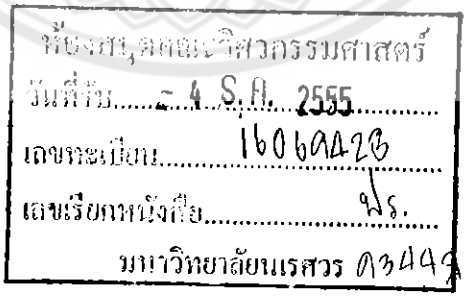
การพัฒนาอุปกรณ์บันทึกข้อมูลและส่วนประสานกราฟฟิกกับผู้ใช้

DATA LOGGER AND GRAPHIC USER INTERFACE DEVELOPMENT

นายคาวี	ศรีภูธร	รหัส 51364248
นายวงศกร	ชวลิตกรัณย์กุล	รหัส 51364477
นางสาวพรรัตน์	ศรีโสภา	รหัส 51383973



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การพัฒนาอุปกรณ์บันทึกข้อมูลและส่วนประสานกราฟฟิกกับผู้ใช้			
ผู้ดำเนินโครงงาน	นายคาวี	ศรีภูธร	รหัส	51364248
	นายวงศกร	ชวลิตกรัณย์กุล	รหัส	51364477
	นางสาวพรรัตน์	ศรีโสภา	รหัส	51383973
ที่ปรึกษาโครงงาน	ดร.พรพิศุทธิ์ วรจิรันตน์			
สาขาวิชา	วิศวกรรมไฟฟ้า			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2554			

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

..... ที่ปรึกษาโครงงาน

(ดร.พรพิศุทธิ์ วรจิรันตน์)

..... กรรมการ

(ดร. พนัส นัถฤทธิ์)

..... กรรมการ

(ดร.พิสุทธิ์ อภิชยกุล)

ชื่อหัวข้อโครงงาน	การพัฒนาอุปกรณ์บันทึกข้อมูลและส่วนประสานกราฟฟิกกับผู้ใช้			
ผู้ดำเนินโครงงาน	นายคาวี	ศรีภูธร	รหัส	51364248
	นายวงศกร	ชวลิตกรัณย์กุล	รหัส	51364477
	นางสาวพรรัตน์	ศรีโสภา	รหัส	51383973
ที่ปรึกษาโครงงาน	ดร.พรพิศุทธิ์ วรจิรันตน์			
สาขาวิชา	วิศวกรรมไฟฟ้า			
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์			
ปีการศึกษา	2554			

บทคัดย่อ

ปริญญานิพนธ์เล่มนี้ ได้พัฒนาอุปกรณ์ที่ใช้สำหรับการบันทึกข้อมูลโดยการนำไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์รุ่น AT-MEGA 2560 มาใช้ในการเขียนโปรแกรมและบันทึกข้อมูล จากเซ็นเซอร์โดยสัญญาณจะต้องมีระดับแรงดัน 0-5 โวลต์ อุปกรณ์บันทึกข้อมูลนี้สามารถต่อเซ็นเซอร์ได้ 16 ตัว และแต่ละช่องสัญญาณสามารถกำหนดเวลาบันทึกข้อมูลได้แตกต่างกันโดยมีช่วงเวลา 1 วินาที ถึง 24 ชั่วโมง และสามารถเลือกให้บันทึกสัญญาณต่อเนื่องได้โดยมีอัตราการบันทึก 9 ครั้งต่อวินาทีโดยบันทึกข้อมูล 1 Channel ข้อมูลที่ได้จะถูกบันทึกผลลงใน SD-Card ผลที่บันทึกอยู่ใน SD-Card นั้นจะแสดงผลออกมาเป็นกราฟทางหน้าจอคอมพิวเตอร์ส่วนบุคคล เมื่อนำ SD-Card มาติดต่อกับคอมพิวเตอร์ ในส่วนของซอฟแวร์ที่รองรับการประมวลผลและแสดงผลในรูปแบบหน้าต่างที่เชื่อมต่อกับผู้ใช้งาน

จากการทดสอบการทำงานของอุปกรณ์บันทึกข้อมูลที่พัฒนาขึ้น พบว่า อุปกรณ์ที่ได้พัฒนาขึ้นมานั้นสามารถบันทึกข้อมูลในช่วงเวลาต่างๆตามที่ได้กำหนดไว้พร้อมกัน 16 ตัวโดยมีการต่อแหล่งจ่ายเพิ่มจากภายนอก

Project title	Data Logger and Graphic User Interface Development		
Name	Mr.Kawee	Sriphutorn	ID. 51364248
	Mr.Vongsakorn	Chavalitkarunkul	ID. 51364477
	Miss.Pornrat	Srisopa	ID. 51383973
Project advisor	Ponpisut Worrajiran, Ph.D		
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2011		

Abstract

Data Logger and Graphic User Interface Development were by using microcontroller AVR ATMEGA 2560 and Visual Basic 6.0. The instrument can record data from sensor signals which has voltage between 0 - 5 volt. This data logger can connect to 16 sensors with additional power supply. In addition, each channel's recording interval can be set to different periods. The intervals can be 1 second to 24 hour or continuous recording which records 9 times per second will be recorded in SD-Card or on the microcontroller board. After the recording session, the data can be downloaded to the computer and the result can be display as a graph or saved in form of Microsoft Excel file

กิตติกรรมประกาศ

ในการทำปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขอกราบขอบพระคุณ ดร.พรพิศุทธิ์ วรจิรันตน์เป็นอย่างสูงที่ได้กรุณาสละเวลามาเป็นอาจารย์ที่ปรึกษาโครงงานและได้ให้ความช่วยเหลือในการให้คำแนะนำและตรวจแก้ไขในการทำปริญญานิพนธ์ฉบับนี้ให้สำเร็จลงได้ด้วยดี และขอกราบขอบพระคุณ คณะกรรมการการสอบโครงงานวิศวกรรม เป็นอย่างสูงที่ได้ให้ความรู้และแก้ไขรวมทั้งข้อเสนอแนะที่เป็นประโยชน์ซึ่งทำให้เนื้อหาของปริญญานิพนธ์ฉบับนี้สมบูรณ์ยิ่งขึ้น

ผู้เขียนขอกราบขอบพระคุณ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ที่อำนวยความสะดวกในการให้ยืมอุปกรณ์และเครื่องมือวัดมาใช้งาน จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

เหนือสิ่งอื่นใด คณะผู้ดำเนินโครงงานขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ผู้มอบความรัก ความเมตตา สติปัญญา คอยให้กำลังใจ และสนับสนุนด้านการศึกษาแก่คณะผู้ดำเนินโครงงาน และขอขอบคุณญาติพี่น้องของคณะผู้ดำเนินโครงงานที่ไม่ได้กล่าวถึงไว้ ณ ที่นี้ด้วย

นายคาวี ศรีภูธร
นายวงศกร ชวลิตกรัณย์กุล
นางสาวพรรัตน์ ศรีโสภา

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญ 1
1.2 วัตถุประสงค์ 2
1.3 ประโยชน์ที่คาดว่าจะได้รับ 2
1.4 ขอบเขตโครงงาน 2
1.5 ขั้นตอนการดำเนินโครงงาน 2
1.6 แผนการดำเนินงาน 3
1.7 งบประมาณในการดำเนินโครงงาน 3

บทที่ 2 หลักการและทฤษฎี 4
2.1 คุณสมบัติและลักษณะของอุปกรณ์บันทึกข้อมูล 4
2.2 บอร์ดไมโครคอนโทรลเลอร์ 5
2.3 การ์ดหน่วยความจำ SD-Card แบบ Micro-SD 6
2.4 Seriperal Periperal Intrface 6
2.5 Flash Memory 8
2.6 SD-Card 9
2.7 การทำงานของ SD-Card 10
2.8 Analog to Digital Conversion 11
2.9 ET-MINI DS 1307 16

สารบัญ (ต่อ)

หน้า

2.10 เครื่องโปรแกรม ET-AVRISP mkII 18
2.11 โปรแกรม Visual Basic 6.0 19
2.12 โปรแกรม AVR Studio 19
2.13 โปรแกรม WinAVR 19

บทที่ 3 วิธีการดำเนินโครงงาน 20
3.1 การพัฒนาโปรแกรมในส่วนของไมโครคอนโทรลเลอร์ 21
3.1.1 การทำงานในส่วนของโปรแกรมหลัก 21
3.1.2 การเขียนโปรแกรมเพื่อใช้งาน ADC 23
3.1.3 การเขียนโปรแกรมเพื่อใช้งาน SD-Card 23
3.1.4 การส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่านทาง พอร์ตอนุกรม 27
3.2 การออกแบบและพัฒนาโปรแกรม Visual Basic 6.0 34
3.2.1 ตั้งค่าการทำงานของอุปกรณ์เก็บข้อมูล 35
3.2.2 การแสดงผลข้อมูล 39
3.3 วิธีการทดลอง 41
3.3.1 การทดสอบส่วนซอฟแวร์ 42
3.3.2 การทดสอบการใช้งานอุปกรณ์เก็บข้อมูล 42

บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง 49
4.1 ผลการทดลองส่วนซอฟแวร์ 49
4.1.1 โปรแกรมที่ใช้บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMAGA 2560 49
4.1.2 วิธีการใช้งานอุปกรณ์บันทึกข้อมูล 50
4.2 ผลการทดลองการใช้งานอุปกรณ์บันทึกข้อมูล 54
4.3 สรุปผลการทดลอง 79

บทที่ 5 สรุปผลการดำเนินโครงงาน 80
5.1 ปัญหาและแนวทางการแก้ปัญหา 80
5.2 ข้อเสนอแนะในการพัฒนา 80

สารบัญ (ต่อ)

หน้า

เอกสารอ้างอิง 81

ภาคผนวก ก คู่มือการใช้งานเซ็นเซอร์ตรวจจับระยะทาง GP2Y0A21YK0F 82

ภาคผนวก ข คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE MEGA 2560 92

ประวัติผู้ดำเนินโครงงาน 121

สารบัญตาราง

ตารางที่ หน้า

3.1 รูปแบบการส่งข้อมูลแบบ HDLC Protocol 9 byte 28

3.2 คำสั่ง PC อ่าน Text file จาก AVR 31

3.3 คำสั่ง AVR ส่ง Text file ให้ PC 31

3.4 คำสั่ง PC ตอบกลับ Command 0x03 33

3.5 คำสั่ง PC ส่งไป Delete Text file 33

3.6 คำสั่ง AVR ตอบกลับ Delete Text file ให้ PC 33

3.7 คำสั่ง PC ส่งค่า Channel setting 33

4.1 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 10 เซนติเมตรของ Channel 1 56

4.2.ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 10 เซนติเมตรของ Channel 2 59

4.3 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 20 เซนติเมตรของ Channel 1 62

4.4 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 20 เซนติเมตรของ Channel 2 65

4.5 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดันไฟฟ้า 2 V ของ Channel 1 68

4.6 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที แรงดันไฟฟ้า 2 V ของ Channel 2 70

4.7 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดันไฟฟ้า 4 V ของ Channel 1 72

4.8 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาในการบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที แรงดันไฟฟ้า 4 V ของ Channel 2 74

4.9 ตารางแสดงจำนวนครั้งของข้อมูลและค่าของแรงดันที่ได้จากการบันทึกข้อมูลแบบต่อเนื่องใน 1 วินาที 77

สารบัญรูป

รูปที่ หน้า

2.1 เครื่องวัดและบันทึกค่าอุณหภูมิและความชื้นสัมพัทธ์ 4
2.2 บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEG 1280/2560 5
2.3 การเชื่อมต่อแบบ SPI 6
2.4 แสดงการเชื่อมต่อแบบ SPI ที่ใช้สายในการรับส่งสัญญาณ 3 หรือ 4 เส้น 8
2.5 การเชื่อมต่อของระบบบัส SPI 8
2.6 รูปแสดงตำแหน่งพินของ SD-Card 10
2.7 แสดงการติดต่อสื่อสารระหว่าง MCU กับ SD Card โดยใช้โหมด SPI 11
2.8 A/D การเปรียบเทียบความละเอียดที่อัตราสุ่มตัวอย่างต่ำ 12
2.9 A/D การเปรียบเทียบความละเอียดที่อัตราสุ่มตัวอย่างสูง 13
2.10 แสดงบล็อกไดอะแกรมของ Successive Approximation ADC 14
2.11 แสดง Flowchart การทำงานของ Successive Approximation ADC 15
2.12 แสดงตัวอย่างการแปลงสัญญาณของ Successive Approximation ADC ขนาด 4 บิต 16
2.13 ET-MINI DS1307 17
2.14 โครงสร้าง ET-MINI DS1307 และตำแหน่งขา Port ใช้งาน 17
2.15 วงจร ET-MINI DS1307 18
2.16 เครื่องโปรแกรม ET-AVRISP mkII 19
3.1 แผนภาพแสดงการเชื่อมต่อวงจรและอุปกรณ์ 20
3.2 แสดง Flowchart การทำงานของโปรแกรมหลัก 21
3.3 ข้อมูลที่บันทึกลง Text File ที่ได้จาก SD-Card 24
3.4 แสดง Flowchart การทำงานของโปรแกรมของ WriteTextFile 24
3.5 แสดง Flowchart การทำงานของโปรแกรมของ ReadTextSendFile 25
3.6 แสดง Flowchart การทำงานของโปรแกรมของ WriteSetting 26
3.7 ตัวอย่างขั้นตอน (ก) การคำนวณหาค่า CRC และ (ข) การตรวจสอบค่า CRC 29
3.8 ตัวอย่างขั้นตอนการตรวจสอบ CRC ของข้อมูลที่มีหลายๆ บล็อก 30
3.9 โครงสร้างของวงจรที่ใช้ในการสร้างข้อมูล CRC แบบ 16 บิต 31
3.10 รูปร่างหน้าต่างหลักของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0 34
3.11 รูปร่างหน้าต่างตั้งค่าการสื่อสารของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0 35

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.12 รูปร่างหน้าต่างตั้งค่าเวลาของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0 35
3.13 รูปร่างหน้าต่างตั้งค่าการทำงานของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0 37
3.14 รูปร่างหน้าต่างแสดงผลข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0 40
3.15 ตัวอย่างหน้าต่างโปรแกรมไฮเปอร์เทอร์มินอล 42
3.16 แสดงการเก็บข้อมูลแรงดันของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรด ... 43
3.17 แสดงการบันทึกข้อมูลของแหล่งจ่าย 45
3.18 แสดงการบันทึกข้อมูลแบบต่อเนื่องโดยจะบันทึกแรงดันไฟฟ้าของแหล่งจ่ายแรงดันไฟฟ้า.. 47
4.1 ผลการทดสอบการส่งข้อมูลของไมโครคอนโทรลเลอร์ 49
4.2 รูปแสดงการเชื่อมต่อของอุปกรณ์บันทึกข้อมูล 50
4.3 หน้าต่างหลักของโปรแกรม 51
4.4 หน้าต่าง Communication 51
4.5 หน้าต่าง Date/Time 51
4.6 หน้าต่าง Channel Setting 52
4.7 หน้าต่างที่ใช้ในการแสดงผล 52
4.8 ตัวอย่าง ไฟล์ Text ที่บันทึกข้อมูล 53
4.9 ตัวอย่างหน้าต่าง Preview การพิมพ์ 53
4.10 ตัวอย่าง ไฟล์ Excel 54
4.11 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด ระยะ 10 เซนติเมตร Channel 1 55
4.12 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 1 57
4.13 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 1 58
4.14 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด ระยะทาง 10 เซนติเมตร Channel 2 58

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.15 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 2 60
4.16 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 2 60
4.17 แสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด ระยะทาง 20 เซนติเมตร Channel 1 61
4.18 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 20 เซนติเมตร Channel 1 63
4.19 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 20 เซนติเมตร Channel 1 64
4.20 แสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด ระยะทาง 20 เซนติเมตร Channel 2 64
4.21 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 20 เซนติเมตร Channel 2 66
4.22 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 20 เซนติเมตร Channel 2 66
4.23 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า 2 V Channel 1 67
4.24 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง บันทึกแรงดัน 2 V Channel 1 69
4.25 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า 2 V Channel 2 69
4.26 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง บันทึกแรงดัน 2 V Channel 2 71
4.27 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า 4 V Channel 1 71
4.28 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง บันทึกแรงดัน 4 V Channel 1 73
4.29 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า 4 V Channel 2 73

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.30 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง บันทึกแรงดัน 4 V Channel 2 75

4.31 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้าที่แสดงข้อมูลการบันทึกแบบต่อเนื่อง 76

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันได้มีการนำอุปกรณ์ที่มีหน้าที่ในการจัดเก็บข้อมูล มาใช้เพื่อให้สะดวกในการจัดเก็บข้อมูล ในลักษณะงานหรือสถานที่ต่างๆ ที่ไม่สามารถมีบุคคลคอยดูแลได้ตลอด 24 ชั่วโมง อุปกรณ์จัดเก็บข้อมูลนั้นสามารถเก็บรวบรวมข้อมูลที่สนใจ ณ เวลาต่างๆ ในช่วงเวลาหนึ่งๆที่ต้องการได้เพื่อนำข้อมูลที่ได้มาใช้ในการวิเคราะห์หรือตรวจสอบการทำงานของระบบ อาทิเช่น เก็บข้อมูลของอุณหภูมิ ภายในห้องเย็นตลอดการส่งสินค้า เพื่อตรวจสอบว่าสินค้าที่ได้นำส่งนั้นอยู่ในช่วงอุณหภูมิที่กำหนดไว้หรือไม่ ซึ่งอุณหภูมิจะมีผลกระทบต่อสินค้าในห้องเย็น เก็บข้อมูลอุตุนิยมวิทยาและอุทกวิทยา คือ ตรวจวัดข้อมูลอุตุนิยมวิทยาและอุทกวิทยาที่สถานีภาคสนาม เช่น การวัดความเข้มแสง อุณหภูมิ ความชื้นสัมพัทธ์ ความกดอากาศ ระดับน้ำและปริมาณน้ำฝน และเก็บข้อมูลที่ตรวจวัดได้ไว้ในอุปกรณ์เก็บข้อมูลและนำข้อมูลที่เก็บเหล่านั้นมาประมวลผลภายหลัง โดยในการบันทึกค่าต่างๆนั้นสามารถกำหนดช่วงเวลาที่จะทำการบันทึกได้ เป็นต้น อุปกรณ์ตัวนี้ช่วยลดเวลาการทำงานที่ต้องให้พนักงานคอยจดบันทึกค่าต่างๆตามช่วงเวลาที่กำหนดไว้ ช่วยลดความผิดพลาดในการจดบันทึก ช่วยจัดเก็บข้อมูลให้เป็นหมวดหมู่ แสดงผลได้อย่างเป็นระเบียบ และสามารถเก็บข้อมูลได้ตลอดเวลา 24 ชั่วโมง แต่ในปัจจุบันที่ไม่ค่อยนิยมใช้อุปกรณ์ชนิดนี้เนื่องมาจากการพกพา การตรวจสอบข้อมูล และมีราคาที่แพง ที่ยากต่อการพกพาและยากต่อการตรวจสอบ เพราะเครื่องมือชนิดนี้ในปัจจุบันนั้นมีขนาดใหญ่ ในการตั้งค่าต้องทำการตั้งค่าที่ตัวเครื่อง กำหนดการทำงานที่ตัวเครื่องทำให้ยุ่งยากต่อการใช้งาน และเมื่อใครมาโดนที่ปุ่มกดบนตัวเครื่องอาจทำให้ค่าที่เราตั้งไว้หายไปได้หรือข้อมูลที่ทำการบันทึกแล้วอาจหายไปได้เมื่อมีการสัมผัสกับปุ่มบนเครื่อง เพราะเมื่อจะนำข้อมูลมาตรวจสอบจะต้องยกเครื่องมาทั้งเครื่องแล้วนำมาต่อกับคอมพิวเตอร์เพื่อทำการตรวจสอบหรืออาจจะทำการตรวจสอบที่หน้าจอของอุปกรณ์เลยก็ได้เช่นกัน อุปกรณ์เก็บข้อมูลในปัจจุบันมีราคาแพงเพราะว่าบางอุปกรณ์ต้องนำเข้าจากต่างประเทศ

คณะผู้จัดทำจึงคิดริเริ่มที่จะทำการพัฒนาอุปกรณ์จัดเก็บข้อมูล เพื่อให้ง่ายต่อการพกพาและการนำเสนอข้อมูลที่ทำการบันทึกเอาไว้โดยคณะผู้จัดทำ จะจัดเก็บข้อมูลไว้ใน SD Card และนำผลที่ได้มาแสดงออกทางหน้าจอคอมพิวเตอร์ส่วนบุคคลในภายหลัง

1.2 วัตถุประสงค์

1.2.1 เพื่อพัฒนาอุปกรณ์ต้นแบบโดยใช้ SD Card เป็นตัวเก็บข้อมูลและแสดงออกทางหน้าจอคอมพิวเตอร์

1.2.2 ช่วยลดการทำงานของพนักงาน

1.2.3 ช่วยลดความผิดพลาดในการจดบันทึก

1.2.4 ช่วยเก็บข้อมูลให้เป็นหมวดหมู่และแสดงผลออกมาอย่างมีระเบียบ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 สามารถนำอุปกรณ์ที่พัฒนาขึ้นมาใช้เก็บข้อมูลตามสถานที่ที่ต้องการได้

1.3.2 มีความรู้และความเกี่ยวกับการพัฒนาอุปกรณ์การเขียนโปรแกรม

1.4 ขอบเขตของโครงงาน

1.4.1 สามารถบันทึกข้อมูลและแปลงอนาล็อกเป็นดิจิตอลได้จำนวน 16 ช่องสัญญาณในวันและเวลาตามที่ต้องการ โดยบันทึกไว้ใน SD Card

1.4.2 สามารถแสดงออกมาเป็นกราฟบนจอคอมพิวเตอร์ส่วนบุคคลเพื่อดูแนวโน้มการเพิ่มและลดของค่าที่สนใจ โดยระดับสัญญาณจะอยู่ที่ 0-5 V

1.5 ขั้นตอนการดำเนินโครงงาน

1.5.1 ศึกษาทฤษฎีที่เกี่ยวข้องกับการจัดเก็บข้อมูลไว้ใน SD Card

1.5.2 ออกแบบและพัฒนาอุปกรณ์การจัดเก็บข้อมูล

1.5.3 ทดสอบและแก้ไขอุปกรณ์ที่พัฒนาขึ้นมา รวมถึงโปรแกรม

1.5.4 ผลการทดลองและวิเคราะห์ผลการทดลอง

1.5.5 สรุปผลการทดลอง

1.5.6 จัดทำรูปเล่มรายงาน

1.6 แผนการดำเนินงาน

กิจกรรม	ปี 2554							ปี 2555				
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. ศึกษาการจัดเก็บข้องมูลไว้ใน SD Card												
2. ศึกษาการใช้บอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR												
3. ศึกษาการใช้โปรแกรม AVR Studio และ โปรแกรม Visual Basic												
4.เขียนโปรแกรมและแก้ไข												
5. ทดสอบอุปกรณ์และปรับปรุงแก้ไข												
6. วิเคราะห์ผลการทดลอง												
7.สรุปผลการทดลอง												
8. จัดทำรูปเล่มรายงาน												

1.7 งบประมาณในการดำเนินโครงงาน

ค่าวัสดุอุปกรณ์	2,000	บาท
ค่าดำเนินงาน	500	บาท
ค่าจัดทำรูปเล่มรายงาน	500	บาท
รวม	3,000	บาท

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

2.1 คุณสมบัติและลักษณะของอุปกรณ์บันทึกข้อมูล

จากการศึกษาเกี่ยวกับอุปกรณ์บันทึกข้อมูลพบว่าระบบบันทึกข้อมูล จะใช้การควบคุมโดยไมโครโปรเซสเซอร์ ซึ่งใช้ระบบบัสสองระบบคือระบบ บัสข้อมูล (Data Bus) ใช้ในการส่งข้อมูล และ บัสของตำแหน่ง (Address Bus) ซอฟแวร์ที่ใช้ในการควบคุมจะถูกเก็บไว้ในหน่วยความจำที่เรียกว่า รอม (ROM: Read Only Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับอ่านเท่านั้นและเก็บในแรม (RAM: Random Access Memory) ชุดคำสั่งโปรแกรมหลักจะถูกเก็บไว้ในหน่วยความจำ รอม ส่วนข้อมูลเข้าตัวแปรในแต่ละช่องสัญญาณ และชุดคำสั่งโปรแกรมอื่นๆ จะถูกเก็บไว้ในหน่วยความจำ แรม ผ่านทางแป้นพิมพ์

โดยพื้นฐานของระบบอุปกรณ์บันทึกข้อมูลจะรับอินพุตเป็นระบบอนาล็อกจากเซ็นเซอร์แล้วทำการเปลี่ยนข้อมูลเป็นระบบดิจิตอล และเก็บข้อมูลไว้ในหน่วยความจำตามช่วงเวลาที่มีการบันทึกโดยอัตโนมัติ ในขณะที่อุปกรณ์เก็บข้อมูลจะรับค่าที่จะบันทึกจากตัวเซนเซอร์ นำมาผ่านตัวแปลงสัญญาณ (Signal Convertor) เพื่อทำการแปลงสัญญาณอนาล็อกที่รับมาเป็นดิจิตอล หลังจากนั้นอุปกรณ์ข้อมูล อาจนำข้อมูลดิจิตอลนั้นมาประมวลผล หรือนำข้อมูลมาเช็คเพื่อทำการส่งสัญญาณไปเตือนผู้ใช้ว่า ข้อมูลมีค่ามากไปหรือน้อยไปแล้วจึงนำข้อมูลที่ได้ไปเก็บบันทึกไว้ในเมมโมรี่ ของตัวเก็บอุปกรณ์ แล้วนำมาแสดงผลบนหน้าจอคอมพิวเตอร์

รูปที่ 2.1 เครื่องวัดและบันทึกค่าอุณหภูมิและความชื้นสัมพัทธ์ [1]

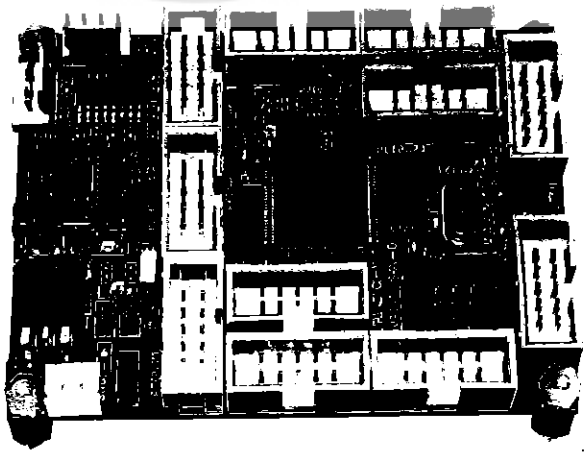
จากรูปที่ 2.1 แสดงตัวอย่างของเครื่องวัดและบันทึกค่าอุณหภูมิและความชื้นชนิดนี้เป็นเครื่องบันทึกข้อมูลที่มีกำหนดช่วงเวลาในการบันทึกได้ตั้งแต่ 1 นาที ถึง 24 ชั่วโมง บันทึกข้อมูล

การวัดด้วยหน่วยความจำสูงถึง 16,000 ข้อมูล แสดงค่าผลการวัดปัจจุบันของอุณหภูมิ หรือความชื้น สัมพัทธ์หรือผลการเตือนที่หน้าจอแสดงผลสามารถป้องกันการสูญหายของข้อมูลหากเกิดแบตเตอรี่หมด

2.2 บอร์ดไมโครคอนโทรลเลอร์

โครงงานนี้ใช้บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA1280/2560 โดยมีไมโครคอนโทรลเลอร์ ATMEGA1280/2560 ซึ่งเป็น MCU ตระกูล AVR MEGA ที่มีระบบ Peripheral I/O ต่างๆ ที่รวบรวมไว้ใน MCU เป็นตัวประมวลผลหลัก ซึ่งในบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA1280/2560 มีคุณสมบัติเด่นๆ คือ

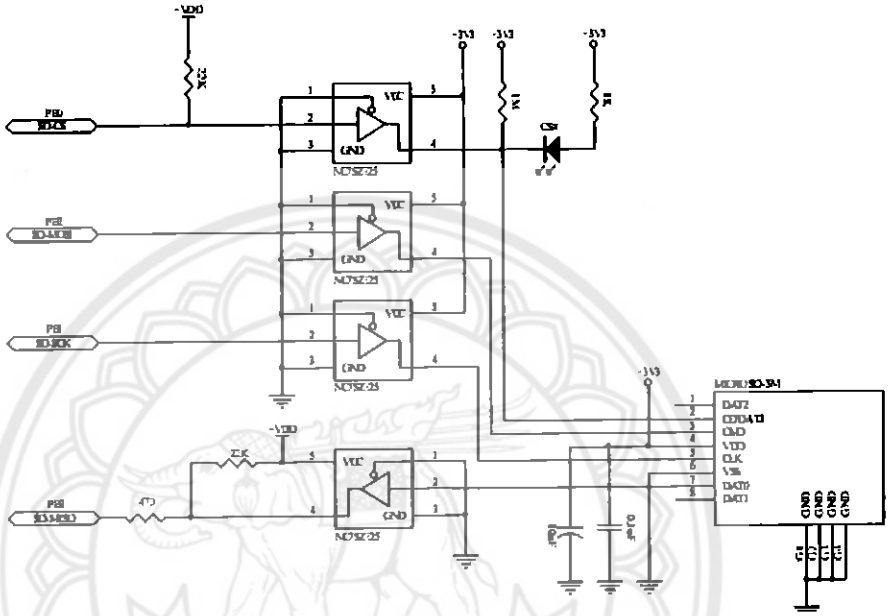
- ใช้ MCU ตระกูล MEGA AVR เบอร์ ATMEGA1280/2560 ของ ATMEL
- มีหน่วยความจำ Flash 128KB (ATMEGA 1280,256K(ATMEGA2560)), 8KB Boot Loader, Static RAM 8KB และ EEPROM 4KB
- ใช้ Crystal 16.00 MHz
- มีวงจร Real Time Counter พร้อม XTAL ค่า 32.768KHz
- รองรับการโปรแกรมแบบ In-System Programming แบบ ISP
- มีวงจรเชื่อมต่อกับ AVR-JTAG ขนาด 10 Pin เพื่อทำการ Debug แบบ Real Time ได้
- Power Supply ใช้แรงดันไฟฟ้า +5VDC พร้อมวงจร Regulate +3V3/3A ภายในบอร์ด พร้อม Jumper เลือกระบบแหล่งจ่ายให้เป็น 3.3V หรือ 5V ได้ตามต้องการ
- มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card (Micro SD) เชื่อมต่อแบบ SPI จำนวน 1 ช่อง
- มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง



