAdmin

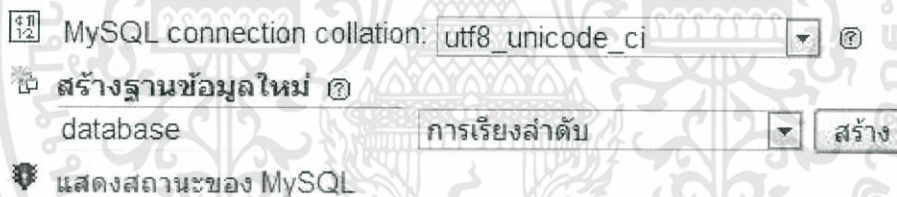
1. PhpMyAdmin มีส่วนประกอบดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



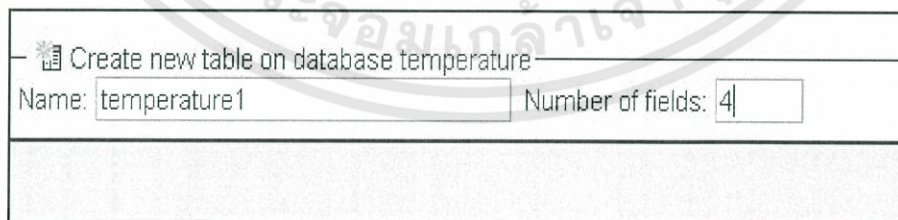
รูปที่ 3.40 อธิบายหน้าแรกของ PhpMyAdmin

1. การสร้างชื่อฐานข้อมูล



รูปที่ 3.41 การสร้างชื่อฐานข้อมูล

2. การสร้างตารางในฐานข้อมูล



รูปที่ 3.42 สร้างตารางโดยกำหนดชื่อและจำนวน fields

หลังจากที่กำหนดชื่อตารางและกำหนด Fields เรียบร้อยแล้ว ต่อไปเป็นการกำหนด คุณสมบัติของ Fields โดยยกตัวอย่างข้อมูลที่ถูกต้องการสร้าง ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ลำดับของเซอเซอร์(Number)
2. ค่าอุณหภูมิ (Temp)
3. วันที่ (Date)
4. เวลา (Time)

Server: localhost Database: temperature Table: temperature1

Field	Type	Length/Values	Collation	Attributes
Number	INT			
Temp	INT			
Date	DATE			
Time	TIME			

รูปที่ 3.43 ภาพแสดงหัวข้อเพื่อกำหนด fields และ คุณสมบัติต่างๆ

Table 'temperature' 'temperature1' has been created.

SQL query:
 CREATE TABLE 'temperature1' (
 'Number' INT NULL AUTO_INCREMENT PRIMARY KEY,
 'Temp' INT NOT NULL,
 'Date' DATE NOT NULL,
 'Time' TIME NOT NULL,
) ENGINE = MYISAM ;

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> Number	int(11)			No		auto_increment	[Icons]
<input type="checkbox"/> Temp	int(11)			No			[Icons]
<input type="checkbox"/> Date	date			No			[Icons]
<input type="checkbox"/> Time	time			No			[Icons]

Check All / Uncheck All With selected: [Icons]

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After Number Go

Indexes:				Space usage		Row Statistics	
Keyname	Type	Cardinality	Action	Field	Type	Usage	Statements Value
PRIMARY	PRIMARY	0	[Icons]	Number	Data	0 B	
Create an index on 1 columns Go				Index	Index	0 B	
				Total		0 B	

รูปที่ 3.44 ตารางการประเมิน

3.3.3.2 ไฟล์โปรแกรมของระบบ

ไฟล์ของระบบมีอยู่ด้วยกัน 2 ไฟล์ ซึ่งในแต่ละไฟล์ทำหน้าที่ต่างกัันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. test_query.php เป็นไฟล์ที่ใช้เก็บค่าอุณหภูมิของห้องปฏิบัติการแต่ละห้อง
2. show_temp.php เป็นไฟล์ที่ใช้แสดงผลอุณหภูมিবน Web Browser