รูปที่ 2.2 บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA1280/2560 [2]

2.3 การ์ดหน่วยความจำ SD Card แบบ Micro-SD

บอร์ด ET-BASEMEGA1280/2560 รองรับการเชื่อมต่อกับการ์ดหน่วยความจำ SD Card แบบ Micro-SD โดยการเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB [0..3] ในการเชื่อมต่อกับบอร์ด ซึ่งในการติดต่อสั่งงานการ์ดนั้นสามารถโปรแกรม Pin I/O ของ PB [0..3] ให้ทำงานในโหมด SPI โดยต้องกำหนดหน้าที่ของขาสัญญาณ PB[0..3] ของ MCU



รูปที่ 2.3 การเชื่อมต่อแบบ SPI [2]

จากรูปที่ 2.3 สามารถอธิบายได้ดังนี้

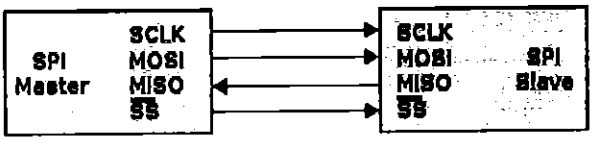
- CD/DAT3 ใช้ PB0 เป็น Output จาก MCU สำหรับเลือกการทำงานของ SD-Card
- CMD ใช้ PB2 โดยขานี้เป็น Output จาก MCU ไปยัง SD-Card
- CLK ใช้ PB1 โดยขานี้เป็น Output จาก MCU ไปยัง SD-Card
- DAT0 ใช้ PB3 โดยขานี้เป็น Input จาก SD-Card เข้ามายัง MCU

2.4 Serial Peripheral Interface (SPI)

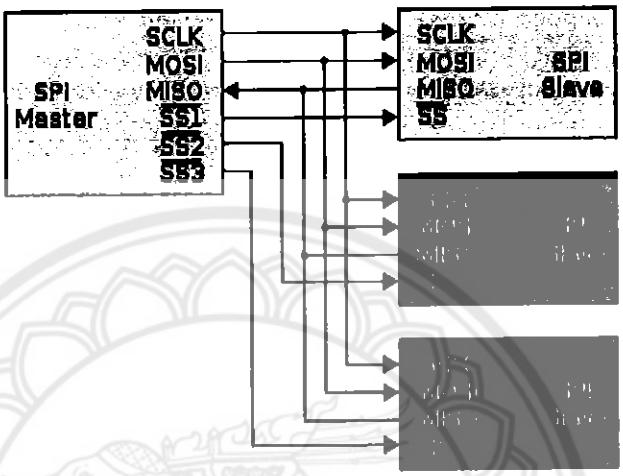
SPI คือ การ Interface อีกแบบหนึ่งที่ย่อมาจาก Serial Peripheral Interface โดย SPI จะใช้สายในการรับส่งสัญญาณ 3 หรือ 4 เส้น ดังรูปที่ 2.4 ในด้านของ UART ทั้งด้านผู้รับและผู้ส่งจะต้องนัดกันก่อนว่าจะใช้ความเร็วในการรับส่งข้อมูล (baud rate) เท่าใด ส่วนใน SPI นั้นจะรับส่งข้อมูลโดยฝั่งหนึ่งจะส่งอัตราความเร็วนาฬิกาพร้อมกับข้อมูลออกไปอีกฝั่งหนึ่ง อัตราความเร็วนาฬิกานี้จะเป็นตัวบ่งบอกความเร็วของผู้รับว่าจะต้องอ่านข้อมูลด้วยความเร็วขนาดไหน อัตราความเร็วนาฬิกาจะถูกส่งมาจากฝั่งที่เป็นตัวบังคับการ (master) ส่วนตัวสนองคำสั่ง (slave) จะไม่มีการส่งอัตราความเร็วนาฬิกาออกไปเพียงแต่เป็นตัวรับค่าจากตัวบังคับการเท่านั้น โดยที่ตัวบังคับการส่งจะส่ง

ความเร็วนาฬิกาออกไปเพียงแต่เป็นตัวรับค่าจากตัวบังคับการเท่านั้น โดยที่ตัวบังคับการส่งจะส่งอัตราความเร็วนาฬิกาออกทางขาต่อรับส่งข้อมูลชื่อ SCK (Serial Clock คือ สัญญาณนาฬิกาที่มาสเตอร์เป็นผู้ส่ง) พร้อมกับส่งข้อมูลออกที่ขาต่อรับส่งข้อมูลชื่อ MOSI (Master Out Slave In คือ เอาท์พุตจากมาสเตอร์ที่ส่งให้สเลฟ) ส่วนตัวสนองคำสั่งจะอ่านอัตราความเร็วนาฬิกาจากขาต่อรับส่งข้อมูล SCK พร้อมกับอ่านข้อมูลที่ขาต่อรับส่งข้อมูล MOSI วิธีดังกล่าวเป็นการส่งข้อมูลทางเดียวอยู่ในขณะที่ข้อมูลถูกส่งออกมาทางเดียวผ่านขาต่อรับส่งข้อมูล MOSI นั้น ในขณะเดียวกันนั้นตัวสนองคำสั่งก็สามารถที่จะส่งข้อมูลกลับคืนได้โดยใช้ขาต่อรับส่งข้อมูลชื่อ MISO (Master Input Slave Output คือ เอาท์พุตจากมาสเตอร์ที่ส่งให้มาสเตอร์) การส่งผ่านข้อมูลกลับไปมานี้เกิดขึ้นในขณะเดียวกัน จะทำการส่งข้อมูลอย่างเดียวหรือรับข้อมูลอย่างเดียวไม่ได้ในระบบ SPI ถ้าเราส่งข้อมูลจำนวนหนึ่งไบต์ออกไป จะได้รับข้อจำนวนหนึ่งไบต์กลับเสมอ เป็นไปได้ที่จะรับส่งข้อมูลขนาดใหญ่กว่าไบต์แต่โดยมาตรฐานเรามักใช้ข้อมูลไบต์เสมอ เป็นไปได้ที่จะรับส่งข้อมูลขนาดใหญ่กว่าไบต์ แต่โดยมาตรฐานเรามักใช้ข้อมูลไบต์เสมอ โดยจะใช้ข้อมูลประเภท 8-บิต (Byte) และ 16-บิต (Short) ในการรับส่ง จากผังทำงานแบบบังคับการ/ตอบสนอง ทำให้เราสามารถเพิ่มต่อตัวตอบสนองเข้ามาในวงจรได้มากกว่าหนึ่งตัวโดยที่ตัวบังคับการจะเลือกสลับข้อมูลไปสู่ตัวตอบสนองตามต้องการ สังเกตว่าเราใช้คำว่าสลับข้อมูล เพราะว่าตามจริงเราไม่ได้ส่งหรือรับข้อมูลทางเดียว แต่เราสลับข้อมูลไปมา(Swap) ตัวบังคับการจะเลือกตัวตอบสนองในทาง ขาต่อรับส่งข้อมูลชื่อ SSEL (Slave Select) ขาต่อรับส่งข้อมูลอันนี้มีชื่ออีกชื่อคือ CS (Chip Select) ตามทฤษฎีตัวบังคับการสามารถติดต่อกับตัวตอบสนองโดยไม่จำกัดจำนวน แต่จะต้องติดต่อส่งข้อมูลไปยังตัวตอบสนองในแต่ละครั้งได้เพียงตัวเดียวเท่านั้น ตัวบังคับการจะใช้สายเชื่อมโยงเพียงสามสาย (SCK, MISO, MOSI) ในการติดต่อกับตัวตอบสนองทั้งหมดที่เชื่อมโยงอยู่ และต้องใช้สายเชื่อมโยง SSEL ต่างหากเพื่อเชื่อมโยงกับตัวตอบสนองแต่ละตัวโดยเฉพาะอุปกรณ์ SPI ที่เป็นตัวตอบสนองบางตัวจะมี ขาต่อรับส่งข้อมูล SSEL หรือ CS มากกว่าหนึ่งขา เป็นต้นว่า โปรเซสเซอร์ VS1053 ที่ใช้แปลข้อมูล MP3 จะมีขาต่อรับส่งสัญญาณสำหรับข้อมูลหนึ่งขา และยังมีขาต่อรับส่งสัญญาณสำหรับออกคำสั่งอีกหนึ่งขาต่างหาก โดยที่ขาทั้งสองจะร่วมใช้ ขาต่อรับส่งข้อมูล 3 ขาคือ SCK, MOSI, MISO [3]

ระบบบัส SPI หรือ การเชื่อมต่ออุปกรณ์อนุกรม (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส ซึ่งมีสัญญาณนาฬิกาเข้ามาเกี่ยวข้อง โดย SPI ถูกพัฒนาขึ้นมาด้วยเหตุผลเดียวกับ I2C (ใช้เพื่อสื่อสารกันระหว่าง IC) เป็นโปรโตคอลที่ต่อเชื่อมได้มากกว่า 2 โนด แต่เปลี่ยนจากการใช้ แอดเดรส (Address) ใส่ไปกับแพ็กเกจข้อมูล (Data Packet) มาเป็นการใช้สายชิพซีเล็ก (Chip Select) แทน เมื่อชิพซีเล็กของสเลฟ (Slave) ตัวไหนเป็นโลว์



รูปที่ 2.4 แสดงการเชื่อมต่อแบบ SPI ที่ใช้สายในการรับส่งสัญญาณ 3 หรือ 4 เส้น [3]



รูปที่ 2.5 การเชื่อมต่อของระบบบัส SPI [3]

จากรูปที่ 2.5 สามารถอธิบายได้ว่า

- MOSI (Master Out–Slave In) คือ ขาที่ใช้ในการส่งข้อมูลออกจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ
- MISO (Master In–Slave Out) คือ ขาที่ใช้ในการรับข้อมูลจากอุปกรณ์สเลฟหรือมาสเตอร์อ่านข้อมูล
- SCK (Serial Clock) คือ สายสัญญาณนาฬิกาที่ใช้งานร่วมกันระหว่างมาสเตอร์กับสเลฟ
- SS (Slave Select) คือ สายสัญญาณสำหรับเลือกอุปกรณ์สเลฟ

2.5 Flash Memory

Flash Memory คือ หน่วยความจำขนาดเล็กประเภทนอนโวลาไทล์ (Nonvolatile คือ หน่วยความจำที่ข้อมูลไม่ถูกลบหรือหายไปเมื่อปิดเครื่องหรือไม่มีไฟเลี้ยง เป็นหน่วยความจำถาวร ซึ่งข้อมูล หรือโปรแกรม ที่เก็บ โปรแกรมพื้นฐานที่สำคัญที่จำเป็นสำหรับการเริ่มทำงานของระบบ เช่น ไบออส ไว้ที่ หน่วยความจำแบบนี้ ได้แก่ หน่วยความจำประเภท รอม (ROM)) ที่สามารถบันทึกข้อมูลลงไปได้โดยที่ไม่ต้องใช้แบตเตอรี่ ข้อมูลไม่มีการสูญหายเมื่อเปิดสวิตช์ ซึ่งมีส่วนที่ใช้บันทึกข้อมูล เรียกว่า Solid State Chips ซึ่งใช้กระบวนการทางไฟฟ้าในการบันทึกข้อมูล มีตัวควบคุมการอ่านและเขียนในตัวเอง หน่วยความจำสำรองแบบ Solid State เป็นหน่วยความจำ ที่เป็น

บันทึกข้อมูลลงไปได้โดยที่ไม่ต้องใช้แบตเตอรี่ ข้อมูลไม่มีการสูญหายเมื่อเปิดสวิตช์ ซึ่งมีส่วนที่ใช้บันทึกข้อมูล เรียกว่า Solid State Chips ซึ่งใช้กระบวนการทางไฟฟ้าในการบันทึกข้อมูล มีตัวควบคุมการอ่านและเขียนในตัวเอง หน่วยความจำสำรองแบบ Solid State เป็นหน่วยความจำ ที่เป็นที่นิยมใช้กันอย่างแพร่หลายในอุปกรณ์ดิจิตอล เช่น กล้องถ่ายภาพดิจิตอล ที่สามารถบันทึกข้อมูลได้มากและรวดเร็ว หรือในอุปกรณ์ที่สามารถบันทึกและเคลื่อนย้ายข้อมูลได้สะดวกรวดเร็ว

แฟลชแบบ NOR แม้จะเขียนและลบข้อมูลได้ช้า แต่ก็เหมาะที่จะนำมาใช้แทน ROM ในงานที่ไม่ต้องเขียนหรือลบข้อมูลบ่อยเช่น BIOS (Basic Input/Output System:คือโปรแกรมที่ปกติจะเก็บเอาไว้ในรอมที่เป็นความจำถาวร หรือกึ่งถาวร (EPROM Erasable Programmable Read Only Memory) และเป็นโปรแกรมที่ไมโครโพรเซสเซอร์ของเครื่องคอมพิวเตอร์ส่วนบุคคลเรียกใช้เป็นโปรแกรมแรกตั้งแต่เปิดเครื่อง โดยไบออสจะทำหน้าที่ในการตรวจอุปกรณ์ที่ต่อไว้ตามตำแหน่งที่ระบุ และทำการโหลดระบบปฏิบัติการจากฮาร์ดดิสก์หรือดิสก์เก็ต ไปที่แรมซึ่งเป็นหน่วยความจำชั่วคราว หลังจากนั้นจะทำหน้าที่บริหารการไหลของข้อมูล ระหว่างระบบปฏิบัติการของคอมพิวเตอร์กับอุปกรณ์ที่เชื่อมต่อหน่วยรับข้อมูลและหน่วยแสดงผล) ของเครื่องคอมพิวเตอร์ และ Firmware ของกล่องเคเบิ้ลทีวี แฟลชแบบ NOR ทนการลบและเขียนข้อมูลได้ 10,000 ครั้ง อุปกรณ์ที่มีคอมพิวเตอร์ฝังในตัวรุ่นแรกๆ จะใช้แฟลชแบบ NOR ทั้งนั้น โตชิบาประกาศตัวแฟลชแบบ NAND ในปี ค.ศ. 1989 มันสามารถลบและเขียนข้อมูลได้รวดเร็วกว่า NOR ชิพมีขนาดเล็กกว่า ความจุสูงกว่าและมีราคาต่ำกว่า NOR แต่มีความทนทานกว่าแฟลชแบบ NOR สิบเท่า ข้อเสียของแฟลชแบบ NAND คือเราไม่สามารถเข้าถึงแต่ละเซลได้โดยตรง ต้องอ่านข้อมูลเป็นบล็อก โดยแต่ละบล็อกปรกติจะมีขนาดหลายร้อยถึงหลายพันบิต จึงไม่เหมาะที่จะใช้แทนที่ ROM เพราะไมโครโปรเซสเซอร์จำเป็นต้องอ่านคำสั่งทีละไบต์

2.6 SD-Card

คำว่า SD-Card หรือ SD นั้นย่อมาจาก Secure Digital เป็นสื่อเก็บข้อมูลที่ได้รับความนิยมมีรูปแบบการ์ดที่เป็นหน่วยความจำแบบ Non-Volatile SD Card ซึ่งนิยมนำมาใช้ร่วมกับอุปกรณ์พกพาขนาดเล็กหลายประเภท ไม่ว่าจะเป็น Digital Cameras, PDAs, Mobile Phones, GPS Receivers, and Video Game Consoles

SD Card ในตอนแรกมีความจุตั้งแต่ 4 MB – 4 GB แต่มีการพัฒนาอย่างต่อเนื่องทำให้ความจุของ SD Card มากขึ้นเรียกว่า SDHC (Secure Digital High Capacity) ซึ่งมีความจุตั้งแต่ 4 GB - 32GB และ SD Card จะพัฒนาไปเป็น SDXC ซึ่งจะมีความจุตั้งแต่ 32 GB – 2 TB ใน SD-Card จะมี 9 พิน ซึ่งในแต่ละพินจะมีการทำงานที่แตกต่างกันไปดังนี้ [5]

พิน 1 ในโหมดของ SPI ทำหน้าที่เป็นขา CS หรือ Chip Select

พิน 2 ในโหมดของ SPI ทำหน้าที่เป็นขา DI หรือ Data Input

พิน 3 ในโหมดของ SPI ทำหน้าที่เป็นขา GND หรือ Ground

พิน 4 ในโหมดของ SPI ทำหน้าที่เป็นขา POWER หรือ +5V

พิน 5 ในโหมดของ SPI ทำหน้าที่เป็นขา CLK หรือ Clock

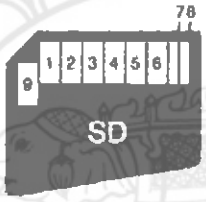
พิน 6 ในโหมดของ SPI ทำหน้าที่เป็นขา GND หรือ Ground

พิน 7 ในโหมดของ SPI ทำหน้าที่เป็นขา DATA0 หรือ Data Out

พิน 8 ในโหมดของ SPI ไม่ใช้

พิน 9 ในโหมดของ SPI ไม่ใช้

ตำแหน่งของ SD-Card จะแสดงในรูปที่ 2.6



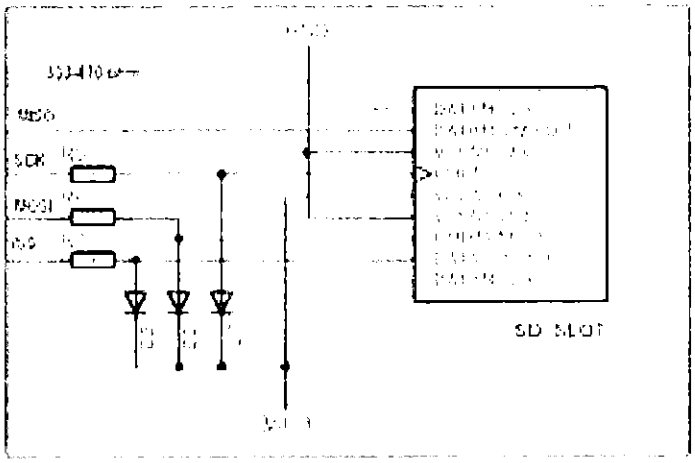
Pin	SD	SPI
1	CD/DAT3	CS
2	CMD	DI
3	VSS1	VSS1
4	VDD	VDD
5	CLK	SCLK
6	VSS2	VSS2
7	DAT0	DO
8	DAT1	X
9	DAT2	X

รูปที่ 2.6 รูปแสดงตำแหน่งพินของ SD-Card [5]

2.7 การทำงานของ SD-Card

ภายใน SD-card จะมี Microcontroller ขนาดเล็กคล้ายกับการทำงานของ CPU ทำหน้าที่คอยดูแลและติดต่อกับอุปกรณ์ต่างๆที่ใช้ SD-Card ในขณะนั้น เช่น PDA Phone หรือโทรศัพท์มือถือ โดย Microcontroller จะทำหน้าที่จัดการเรื่องข้อมูลต่างๆที่ถูกอ่านและถูกเขียนของใน Flash โดย CPU หรือ Microcontroller จะทำงานเฉพาะในช่วงที่มีการใช้งานข้อมูลใน Card เท่านั้น การทำงานของ SD Card จะเร็วหรือช้านั้นก็ขึ้นอยู่กับความเร็วของ CPU ในเครื่อง PDA Phone หรือ อุปกรณ์ต่างๆด้วย เช่น เมื่อมีการใช้งาน SD-Card บน โทรศัพท์ ตัวโทรศัพท์จะส่งข้อมูลแจ้งไปยัง SD-card ว่า Microcontroller นั้นสามารถรองรับความเร็วสูงสุดที่เท่าไรก็ให้ใช้ความเร็วนั้น เพราะฉะนั้นหาก Clock ของโทรศัพท์ยิ่งเร็ว การทำงานของ Card ก็จะเร็วตามไปด้วย โดยข้อมูลจะถูกส่งในแต่ละวินาทีได้รวดเร็วขึ้น

การติดต่อสื่อสารระหว่าง MCU กับ SD-Card โดยใช้โหมด SPI ในการทำงานในโหมดนี้ใช้สายสัญญาณเพียง 3 เส้นดังแสดงในรูปที่ 2.7 แต่ความเร็วในการรับส่งข้อมูลก็จะช้าลงไปด้วย แต่เมื่อเทียบกับการใช้งาน ส่วนใหญ่แล้วไม่ต้องการความเร็วมากนัก ความเร็วในการรับส่งข้อมูลจึงไม่เป็นปัญหาอะไร [5]



รูปที่ 2.7 แสดงการติดต่อสื่อสารระหว่าง MCU กับ SD Card โดยใช้โหมด SPI [5]

2.8 Analog to Digital conversion (ADC)

Analog to Digital Conversion (ADC) คือระบบการแปลงสัญญาณ Analog ให้เป็นสัญญาณ Digital ซึ่งจะค่าเท่ากับสัญญาณ Analog นั้น

Resolution คือการเปลี่ยนแปลงค่าที่น้อยที่สุดของแรงดันที่สามารถ ตรวจจับได้ โดยระบบ และจะถูกแสดงเป็น Digital Code ซึ่ง Resolution คือจำนวนของ Digital Code ทั้งหมด

$$\text{Resolution} = n \text{ bit}$$

Quantization Level คือค่าระดับทั้งหมดของจำนวนค่า Analog

$$\text{Quantization Level} = 2n$$

Quantization Step คือจำนวนขั้นระหว่าง Quantization Level ทั้งหมด

$$\text{Quantization step} = 2n - 1$$

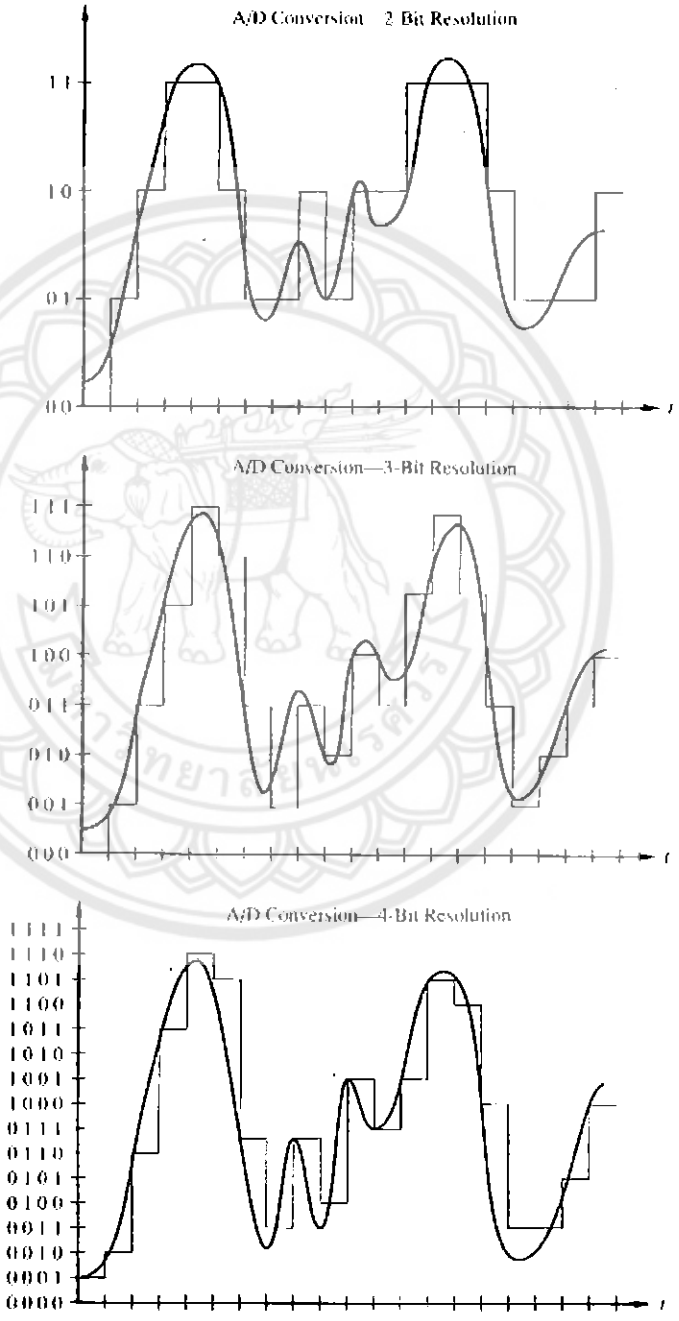
Voltage Resolution คือค่าแรงดันในแต่ละช่วงของ Step

Sampling Rate เป็นปริมาณที่มักจะถูกใช้กับ Analog to Digital Conversion ซึ่ง Sampling rate เป็นจำนวนของเวลาต่อวินาทีที่สัญญาณ Analog ถูกเปลี่ยนเป็น Digital Code สำหรับคุณสมบัติของ Analog to Digital Conversion ค่าที่น้อยที่สุดของ Sampling rate จะมีค่าเป็นสองเท่าของความถี่ Analog ที่สูงที่สุด ซึ่ง Sampling rate ที่น้อยที่สุดนี้ มักจะเรียกเป็น Nyquist Sampling Rate ตามรูปที่ 2 แสดงผลของ Analog to Digital Conversion ที่เป็นสองเท่าของ Sampling rate ในรูปที่ 1

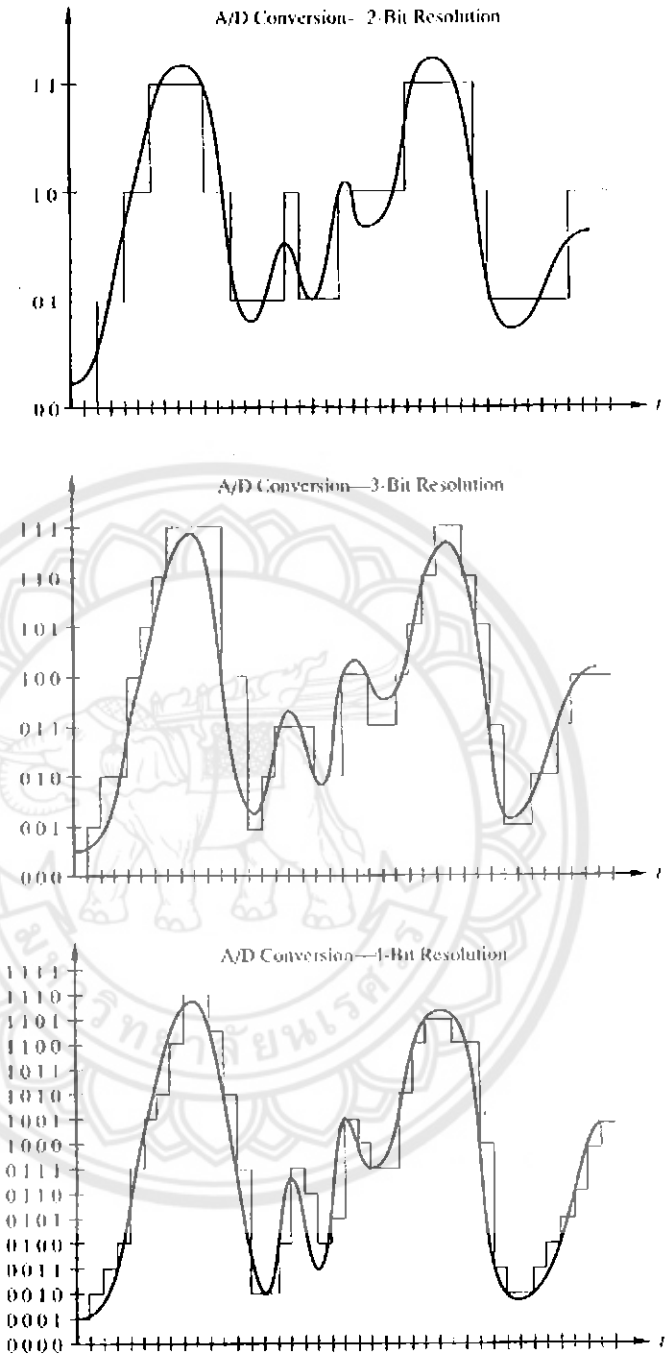
ถ้า Sampling Frequency น้อยกว่าสองเท่าของความถี่ Analog ที่สูงที่สุด จะเกิด Aliasing error ขึ้น Aliasing error จะทำให้เกิดความผิดเพี้ยนของความถี่สูงของสัญญาณ เนื่องจากตัว Spectral ของสัญญาณที่ถูก Sampling ซึ่ง Aliasing Error เป็นปรากฏการณ์ ที่มีสาเหตุมาจาก ส่วนประกอบของความถี่ที่เกิดขึ้นจากสัญญาณ Sampling จะถูกสอดแทรกจากความถี่ ของสัญญาณที่จะถูก

Sampling เราสามารถที่จะหลีกเลี่ยง Aliasing Error ได้โดยการใช้ความถี่ Sampling ที่น้อยกว่า 2 เท่าของ Analog Frequency ที่สูงที่สุด

พื้นฐานของ Analog to Digital Converter ตามรูปที่ 2.8 ซึ่งรับค่า Analog input voltage และทำการแปลงให้อยู่ในรูปของ Digital Code ขนาด 4 bit



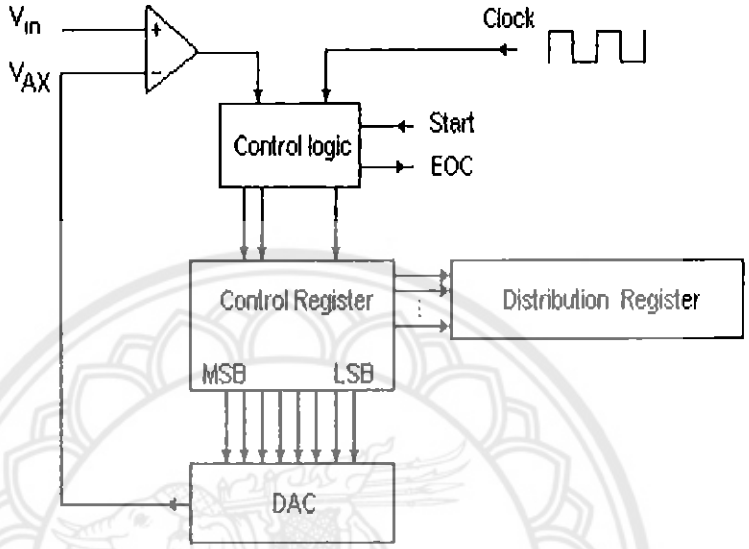
รูปที่ 2.8 A/D การเปรียบเทียบความละเอียดที่อัตราสุ่มตัวอย่างต่ำ [6]



รูปที่ 2.9 A/D การเปรียบเทียบความละเอียดที่อัตราสุ่มตัวอย่างสูง [6]

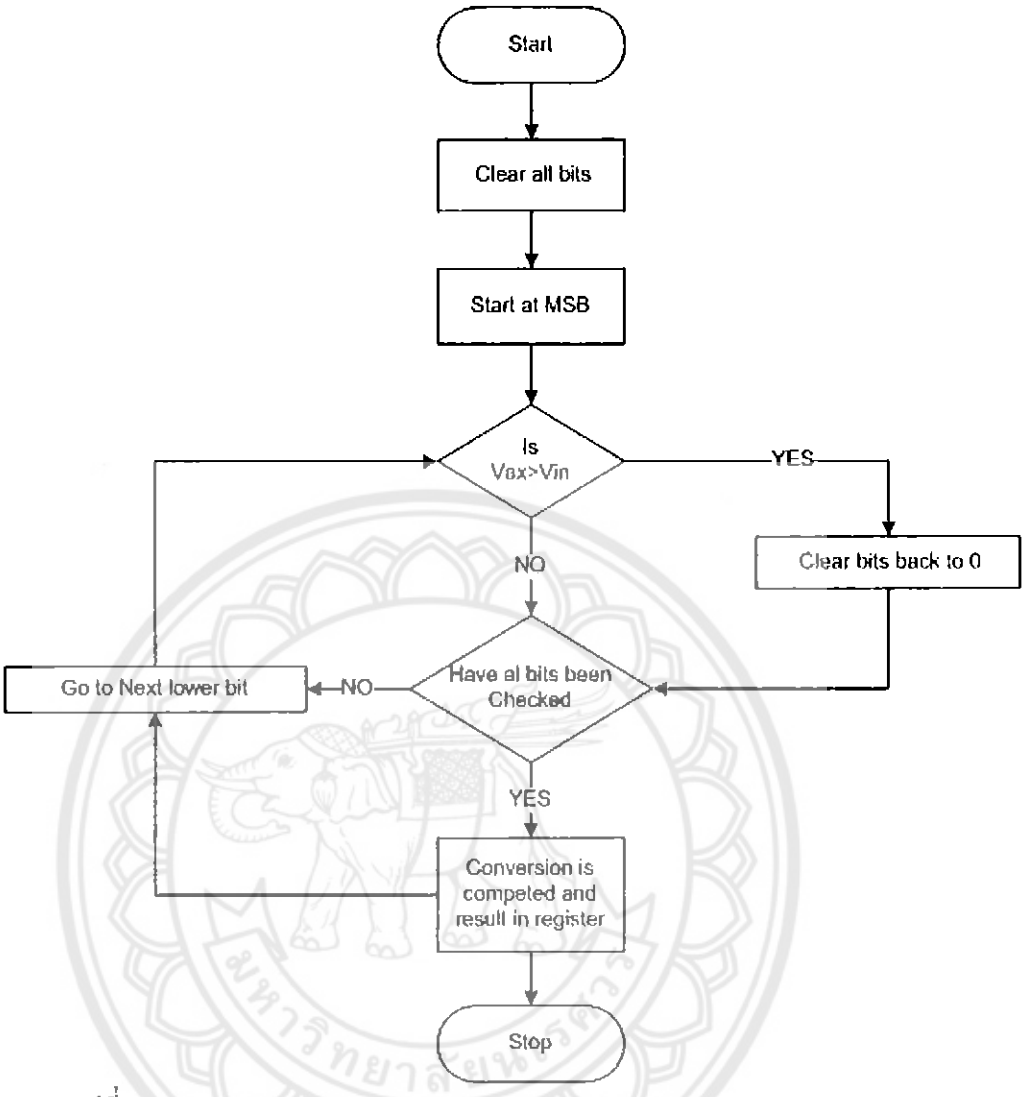
Successive Approximation ADC (SAC) วงจรแปลงสัญญาณชนิดประมาณค่า เป็นวงจรที่อาศัยการทดสอบผลแต่ละบิตของค่า ข้อมูลดิจิตอล การทำงานอาศัยการควบคุมลอจิก "1" หรือ "0" จาก Control Register ส่งไปยังวงจรแปลงสัญญาณ DAC แปลงให้เป็นสัญญาณ Analog Voltage แล้วทำการเปรียบเทียบกับ Input Voltage ที่ป้อนเข้ามาเมื่อผลลัพธ์ ได้เป็น "1" แสดงว่าสัญญาณเข้าสูงกว่าสัญญาณเปรียบเทียบ กำหนดให้บิต (Bit) ทดสอบเป็น "1" แล้วจึงตรวจสอบบิต ต่อไป และมี

Distribution Register ทำหน้าที่จดจำสภาวะการตรวจแต่ละบิต โดยเรียงจากบิต ที่มีน้ำหนักมากสุด (MSB) ไปยังบิต ที่มีน้ำหนักต่ำสุด (LSB) ตามลำดับ ซึ่งสามารถแสดงได้ดังรูปที่ 2.10 แสดงบล็อกไดอะแกรมของ Successive Approximation ADC



รูปที่ 2.10 แสดงบล็อกไดอะแกรมของ Successive Approximation ADC [6]

จากรูปบล็อกไดอะแกรมสามารถอธิบายการทำงานเป็น Flowchart เพื่อทำให้ สามารถเข้าใจการทำงานของ Successive Approximation ADC ได้ง่ายขึ้น



รูปที่ 2.11 แสดง Flowchart การทำงานของ Successive Approximation ADC [6]

จากรูปที่ 2.11 แสดง Flowchart การทำงานของ Successive Approximation ADC Flowchart สามารถอธิบายการทำงานได้ดังนี้

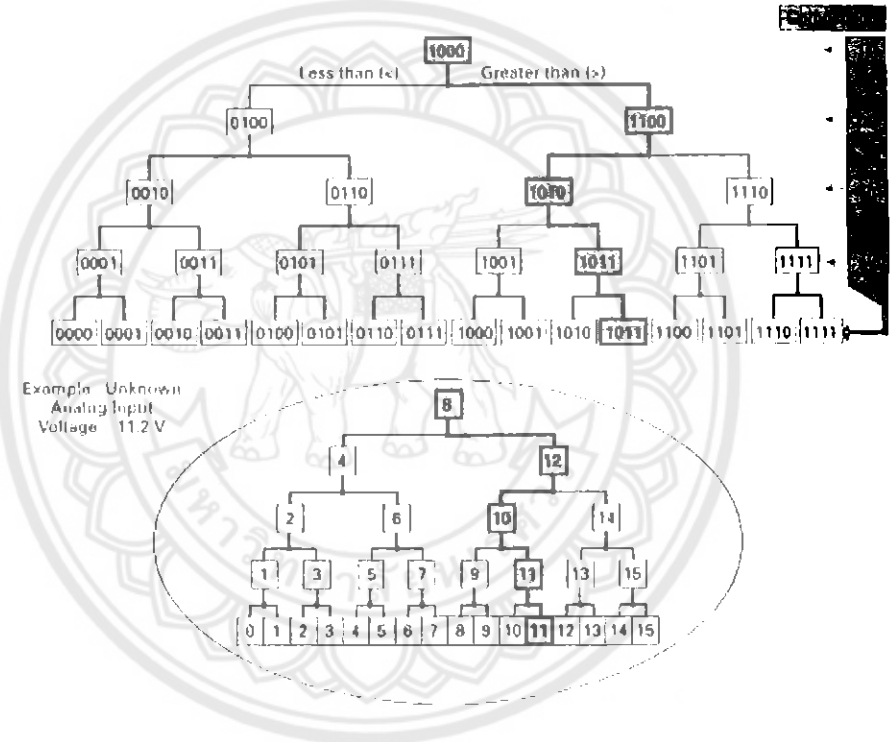
1. เมื่อมีสัญญาณ Start เข้ามาที่ Control Logic จะทำให้ Successive Approximation ADC เริ่มทำงาน
2. Control Logic จะ Clear ทุกบิต ใน Register ให้เป็น "0"
3. เริ่มการทำงานที่บิตที่มีนัยสำคัญมากที่สุดก่อน (MSB)
4. ทำการ Set ให้บิต MSB เป็น "1"
5. แล้วนำมาเช็คเงื่อนไขการทำงาน ซึ่งใน Circuit ก็คือ Op-Amp มีหน้าที่ในการ เปรียบเทียบค่าแรงดัน

   - ถ้า Vax > Vin จริง จะ Set Bit เดิม ให้เป็น "0" แล้วเช็คบิตถัดไป และ Set Bit นั้น ให้เป็น "1" จากนั้นเข้ามาเช็คบิตเงื่อนไขอีก

- ถ้า Vax > Vin ไม่จริง จะยังคงบิตนั้น ให้เป็น "1" อยู่ แล้วเช็คบิตถัดไป และ Set Bit นั้น ให้เป็น "1" จากนั้นเข้ามาเช็คเงื่อนไขอีก

6. ทำตามข้อ 5. ไปเรื่อยๆ จนกว่า Distribution Register จะเช็คว่า ทำงานจนถึงบิต ที่มีนัยสำคัญน้อยที่สุด (LSB) แล้ว แสดงว่าการ Convert เสร็จสิ้น ซึ่งผลลัพธ์จะถูกเก็บไว้ใน Control Register
7. เมื่อ Convert เสร็จแล้ว จะมีการส่งสัญญาณ End of Convert (EOC) เพื่อบอกให้ทราบว่าขณะนี้ได้ทำการแปลงสัญญาณ Analog ให้เป็น Digital เสร็จเรียบร้อยแล้ว

ตัวอย่างการแปลงสัญญาณของ Successive Approximation ADC สามารถแสดงได้ดังนี้

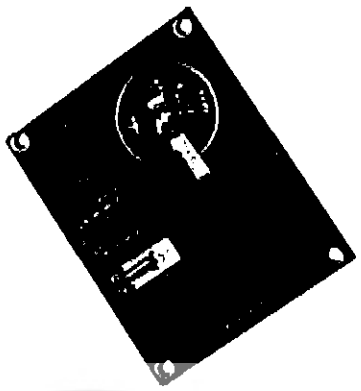


รูปที่ 2.12 แสดงตัวอย่างการแปลงสัญญาณของ Successive Approximation ADC ขนาด 4บิต [6]

2.9 ET-MINI DS1307

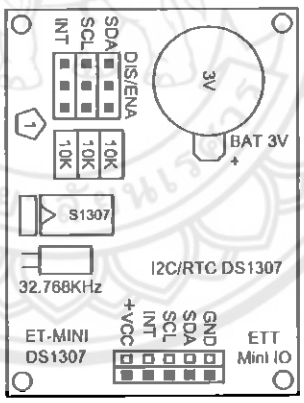
Module ET-MINI DS1307 เป็นชุด I^2C ประเภท Real Time Clock (RTC) ซึ่งทำหน้าที่เกี่ยวกับฐานเวลาในลักษณะของนาฬิกาเวลาที่แสดงออกมาในรูปแบบ ชั่วโมง นาที วินาที และปฏิทินในรูปแบบ วัน เดือนและ ปี ในโมดูลนี้จะมีขั้วต่อ SDA และ SCL สำหรับต่อเข้ากับขา SDA และ SCL ของ MCU ตามลำดับ ส่วนขั้วต่อ INT จะต่อเข้ากับ MCU ก็ต่อเมื่อต้องการใช้งาน Interrupt โดยให้ต่อเข้ากับขาของ MCU ที่ผู้ใช้ได้กำหนดให้มีการรับสัญญาณ INT จากภายนอกไว้ ส่วนไฟเลี้ยงโมดูลจะอยู่ที่ VDC 3V-5V เบอร์นี้จะมี Control Byte อยู่ที่ "1101000x" นอกจากนี้ก็จะ

มีในส่วนของที่สำหรับใส่แบตเตอรี่เพื่อใช้ในการ Back Up ฐานเวลา ให้นาฬิกายังคงเดินได้อย่างถูกต้อง เมื่อไม่มีการจ่ายไฟเลี้ยงให้กับ Module

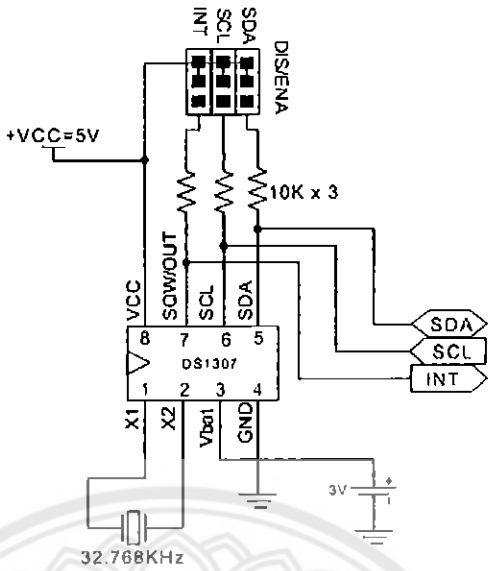


รูปที่ 2.13 ET-MINI DS1307 [7]

ก่อนการใช้งานโมดูลนี้จะต้องทำการ Set Jumper SDA, SCL และ INT มาทางด้าน ENA ดังรูปที่ 2.4 เพื่อเป็นการต่อ R Pull Up ให้กับทั้ง 3 ขาของ I^2C ถ้ามีการต่อ R Pull Up ใน Line SDA และ SCL จากภายนอกหรือจาก Module อื่นไว้แล้วให้ Set Jumper ทั้ง 3 มาทางด้าน DIS เพื่อไม่ต้องต่อ R Pull Up เข้าไปอีก



รูปที่ 2.14 โครงสร้าง ET-MINI DS1307 และตำแหน่งขา Port ใช้งาน [8]

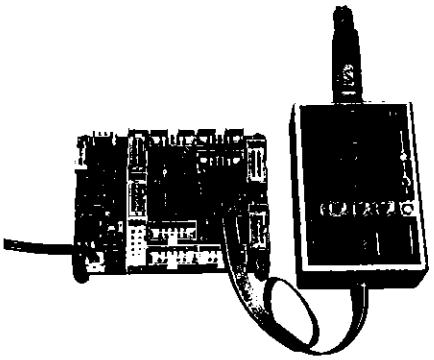


รูปที่ 2.15 วงจร ET-MINI DS1307 [8]

2.10 เครื่องโปรแกรม ET-AVRISP mkII

ET-AVRISP mkII เป็นเครื่องโปรแกรมไมโครคอนโทรลเลอร์ตระกูล AVR โดยใช้ Protocol เดียวกันกับ AVRISP mkII ของ ATMEL โดยเชื่อมต่อกับคอมพิวเตอร์ PC ผ่านทางพอร์ต USB โดยคุณสมบัติของเครื่องโปรแกรม ET-AVRISP mkII จะมีคุณสมบัติการทำงานใกล้เคียงกันกับ AVRISP mkII ของ ATMEL โดยสามารถใช้โปรแกรม MCU ตระกูล AVR ได้ทั้งแบบที่เป็น ISP และ PDI และ TPI โดยใช้งานร่วมกับชุดโปรแกรม AVRStudio ของ ATMEL

- ใช้งานร่วมกับโปรแกรม AVRStudio
- รองรับการโปรแกรม MCU ตระกูล AVR แบบ In-Circuit Program ทั้งแบบ ISP, PDI และ TPI
- สามารถโปรแกรมได้ทั้งหน่วยความจำ Flash และ EEPROM ในตัว MCU
- สามารถโปรแกรม Fuse Bit และ Lock Bit ของ MCU ได้
- สามารถใช้ได้กับ Target Board ที่ใช้แหล่งจ่ายตั้งแต่ 1.8V-5.5V
- สามารถปรับความเร็วของ ISP ในการโปรแกรมได้ตั้งแต่ 50Hz-8MHz
- เชื่อมต่อกับคอมพิวเตอร์ PC ทางพอร์ต USB 2.0 แบบ Full Speed (12Mbps)
- ใช้แหล่งจ่ายไฟจากพอร์ต USB



รูปที่ 2.16 เครื่องโปรแกรม ET-AVRISP mkII [9]

2.11 โปรแกรม Visual Basic 6.0

Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event-Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น [10]

2.12 โปรแกรม AVR Studio4

AVR Studio เป็นซอฟต์แวร์สำหรับสร้างโปรแกรมควบคุม โดยภายใน AVR Studio ได้บรรจุเครื่องช่วยอำนวยความสะดวกต่อการพัฒนาโปรแกรมภาษาซีและแอสแซมบลี ทั้งยังสามารถเชื่อมโยงการทำงานในส่วนต่างๆ กับคอมไพเลอร์ที่กำหนดได้ด้วย โดย ซีคอมไพเลอร์ที่นำมาใช้ร่วมกันคือ WinAVR ทั้งยังสามารถติดต่อกับซอฟต์แวร์สำหรับโปแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ AVR ดังนั้นซอฟต์แวร์ AVR Studio จึงเหมาะอย่างยิ่งในการนำมาใช้พัฒนาโปรแกรมสำหรับไมโครคอนโทรลเลอร์ AVR โดย AVR Studio [11]

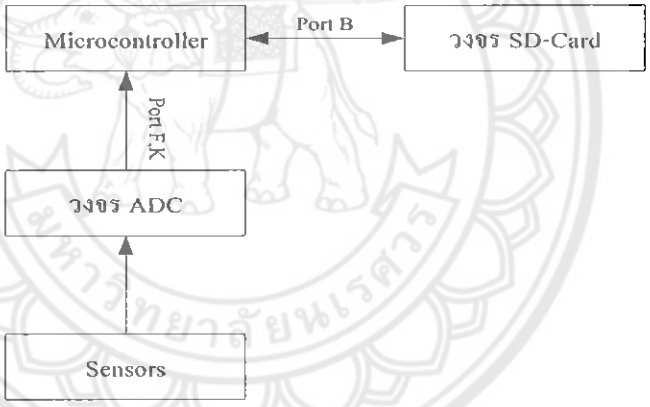
2.13 โปรแกรม WinAVR

WinAVR เป็นซอฟต์แวร์ ซีคอมไพเลอร์หรือตัวแปลโปรแกรมภาษาซี สำหรับไมโครคอนโทรลเลอร์ AVR โดย WinAVR เป็นซอฟต์แวร์แบบโอเพ่นซอร์ส (open source) พัฒนาต่อจาก GNU GCC คอมไพเลอร์ เมื่อทำการติดตั้ง WinAVR แล้วจะสามารถเชื่อมโยงการทำงานเข้ากับ AVR Studio ได้ ดังนั้นจึงสามารถทำการเขียนโปรแกรมภาษาซีบน AVR Studio แล้วทำการคอมไพล์โปรแกรมด้วย WinAVR ได้อย่างต่อเนื่องโดยผลลัพธ์ของการคอมไพล์จะได้ไฟล์เป็นนามสกุล .hex อันเป็นไฟล์รหัสภาษาเครื่องหรือที่เรียกว่า “แมชชีนโค้ด” โดยเป็นไฟล์ผลลัพธ์ที่ได้จากการพัฒนาสามารถนำดาวน์โหลดลงสู่ไมโครคอนโทรลเลอร์ต่อไปได้ทันที [11]

บทที่ 3
วิธีการดำเนินโครงงาน

การออกแบบและพัฒนาอุปกรณ์บันทึกข้อมูลหลังจากการศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้อง ผู้ดำเนินโครงงานได้ออกแบบและพัฒนาอุปกรณ์ต้นแบบสำหรับการบันทึกข้อมูลโดยบันทึกข้อมูลเป็นแรงดันไฟฟ้าในส่วนของซอฟต์แวร์

ในการออกแบบและพัฒนาซอฟแวร์ของอุปกรณ์บันทึกข้อมูล สามารถเขียนแผนภาพแผนภาพแสดงการเชื่อมต่อวงจรและอุปกรณ์ ได้ดังรูป 3.1 โดยเริ่มจากการออกแบบและพัฒนาโปรแกรมภาษาซีในการรับข้อมูลสัญญาณอนาล็อกที่รับจากเซ็นเซอร์ต่อมาเขียนโปรแกรมภาษาซีสนับสนุนการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลและบันทึกลง SD-Card จากนั้นเขียนโปรแกรมภาษาซีส่งข้อมูลเหล่านี้ไปยังหน่วยความจำของคอมพิวเตอร์ส่วนบุคคลเพื่อแสดงผลบนจอมอนิเตอร์ด้วยโปรแกรม Visual Basic 6.0



รูปที่ 3.1 แผนภาพแสดงการเชื่อมต่อวงจรและอุปกรณ์

Microcontroller จะทำหน้าทีรับสัญญาณดิจิตอลจากวงจร ADC และทำการแปลงสัญญาณดิจิตอลให้เป็นข้อมูลแรงดันไฟฟ้าแล้วส่งไปบันทึกลงใน SD-Card

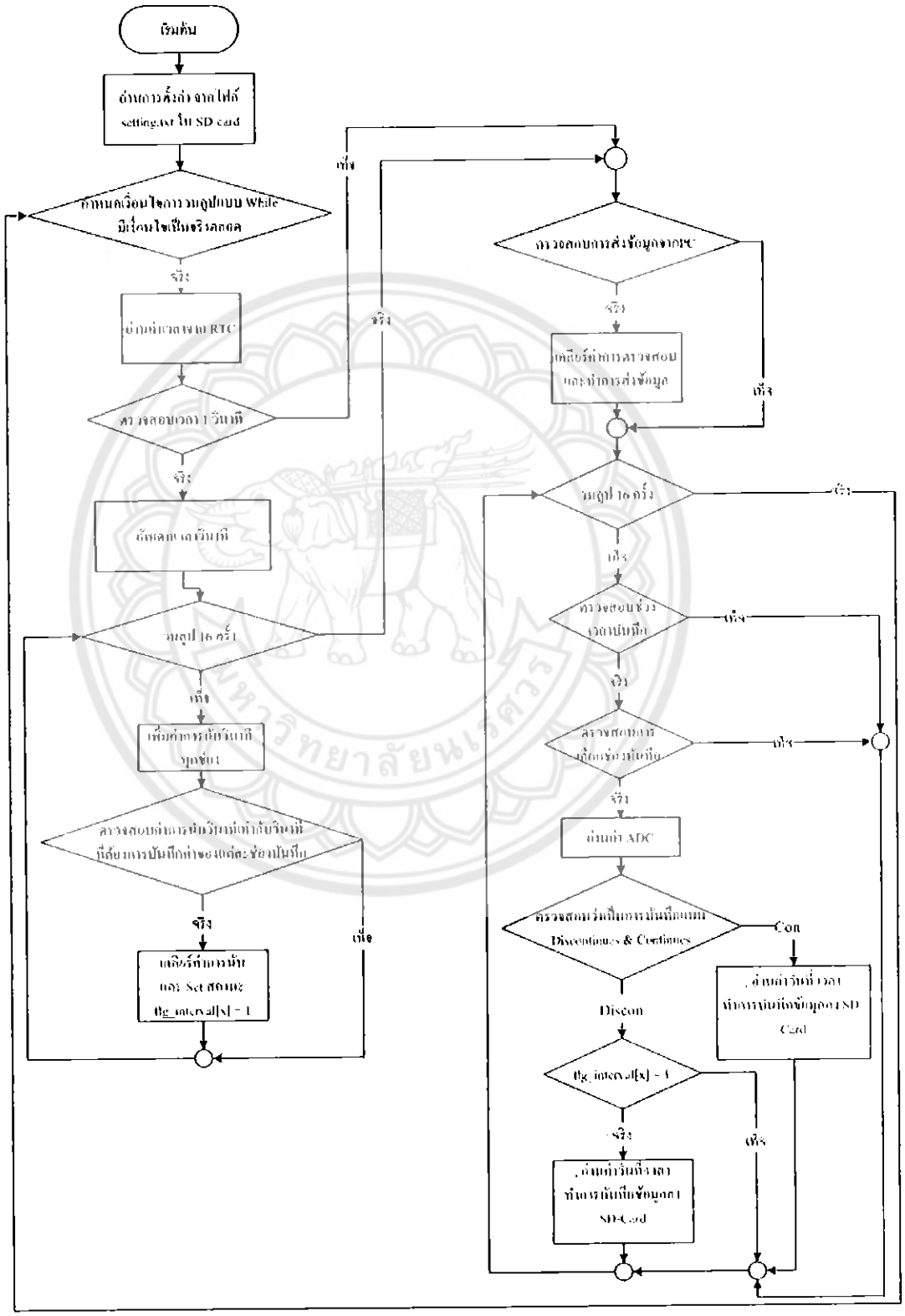
SD-Card จะมีหน้าจัดเก็บข้อมูลการบันทึกแรงดันไฟฟ้าและข้อมูลการตั้งค่าในการบันทึกข้อมูลจะเชื่อมต่อด้วย port B

วงจร ADC มีหน้าที่แปลงสัญญาณอนาล็อกที่ได้จาก Sensors ให้เป็นสัญญาณดิจิตอลแล้วส่งเข้า Microcontroller จะเชื่อมต่อด้วย port F และ K

Sensors เป็นวงจรที่ทำหน้าที่ตรวจจับการเปลี่ยนแปลงของสิ่งแวดล้อม เช่น แสง เสียง ที่อยู่รอบตัวมาเป็นสัญญาณทางไฟฟ้าแล้วส่งไปยังส่วนควบคุมของ Microcontroller

3.1 การพัฒนาโปรแกรมในส่วนของไมโครคอนโทรลเลอร์

3.1.1 การทำงานในส่วนโปรแกรมหลัก



รูปที่ 3.2 แสดง Flowchart การทำงานของโปรแกรมหลัก

จากรูปที่ 3.2 แสดง Flowchart การทำงานของโปรแกรมหลัก สามารถอธิบายการทำงานได้ดังนี้

1. เริ่ม Start เข้ามาอ่านค่าเริ่มต้นต่างๆและอ่านการตั้งค่าจากไฟล์ setting.txt ใน SD-Card
2. กำหนดเงื่อนไขการวนลูปแบบ While มีเงื่อนไขเป็นจริงตลอด
3. เริ่มทำการอ่านค่าวันที่และเวลาจาก RTC
4. ตรวจสอบการนับเป็นวินาที เนื่องจากลูปของ While จะวนลูปเร็วกว่า 1 วินาที ทำให้ค่าการนับมากกว่าความจริง
   - ถ้าจริง เพิ่มค่าวินาทีเก่าให้เท่ากับวินาทีใหม่ เริ่มทำการตรวจสอบค่าการนับวินาทีเท่ากับค่าวินาทีที่ต้องการบันทึกค่าทั้ง 16 ช่อง แล้วเตรียมการบันทึกแบบ Discontinues
   - ถ้าเท็จ ให้ออกจาก เงื่อนไขนี้
5. ตรวจสอบการรับข้อมูลจาก PC
   - ถ้าจริง เริ่มทำการรับข้อมูลจาก PC
   - ถ้าเท็จ ให้ออกจาก เงื่อนไขนี้
6. ตรวจสอบช่วงเวลาการบันทึกของแต่ละช่อง
   - ถ้าจริง ตรวจสอบการเลือกใช้งานช่องบันทึก เริ่มอ่านค่า ADC แล้วตรวจสอบว่าเป็นการบันทึกแบบ Discontinues หรือ Continues และทำการบันทึกข้อมูลลง SD-Card
   - ถ้าเท็จ ให้ออกจาก เงื่อนไขนี้

3.1.2 การเขียนโปรแกรมเพื่อใช้งาน ADC (Analog-to-Digital Converter)

สำหรับการแปลง ADC ของบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA2560 จะใช้การแปลงแบบ Successive Approximation ADC (SAC)

สำหรับวงจรแปลงสัญญาณชนิด Successive Approximation ADC ขนาด "n" Bit จะมีความละเอียด (Resolution) เท่ากับ 1 LSB

ช่วงระยะเวลาแต่ละสเตป (Step time)

$$T_s = T_o = 1/f_o$$

ระยะเวลาการแปลงสัญญาณ (Conversion time) จะขึ้นอยู่กับจำนวนบิต ของการเปลี่ยนแปลง ดังนั้นระยะเวลาการเปลี่ยนแปลงสูงสุดคือ

$$T_{c(\max)} = n * T_s$$

สำหรับการแปลง ADC ของบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA2560 ผลที่ได้คือ

$$ADC = \frac{V_{IN} * 1024}{V_{REF}}$$

โดยที่ V_{IN} คือ แรงดันไฟฟ้าอินพุต

V_{REF} คือ แรงดันไฟฟ้าอ้างอิง (5 V)