Today is 13-Feb-2014 11:07:17: am

ตารางแสดงค่าอุณหภูมิของแต่ละห้องปฏิบัติการในช่วงเวลาต่างๆ

Date/Temperature	Control System Lab.	Process Instrument Lab.	Factory Automation Lab.	Electrical Machine Lab.	Basic Measurement Lab.	Electronics Circuit Lab.	Microcontroller Lab.	Sensor Transducer Lab.	Instrument System Lab.	Outside
2014-02-05 18:35:30	11 °C	22 °C	33 °C	44 °C	55 °C	66 °C	77 °C	88 °C	99 °C	50°C
2014-02-05 18:37:12	11.1 °C	12.2 °C	13.2 °C	14.2 °C	12.5 °C	15.6 °C	16.7 °C	17.8 °C	22.1 °C	18.5°C
2014-02-05 18:39:59	26.3 °C	22.5 °C	23.1 °C	24.7 °C	28.6 °C	26.8 °C	30.2 °C	25.4 °C	22.9 °C	24.7°C
2014-02-06 17:41:02	12 °C	22 °C	23 °C	21 °C	24 °C	25 °C	33 °C	32 °C	31 °C	22°C
2014-02-06 17:42:20	23.4 °C	22.1 °C	25.6 °C	24 °C	27.8 °C	26.5 °C	21.9 °C	29.3 °C	28.2 °C	24°C

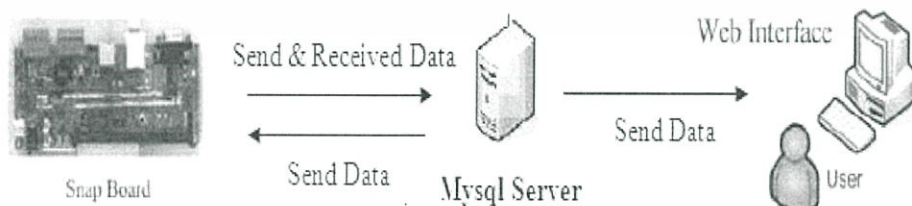
รูปที่ 3.45 ตารางแสดงค่าอุณหภูมิ

3.3.3.3 วิธีการสร้างไฟล์ระบบโปรแกรม

การพัฒนาโปรแกรมเลือกใช้ภาษา PHP เนื่องจากเป็นภาษาที่นิยมใช้ในการสร้าง Web Application และเป็น Open Source ที่ไม่ต้องมีค่าใช้จ่ายในการนำมาพัฒนา และเก็บข้อมูลโดยใช้ MySQL Database ซึ่งเป็นโปรแกรมระบบฐานข้อมูลที่สามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่ายเช่นเดียวกัน การสร้างไฟล์ระบบโปรแกรมจะสร้างจากภาษา PHP โดยมีการสร้างไฟล์ต่างๆ ตามลำดับขั้นตอนดังนี้

1. connect.php สำหรับการติดต่อกับฐานข้อมูล ซึ่งเป็นไฟล์ใช้ร่วมกับไฟล์อื่นๆ ทั้งหมด เนื่องจากทุกไฟล์มีการใช้งานและติดต่อกับระบบฐานข้อมูล
2. result.php แสดงค่าอุณหภูมิที่ได้รับจากเซนเซอร์วัดอุณหภูมิ

3.4 เขียนโปรแกรมภาษา Java บน SNAP เพื่อติดต่อกับเครื่องแม่ข่าย



รูปที่ 3.46 การทำงานของระบบโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การออกแบบระบบและการพัฒนาระบบมีการทำงานดังต่อไปนี้

- SNAP รับค่าอุณหภูมิมาจากรีจิสเตอร์อุณหภูมิ
- SNAP ส่งข้อมูลไปยัง MySQL Server
- MySQL Server จะทำการเช็คข้อมูลแล้วนำข้อมูลที่ได้อัพเดทลงในฐานข้อมูล
- หากเกิดข้อผิดพลาดในการอัปเดตข้อมูล MySQL Server จะทำการส่งสถานะ

error

- กลับไปยัง SNAP เพื่อรับข้อมูลใหม่
- เมื่อบันทึกข้อมูลลงในฐานข้อมูลของ Server แล้ว ผู้ดูแลระบบและผู้ใช้งาน
- สามารถเรียกดูผลการประเมินผ่านทางเว็บเบราว์เซอร์ได้
- จบการทำงานของระบบ