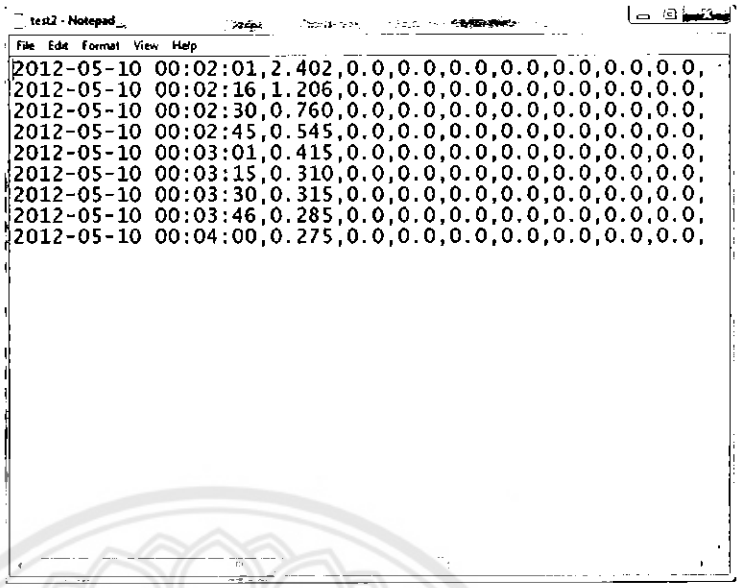
1024 คือ 2^n โดยที่ $n = 10$ bit

3.1.3 การเขียนโปรแกรมเพื่อใช้งาน SD-Card

การบันทึกค่าและการอ่านค่าจาก SD-Card จะใช้ Text Files เป็นไฟล์เก็บข้อมูลที่เป็นตัวอักษร ซึ่งก็คือไฟล์ข้อความนั่นเอง โดยหากเก็บในรูปแบบของรหัส ASCII (หรือ ANSI) ข้อมูลแต่ละ byte จะแทนตัวอักษร 1 ตัว หากเก็บในรูปแบบของรหัส Unicode ข้อมูลทุกๆ 2 bytes จะแทนตัวอักษร 1 ตัว รูปแบบการเก็บตัวอักษรจะแตกต่างกันไปตามมาตรฐานรหัสที่ใช้เก็บ ซึ่งนอกเหนือจาก ASCII และ Unicode ก็ยังมีรหัสแบบ UTF-8 เป็นต้น

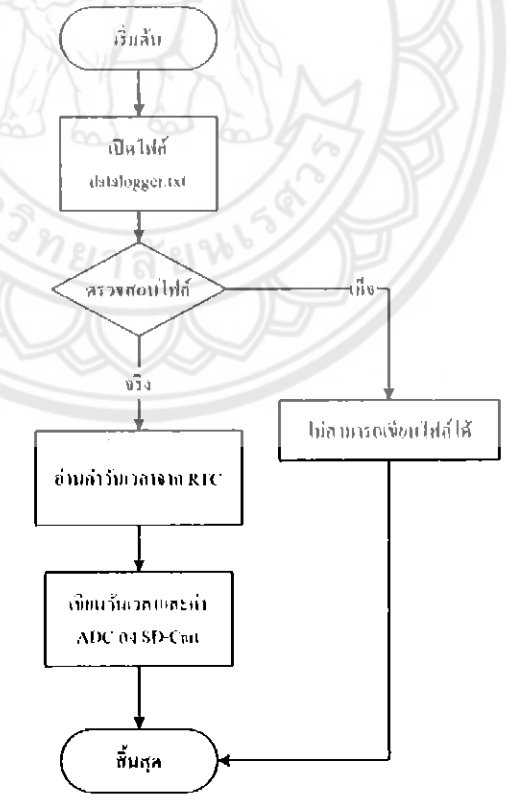
การทำงานกับ Text file มีการดำเนินการหลักๆ อยู่ 2 อย่างคือ

1. การอ่านข้อมูลจากไฟล์
2. การบันทึกข้อมูลลงในไฟล์



รูปที่ 3.3 ข้อมูลที่บันทึกลง Text File ที่ได้จาก SD-Card

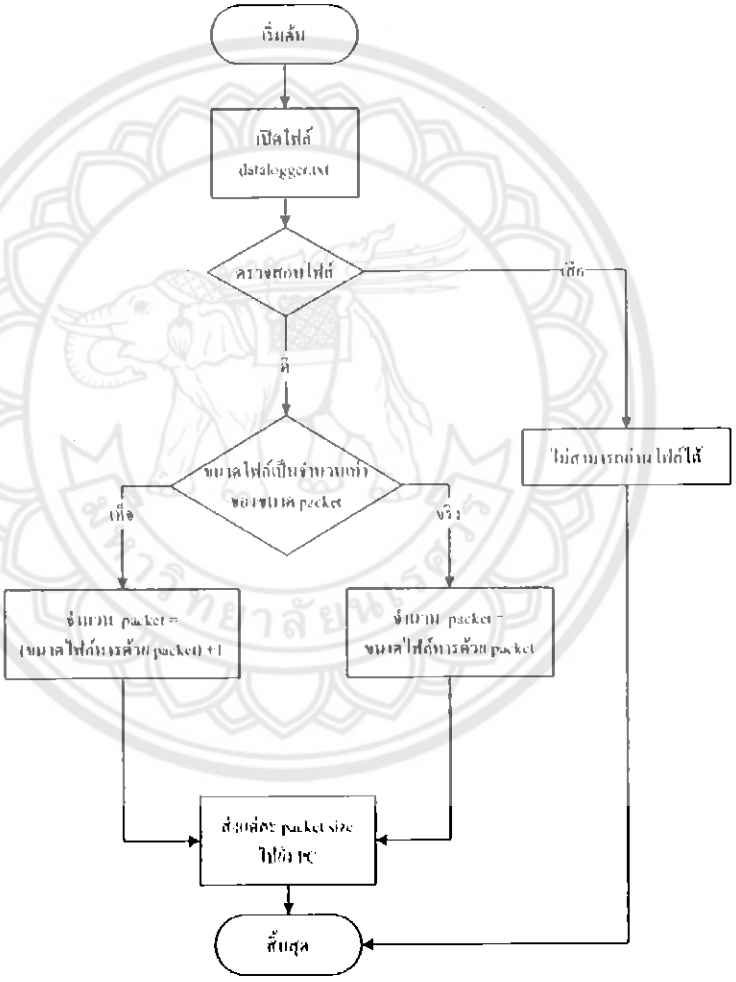
จากรูปที่ 3.5 ข้อมูลที่บันทึกลง Text File จะเริ่มต้นที่ วันที่ เวลาและแรงดันไฟฟ้าตั้งแต่ Channel ที่ 1 ถึง 16 ตามลำดับ



รูปที่ 3.4 แสดง Flowchart การทำงานของโปรแกรมของ WriteTextFile

จากรูปที่ 3.4 แสดง Flowchart การทำงานของ WriteTextFile Flowchart สามารถอธิบายการทำงานได้ดังนี้

1. เริ่ม Start เข้ามาเปิดไฟล์ datalogger.txt
2. ตรวจสอบการเปิดไฟล์
   - ถ้าจริง เริ่มการอ่านค่า ADC,วันที่และเวลาและเริ่มทำการเขียนค่า ADC, วันที่และเวลาลงไฟล์ datalogger.txt ใน SD-Card
   - ถ้าเท็จ เปิดไฟล์ไม่ได้จะสิ้นสุดการทำงาน

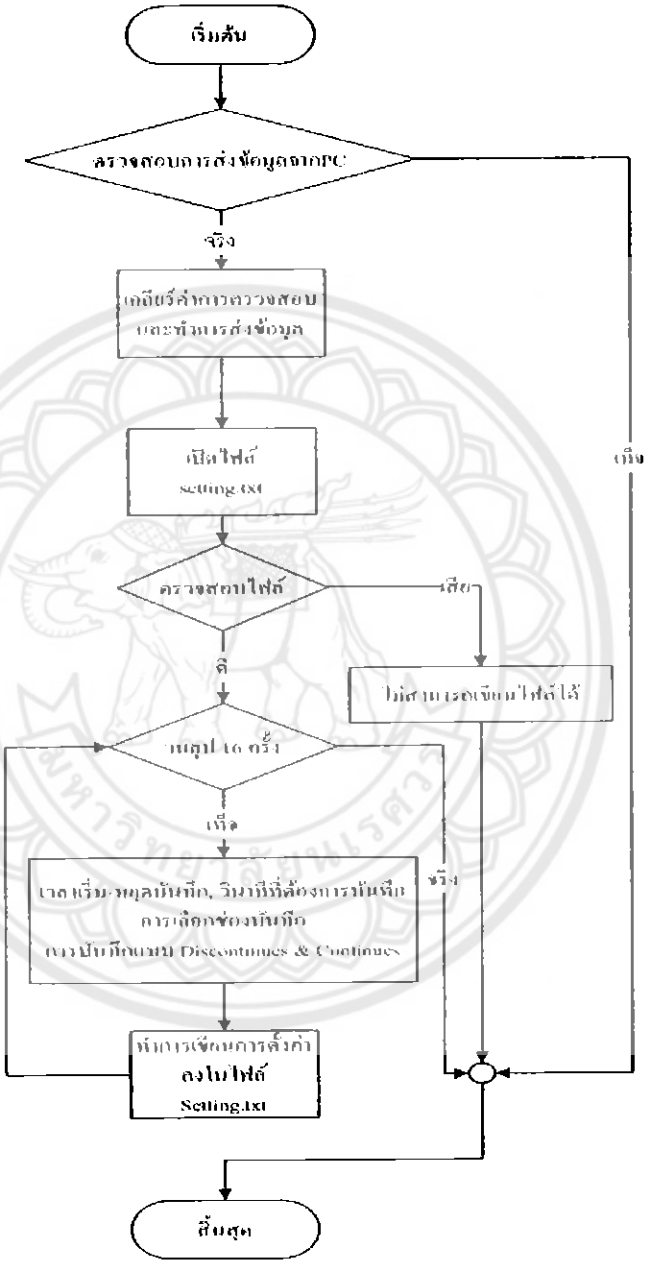


รูปที่ 3.5 แสดง Flowchart การทำงานของโปรแกรมของ ReadTextSendFile

จากรูปที่ 3.5 แสดง Flowchart การทำงานของ ReadTextSendFile Flowchart สามารถอธิบายการทำงานได้ดังนี้

1. เริ่ม Start เข้ามาเปิดไฟล์ datalogger.txt
2. ตรวจสอบการเปิดไฟล์

- ถ้าดี เริ่มการหาขนาดไฟล์ทั้งหมดและจำนวนของ Packet Size และ เริ่มทำการส่งแต่ละ Packet Size ไปยัง PC โดยใช้โปรแกรม Visual Basic 6.0 แสดงผล
- ถ้าเสีย เปิดไฟล์ไม่ได้จะสิ้นสุดการทำงาน



รูปที่ 3.6 แสดง Flowchart การทำงานของโปรแกรมของ WriteSetting

จากรูปที่ 3.6 แสดง Flowchart การทำงานของ WriteSetting Flowchart สามารถอธิบายการทำงานได้ดังนี้

1. เริ่ม Start เข้ามาเปิดไฟล์ setting.txt
2. ตรวจสอบการเปิดไฟล์
   - ถ้าดี เริ่มการวนลูป 16 ครั้ง เพื่อเก็บค่า เวลาเริ่ม-หยุดบันทึก, วินาทีที่ต้องการบันทึกการเลือกช่องบันทึกการบันทึกแบบ Discontinues & Continues ทำการเขียนการตั้งค่าลงในไฟล์ Setting.txt
   - ถ้าเสีย เปิดไฟล์ไม่ได้จะสิ้นสุดการทำงาน

3.1.4 การส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

ในการส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม จะต้องเริ่มต้นด้วยการกำหนดอัตราการรับส่งข้อมูลระหว่างไมโครคอนโทรเลอร์กับคอมพิวเตอร์ก่อน โดยจะต้องมีการกำหนดให้ตรงกัน หากต้องการให้อุปกรณ์ทั้งสองที่ติดต่อกันโดยรับส่งข้อมูลที่เฉพาะเจาะจง จะต้องประกอบไปด้วย ลักษณะการส่งข้อมูล รูปแบบการส่งข้อมูล การเริ่มต้นการติดต่อ และการยกเลิกการติดต่อ สิ่งเหล่านี้เรียกว่า โปรโตคอล [14]

โปรโตคอลเป็นการตกลงกันระหว่างอุปกรณ์ทั้งฝ่ายรับและฝ่ายส่ง หรือการรับข้อมูลระหว่างกันว่าจะใช้การส่งข้อมูลกันในลักษณะไหน เพราะหากไม่มีการกำหนดรูปแบบการรับ-ส่งที่เป็นมาตรฐานเดียวกันแล้ว จะทำให้ไม่สามารถสื่อสารกันได้ ในการรับส่งแบบโปรโตคอลที่ใช้จะใช้ในรูปแบบ HDLC Protocol (High-Level Data Link Control) [14]

HDLC Protocol ที่ช่วยเพิ่มความเร็วในการส่ง-รับข้อมูลแบบซิงโครนัส สามารถใช้ได้กับการเชื่อมโยงการส่งข้อมูลทั้งแบบจุดต่อจุด และแบบหลายจุด การส่ง-รับข้อมูลเป็นได้ทั้ง Half Duplex และ Full Duplex ข้อมูลที่ส่ง-รับเป็นแบบบิตข้อมูล (Bit-Oriented) ใช้วิธีการส่งผ่านข้อมูลแบบซิงโครนัส [14]

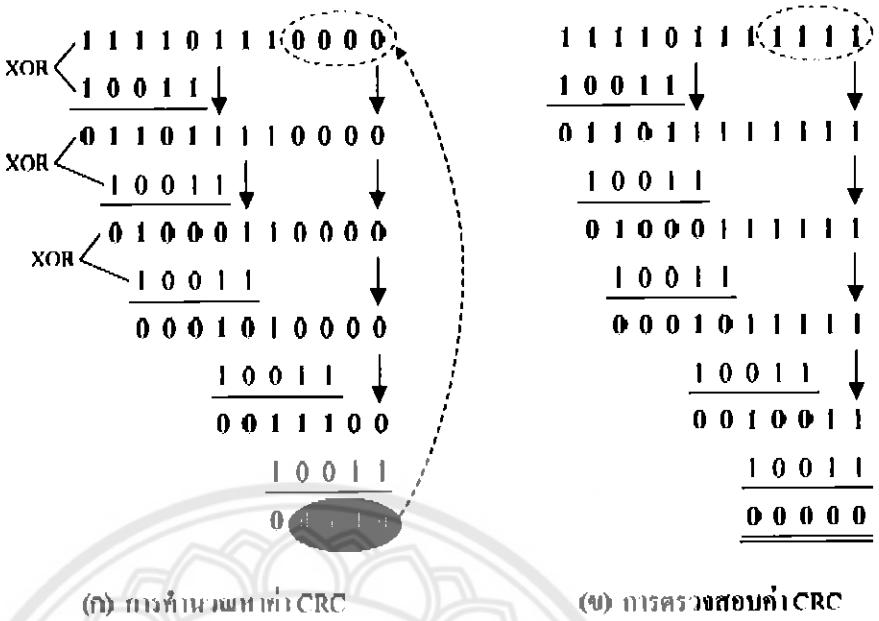
คุณสมบัติพิเศษอย่างหนึ่งของโปรโตคอล HDLC คือแทนที่จะต้องส่งข้อมูลไปทีละ 1 บล็อก แล้วต้องคอยตอบรับแบบบล็อกต่อบล็อก เช่น ในการสื่อสารในโปรโตคอล SDLC (Synchronous Data Link Control) สถานีปฐมภูมิสามารถส่งข้อมูล ได้ทีละหลาย ๆ บล็อกพร้อมกันแล้วค่อยรอการตอบรับจากสถานีทุติยภูมิว่าผลรับข้อมูลเป็นอย่างไร ซึ่งทำให้ลด การเสียเวลาในการตอบรับบล็อก ซึ่งอาจจะมีการตอบรับเพียงครั้งเดียวสำหรับการส่งข้อมูลทั้งหมดก็ได้ วิธีการ นี้เรียกว่า "Sliding Windows" [14]

โปรโตคอล HDLC ให้ประสิทธิภาพในแง่ที่ประหยัดเวลา สามารถเพิ่มความเร็วในการส่งข้อมูล ได้ดีกว่าโปรโตคอล SDLC ถ้าไม่มีความผิดพลาด หรือมีความผิดพลาดในการส่งข้อมูลน้อย แต่ถ้ามีความผิด พลาดเกิดขึ้นมากจะทำให้เสียเวลาในการส่งข้อมูลย้อนหลังมาให้กับสถานีทุติยภูมิใหม่บ่อย ๆ ทำให้ยิ่งเสียเวลา กว่าการใช้โปรโตคอล SDLC [14]

ตารางที่ 3.1 รูปแบบการส่งข้อมูลแบบ HDLC Protocol 9 byte

			Heximal	Decimal
Byte Start	1 byte	0x7E	0x7E	126
Address Destination	1 byte	0x01 (your define)	0x01-0xFF	0 – 255
Command	1 byte	0x01 (your define)	0x01-0xFF	0 – 255
Offset	1 byte	0x01 (your define)	0x01-0xFF	0 – 255
Data size	2 bytes	[byte high] [byte low]	0x0000-0xFFFF	0 – 65535
Data	n bytes	[byte 1][byte 2][byte 3]...[byte n]	(your define)	
CRC	2 bytes	[byte high] [byte low](Calculate Data only)	0x0000-0xFFFF	0 – 65535
Byte Stop	1 byte	0x7E	0x7E	126

CRC ย่อมาจาก Cyclic Redundancy Check เป็นวิธีการนำข้อมูลชุดหนึ่งมาคำนวณเป็นตัวเลขตัวหนึ่งเพื่อใช้ในการตรวจสอบว่าข้อมูลชุดนั้นมีการเปลี่ยนแปลงหรือไม่ หลังจากผ่านกระบวนการอย่างใดอย่างหนึ่ง เช่น การจัดเก็บและอ่านข้อมูลภายในอุปกรณ์ฮาร์ดดิสก์ไดรฟ์ การส่งข้อมูลผ่านระบบเครือข่าย หรือการรับส่งข้อมูลในระบบ RFID เป็นต้น โดยทั่วไปการตรวจสอบข้อมูลด้วยวิธี CRC จะมีความถูกต้องแม่นยำมากกว่าวิธีการหาผลรวมตรวจสอบ (Checksum) มาก นอกจากนี้การตรวจสอบ CRC สามารถใช้งานได้ดีกับบล็อกข้อมูลขนาดใหญ่ที่มีจำนวนข้อมูลหลายๆ ไบต์ได้อย่างมีประสิทธิภาพหลักการทำงานของการตรวจสอบ CRC จะเริ่มจากการจัดข้อมูลให้อยู่ในรูปของพหุนาม (Polynomial) เช่น ข้อมูล "10111" จะมีรูปพหุนามคือ จากนั้นนำข้อมูลที่ต้องการจะคำนวณหาผลรวมตรวจสอบ CRC มาหารด้วยพหุนามตัวกำเนิด (Generating Polynomial) โดยเศษที่ได้จากการหารก็คือค่า CRC นั่นเอง โดยทั่วไปถ้าต้องการค่า CRC ขนาด n บิต ก็จะต้องใช้พหุนามตัวกำเนิดในรูปของเลขฐานสองที่มีจำนวน $n+1$ บิต ในทางปฏิบัติการหาค่า



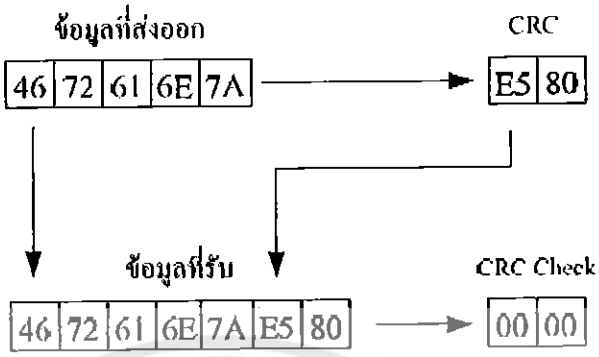
(ก) การคำนวณหาค่า CRC (ข) การตรวจสอบค่า CRC

รูปที่ 3.7 ตัวอย่างขั้นตอน (ก) การคำนวณหาค่า CRC และ (ข) การตรวจสอบค่า CRC [12]

ขั้นตอนแรกให้นำข้อมูลทั้งหมดมาเรียงตามแนวนอนและเพิ่มค่าเริ่มต้นเท่ากับจำนวนบิตของค่า CRC ที่ต้องการ นั่นคือ 4 บิตต่อท้ายบิตข้อมูล ซึ่งในตอนแรกค่าเริ่มต้นจะมีค่าเท่ากับค่าศูนย์ทั้งหมด10 ดังแสดงในภาพที่ 3.7 (ก) จากนั้นนำพหุนามตัวกำเนิดมาวางไว้ใต้ข้อมูลโดยให้บิตข้อมูลตัวแรกอยู่ในตำแหน่งที่ตรงกัน แล้วทำการ XOR ก็จะได้ผลลัพธ์เป็น "011011110000" ถ้าบิตข้อมูลตัวแรกทางด้านซ้ายมือของผลลัพธ์ที่ได้มีค่าเป็น 0 ก็ให้เลื่อนพหุนามตัวกำเนิดไปทางด้านขวา 1 บิต จนกระทั่งพบกับบิตข้อมูลของผลลัพธ์ที่ได้มีค่าเป็น 1 แล้วจึงจะทำการ XOR กันอีกครั้งทำการคำนวณแบบนี้ไปเรื่อยๆ จนกระทั่งบิตด้านขวาสุดของพหุนามตัวกำเนิดอยู่ในตำแหน่งเดียวกันกับบิตตัวสุดท้ายของข้อมูล ก็ให้ทำการ XOR กันอีกเป็นครั้งสุดท้าย ก็จะได้ผลลัพธ์เป็นค่า CRC จำนวน4 บิตตามที่ต้องการ ซึ่งในที่นี้ค่า CRC คือ "1111" [12]

การคำนวณหาค่า CRC ของข้อมูลที่ต่อกันหลายๆ บล็อกจะถูกกระทำทีละบล็อก โดยในตอนแรกค่าเริ่มต้นมีค่าเท่ากับค่าศูนย์ทั้งหมดสำหรับการคำนวณหาค่า CRC ของบล็อกข้อมูลที่หนึ่ง จากนั้นค่า CRC ที่ได้จะถูกนำมาใช้เป็นค่าเริ่มต้นของบล็อกข้อมูลที่สอง และค่า CRC ที่ได้ก็จะถูกนำมาใช้เป็นค่าเริ่มต้นของบล็อกข้อมูลที่สาม ทำแบบนี้ไปเรื่อยๆ จนถึงบล็อกข้อมูลสุดท้าย ก็จะได้ค่า CRC ของข้อมูลทั้งหมด [12]

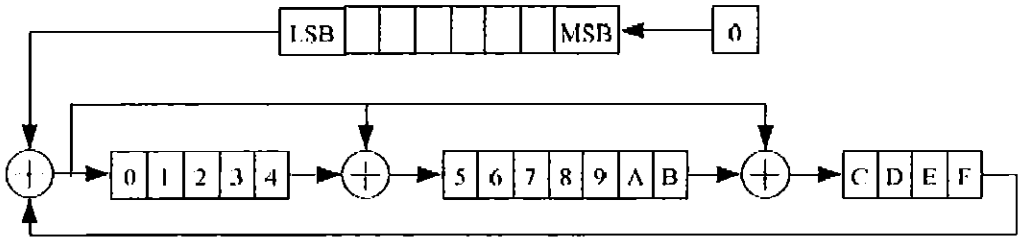
นำมาใช้เป็นค่าเริ่มต้นของบล็อกข้อมูลที่สาม ทำแบบนี้ไปเรื่อยๆ จนถึงบล็อกข้อมูลสุดท้าย ก็จะได้ค่า CRC ของข้อมูลทั้งหมด [12]



รูปที่ 3.8 ตัวอย่างขั้นตอนการตรวจสอบ CRC ของข้อมูลที่มีหลายๆ บล็อก [12]

เมื่อต้องการส่งบล็อกข้อมูลไปยังวงจรภาครับ ก็จะนำค่า CRC ที่คำนวณได้มาต่อท้ายบล็อกข้อมูลก่อนที่จะส่งไปยังวงจรภาครับ ในการตรวจสอบว่าข้อมูลที่วงจรภาครับได้รับว่ามีข้อผิดพลาดเกิดขึ้นหรือไม่ ก็สามารถทำได้โดยการคำนวณหาค่า CRC ของบล็อกข้อมูลที่ได้รับ โดยข้อมูลที่ได้รับจะไม่มีข้อผิดพลาดก็ต่อเมื่อผลลัพธ์ที่ได้จากการคำนวณหาค่า CRC มีค่าเป็นค่าศูนย์ทั้งหมด ดังแสดงในรูปที่ 3.7 (ข) นอกจากนี้รูปที่ 3.8 แสดงตัวอย่างขั้นตอนการตรวจสอบ CRC ของข้อมูลที่มีหลายๆบล็อก ซึ่งสามารถตรวจสอบความถูกต้องของข้อมูลได้อย่างรวดเร็ว และวงจรภาครับไม่ต้องทราบล่วงหน้าว่าค่า CRC ที่ส่งมาจากวงจรภาคส่งคือค่าอะไร กล่าวคือถ้าค่า CRC ที่วงจรภาครับคำนวณได้มีค่าเท่ากับค่าศูนย์ทั้งหมด แสดงว่าข้อมูลที่รับมาทั้งหมดไม่มีข้อผิดพลาดเกิดขึ้น [12]

ในทางปฏิบัติการตรวจสอบ CRC เพื่อคำนวณหาค่าศูนย์จะมีความซับซ้อนน้อยกว่าการเปรียบเทียบค่าผลรวมตรวจสอบ จึงเป็นเหตุผลทำให้การตรวจสอบ CRC เป็นที่นิยมใช้งานในหลายๆ งานประยุกต์ ข้อดีของการตรวจสอบ CRC คือมีความน่าเชื่อถือสูงในการตรวจหาข้อผิดพลาดของข้อมูล โดยข้อมูล CRC ขนาด 16 บิต สามารถใช้ในการตรวจสอบบล็อกข้อมูลขนาดความยาว 4กิโลไบต์ ได้อย่างมีประสิทธิภาพ อย่างไรก็ตามเนื่องจากข้อมูลที่รับส่งในระบบ RFID มีขนาดความยาวน้อยกว่า 4 กิโลไบต์ จึงทำให้สามารถใช้งานการตรวจสอบ CRC แบบ 8 หรือ 12 บิตก็เพียงพอ ตัวอย่างพหุนามตัวกำเนิดที่ใช้ในการหาค่า CRC แบบ 8 บิต คือ $x^8 + x^4 + x^3 + x^2 + 1$ และ CRC แบบ 16 บิต คือ $x^{16} + x^{12} + x^5 + 1$ รูปที่ 3.6 แสดงโครงสร้างของวงจรที่ใช้ในการสร้างข้อมูล CRC แบบ 16 บิต ซึ่งจะประกอบไปด้วยรีจิสเตอร์แบบเลื่อน (Shift Register) และตัวดำเนินการ XOR [12]



รูปที่ 3.9 โครงสร้างของวงจรที่ใช้ในการสร้างข้อมูล CRC แบบ 16 บิต [12]

การทำงานกับไฟล์ จะมีขั้นตอนหลักๆ คือ

1. เปิดไฟล์ โดยระบุชื่อหรือ path ของไฟล์
2. ทำการอ่านข้อมูลจากไฟล์ หรือเขียนข้อมูลลงไปในไฟล์
3. ปิดไฟล์

Command=0x03 คือ คำสั่งเมื่อ PC ต้องการอ่านค่าจาก Text File จาก AVR โดยจะส่ง "READTEXT" จากนั้น AVR จะส่ง Text File ให้ PC โดยแต่ละ Packet จะมีขนาด 524 byte แบ่งออกเป็น ลำดับ Packet 4 byte, จำนวน Packet 4 byte และ Data Size 4 byte ที่เหลือจะเป็นขนาดของข้อมูลใน Text File สูงสุดไม่เกิน 512 byte ดังตารางที่ 3.2 และ 3.3

ตารางที่ 3.2 คำสั่ง PC อ่าน Text file จาก AVR

Command	Ofset	Data Size	Data		
0x03	0x01	9 byte	"READ TEXT"		Ascii

ตารางที่ 3.3 คำสั่ง AVR ส่ง Text file ให้ PC

Command	Offset	Data Size	Data			
0x03	0x01	524 byte	[ลำดับ Packet 4 bytes][จำนวน Packet ทั้งหมด 4 bytes][DataSize 4 byte][dataของ Text file 512 byte]	ลำดับ Packet	"0"-"9999"	Ascii
				จำนวน Packet ทั้งหมด	"0"-"9999"	Ascii

ตารางที่ 3.3 (ต่อ) คำสั่ง AVR ส่ง Text file ให้ PC

			ตัวอย่างเช่น Text file ขนาด 1200 bytes จะแบ่งเป็น Packet ละ 512 byte จะได้จำนวน packet ทั้งหมดดังนี้	DataSize ของแต่ละ Packet	"0"-"9999"	Ascii
			1200 / 512 = 3 packet โดยรวม Packet ที่เป็น เศษด้วย			
			โดย ลำดับ Packetที่ 1 คือ 512 byte แรก	data ของ Text file 512 byte		Ascii
			ลำดับ Packetที่ 2 คือ 512 byte ต่อจาก Packetที่ 1			
			ลำดับ Packetที่ 3 คือ 176 byte ต่อจาก Packetที่ 2			
			ละ Packet ดังนี้			
			Packet ที่ 1			
			["0000"]["0003"]["0512"][data จำนวน 512 bytes]			
			Packet ที่ 2			
			["0001"]["0003"]["0512"][data จำนวน 512 bytes]			
			Packet ที่ 3			
			["0002"]["0003"]["0176"][data จำนวน 176 bytes]			

ในการส่งแต่ละ Packet PC จะต้องตอบ Command ดังตารางที่ 3.4

ตารางที่ 3.4 คำสั่ง PC ตอบกลับ Command 0x03

Command	Offset	Data Size	Data	ลำดับ Packet	"0"-"9999"	Ascii
0x03	0x02	8 byte	[ลำดับ Packet 4 bytes][จำนวน Packet ทั้งหมด 4 bytes]	จำนวน Packet ทั้งหมด	"0"-"9999"	Ascii

Command=0x04 คือ คำสั่งที่ PC ส่งไปลบ Text File ใน SD-Card โดยจะส่ง "DELETE TEXT" จากนั้น AVR จะตอบกลับมาว่าลบเรียบร้อยแล้ว โดยจะส่งคำว่า "DELETE COMPLETE" ดังตารางที่ 3.5 และ 3.6

ตารางที่ 3.5 คำสั่ง PC ส่งไป Delete Text file

Command	Offset	Data Size	Data		
0x04	0x01	11 byte	"DELETE TEXT"		Ascii

ตารางที่ 3.6 คำสั่ง AVR ตอบกลับ Delete Text file ให้ PC

Command	Offset	Data Size	Data		
0x04	0x01	15 byte	"DELETE COMPLETE"		Ascii

คำสั่ง Command = 0x05 คือคำสั่ง PC ส่งวันที่เวลาเริ่ม, วันที่เวลาหยุดบันทึก, ช่วงเวลาที่บันทึก, บันทึกแบบต่อเนื่อง (ถ้าเลือกบันทึกต่อเนื่องจะส่ง C ถ้าไม่เลือกจะส่ง U) และ Channel 0-16 (ถ้าเลือก Channel จะส่ง C ถ้าไม่เลือกจะส่ง U) ดังตารางที่ 3.7

ตารางที่ 3.7 คำสั่ง PC ส่งค่า Channel setting

Command	Offset	Data Size	Data			
0x05	0x01	26 byte	(วันที่เวลาเริ่ม12 byte) = [YY 2 byte][MM 2 byte] [DD 2 byte][hh 2 byte][nn 2 byte][ss 2 byte]			Ascii
			(วันที่เวลาหยุด12 byte)[YY 2 byte][MM 2 byte] [DD 2 byte][hh 2 byte][nn 2 byte][ss 2 byte]			Ascii

			interval (วินาที) [3 byte]	interval	"0"-"999"	Ascii
			continus record [1 byte]	continus record	"U" = unchecked, "C" = checked	Ascii
			Channel No [2 byte]	Channel No	"0"-"16"	Ascii
			Check Used [1 byte]	Check Used	"U" = unchecked, "C" = checked	Ascii

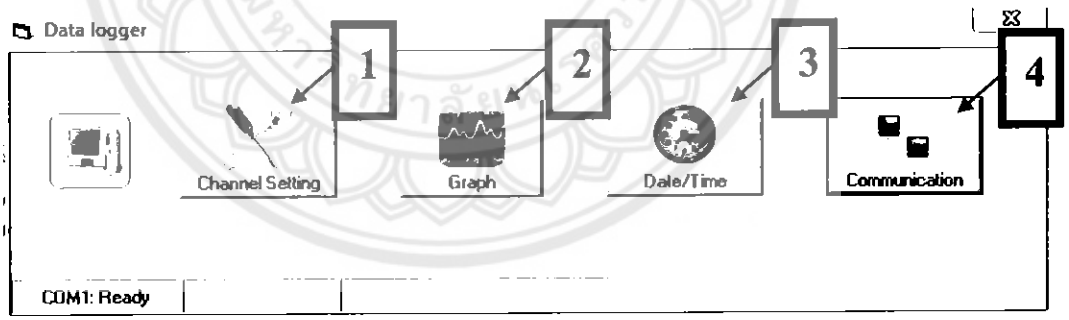
3.2 การออกแบบและพัฒนาโปรแกรม Visual Basic 6.0

ในการออกแบบและพัฒนาโปรแกรม Visual Basic 6.0 แบ่งออกเป็น 2 ส่วน คือ

1.ส่วนสำหรับการตั้งค่าการทำงานของอุปกรณ์เก็บข้อมูล

2.ส่วนแสดงผลของข้อมูล

โดยมีหน้าต่างหลักแสดงดังรูปที่ 3.13



รูปที่ 3.10 รูปร่างหน้าต่างหลักของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0

จากรูปที่ 3.10 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 คือ ปุ่ม Channel Setting ใช้แสดงหน้าต่าง Channel Setting

ตำแหน่งที่ 2 คือ ปุ่ม Graph ใช้แสดงหน้าต่าง Graph

ตำแหน่งที่ 3 คือ ปุ่ม Date/Time ใช้แสดงหน้าต่าง Date/Time

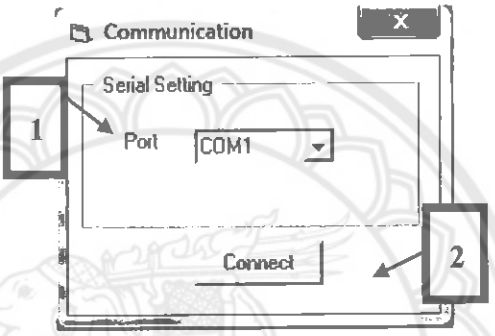
ตำแหน่งที่ 4 คือ ปุ่ม Communication ใช้แสดงหน้าต่าง Communication

3.2.1 ตั้งค่าการทำงานของอุปกรณ์เก็บข้อมูล

การออกแบบและพัฒนาโปรแกรม Visual Basic 6.0 มีลำดับขั้นตอนการทำงานของโปรแกรมที่ใช้ในการแสดงผลของสัญญาณดังนี้

1. เชื่อมต่อพอร์ทที่ใช้ในการรับส่งข้อมูล
2. ตั้งค่าวันที่และเวลาให้กับอุปกรณ์
3. ตั้งค่าการทำงานในการบันทึกข้อมูล

ในการออกแบบรูปร่างหน้าต่างของจอมอนิเตอร์ตั้งค่าต่างๆของอุปกรณ์บันทึกข้อมูลนั้นแสดงในรูปที่ 3.11

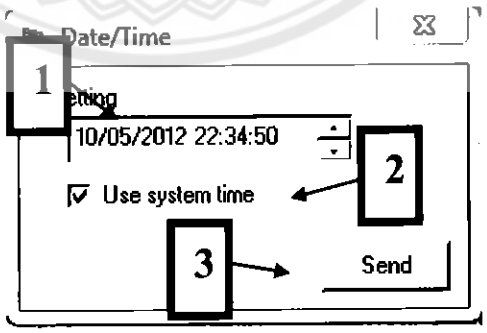


รูปที่ 3.11 รูปร่างหน้าต่างตั้งค่าการสื่อสารของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0

จากรูปที่ 3.11 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 กรอบข้อความ ใช้กำหนดพอร์ท

ตำแหน่งที่ 2 ปุ่ม Connect ใช้กำหนดการเชื่อมต่อระหว่าง PC กับไมโครคอนโทรลเลอร์



รูปที่ 3.12 รูปร่างหน้าต่างตั้งค่าเวลาของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0

จากรูปที่ 3.12 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 กรอบข้อความ ใช้กำหนดวันที่และเวลา

จากรูปที่ 3.12 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 กรอบข้อความ ใช้กำหนดวันที่และเวลา

ตำแหน่งที่ 2 กล่องตัวเลือก เลือกเมื่อต้องการให้เวลาตรงกับเวลาในคอมพิวเตอร์

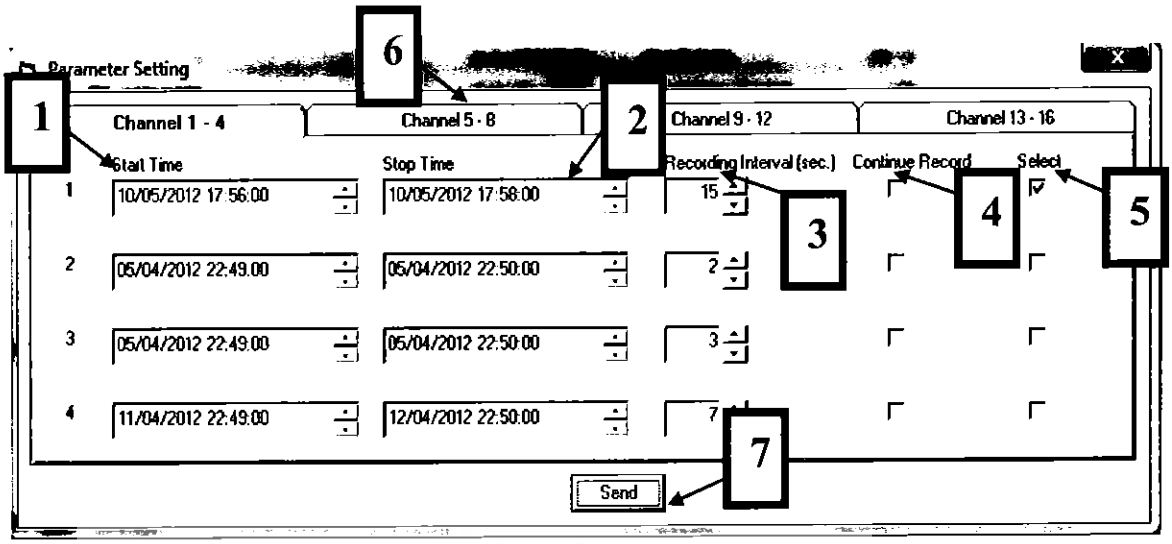
ตำแหน่งที่ 3 ปุ่ม Send ใช้ส่งข้อมูลจาก PC ไปยังไมโครคอนโทรลเลอร์

จากรูปที่ 3.14 หน้าต่างตั้งเวลาของอุปกรณ์เก็บข้อมูลนั้นจะถูกนำไปใช้ในกรณีที่มีการนำอุปกรณ์นี้ไปใช้ยังต่างประเทศเพื่อทำการตั้งเวลาให้ตรงกับประเทศนั้น

ตัวอย่าง Code ของปุ่ม Send

Private Sub cmdSend_Click()	ใช้งานฟังก์ชันเมื่อมีการกดปุ่ม cmdSend
EnableButton (False)	ไม่สามารถกดปุ่มได้
Screen.MousePointer = vbHourglass	กำหนดให้เมาส์แสดงเป็นลักษณะนาฬิกาทราย
On Error Resume Next	ทำต่อไป เมื่อข้อผิดพลาดเกิดขึ้นในขณะนั้น
Dim DataDateTime As String	ประกาศตัวแปร DataDateTime เพื่อใช้เก็บค่าตัวอักษร

MainForm.StatusBar1.Panels(3).Text = ""
If Not MainForm.MSComm.PortOpen Then
MainForm.MSComm.PortOpen = True
MainForm.MSComm.InputMode = comInputModeBinary
MainForm.StatusBar1.Panels(1).Text = frmCommunication.cbbPortName.Text & ": Connected"
End If
DataDateTime = Format$(dtpkDateTime.Value, "yymmddhhnnss")
MainForm.InitialDataReceive
Call Module.CommPortSendData(&H1, &H1, &H1, DataDateTime)
Screen.MousePointer = vbDefault
EnableButton (True)
End Sub
	ถ้ามีการหมุนปุ่ม Send จะมีข้อความ ": Connected" ปรากฏบน StatusBar ช่องที่ 1 ของหน้าหลัก และจะส่งค่า DataDateTime ไปยังไมโครคอนโทรลเลอร์ โดยอาศัย Module.CommPortSendData ในการส่ง



รูปที่ 3.13 รูปร่างหน้าต่างตั้งค่าการทำงานของอุปกรณ์เก็บข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0

จากรูปที่ 3.13 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 กรอบข้อความ ใช้กำหนดเวลาเริ่มต้นการบันทึกข้อมูล

ตำแหน่งที่ 2 กรอบข้อความ ใช้กำหนดเวลาหยุดการบันทึกข้อมูล

ตำแหน่งที่ 3 กรอบข้อความ ใช้กำหนดช่วงเวลาในการบันทึกข้อมูล

ตำแหน่งที่ 4 กล่องตัวเลือก เลือกเมื่อใช้ในการบันทึกข้อมูลต่อเนื่อง

ตำแหน่งที่ 5 กล่องตัวเลือก เลือก Channel ที่ต้องการใช้งาน

ตำแหน่งที่ 6 แท็บหน้าต่าง ใช้แสดง Channel ที่ต้องการตั้งค่า

ตำแหน่งที่ 7 ปุ่ม Send ใช้ส่งข้อมูลจาก PC ไปยังไมโครคอนโทรลเลอร์

ตัวอย่าง Code ของปุ่ม Send

Private Sub cmdSend_Click()	ใช้งานฟังก์ชันเมื่อมีการกดปุ่ม cmdSend
On Error Resume Next	ทำต่อไป เมื่อข้อผิดพลาดเกิดขึ้นในขณะนั้น
EnableButton (False)	ไม่สามารถกดปุ่มได้
Dim ParameterID As String	ประกาศตัวแปร ParameterID เพื่อใช้เก็บค่าตัวอักษร
Dim RecordInterval As String	ประกาศตัวแปร RecordInterval เพื่อใช้เก็บค่า
Dim IsContinueRecord As String	ตัวอักษร
Dim IsSelect As String	ประกาศตัวแปร D ContinueRecord เพื่อใช้เก็บค่า
Dim intRecordInterval As Integer	ตัวอักษร
Dim Index As Integer	ประกาศตัวแปร IsSelect เพื่อใช้เก็บค่าตัวอักษร

	ประกาศตัว intRecordInterval เพื่อใช้เก็บค่าตัวอักษร ประกาศตัวแปร Index เพื่อใช้เก็บค่าตัวอักษร

MainForm.StatusBar1.Panels(3).Text = ""
If Not MainForm.MSComm.PortOpen Then
MainForm.MSComm.PortOpen = True
MainForm.MSComm.InputMode = comInputModeBinary
MainForm.StatusBar1.Panels(1).Text = frmCommunication.cbbPortName.Text & ": Connected"
End If
For Index = 0 To 15
intRecordInterval = CInt(txtRecordInterval(Index).Text)
RecordInterval = Format$(intRecordInterval, "000")
If (chkContinueRecord(Index).Value = Checked) Then
IsContinueRecord = "C"
Else
IsContinueRecord = "U"
End If
intRecordInterval = Index + 1
ParameterID = Format$(intRecordInterval, "00")
If (chkSelect(Index).Value = Checked) Then
IsSelect = "C"
Else
	ถ้ามีการหมุนปุ่ม Send จะมีข้อความ ": Connected" ปรากฏบน StatusBar ช่องที่ 1 ของหน้าหลัก
	จากนั้นทำการตรวจสอบค่า IsContinueRecord ว่าเป็นแบบ Continues (C) หรือ Discontinues (U) และจากนั้นทำการตรวจสอบค่า IsSelect ว่าเลือกใช้งาน (C) หรือ ไม่เลือกใช้งาน (U) จะส่งค่า DataSend ไปยังไมโครคอนโทรลเลอร์ โดยอาศัย Module.CommPortSendData ในการส่ง วนลูปส่งค่า เวลาเริ่ม-หยุดบันทึก, วินาทีที่ต้องการบันทึก, การเลือกช่องบันทึก, การบันทึกแบบ Discontinues & Continues ของแต่ละช่องจนครบ 16 ช่อง

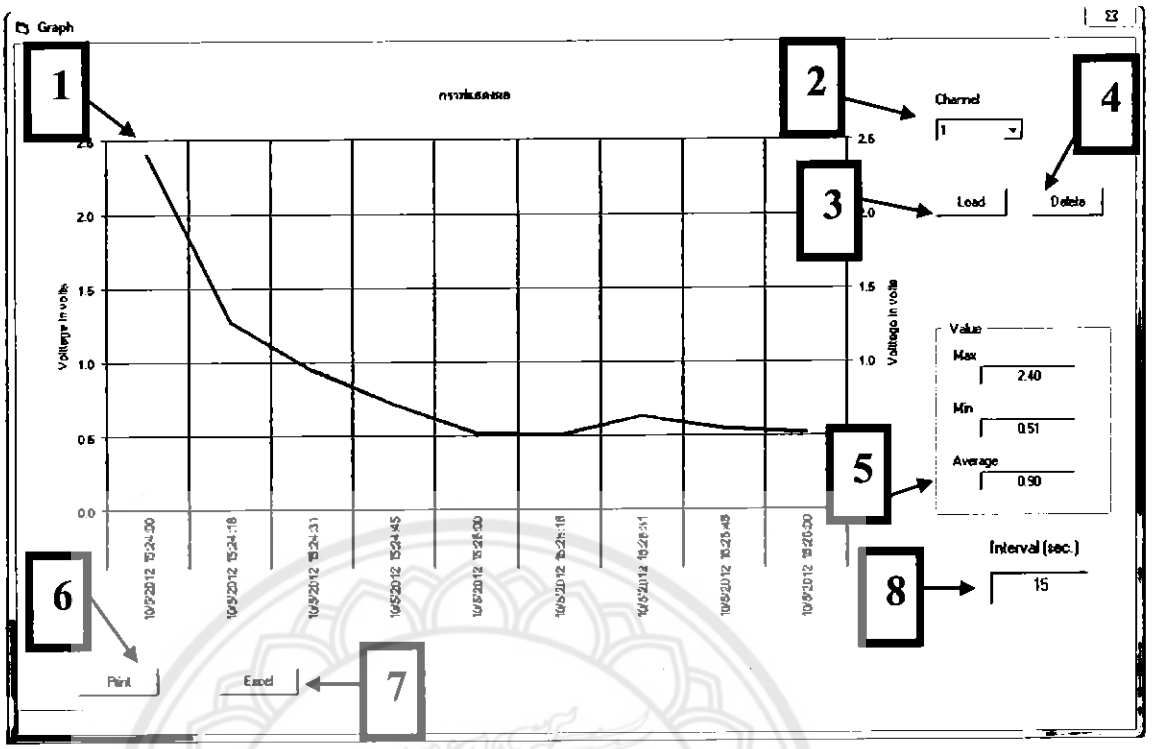
```
            IsSelect = "U"
        End If
        DataSend =
Format$(dtpkStartDateTime(Index).Value,
"yymmddhhnnss") _
                        &
Format$(dtpkStopDateTime(Index).Value,
"yymmddhhnnss") _
                    & RecordInterval &
IsContinueRecord & ParameterID & IsSelect
        MainForm.InitialDataReceive
        Call Module.CommPortSendData(&H1,
&H5, &H1, DataSend)
        SaveSetting (Index)
        Delay (2)
    Next Index
    EnableButton (True)
End Sub
```

3.2.2 การแสดงผลข้อมูล

ในการออกแบบและพัฒนาโปรแกรม Visual Basic 6.0 มีลำดับขั้นตอนการทำงานของโปรแกรมที่ใช้ในการแสดงผลสัญญาณดังนี้

1. รับข้อมูลที่บันทึกจาก SD-card
2. แสดงผลข้อมูลมีดังนี้
    2.1 เป็นกราฟระหว่างวันที่และเวลากับแรงดันไฟฟ้าที่บันทึกได้
    2.2 แรงดันไฟฟ้าสูงสุด, แรงดันไฟฟ้าต่ำสุดและแรงดันไฟฟ้าเฉลี่ย
    2.3 ช่วงเวลาในการบันทึก

ในการออกแบบรูปร่างหน้าต่างของจอมอนิเตอร์ตั้งค่าต่างๆของอุปกรณ์บันทึกข้อมูลนั้นแสดงในรูปที่ 3.16



รูปที่ 3.14 รูปร่างหน้าต่างแสดงผลข้อมูลที่พัฒนาจากโปรแกรม Visual Basic 6.0

จากรูปที่ 3.14 สามารถอธิบายรายละเอียดของแต่ละตำแหน่งได้ดังนี้

ตำแหน่งที่ 1 กล่องแสดงกราฟ ทำหน้าที่เป็นหน้าจอหลักในการแสดงผลข้อมูล

ตำแหน่งที่ 2 กรอบข้อความ ทำหน้าที่เลือกพารามิเตอร์ที่ใช้งาน

ตำแหน่งที่ 3 ปุ่มโหมดคำสั่ง ทำหน้าที่เชื่อมต่อการรับข้อมูลจาก SD card

ตำแหน่งที่ 4 ปุ่มกดทำหน้าที่ลบข้อมูลที่บันทึกใน SD card

ตำแหน่งที่ 5 กรอบข้อความ ทำหน้าที่แสดงค่าขนาดสูงสุด ค่าขนาดต่ำสุด ค่าขนาดเฉลี่ย

ตำแหน่งที่ 6 ปุ่ม Print เมื่อต้องการที่จะพิมพ์กราฟข้อมูลที่บันทึก

ตำแหน่งที่ 7 ปุ่มเซฟเป็นไฟล์ Excel เฉพาะ Channel ที่เลือกแสดงกราฟ

ตำแหน่งที่ 8 กรอบข้อความทำหน้าที่แสดงช่วงเวลาที่บันทึกของแต่ละ Channel

ตัวอย่าง Code ของปุ่ม Load

Private Sub cmdLoadData_Click() EnableButton (False) frmGetText.cmdClear_Click Screen.MousePointer = vbHourglass On Error Resume Next	ใช้งานฟังก์ชันเมื่อมีการกดปุ่ม cmdLoadData -ไม่สามารถกดปุ่มได้ -ล้างข้อมูลบนฟอร์ม -กำหนดให้เมาส์แสดงเป็นลักษณะนาฬิกาทราย -ทำต่อไป เมื่อข้อผิดพลาดเกิดขึ้นในขณะนั้น

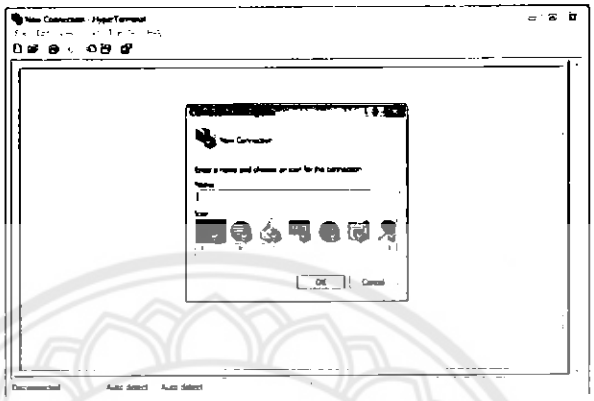
StatusBar1.Panels(1).Text = ""
If Not MainForm.MSComm.PortOpen Then
MainForm.MSComm.PortOpen = True
MainForm.MSComm.InputMode = comInputModeBinary
MainForm.StatusBar1.Panels(1).Text = frmCommunication.cbbPortName.Text & ": Connected"
End If
DataSend = "READ TEXT"
MainForm.InitialDataReceive
Call Module.CommPortSendData(&H1, &H3, &H1, DataSend)
Screen.MousePointer = vbDefault
EnableButton (True)
End Sub
	ถ้ามีการหมุนปุ่ม Send จะมีข้อความ ": Connected" ปรากฏบน StatusBar ช่องที่ 1 ของหน้าหลัก
	จากนั้นทำการส่ง DataSend = "READ TEXT"
	ไปยังไมโครคอนโทรลเลอร์ โดยอาศัย Module.CommPortSendData ในการส่ง เพื่อสั่งอ่านไฟล์จาก SD-Card

3.3 วิธีการทดสอบ

เพื่อตรวจสอบฟังก์ชันการทำงานของอุปกรณ์เก็บข้อมูลที่พัฒนาขึ้นด้านซอฟแวร์ จึงมีความจำเป็นต้องทดสอบประสิทธิภาพการทำงานของอุปกรณ์เก็บข้อมูลที่พัฒนาขึ้นหัวข้อย่อยนี้ จะกล่าวถึงการออกแบบการทดสอบอุปกรณ์ดังกล่าว 2 ส่วนหลัก คือ การทดสอบส่วนซอฟแวร์ การทดสอบการใช้งานอุปกรณ์เก็บข้อมูล

3.3.1 การทดสอบส่วนซอฟแวร์

เริ่มต้นด้วยการเขียนรหัสโปรแกรมภาษาซี ที่ใช้สำหรับการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลและนำข้อมูลมาบันทึกลง SD-Card โดยใช้โปรแกรมไฮเปอร์เทอร์มินอลแสดงผลดังรูปที่ 3.15



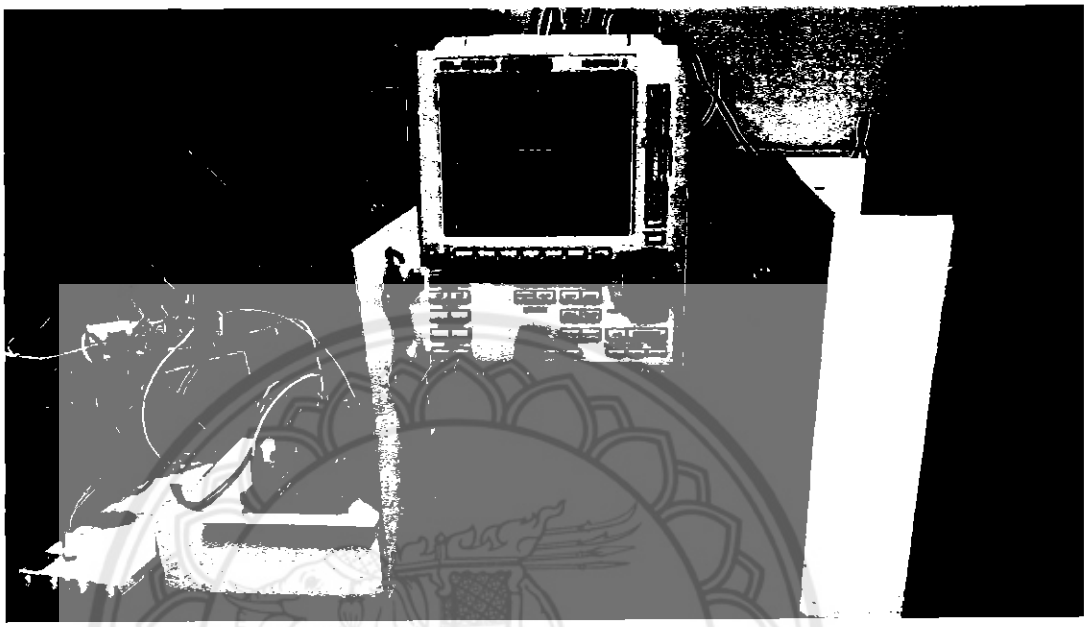
รูปที่ 3.15 ตัวอย่างหน้าต่างโปรแกรมไฮเปอร์เทอร์มินอล

เมื่อเชื่อมต่อการทำงานระหว่างบอร์ดไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ได้แล้ว จึงทำการออกแบบและเขียนโปรแกรม Visual Basic 6.0 เพื่อทำการแสดงผลสัญญาณ ซึ่งมีรหัสโปรแกรมแสดงผลสัญญาณ จากนั้นทำการทดสอบโปรแกรมขั้นสุดท้ายด้วยการต่อบอร์ดไมโครคอนโทรลเลอร์เข้ากับคอมพิวเตอร์ส่วนบุคคล เพื่อแสดงผลสัญญาณผ่านทางหน้าต่างจอมอนิเตอร์ของคอมพิวเตอร์ส่วนบุคคล

3.3.2 การทดสอบการใช้งานอุปกรณ์เก็บข้อมูล

เมื่อทำการทดสอบในส่วนของซอฟแวร์เสร็จเรียบร้อยแล้ว จึงนำอุปกรณ์เก็บข้อมูลและโปรแกรมที่พัฒนาขึ้นทั้งหมด มาทำการทดสอบ 3 การทดลองดังนี้

การทดลองที่ 1 เก็บข้อมูลแรงดันของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรด เพื่อหาประสิทธิภาพการทำงานของอุปกรณ์เก็บข้อมูล การทดลองนี้จะใช้กระดาษสีขาวในการทดลอง โดยมีขั้นตอนการทดสอบเก็บข้อมูล 6 ขั้นตอน ดังแสดงในรูปที่ 3.18



รูปที่ 3.16 แสดงการเก็บข้อมูลแรงดันของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรด

ขั้นตอนที่ 1 ต่อโมดูลตรวจจับระยะทางแบบอินฟราเรด 2 ตัว กับบอร์ดไมโครคอนโทรลเลอร์ โดยต่อเอาต์พุทของโมดูลตรวจจับระยะทางแบบอินฟราเรดตัวที่ 1 เข้ากับพอร์ท F ที่พิน 1 และต่อเอาต์พุทของโมดูลตรวจจับระยะทางแบบอินฟราเรดตัวที่ 2 เข้ากับพอร์ท F ที่พิน 2 ซึ่งคือ Channel 1 และ Channel 2 โดยที่ Channel 1 ต่อ Vcc ที่พิน 9, GND ที่พิน 10 และChannel 2 ต่อ Vcc ที่พิน 19, GND ที่พิน 20

ขั้นตอนที่2 ทำการเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ เข้ากับคอมพิวเตอร์ส่วนบุคคล โดยนำส่วนหัวของสายพอร์ทอนุกรม (RS232) ต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ และนำส่วนปลายต่อเข้ากับอุปกรณ์แปลงจากพอร์ทอนุกรมให้เป็นพอร์ทยูเอสบี เพื่อนำไปต่อเข้ากับคอมพิวเตอร์ส่วนบุคคล