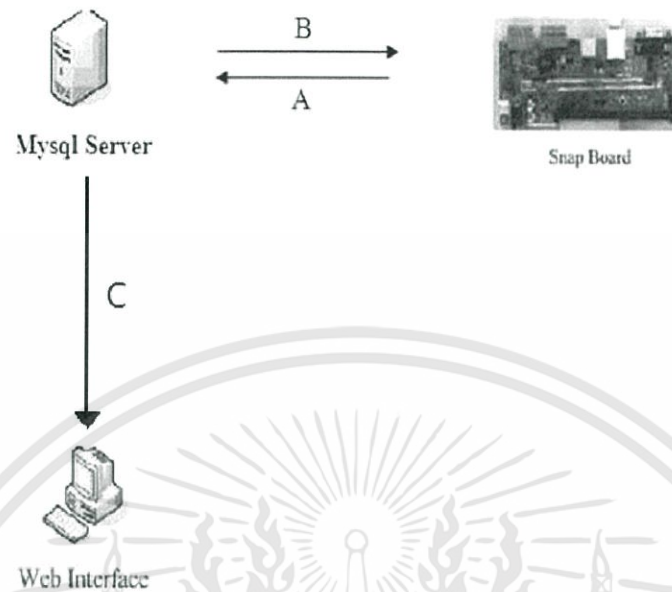
3.4.1.1 การให้ SNAP ส่งข้อมูลไปยังเครื่อง MySQL Server มีการออกแบบได้

ดังต่อไปนี้

การเชื่อมต่อระหว่าง SNAP กับ MySQL จะมีการเชื่อมต่อดังนี้

- การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ซึ่งมี IP Address 192.168.1.141 กับ SNAP ซึ่งมี IP Address 192.168.1.2 โดยเครื่องคอมพิวเตอร์จะทำการเชื่อมต่อกับ SNAP ผ่านโปรแกรม Telnet โดยผู้ใช้สามารถใส่ Username และ Password ได้ หลังจากเครื่องคอมพิวเตอร์เชื่อมต่อกับ SNAP
- การเชื่อมต่อระหว่าง SNAP ซึ่งมี IP Address 192.168.2.2 กับ MySQL Server ซึ่งมี IP Address 192.168.1.141 โดย SNAP จะทำการเปิด Port 1027 ติดต่อไปยัง MySQL Server ซึ่งจะเปิด Port 8686 โดยการเขียนโปรแกรมการเปิด Port 8686 ไว้ในโปรแกรม ซึ่ง SNAP จะส่ง user id และสถานะของข้อมูลไปยัง MySQL Server เมื่อ MySQL Server ได้รับการ Request คือ เลข 1 จาก SNAP ตัว MySQL Server จะทำการเช็คสถานะของข้อมูลว่าถูกต้องหรือไม่ ถ้าสถานะของข้อมูลถูกต้อง ระบบจะทำการอัปเดตข้อมูลของ อุณหภูมิ พร้อมกับ insert ข้อมูล user ลงในตารางฐานข้อมูล และเมื่ออัปเดตค่าแล้ว MySQL Server จะส่งเลข 002 กลับไปยัง SNAP ถ้าสถานะของข้อมูลผิด MySQL Server จะส่งเลขเป็น 000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- A Response UID: CHOICE to Mysql Server
- B Mysql Server Send Status (000 : Is Error , 002 : Is success) to Snap Server
- C Generate Report

รูปที่ 3.47 Diagram ของระบบ

ในส่วนของการแสดงผลระบบจะสามารถดูค่าอุณหภูมิโดยเข้า Link ของ MySQL Server โดยใช้ฟังก์ชันของ PHP ในการดึงข้อมูลมาแสดงผล

3.5 โปรแกรมในส่วนของ Server

3.5.1 โปรแกรมในส่วนนี้จะประกอบด้วย 3 ส่วน คือ Main Program, Function Program และ Database Program

3.5.1.1 Main Program

เป็นการประกาศ FunctionMain โดยใช้ object type server เรียก functionRun() เพื่อทำงานโปรแกรมนี้เป็นโปรแกรมหลักที่ใช้ในการประกาศ Functionmain โดยใช้ object type server เรียก functionRun() เพื่อทำงาน เป็นโปรแกรมหลักที่ใช้ในการเริ่มโปรแกรม และทำงานในส่วนของโปรแกรมอื่นๆ

3.5.1.2 Function Program

มี 2 ส่วน คือ ส่วนของการติดต่อ Server และการตัดข้อความของ Message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ทาง Snap Server ส่งค่าเข้ามา

- เป็นส่วนของการติดต่อระดับ socket ระหว่าง Serve กับ Snap
- การตัดข้อความของ message ที่ทาง Snap Serve ส่งเข้ามา

3.5.1.3 โปรแกรมส่วนของ Database

- การเลือกข้อมูลขึ้นมาเพื่อตรวจสอบและเก็บข้อมูลแล้วนำค่าไปอัปเดต
- อัปเดตข้อมูลบน Web browser



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการดำเนินงาน

4.1 บทนำ

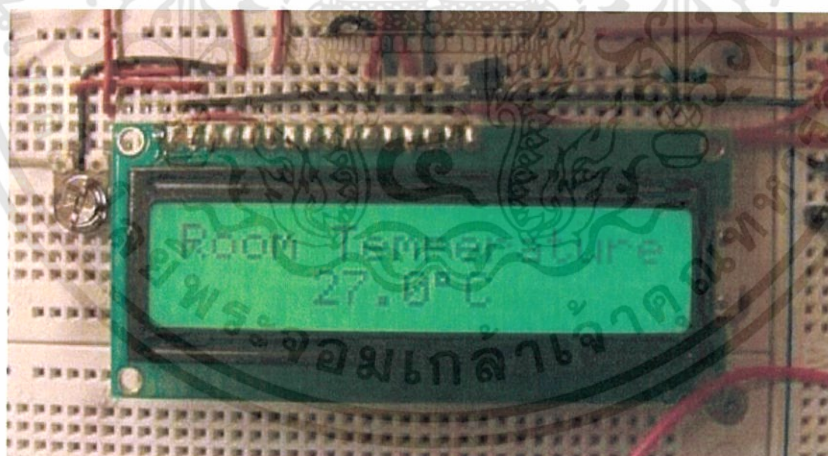
ในบทนี้จะกล่าวถึงรายละเอียดการทดลองและผลการทำงานของโครงการ โดยเครื่องตรวจวัดอุณหภูมิจะต้องสามารถตรวจรับค่าอุณหภูมิและแสดงผลของอุณหภูมิตนจอ LCD ได้ และสามารถส่งค่าอุณหภูมินั้นไปที่ SNAP ผ่านการส่งข้อมูลแบบ CAN และสามารถเรียกดูค่าอุณหภูมิที่เว็บเบราว์เซอร์ได้

4.2 การทดลองและผลการทดลอง

4.2.1 การทดลองแสดงค่าของอุณหภูมิตนจอ LCD

ในการทดลองนี้เป็นการทดลองให้ตัวเซนเซอร์ตรวจวัดอุณหภูมิหรือ IC DS18B20 ตรวจวัดค่าอุณหภูมิที่อุณหภูมิห้องและแสดงค่าของอุณหภูมิตนจอ LCD

เมื่อทำการเปิดเครื่องตรวจวัดอุณหภูมิ ตัวเซนเซอร์ตรวจวัดอุณหภูมิ หรือ IC DS18B20 จะทำการวัดค่าอุณหภูมิ แล้วส่งค่าอุณหภูมิตนจอ LCD ดังรูปที่ 4.1