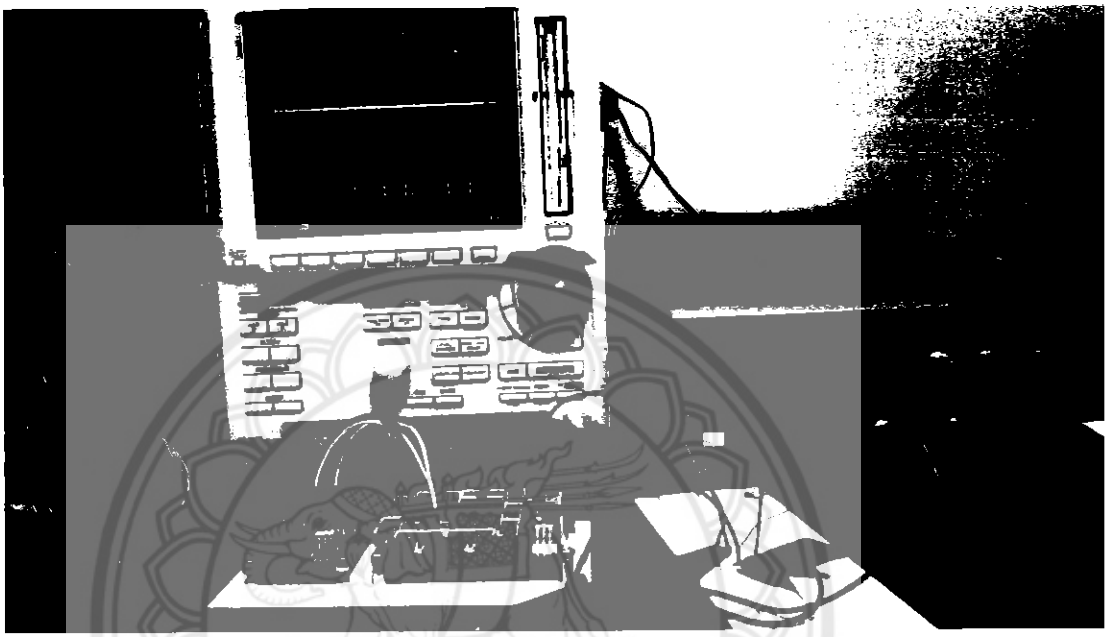
ขั้นตอนที่ 3 ตั้งค่าการเชื่อมต่อที่คอมพอร์ตที่ 1 (โดยจะกำหนดบอร์ดเรทเท่ากับ 9600) และตั้งค่าวันที่และเวลา

ขั้นตอนที่ 4 กำหนดค่าระยะที่จะบันทึกภายใน 2 นาที คือ 10 เซนติเมตร โดย Channel1 บันทึกช่วงเวลา 5 วินาที และ Channel2 บันทึกช่วงเวลา 7 วินาที

ขั้นตอนที่ 5 นำข้อมูลที่บันทึกที่ได้จากอุปกรณ์เก็บข้อมูล มาแสดงผลผ่านทางจอมอร์นิเตอร์ของคอมพิวเตอร์ส่วนบุคคล โดยมีการเปรียบเทียบค่าแรงดันไฟฟ้าที่บันทึกได้กับแรงดันไฟฟ้าที่วัดได้ออสซิลโลสโคปและเปรียบเทียบค่าระยะทางที่ได้กำหนดเทียบจากกราฟในคู่มือการใช้งานของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรดกับระยะทางที่กำหนดไว้คือ 10 เซนติเมตร เพื่อทำการทดสอบการบันที่ข้อมูลและความถูกต้องของข้อมูลของอุปกรณ์เก็บข้อมูลที่พัฒนาขึ้น

ขั้นตอนที่ 6 ทำการทดลองตั้งแต่ขั้นตอนที่ 1 ถึง 5 โดยเปลี่ยนระยะทางเป็น 20 เซนติเมตร

การทดลองที่ 2 บันทึกข้อมูลแรงดันไฟฟ้าของแหล่งจ่ายแรงดันโดยมีขั้นตอนการทดลองบันทึกข้อมูล 6 ขั้นตอนดังนี้



รูปที่ 3.17 แสดงการบันทึกข้อมูลแรงดันไฟฟ้าของแหล่งจ่ายแรงดัน

ขั้นตอนที่ 1 ต่อแรงดันไฟฟ้าที่ได้จากแหล่งจ่ายเข้ากับพอร์ท F ที่พิน 1 และ ที่พิน 2 ซึ่งคือ Channel 1 และ Channel 2 ต่อ GND ที่พิน 10

ขั้นตอนที่2 ทำการเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ เข้ากับคอมพิวเตอร์ส่วนบุคคล โดยนำส่วนหัวของสายพอร์ทอนุกรม (RS232) ต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ และนำส่วนปลายต่อเข้ากับอุปกรณ์แปลงจากพอร์ทอนุกรมให้เป็นพอร์ทยูเอสบี เพื่อนำไปต่อเข้ากับคอมพิวเตอร์ส่วนบุคคล

ขั้นตอนที่ 3 ตั้งค่าการเชื่อมต่อที่คอมพอร์ตที่ 1 (โดยจะกำหนดบอร์ดเรทเท่ากับ 9600) และตั้งค่าวันที่และเวลา

ขั้นตอนที่ 4 กำหนดค่าแรงดันไฟฟ้าที่จะบันทึกภายใน 2 นาที คือ 2 V โดย Channel1 บันทึกช่วงเวลา 5วินาที และ Channel2 บันทึกช่วงเวลา 7 วินาที

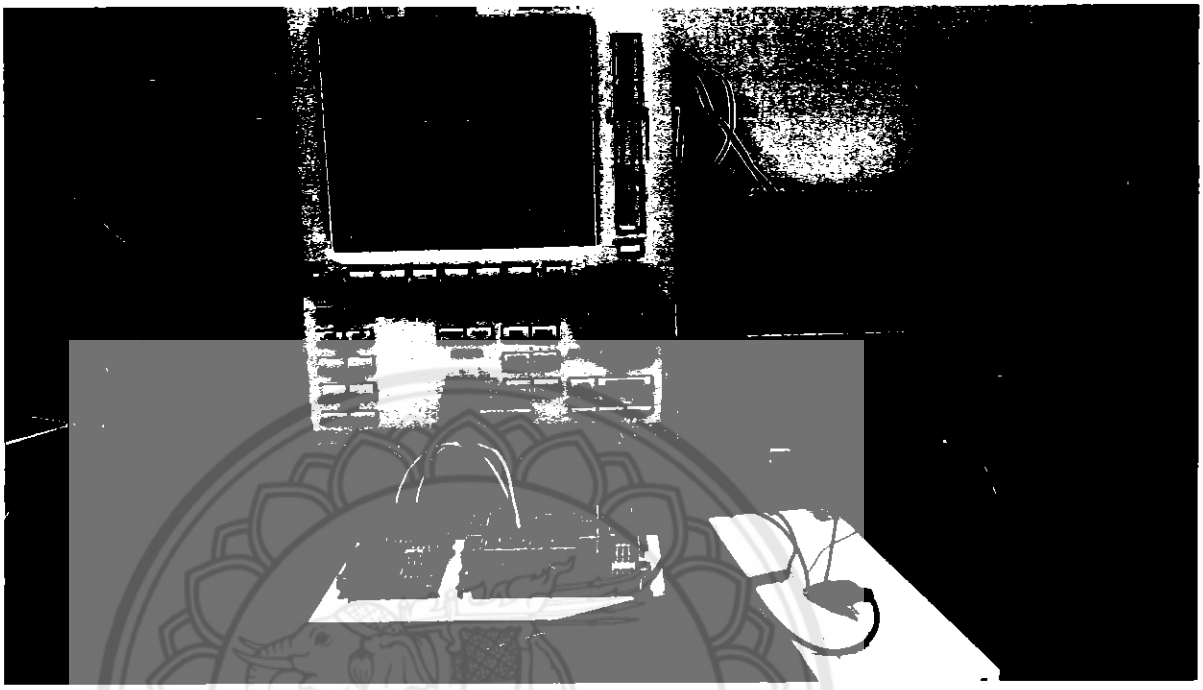
ขั้นตอนที่ 5 นำข้อมูลที่บันทึกที่ได้จากอุปกรณ์เก็บข้อมูล มาแสดงผลผ่านทางจอมอร์นิเตอร์ของคอมพิวเตอร์ส่วนบุคคล โดยมีการเปรียบเทียบค่าแรงดันไฟฟ้าที่บันทึกได้กับแรงดันไฟฟ้าที่วัดได้จากออสซิลโลสโคปและเปรียบเทียบค่าแรงดันไฟฟ้าที่บันทึกได้

แรงดันไฟฟ้าที่วัดได้จากออสซิลโลสโคปและเปรียบเทียบค่าแรงดันไฟฟ้าที่บันทึกได้กับแรงดันไฟฟ้าที่กำหนดไว้คือ 2 V เพื่อทำการทดสอบการบันที่ข้อมูลและความถูกต้องของข้อมูลของอุปกรณ์เก็บข้อมูลที่พัฒนาขึ้น

ขั้นตอนที่ 6 ทำการทดลองตั้งแต่ขั้นตอนที่ 1 ถึง 5 โดยเปลี่ยนแรงดันไฟฟ้าเป็น 4 V

มหาวิทยาลัยนเรศวร

การทดลองที่ 3 บันทึกข้อมูลแบบต่อเนื่องโดยจะบันทึกแรงดันไฟฟ้าของแหล่งจ่ายแรงดันไฟฟ้าโดยมีขั้นตอนการทดลองบันทึกข้อมูล 5 ขั้นตอนดังนี้



รูปที่ 3.18 แสดงการบันทึกข้อมูลแบบต่อเนื่องโดยจะบันทึกแรงดันไฟฟ้าของแหล่งจ่ายแรงดันไฟฟ้า

ขั้นตอนที่ 1 ต่อแรงดันไฟฟ้าที่ได้จากแหล่งจ่ายเข้ากับพอร์ต F ที่พิน 1 ซึ่งคือ Channel 1 ต่อ GND ที่พิน 10

ขั้นตอนที่2 ทำการเชื่อมต่อระหว่างบอร์ดไมโครคอนโทรลเลอร์ เข้ากับคอมพิวเตอร์ส่วนบุคคล โดยนำส่วนหัวของสายพอร์ทอนุกรม (RS232) ต่อเข้ากับบอร์ดไมโครคอนโทรลเลอร์ และนำส่วนปลายต่อเข้ากับอุปกรณ์แปลงจากพอร์ทอนุกรมให้เป็นพอร์ทยูเอสบี เพื่อนำไปต่อเข้ากับคอมพิวเตอร์ส่วนบุคคล

ขั้นตอนที่ 3 ตั้งค่าการเชื่อมต่อที่คอมพอร์ตที่ 1 (โดยจะกำหนดบอร์ดเรทเท่ากับ 9600) และตั้งค่าวันที่และเวลา

ขั้นตอนที่ 4 กำหนดค่าแรงดันไฟฟ้าที่จะบันทึกภายใน 10 วินาที คือ 2 V โดย Channel1 บันทึกแบบต่อเนื่อง

ขั้นตอนที่ 5 นำข้อมูลที่บันทึกที่ได้จากอุปกรณ์เก็บข้อมูล มาแสดงผลผ่านทางจอมอร์นิเตอร์ของคอมพิวเตอร์ส่วนบุคคล โดยมีการตรวจสอบว่าบันทึกได้กี่ครั้งต่อวินาที

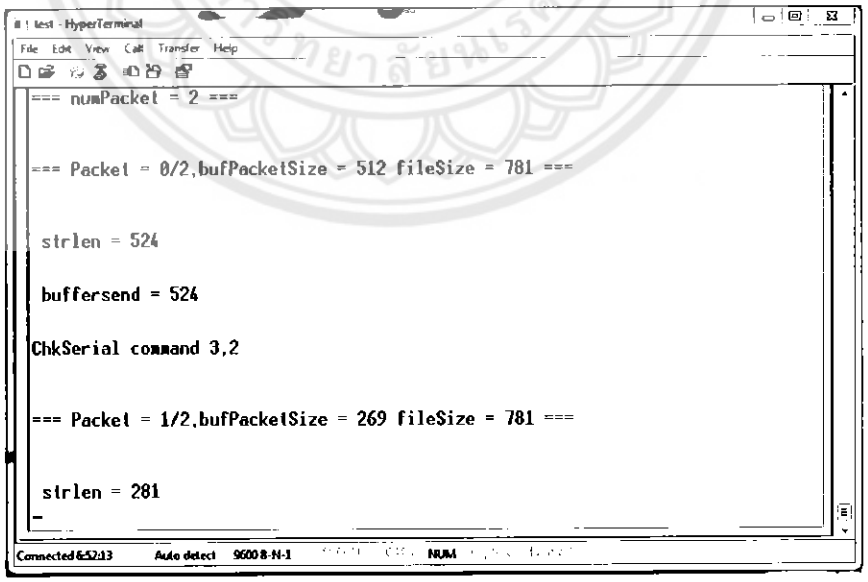
บทที่ 4
ผลการทดลองและสรุปผลการทดลอง

บทนี้ จะกล่าวถึงผลการทดลองและวิเคราะห์ผลการทดลองที่ได้รับจากการทดสอบประสิทธิภาพการทำงานของอุปกรณ์เก็บข้อมูลที่พัฒนาขึ้นตามที่กล่าวมาแล้วในบทที่ 3 ซึ่งแบ่งผลการทดลองและสรุปผลการทดลองออกเป็น 2 กรณี คือ ผลการทดลองส่วนซอฟแวร์ ผลการทดลองการใช้งานอุปกรณ์บันทึกข้อมูล

4.1 ผลการทดลองส่วนซอฟแวร์

4.1.1 โปรแกรมที่ใช้ในบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASEMEGA1280/2560

เมื่อทำการเขียนโปรแกรมภาษาซีซึ่งแสดงไว้ในภาคผนวก ก ลงบันทึกลงไปยังบอร์ดไมโครคอนโทรลเลอร์ พบว่า บอร์ดไมโครคอนโทรลเลอร์นั้นสามารถแปลงข้อมูลจากข้อมูลอนาลอกเป็นข้อมูลดิจิตอลและมีการบันทึกข้อมูลลงในSD-Card เมื่อมีการเรียกข้อมูลในหน่วยความจำ SD-Card จะทำให้ไมโครคอนโทรลเลอร์ส่งข้อมูลชุดนั้นออกไปยังช่องทางการติดต่อสื่อสารแบบอนุกรม และผ่านตัวแปลงจากการสื่อสารแบบอนุกรมให้เป็นการสื่อสารแบบยูเอสบี เพื่อนำไปแสดงผลการทดสอบการส่งข้อมูลจากไมโครคอนโทรลเลอร์ที่โปรแกรมไฮเปอร์เทอร์มินอล ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 ผลการทดสอบการส่งข้อมูลของไมโครคอนโทรลเลอร์

พิจารณารูปที่ 4.1 ผลการทดสอบการส่งข้อมูลของไมโครคอนโทรลเลอร์ โดยใช้โปรแกรมไฮเปอร์เทอร์มินอลในการแสดงผลทดสอบการส่งข้อมูล พบว่าไมโครคอนโทรลเลอร์ส่งข้อความ numPacket =2 (มีอยู่ 2 Packet), bufPacketsize = 512 (แต่ละ Packet มีขนาด 512 byte), strlen = 524 (การรวมตัวแปลเพื่อส่ง) มายังโปรแกรมไฮเปอร์เทอร์มินอล ซึ่งแสดงให้เห็นว่าโปรแกรมภาษาซีที่บันทึกลงในไมโครคอนโทรลเลอร์ สามารถติดต่อสื่อสารมายังยังคอมพิวเตอร์ส่วนบุคคลได้

4.1.2 วิธีการใช้งานอุปกรณ์บันทึกข้อมูล

ในการใช้งานของอุปกรณ์เก็บข้อมูลประกอบด้วย 2 ส่วนดังนี้

1.ส่วนการเชื่อมต่ออุปกรณ์

รูปที่ 4.2 แสดงการเชื่อมต่อของอุปกรณ์บันทึกข้อมูล

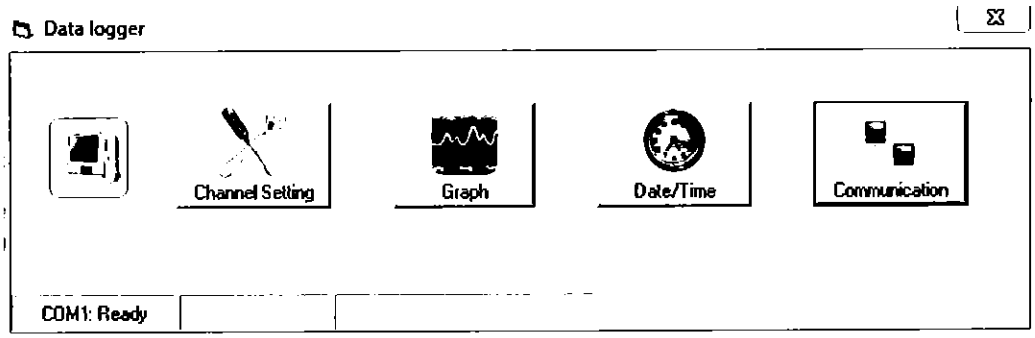
1.1 ต่อสาย RS232 กับพอร์ท UART0 ของบอร์ดไมโครคอนโทรลเลอร์และต่อเข้ากับคอมพิวเตอร์ โดยผ่านตัวแปลงเป็นยูเอสบี

1.2 ต่ออินพุทแรงดันฟ้าที่ต้องการบันทึกเข้าที่พินของพอร์ท F (channel1-8) และ K (channel9-16)

1.3 ต่อไฟเลี้ยง 5 V ให้กับบอดไมโครคอนโทรลเลอร์

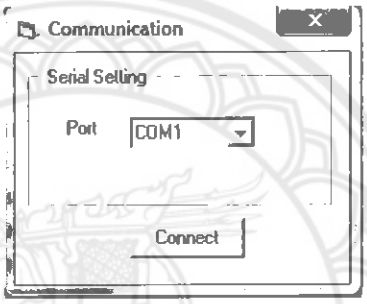
2.ส่วนของโปรแกรม

2.1 เปิดหน้าต่างหลักของโปรแกรมดังรูปที่



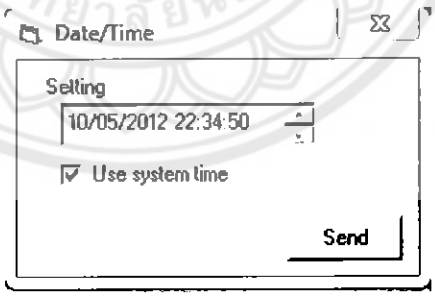
รูปที่ 4.3 หน้าต่างหลักของโปรแกรม

2.2 กดปุ่ม Communication จะแสดงหน้าต่างเพื่อตั้งค่าการเชื่อมต่อดังรูปที่ 4.4



รูปที่ 4.4 หน้าต่าง Communication

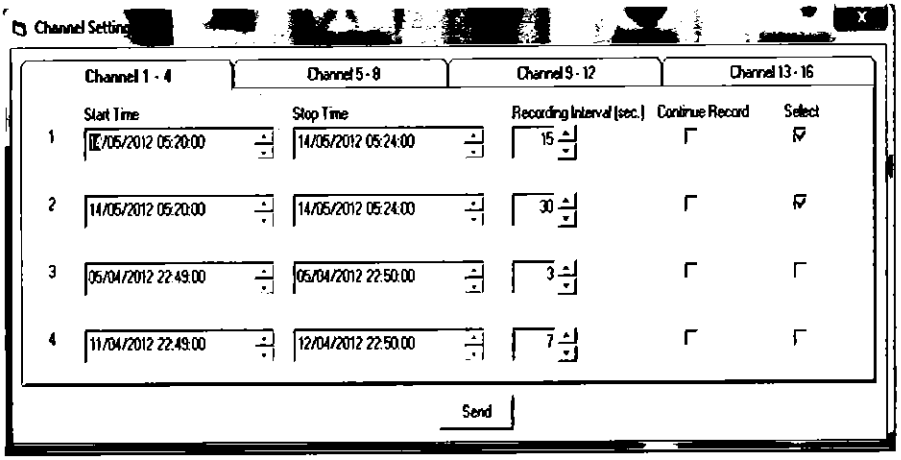
2.3 กดปุ่ม Date/Time จะแสดงหน้าต่างเพื่อตั้งค่าวันที่และเวลาให้กับอุปกรณ์ ดังรูปที่ 4.5



รูปที่ 4.5 หน้าต่าง Date/Time

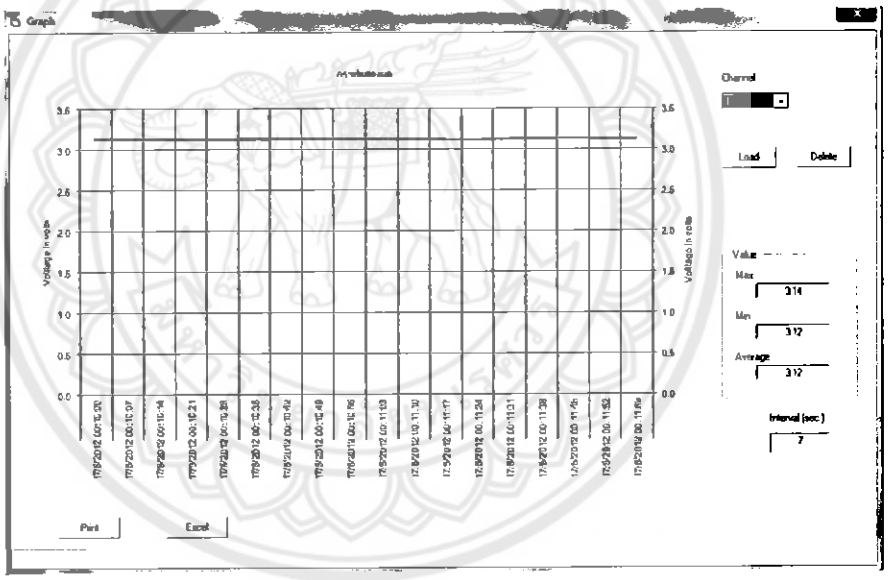
2.4 กดปุ่ม Channel Setting จะแสดงหน้าต่างเพื่อตั้งค่าช่วงเวลาการใช้งานในการบันทึกและกดปุ่ม send ดังรูปที่ 4.6

2.5 ถอดสายเชื่อมต่อที่จากคอมพิวเตอร์ออกแล้วนำ Data loggerที่พัฒนาขึ้นไปวางที่สถานที่ต้องการทำการบันทึก ทิ้งไว้จนครบเวลาที่กำหนดและนำกลับมาเชื่อมต่อกลับคอมพิวเตอร์เพื่อถ่ายโอนข้อมูลที่บันทึกได้จาก Data logger เข้าสู่เครื่องคอมพิวเตอร์



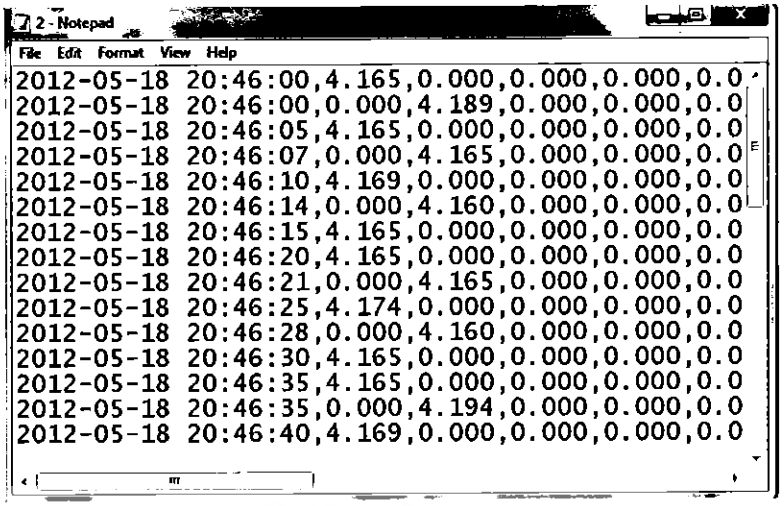
รูปที่ 4.6 หน้าต่าง Channel Setting

2.6 กดปุ่ม Graph จะแสดงหน้าต่างแสดงผลดังรูปที่ 4.7



รูปที่ 4.7 หน้าต่างที่ใช้ในการแสดงผล

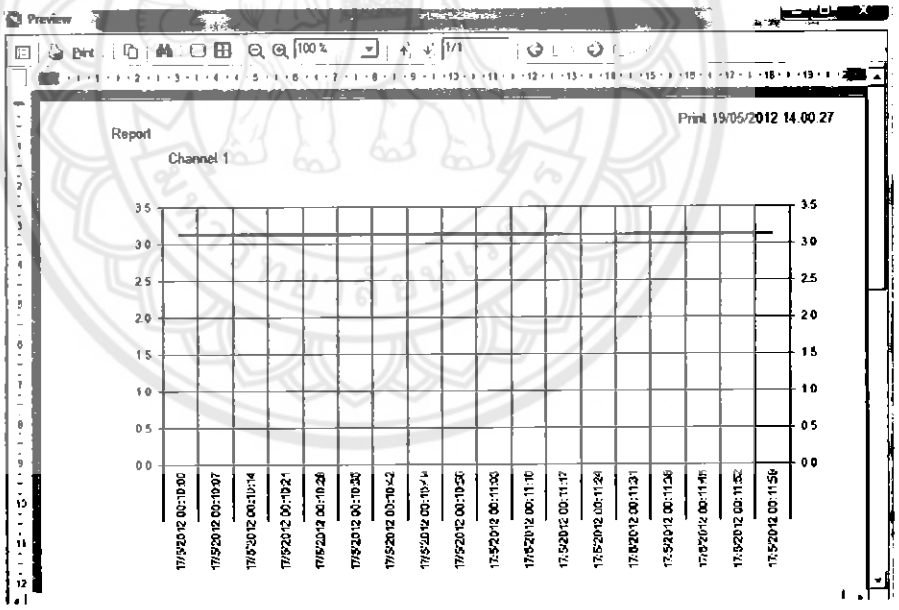
2.6.1 กดปุ่ม Load จะมีหน้าต่างขึ้นให้บันทึกไฟล์ Text ไว้ในที่ที่ต้องการ ตัวอย่างไฟล์ text ที่บันทึกข้อมูลดังรูปที่ 4.8



รูปที่ 4.8 ตัวอย่าง ไฟล์ Text ที่บันทึกข้อมูล

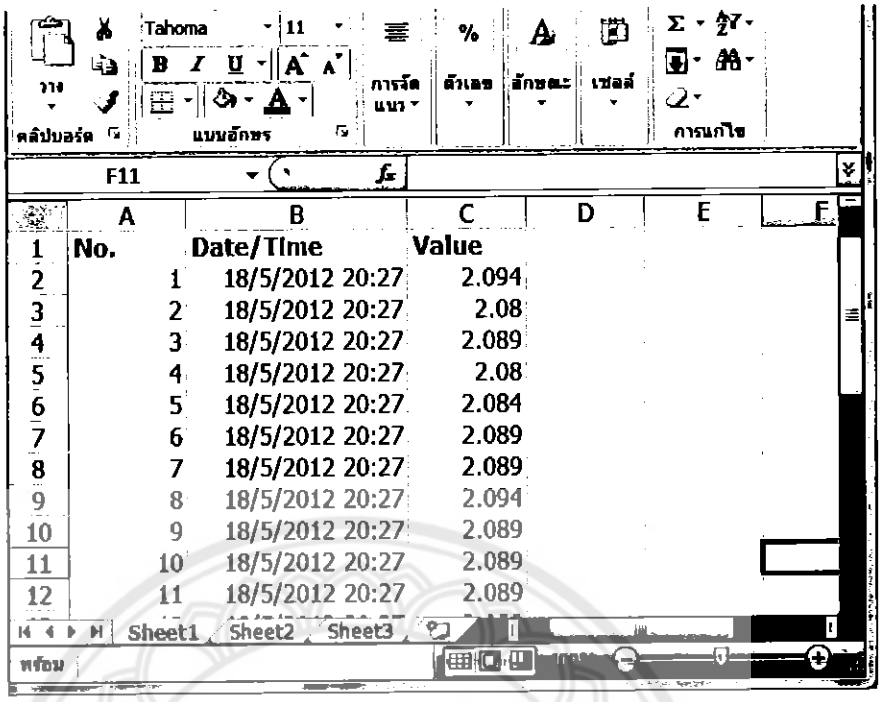
2.6.2 เลือก Channel ที่ต้องการแสดงผล

2.6.3 กดปุ่ม Print จะมีหน้าต่าง Preview ขึ้นมาเพื่อตั้งค่าและทำการพิมพ์ต่อไปดังตัวอย่างรูปที่ 4.9



รูปที่ 4.9 ตัวอย่างหน้าต่าง Preview การพิมพ์

2.6.4 กดปุ่ม Excel จะมีหน้าต่างขึ้นให้บันทึกไฟล์ Excel ไว้ในที่ที่ต้องการซึ่งค่าที่บันทึกจะเป็นค่าของ Channel นั้น ตัวอย่างไฟล์ Excel ดังรูปที่ 4.10



	A	B	C	D	E	F
1	No.	Date/Time	Value			
2	1	18/5/2012 20:27	2.094			
3	2	18/5/2012 20:27	2.08			
4	3	18/5/2012 20:27	2.089			
5	4	18/5/2012 20:27	2.08			
6	5	18/5/2012 20:27	2.084			
7	6	18/5/2012 20:27	2.089			
8	7	18/5/2012 20:27	2.089			
9	8	18/5/2012 20:27	2.094			
10	9	18/5/2012 20:27	2.089			
11	10	18/5/2012 20:27	2.089			
12	11	18/5/2012 20:27	2.089			