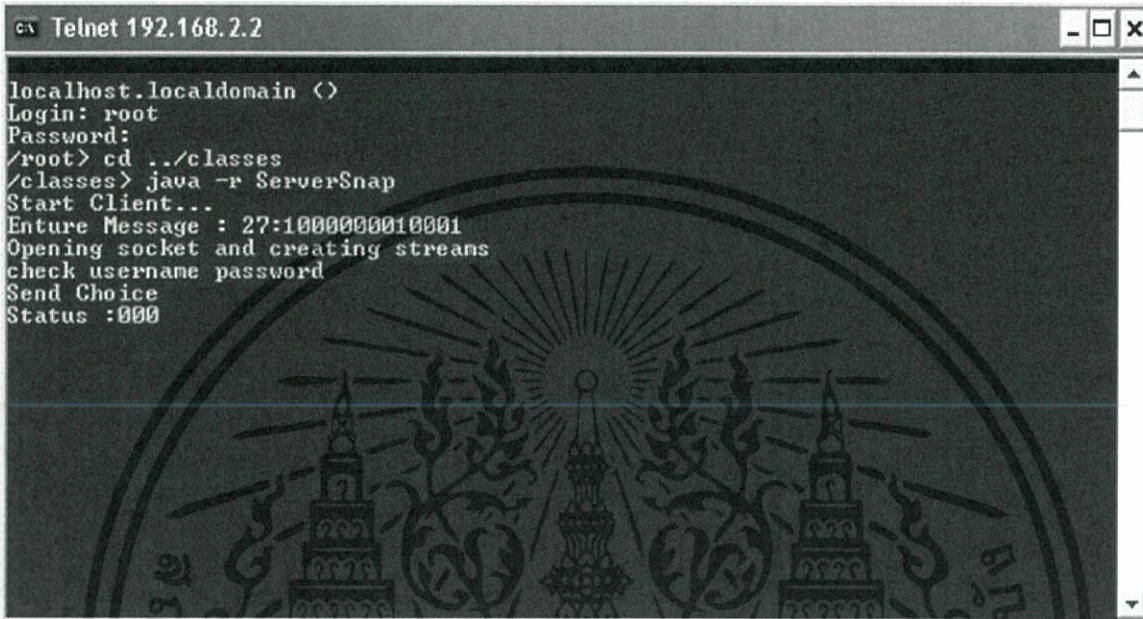
รูปที่ 4.1 เครื่องตรวจวัดอุณหภูมิ

4.2.2 การทดลอง SNAP

ในการทดลองนี้เป็นการทดลองการส่งค่าของอุณหภูมิตนจอ LCD ที่รับมาจาก CANBUS ไปยัง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงค่าอุณหภูมิที่รับมาจากเครื่องวัดอุณหภูมิผ่าน CANBUS โดยโปรแกรม Telnet ซึ่งค่าที่รับมาเป็น 27 องศาเซลเซียส ดังรูปที่ 4.2



```

c:\ Telnet 192.168.2.2
localhost.localdomain (>)
Login: root
Password:
/root> cd ../classes
/classes> java -r ServerSnap
Start Client...
Enture Message : 27:1000000010001
Opening socket and creating streans
check username password
Send Choice
Status :000
  
```

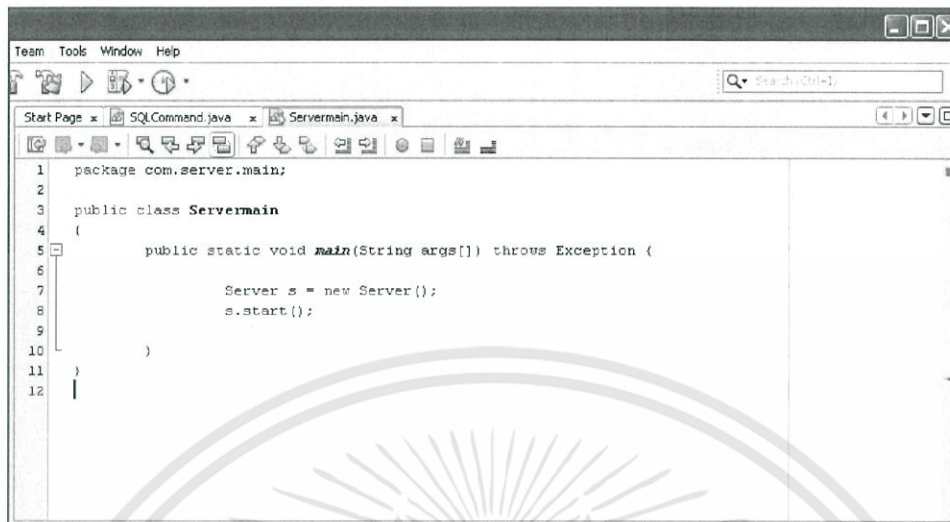
รูปที่ 4.2 แสดงค่าของอุณหภูมิที่รับมาจาก CAN BUS ไปยัง Server

4.2.3 การทดลอง Server

จากการออกแบบระบบในบทที่ 3 สามารถนำเอาสิ่งที่ผ่านการออกแบบมาพัฒนาเป็นโปรแกรมช่วยสร้างรายงาน ซึ่งสามารถแสดงให้เห็นแต่ละหน้าจอ ทำให้ง่ายต่อความเข้าใจต่อผู้ใช้ระบบ ซึ่งจะประกอบไปด้วยขั้นตอนการใช้งานของระบบ

4.2.3.1 การทดลองการรันโปรแกรม Servermain บนเครื่อง Server

ทำการ Run โปรแกรม Servermain บนเครื่อง server โดยโปรแกรม Servermain จะทำการติดต่อกับฐานข้อมูล ด้วย user = root , password = 1234 โดยอัตโนมัติ



```

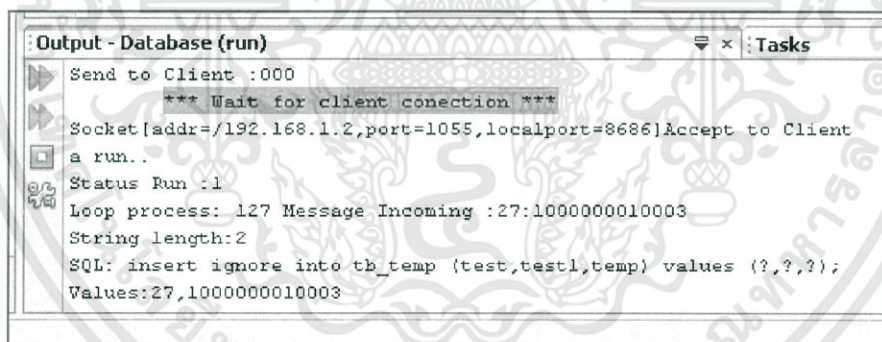
1 package com.server.main;
2
3 public class Servermain
4 {
5     public static void main(String args[]) throws Exception {
6
7         Server s = new Server();
8         s.start();
9
10    }
11 }
12

```

รูปที่ 4.3 แสดงโปรแกรม Servermain

4.2.3.2 ผลการทดลองการรันโปรแกรม Servermain บนเครื่อง Server

เมื่อโปรแกรม Servermain แสดงผลว่า *** Wait for client conection
 *** หมายความว่า Server พร้อมรับข้อมูลจาก SNAP แล้ว



```

Output - Database (run)
Send to Client :000
*** Wait for client conection ***
Socket[addr=/192.168.1.2,port=1055,localport=8686]Accept to Client
a run..
Status Run :1
Loop process: 127 Message Incoming :27:1000000010003
String length:2
SQL: insert ignore into tb_temp (test,test1,temp) values (?,?,?);
Values:27,1000000010003

```

รูปที่ 4.4 แสดงผลการรันโปรแกรม Servermain

4.2.3.3 ผลการทดลองการสร้างฐานข้อมูล

โดยที่ค่าที่เข้ามานั้น จะมีข้อมูลของ วันที่ เวลา และค่าของอุณหภูมิ ที่ส่งมาจาก
 Servermain