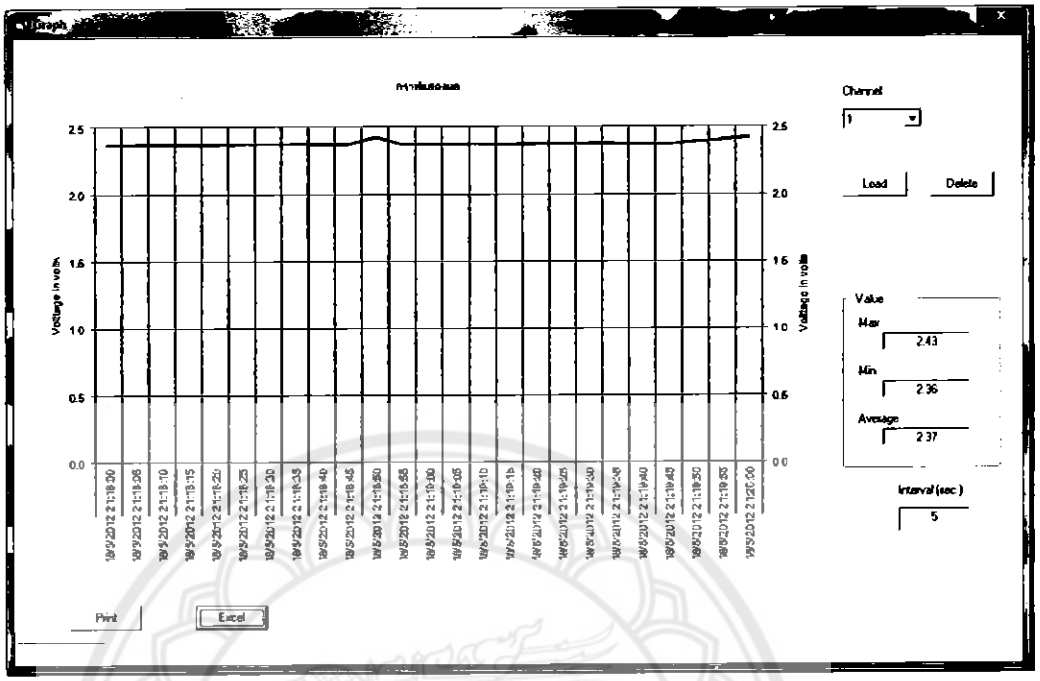
รูปที่ 4.10 ตัวอย่างไฟล์ Excel

2.6.5 กดปุ่ม Delete จะทำการลบค่าที่ตั้งไว้ใช้ในการบันทึกข้อมูลและข้อมูลที่บันทึกใน SD-Card

4.2 ผลการทดลองการใช้งานอุปกรณ์บันทึกข้อมูล

จากการทดลองที่ 1 เก็บข้อมูลแรงดันของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรดในการเปลี่ยนค่าแรงดันไฟฟ้าให้เป็นระยะทางจะเปลี่ยนโดยกำหนดเทียบจากกราฟในคู่มือการใช้งานของ GP2Y0A21K0F โมดูลตรวจจับระยะทางแบบอินฟราเรด

Channel 1 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 10 เซนติเมตร



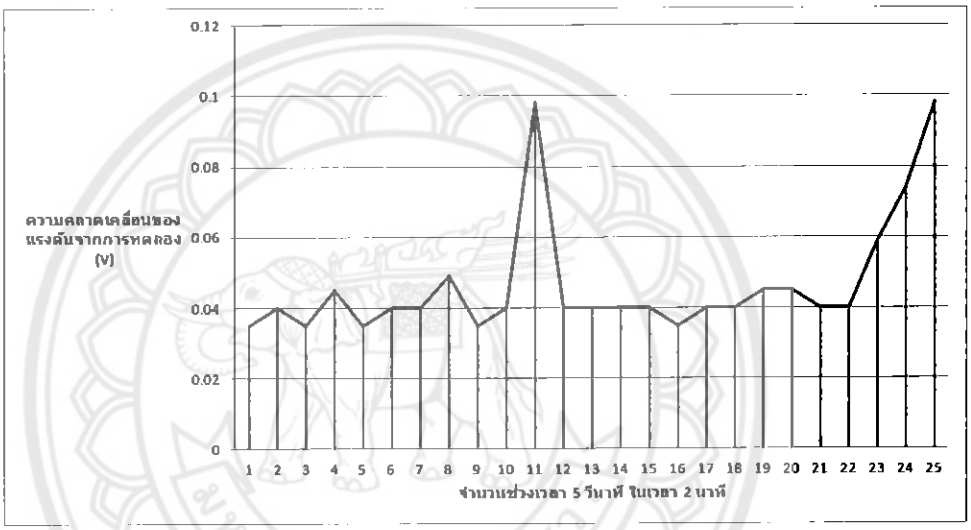
รูปที่ 4.11 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด

ตารางที่ 4.1 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 10 เซนติเมตรของ Channel1

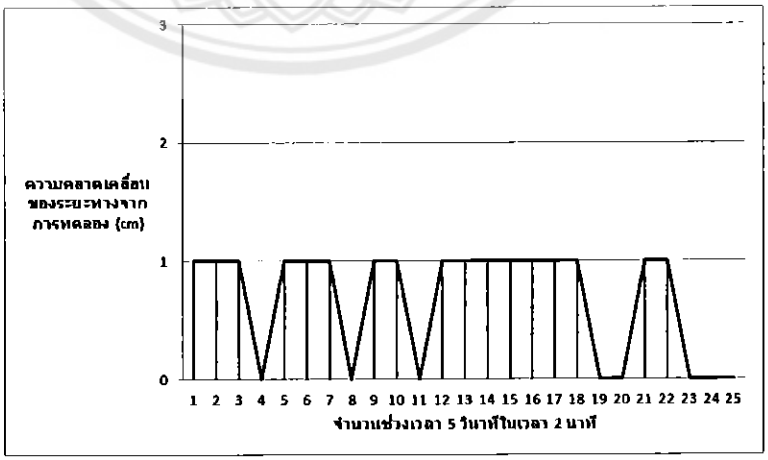
จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน(V)	ระยะทางที่ได้จากค่าแรงดันที่วัดได้(cm)	ระยะทางที่กำหนด (cm)	ความคลาดเคลื่อน (cm)
1	2.363	2.328	0.035	11	10	1
2	2.368	2.328	0.04	11	10	1
3	2.363	2.328	0.035	11	10	1
4	2.373	2.328	0.045	10	10	0
5	2.363	2.328	0.035	11	10	1
6	2.368	2.328	0.04	11	10	1
7	2.368	2.328	0.04	11	10	1
8	2.377	2.328	0.049	10	10	0
9	2.363	2.328	0.035	11	10	1
10	2.368	2.328	0.04	11	10	1
11	2.426	2.328	0.098	10	10	0
12	2.368	2.328	0.04	11	10	1
13	2.368	2.328	0.04	11	10	1
14	2.368	2.328	0.04	11	10	1
15	2.368	2.328	0.04	11	10	1
16	2.363	2.328	0.035	11	10	1
17	2.368	2.328	0.04	11	10	1
18	2.368	2.328	0.04	11	10	1
19	2.373	2.328	0.045	10	10	0
20	2.373	2.328	0.045	10	10	0
21	2.368	2.328	0.04	11	10	1

ตารางที่ 4.1 (ต่อ) ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 10 เซนติเมตรของ Channel1

22	2.368	2.328	0.04	11	10	1
23	2.387	2.328	0.059	10	10	0
24	2.402	2.328	0.074	10	10	0
25	2.426	2.328	0.098	10	10	0
ค่าเฉลี่ย	2.37472	2.328	0.04672	10.68	10	

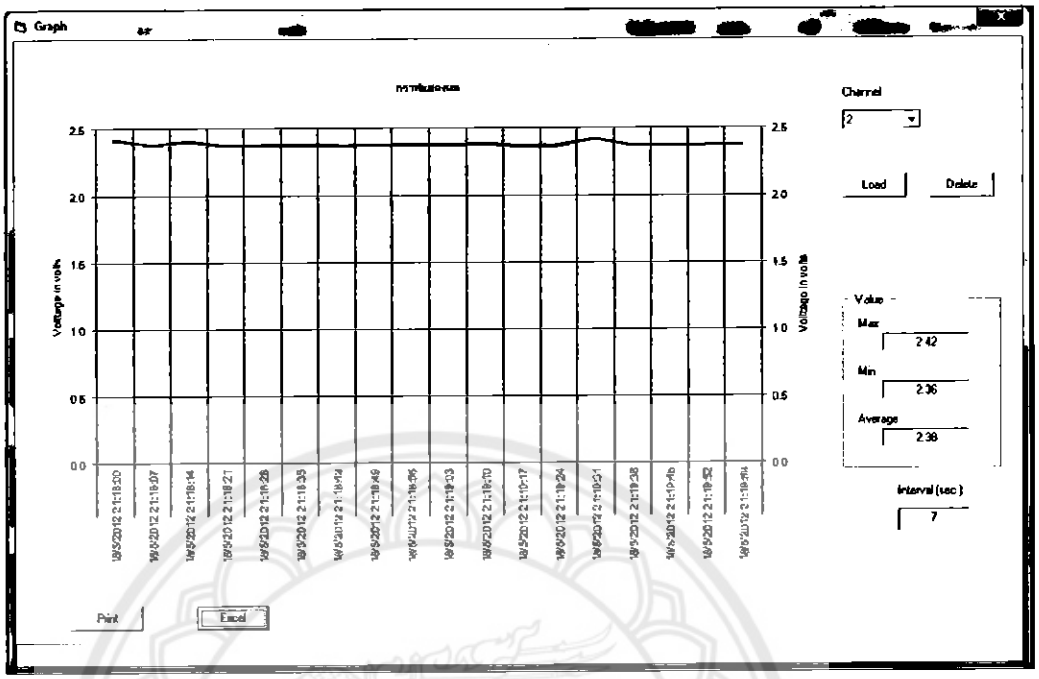


รูปที่ 4.12 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 1



รูปที่ 4.13 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 10 เซนติเมตร Channel 1

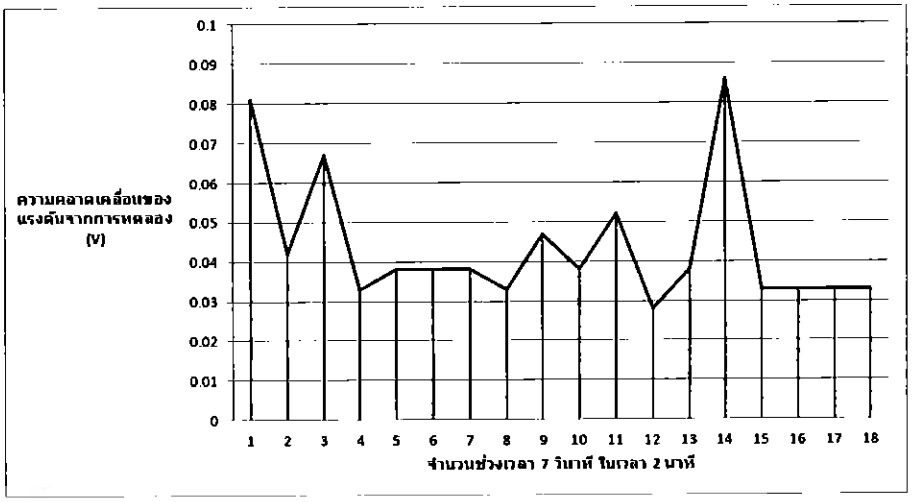
Channel 2 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 10 เซนติเมตร



รูปที่ 4.14 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด

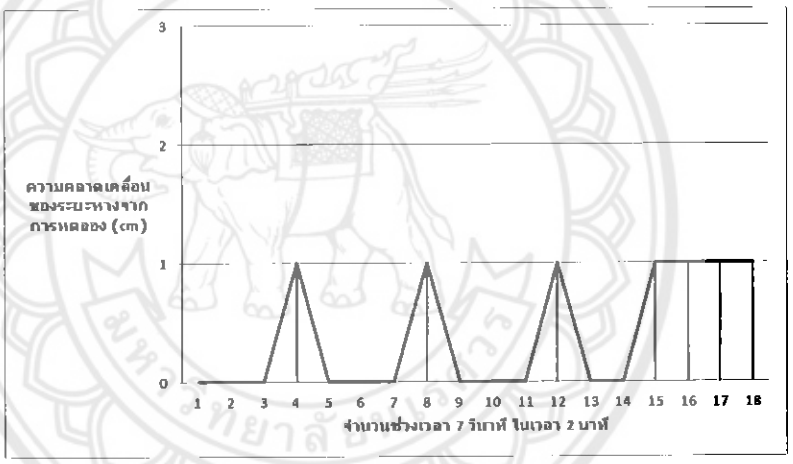
ตารางที่ 4.2 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 10 เซนติเมตรของ Channel 2

จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน(V)	ระยะทางที่ได้จากค่าแรงดันที่วัดได้(cm)	ระยะทางที่กำหนด (cm)	ความคลาดเคลื่อน(cm)
1	2.416	2.335	0.081	10	10	0
2	2.377	2.335	0.042	10	10	0
3	2.402	2.335	0.067	10	10	0
4	2.368	2.335	0.033	11	10	1
5	2.373	2.335	0.038	10	10	0
6	2.373	2.335	0.038	10	10	0
7	2.373	2.335	0.038	10	10	0
8	2.368	2.335	0.033	11	10	1
9	2.382	2.335	0.047	10	10	0
10	2.373	2.335	0.038	10	10	0
11	2.387	2.335	0.052	10	10	0
12	2.363	2.335	0.028	11	10	1
13	2.373	2.335	0.038	10	10	0
14	2.421	2.335	0.086	10	10	0
15	2.368	2.335	0.033	11	10	1
16	2.368	2.335	0.033	11	10	1
17	2.368	2.335	0.033	11	10	1
18	2.368	2.335	0.033	11	10	1
ค่าเฉลี่ย	2.378944	2.335	0.043944	10.38889	10	



รูปที่ 4.15 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 10 เซนติเมตร

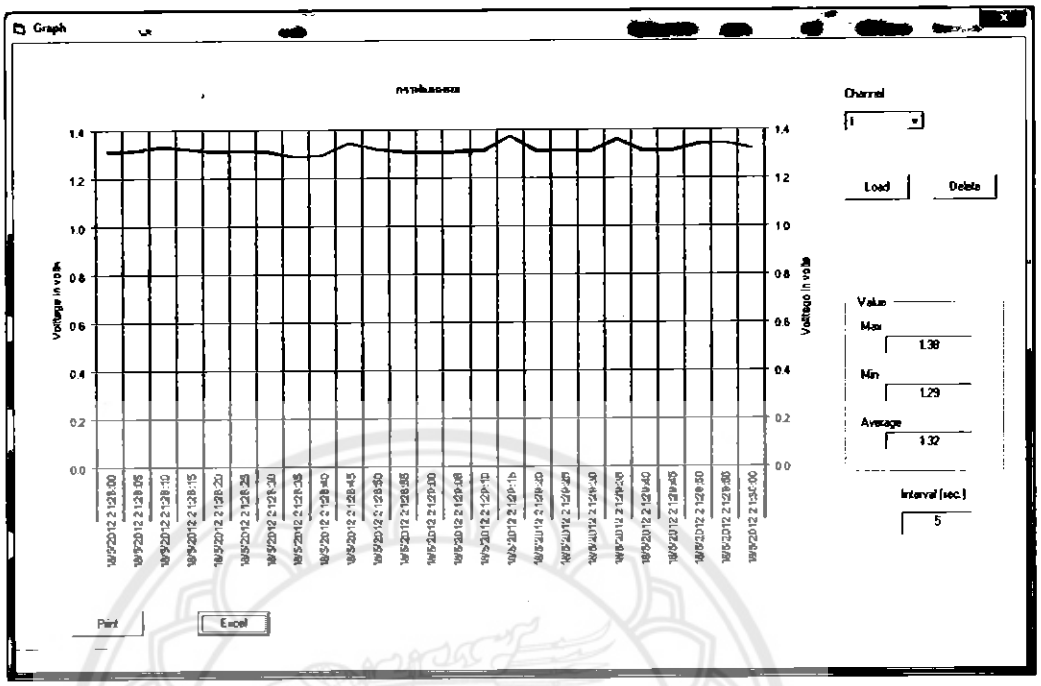
Channel 2



รูปที่ 4.16 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 10 เซนติเมตร

Channel 2

Channel 1 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 20 เซนติเมตร



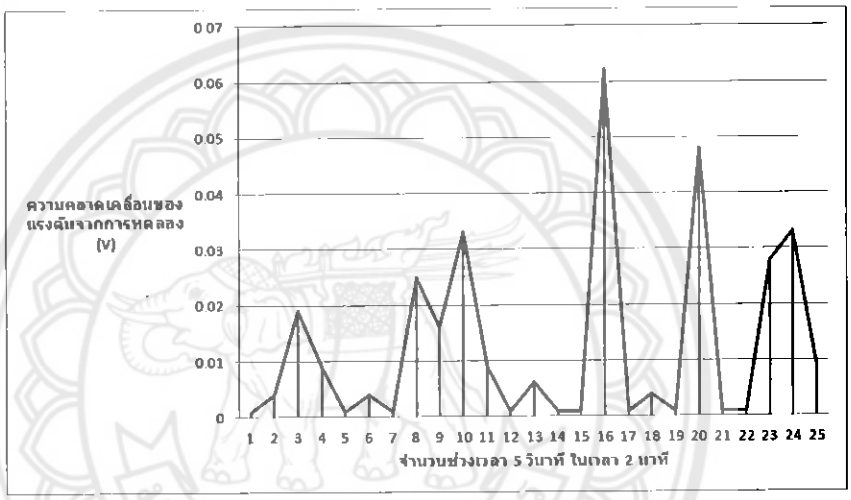
รูปที่ 4.17 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด

ตารางที่ 4.3 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 20 เซนติเมตรของ Channel 1

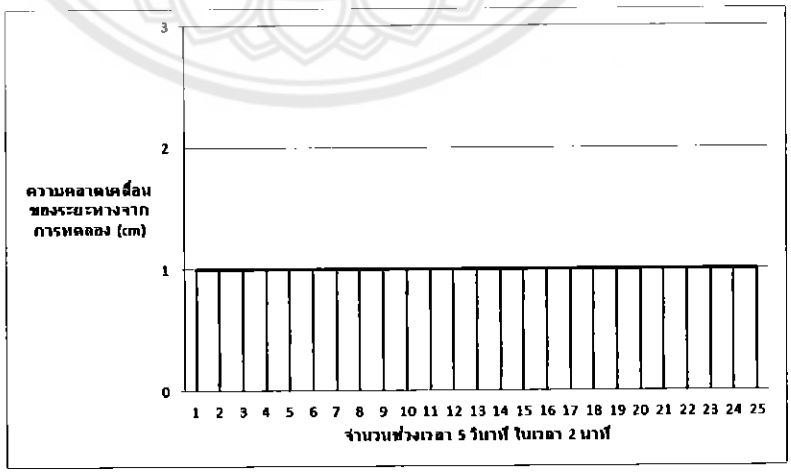
จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้(V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน(V)	ระยะทางที่ได้จากค่าแรงดันที่วัดได้(cm)	ระยะทางที่กำหนด (cm)	ความคลาดเคลื่อน(cm)
1	1.313	1.314	0.001	21	20	1
2	1.318	1.314	0.004	21	20	1
3	1.333	1.314	0.019	21	20	1
4	1.323	1.314	0.009	21	20	1
5	1.313	1.314	0.001	21	20	1
6	1.318	1.314	0.004	21	20	1
7	1.313	1.314	0.001	21	20	1
8	1.289	1.314	0.025	22	20	1
9	1.298	1.314	0.016	22	20	1
10	1.347	1.314	0.033	21	20	1
11	1.323	1.314	0.009	21	20	1
12	1.313	1.314	0.001	21	20	1
13	1.308	1.314	0.006	21	20	1
14	1.313	1.314	0.001	21	20	1
15	1.313	1.314	0.001	21	20	1
16	1.376	1.314	0.062	21	20	1
17	1.313	1.314	0.001	21	20	1
18	1.318	1.314	0.004	21	20	1
19	1.313	1.314	0.001	21	20	1
20	1.362	1.314	0.048	21	20	1
21	1.313	1.314	0.001	21	20	1

ตารางที่ 4.3 (ต่อ) ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที ระยะทาง 20 เซนติเมตรของ Channel 1

22	1.313	1.314	0.001	21	20	1
23	1.342	1.314	0.028	21	20	1
24	1.347	1.314	0.033	21	20	1
25	1.323	1.314	0.009	21	20	1
ค่าเฉลี่ย	1.3222	1.314	0.01276	21.08	20	

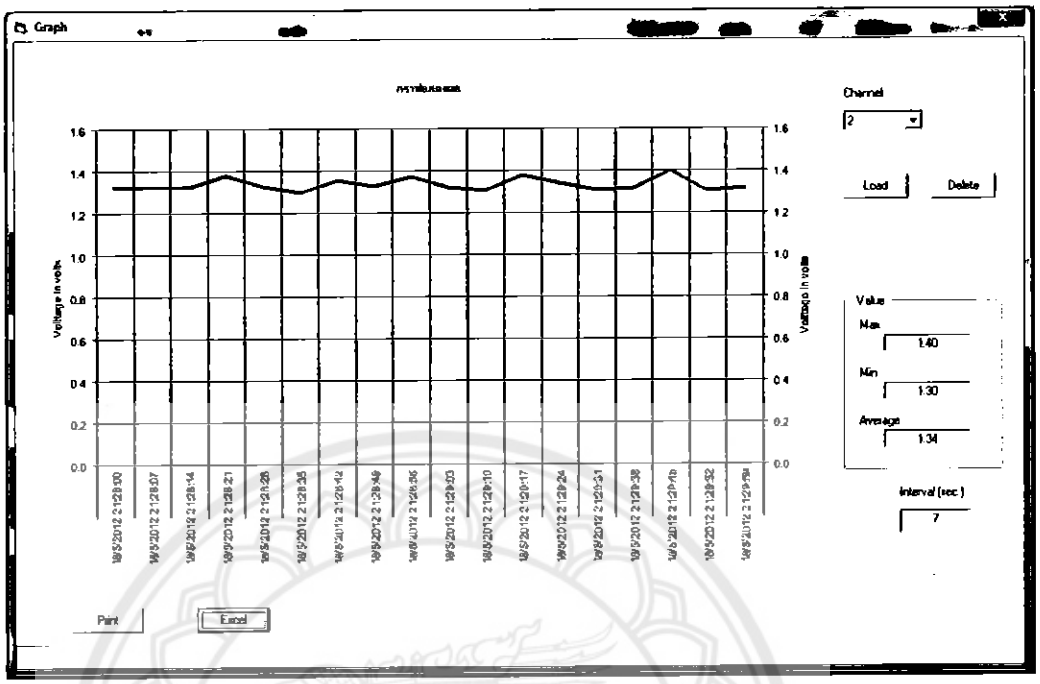


รูปที่ 4.18 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 20 เซนติเมตร Channel 1



รูปที่ 4.19 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 20 เซนติเมตร Channel1

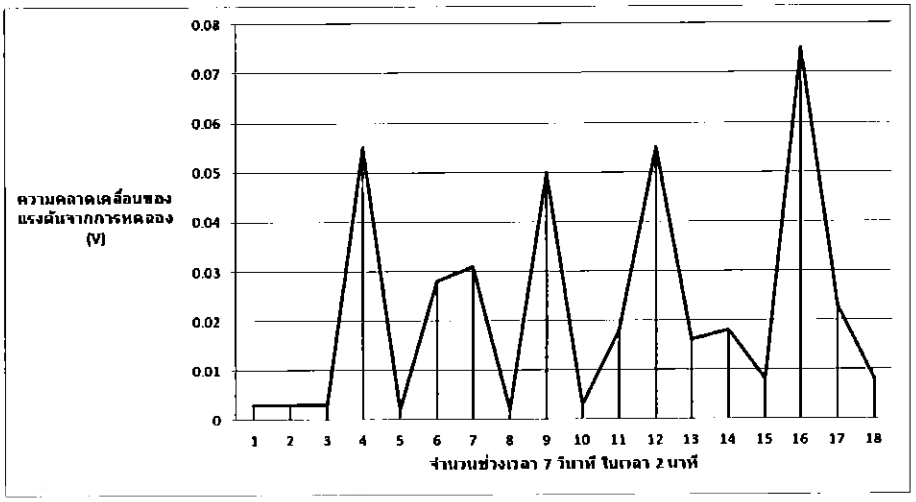
Channel2 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 20 เซนติเมตร



รูปที่ 4.20 กราฟแสดงข้อมูลแรงดันของโมดูลตรวจจับข้อมูลแบบอินฟราเรด

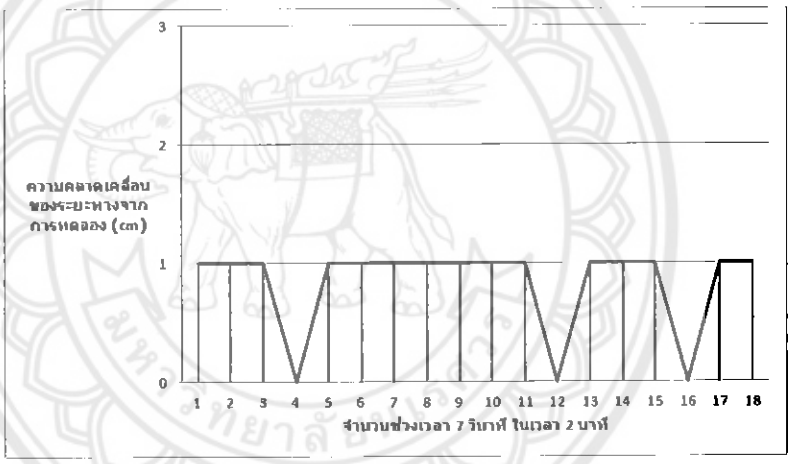
ตารางที่ 4.4 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที ระยะทาง 20 เซนติเมตรของ Channel 2

จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้(V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน (V)	ระยะทางที่ได้จากค่าแรงดันที่วัดได้(cm)	ระยะทางที่กำหนด (cm)	ความคลาดเคลื่อน (cm)
1	1.323	1.326	0.003	21	20	1
2	1.323	1.326	0.003	21	20	1
3	1.323	1.326	0.003	21	20	1
4	1.381	1.326	0.055	20	20	0
5	1.328	1.326	0.002	21	20	1
6	1.298	1.326	0.028	22	20	1
7	1.357	1.326	0.031	21	20	1
8	1.328	1.326	0.002	21	20	1
9	1.376	1.326	0.05	21	20	1
10	1.323	1.326	0.003	21	20	1
11	1.308	1.326	0.018	21	20	1
12	1.381	1.326	0.055	20	20	0
13	1.342	1.326	0.016	21	20	1
14	1.308	1.326	0.018	21	20	1
15	1.318	1.326	0.008	21	20	1
16	1.401	1.326	0.075	20	20	0
17	1.303	1.326	0.023	21	20	1
18	1.318	1.326	0.008	21	20	1
ค่าเฉลี่ย	1.3355	1.326	0.022278	20.88889	20	



รูปที่ 4.21 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง ระยะทาง 20 เซนติเมตร

Channel 2

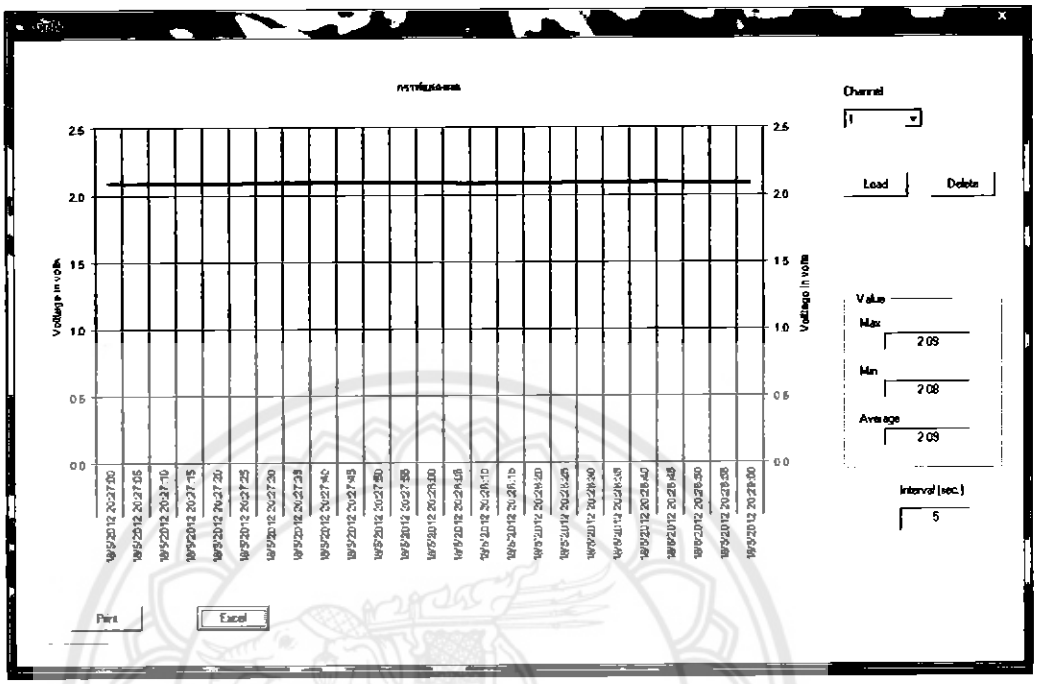


รูปที่ 4.22 กราฟแสดงความคลาดเคลื่อนของระยะทางจากการทดลอง ระยะทาง 20 เซนติเมตร

Channel 2

จากการทดลองที่ 2 บันทึกข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

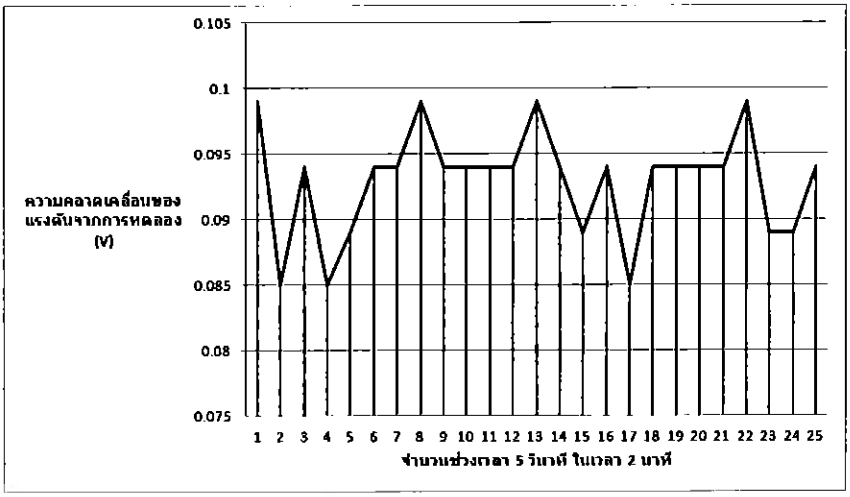
Channel 1 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดันไฟฟ้า 2 V