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

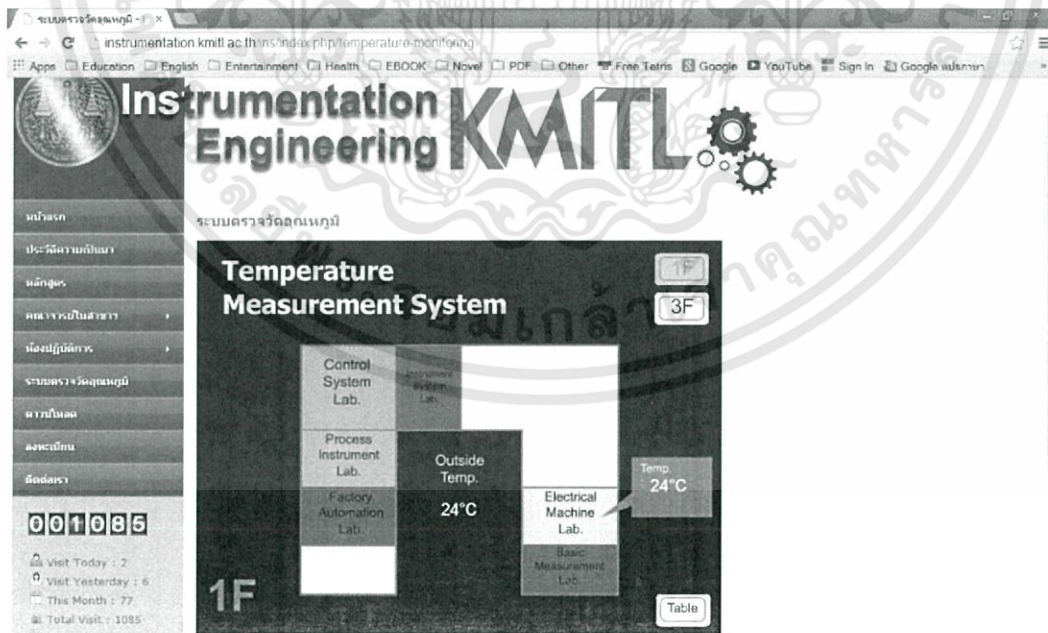


รูปที่ 4.5 แสดงผลการรับข้อมูลลง Database

4.2.3.4 ผลการทดลองการสร้าง Web browser

ผลการทดลองการสร้าง Web browser เป็นการแสดงค่าของอุณหภูมิที่รับมาของตัวเซนเซอร์วัดอุณหภูมิ โดยที่จะแสดง วันที่ เวลา และค่าอุณหภูมิมีหน่วยเป็นองศาเซลเซียส

4.2.3.5 ผลการทดลองการเรียกดูค่าของอุณหภูมิก่อนหน้า



รูปที่ 4.6 แสดงผลค่าของอุณหภูมิก่อนหน้า Web browser โดยใช้ Adobe FLASH CS3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผล และปัญหา

5.1 สรุปผล

จากการทดลองของโครงการระบบตรวจวัดอุณหภูมิ ผลที่ได้ของโครงการเป็นไปตามวัตถุประสงค์ กล่าวคือ เมื่อเซนเซอร์ตรวจวัดอุณหภูมิตัวรับค่าอุณหภูมิแล้ว สามารถเรียกดูข้อมูลดังกล่าวบนระบบเครือข่ายได้

5.2 ปัญหา

5.2.1 ปัญหาในการทดลองและแนวทางแก้ไขมีดังนี้

- การเขียนโปรแกรมในการสร้างชุดคำสั่งเพื่อให้เครื่องวัดอุณหภูมิตัวรับค่าอุณหภูมิ และส่งค่าอุณหภูมิเพื่อแสดงค่าของอุณหภูมิต่อระบบเครือข่ายนั้น ควรศึกษาอุปกรณ์และการต่อวงจรต่าง ๆ อย่างละเอียด เนื่องจากชุดคำสั่งแต่ละคำสั่งมีเงื่อนไขที่แตกต่างออกไป เพราะถ้าส่งชุดคำสั่งผิดข้อมูลที่ได้กลับมาก็จะเกิดการผิดพลาด
- การเขียนโปรแกรมการส่งข้อมูลผ่าน CAN BUS ควรศึกษารูปแบบวิธีการส่งและคอมไพเลอร์ที่ใช้ในการเขียนโปรแกรมการส่งข้อมูล เนื่องจากคอมไพเลอร์แต่ละตัวจะมีรูปแบบการเขียนโปรแกรมการส่งข้อมูลผ่าน CAN BUS ไม่เหมือนกัน
- การส่งข้อมูลผ่าน CAN BUS โหนดทั้งสองจะต้องกำหนดค่าอัตราการส่งข้อมูลต้องเท่ากันการส่งข้อมูลต้องเท่ากันการส่งข้อมูลจึงจะไม่ผิดพลาด
- เนื่องจาก SNAP เป็น UNIX และออกแบบให้เขียนโปรแกรมควบคุมได้เพียงภาษาเดียวเท่านั้น ซึ่งทำให้ยากต่อการเข้าใจและพัฒนาโปรแกรม

บรรณานุกรม

กิตติ ภัคดีวัฒน์กุล. 2546. คัมภีร์ JAVA เล่ม1. กรุงเทพฯ: บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด

บัญชา ปะสีละเตสัง. 2553. พัฒนาเว็บแอปพลิเคชันด้วย PHP ร่วมกับ MySQL และ Dreamweaver. กรุงเทพฯ: บริษัท ซีเอ็ดยูเคชั่นจำกัด(มหาชน)

Microchip Technology Inc. 2002. Controller Area Network (CAN) Basics. [Online]. Available: [http://www.ee.uidaho.edu/ee/classes/ECE341/refs/AN713...\(CAN\)%20Basics.pdf](http://www.ee.uidaho.edu/ee/classes/ECE341/refs/AN713...(CAN)%20Basics.pdf)

Imsys Technologies. SNAP - Simple Network Application Platform. [Online]. Available: <http://www.imsystech.com>