รูปที่ 4.23 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

ตารางที่ 4.5 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดันไฟฟ้า 2 V ของ Channel1

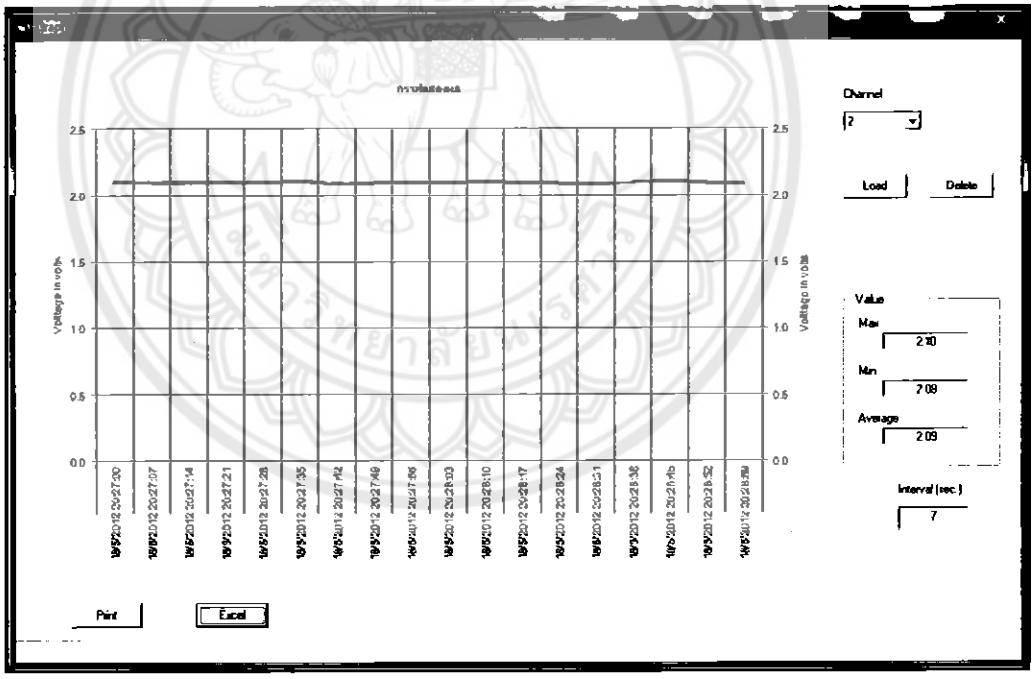
จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป(V)	ความคลาดเคลื่อน (V)
1	2.094	1.995	0.099
2	2.08	1.995	0.085
3	2.089	1.995	0.094
4	2.08	1.995	0.085
5	2.084	1.995	0.089
6	2.089	1.995	0.094
7	2.089	1.995	0.094
8	2.094	1.995	0.099
9	2.089	1.995	0.094
10	2.089	1.995	0.094
11	2.089	1.995	0.094
12	2.089	1.995	0.094
13	2.094	1.995	0.099
14	2.089	1.995	0.094
15	2.084	1.995	0.089
16	2.089	1.995	0.094
17	2.08	1.995	0.085
18	2.089	1.995	0.094
19	2.089	1.995	0.094
20	2.089	1.995	0.094
21	2.089	1.995	0.094
22	2.094	1.995	0.099
23	2.084	1.995	0.089
24	2.084	1.995	0.089
25	2.089	1.995	0.094
ค่าเฉลี่ย	2.08792	1.995	0.09292



รูปที่ 4.24 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง **บันทึกแรงดัน** 2 V

Channel1

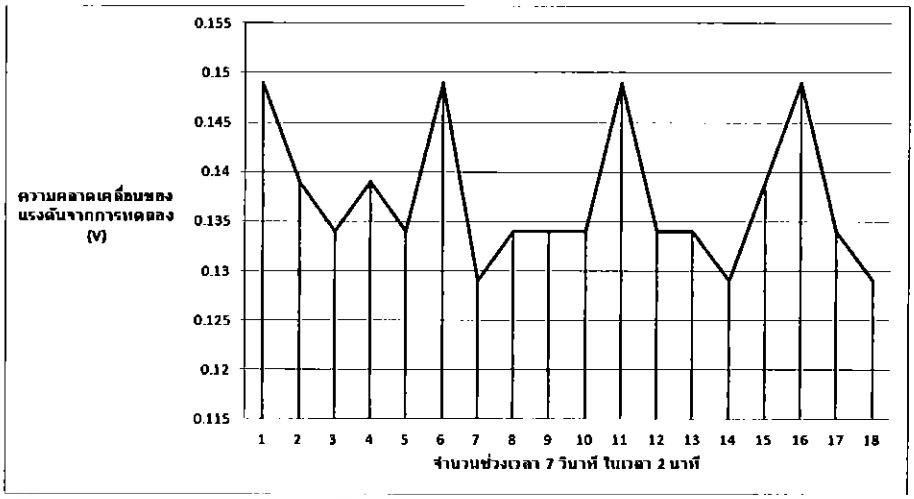
Channel 2 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที แรงดัน**ไฟฟ้า** 2 V



รูปที่ 4.25 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

ตารางที่ 4.6 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7วินาที แรงดันไฟฟ้า 2 V ของ Channel2

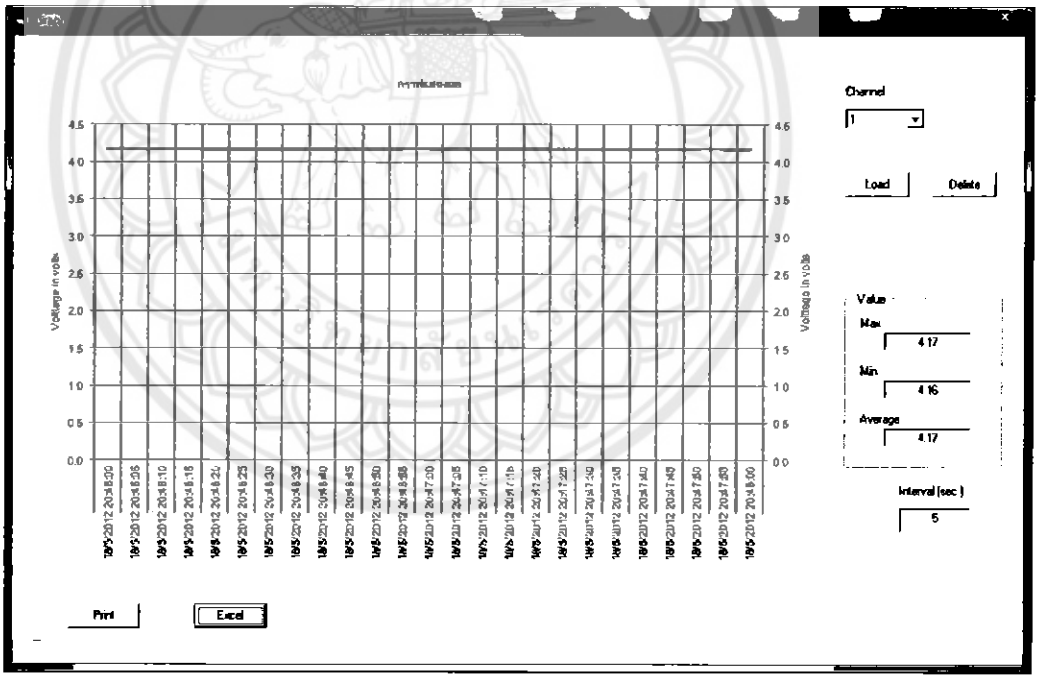
จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน(V)
1	2.104	1.955	0.149
2	2.094	1.955	0.139
3	2.089	1.955	0.134
4	2.094	1.955	0.139
5	2.089	1.955	0.134
6	2.104	1.955	0.149
7	2.084	1.955	0.129
8	2.089	1.955	0.134
9	2.089	1.955	0.134
10	2.089	1.955	0.134
11	2.104	1.955	0.149
12	2.089	1.955	0.134
13	2.089	1.955	0.134
14	2.084	1.955	0.129
15	2.094	1.955	0.139
16	2.104	1.955	0.149
17	2.089	1.955	0.134
18	2.084	1.955	0.129
ค่าเฉลี่ย	2.092333	1.955	0.137333



รูปที่ 4.26 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง **บันทึกแรงดัน 2 V**

Channel2

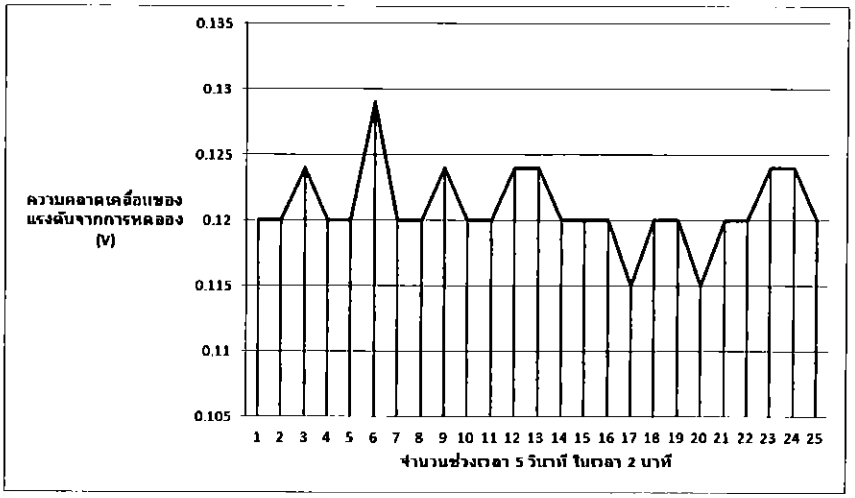
Channel 1 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดัน**ไฟฟ้า** 4 V



รูปที่ 4.27 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

ตารางที่ 4.7 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 5 วินาที แรงดันไฟฟ้า 4 V ของ Channel 1

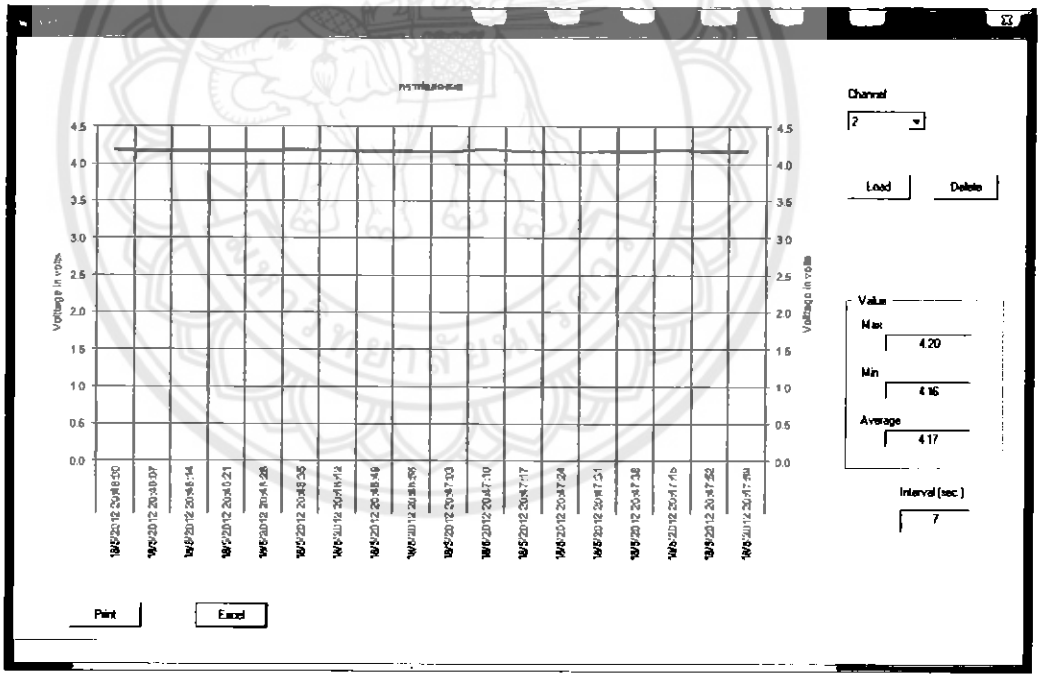
จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้(V)	แรงดันที่วัดได้จากออสซิโลสโคป(V)	ความคลาดเคลื่อน(V)	จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป(V)	ความคลาดเคลื่อน (V)
1	4.165	4.045	0.12	14	4.165	4.045	0.12
2	4.165	4.045	0.12	15	4.165	4.045	0.12
3	4.169	4.045	0.124	16	4.165	4.045	0.12
4	4.165	4.045	0.12	17	4.16	4.045	0.115
5	4.165	4.045	0.12	18	4.165	4.045	0.12
6	4.174	4.045	0.129	19	4.165	4.045	0.12
7	4.165	4.045	0.12	20	4.16	4.045	0.115
8	4.165	4.045	0.12	21	4.165	4.045	0.12
9	4.169	4.045	0.124	22	4.165	4.045	0.12
10	4.165	4.045	0.12	23	4.169	4.045	0.124
11	4.165	4.045	0.12	24	4.169	4.045	0.124
12	4.169	4.045	0.124	25	4.165	4.045	0.12
13	4.169	4.045	0.124	ค่าเฉลี่ย	4.16592	4.045	0.12092



รูปที่ 4.28 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง **บันทึกแรงดัน 4 V**

Channel1

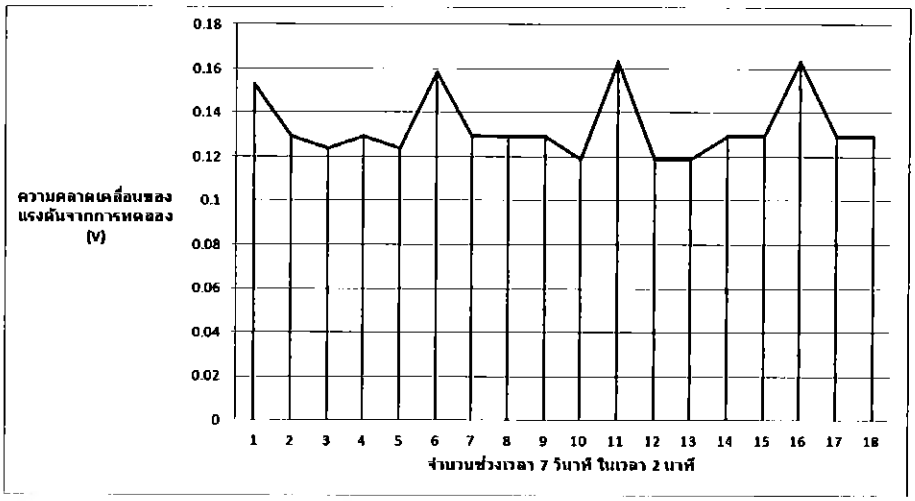
Channel 2 ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7 วินาที แรงดันไฟฟ้า 4 V



รูปที่ 4.29 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

ตารางที่ 4.8 ตารางค่าของข้อมูลที่ได้จากการทดลอง ใช้เวลาบันทึกข้อมูล 2 นาที ช่วงเวลาในการบันทึก 7วินาที แรงดันไฟฟ้า 4 V ของ Channel2

จำนวนช่วงเวลาภายใน 2 นาที	แรงดันที่บันทึกได้ (V)	แรงดันที่วัดได้จากออสซิโลสโคป (V)	ความคลาดเคลื่อน(V)
1	4.189	4.036	0.153
2	4.165	4.036	0.129
3	4.16	4.036	0.124
4	4.165	4.036	0.129
5	4.16	4.036	0.124
6	4.194	4.036	0.158
7	4.165	4.036	0.129
8	4.165	4.036	0.129
9	4.165	4.036	0.129
10	4.155	4.036	0.119
11	4.199	4.036	0.163
12	4.155	4.036	0.119
13	4.155	4.036	0.119
14	4.165	4.036	0.129
15	4.165	4.036	0.129
16	4.199	4.036	0.163
17	4.165	4.036	0.129
18	4.165	4.036	0.129
ค่าเฉลี่ย	4.1695	4.036	0.1335

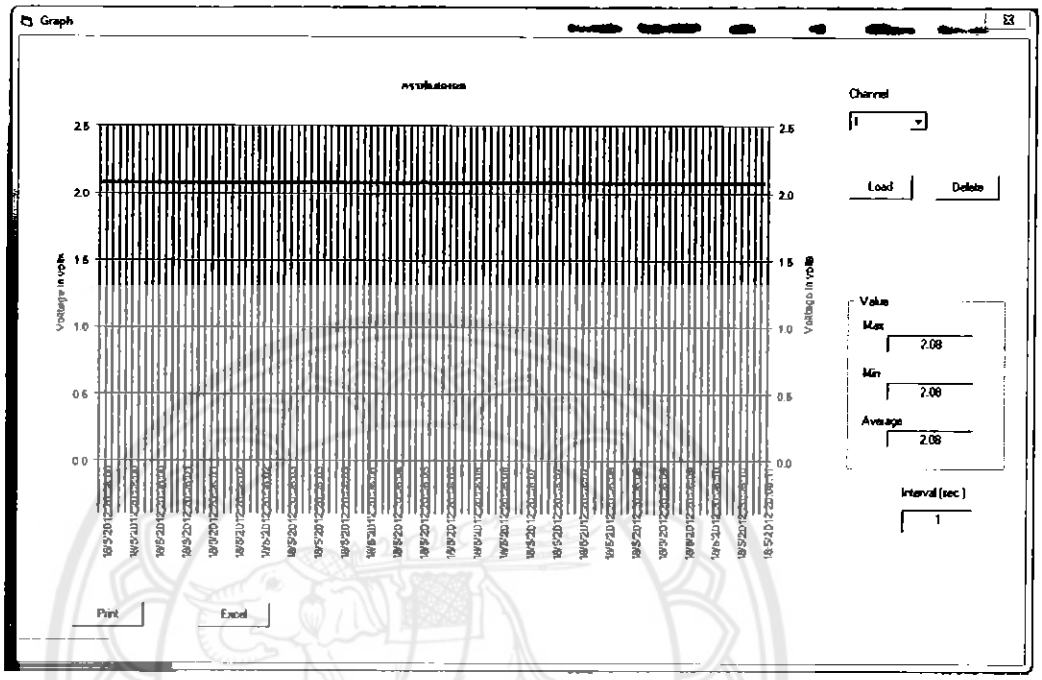


รูปที่ 4.30 กราฟแสดงความคลาดเคลื่อนของแรงดันไฟฟ้าจากการทดลอง **บันทึกแรงดัน 4 V**

Channel 2

จากการทดลองที่ 3 บันทึกข้อมูลแบบต่อเนื่องโดยจะบันทึกแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้า

Channel 1 ใช้เวลาบันทึกข้อมูล 10 วินาที



รูปที่ 4.31 กราฟแสดงข้อมูลแรงดันไฟฟ้าจากแหล่งจ่ายแรงดันไฟฟ้าที่แสดงข้อมูลการบันทึกแบบต่อเนื่อง

ตารางที่ 4.9 ตารางแสดงจำนวนครั้งของข้อมูลและค่าของแรงดันที่ได้จากการบันทึกแบบต่อเนื่อง ใน 1 วินาที

ครั้งที่	วัน เดือน ปี/เวลา	ค่าของแรงดัน (V)
1	20:08:00	2.075
2	20:08:00	2.075
3	20:08:00	2.075
4	20:08:00	2.075
5	20:08:00	2.075
6	20:08:00	2.075
7	20:08:00	2.08
8	20:08:00	2.075
9	20:08:00	2.075
10	20:08:01	2.075
11	20:08:01	2.075
12	20:08:01	2.075
13	20:08:01	2.075
14	20:08:01	2.075
15	20:08:01	2.075
16	20:08:01	2.075
17	20:08:01	2.075
18	20:08:01	2.075
19	20:08:02	2.075
20	20:08:02	2.075
21	20:08:02	2.075
22	20:08:02	2.075
23	20:08:02	2.075
24	20:08:02	2.075

ครั้งที่	วัน เดือน ปี/เวลา	ค่าของแรงดัน (V)
52	20:08:05	2.08
53	20:08:05	2.075
54	20:08:05	2.075
55	20:08:06	2.075
56	20:08:06	2.075
57	20:08:06	2.075
58	20:08:06	2.075
59	20:08:06	2.08
60	20:08:06	2.075
61	20:08:06	2.075
62	20:08:06	2.075
63	20:08:06	2.075
64	20:08:07	2.075
65	20:08:07	2.075
66	20:08:07	2.075
67	20:08:07	2.075
68	20:08:07	2.075
69	20:08:07	2.075
70	20:08:07	2.075
71	20:08:07	2.075
72	20:08:07	2.075
73	20:08:07	2.075
74	20:08:08	2.075
75	20:08:08	2.075

ตารางที่ 4.9 (ต่อ) ตารางแสดงจำนวนครั้งของข้อมูลและค่าของแรงดันที่ได้จากการบันทึกแบบต่อเนื่อง ใน 1 วินาที

25	20:08:02	2.075
26	20:08:02	2.075
27	20:08:02	2.075
28	20:08:02	2.075
29	20:08:03	2.075
30	20:08:03	2.075
31	20:08:03	2.075
32	20:08:03	2.075
33	20:08:03	2.08
34	20:08:03	2.075
35	20:08:03	2.075
36	20:08:03	2.075
37	20:08:03	2.075
38	20:08:04	2.075
39	20:08:04	2.075
40	20:08:04	2.075
41	20:08:04	2.075
42	20:08:04	2.075
43	20:08:04	2.075
44	20:08:04	2.075
45	20:08:04	2.075
46	20:08:04	2.075
47	20:08:05	2.075
48	20:08:05	2.08
49	20:08:05	2.08

76	20:08:08	2.075
77	20:08:08	2.075
78	20:08:08	2.075
79	20:08:08	2.075
80	20:08:08	2.075
81	20:08:08	2.075
82	20:08:08	2.075
83	20:08:09	2.075
84	20:08:09	2.075
85	20:08:09	2.075
86	20:08:09	2.075
87	20:08:09	2.075
88	20:08:09	2.075
89	20:08:09	2.075
90	20:08:09	2.075
91	20:08:09	2.075
92	20:08:10	2.075
93	20:08:10	2.075
94	20:08:10	2.075
95	20:08:10	2.075
96	20:08:10	2.075
97	20:08:10	2.075
98	20:08:10	2.075
99	20:08:10	2.075
100	20:08:10	2.075

ตารางที่ 4.9 (ต่อ) ตารางแสดงจำนวนครั้งของข้อมูลและค่าของแรงดันที่ได้จากการบันทึกแบบต่อเนื่องใน 1 วินาที

50	20:08:05	2.075
51	20:08:05	2.075

101	20:08:11	2.075
ค่าเฉลี่ย	2.075297	

4.3 สรุปผลการทดลอง

จากการทดลองในส่วนของซอฟแวร์ที่ใช้ในไมโครคอนโทรลเลอร์โดยใช้โปรแกรมในการแปลงข้อมูลจากอนาล็อกเป็นดิจิตอลนั้น จะส่งข้อมูลผ่านไฮเปอร์เทอร์มินอลโดยส่งข้อมูลผ่านช่องทางการสื่อสารแบบอนุกรม สำหรับค่าแรงดันที่ได้จากการทดลองนั้นจะเห็นว่าค่ามีค่าคลาดเคลื่อนเกิดขึ้นในทั้ง 2 Channels ระหว่าง 0.012-0.137 v

ส่วนการบันทึกแบบต่อเนื่องนั้นจะบันทึกได้ทั้งหมด 9 ครั้งต่อ 1 วินาทีโดยบันทึกข้อมูล 1 Channel

ในการกำหนดเวลานั้นสามารถเริ่มได้ตั้งแต่ 1 วินาที จนถึง 24 ชั่วโมงตามที่เราต้องการ ส่วนค่าแรงดันที่เราทำการป้อนให้กับบอร์ดนั้นจะใช้ค่าเริ่มต้นที่ 0 จนถึง 5 มีความละเอียดอยู่ที่ 1 mV

จากการใช้งานอุปกรณ์บันทึกข้อมูลพบว่าอุปกรณ์บันทึกข้อมูลที่พัฒนาขึ้นสามารถรับค่าแรงดันไฟฟ้าของ GP2Y0A21 โมดูลตรวจจับระยะทางได้ และเมื่อนำค่าแรงดันไฟฟ้าที่บันทึกได้มาทำการเปรียบเทียบกับคู่มือของเซ็นเซอร์เพื่อแปลงเป็นระยะทางแล้วนำมาเปรียบเทียบกับระยะทางที่กำหนดจะมีความคลาดเคลื่อนเล็กน้อย

บทที่ 5
สรุปผลการดำเนินโครงงาน

ในบทนี้ได้กล่าวถึงสรุปของผลการดำเนินงานที่ได้ทำมาตลอดปีการศึกษาว่าได้เจอปัญหาหรือข้อผิดพลาดใดบ้างที่เกิดขึ้นในระหว่างการดำเนินโครงงาน เพื่อเสนอแนะแนวทางในการแก้ไขและพัฒนาให้อุปกรณ์บันทึกข้อมูลมีประสิทธิภาพที่ดีขึ้น จากโครงงานทำให้ได้ข้อสรุปผลการดำเนินงานดังนี้

1. ข้อมูลที่จัดเก็บไว้ใน SD-Card นั้นสามารถนำมาแสดงและประมวลผลได้โดยผลที่ได้นั้นสามารถนำไปประยุกต์ใช้กับงานอื่นเพื่อให้เกิดประโยชน์ได้
2. เกิดความรู้ความเข้าใจในการใช้งานอุปกรณ์ด้านอิเล็กทรอนิกส์ชนิดต่างๆ
3. เกิดความรู้ความเข้าใจในโปรแกรมที่นำมาใช้ในการทำโครงงาน คือ โปรแกรม AVR Studio และ โปรแกรม Visual Basic 6.0

5.1 ปัญหาและแนวทางการแก้ปัญหา

อุปกรณ์ยังมีความล่าช้าในการส่งข้อมูลที่บันทึกจำนวนมากไปยังคอมพิวเตอร์ เนื่องจากในการส่งสามารถส่งได้ Packet ละ 512 Byte ซึ่งการแก้ไขปัญหานี้ ต้องใช้ระยะเวลาที่ค่อนข้างนานพอสมควร

5.2 ข้อเสนอแนะในการพัฒนา

อุปกรณ์บันทึกข้อมูลนี้สามารถใช้ได้กับเซนเซอร์ที่มีแรงดันไฟฟ้าสูงสุดได้ไม่เกิน 5 โวลต์ ดังนั้นในการพัฒนาควรเพิ่มวงจรแปลงแรงดันที่รับเข้ามามากกว่า 5 โวลต์ให้เป็น 5 โวลต์ เพื่อให้อุปกรณ์บันทึกข้อมูลนี้ สามารถใช้ได้กับเซนเซอร์ที่มีแรงดันมากกว่า 5 โวลต์ได้ และพัฒนาโดยการเพิ่มแหล่งจ่ายจากภายนอกให้กับเซ็นเซอร์

เอกสารอ้างอิง

[1] บริษัท เมเชอร์ วัน เทคโนโลยีจำกัด (2555) ศูนย์รวมเครื่องมือทางอุตสาหกรรม จากเว็บไซต์: http://www.measure1.co.th/?lang=th

[2] บริษัท อีทีที จำกัด (2553) Product AVR microcontroller จากเว็บไซต์: http://www.ett.co.th/prod2010/ET-AVR/man-ET-BASE-MEGA1280-2560.pdf

[3] Wikipedia (2554) การเชื่อมต่อแบบอนุกรมอุปกรณ์คอมพิวเตอร์หรือ SPI จากเว็บไซต์: http://en.wikipedia.org.th.mk.gd/wiki/Serial_Peripheral_Interface_Bus

[4] มนตรี เสาวดี และ อัคภณ เกตุประดิษฐ์ (2552) ปริญญานิพนธ์ "มาตรวัดไฟฟ้า 1 เฟสระบบดิจิตอลแบบไร้สาย" (หน้า 19) จากเว็บไซต์: http://library.kmutnb.ac.th/projects/B15412301.pdf

[5] สังวาล บกสุวรรณ และ สุรวัฒ บกสุวรรณ (2554) Work Experiences "SD Card Interface" จากเว็บไซต์: http://mcumaster.org/Projects/SDCard.html

[6] Analog to Digital conversion (ADC) และ Successive Approximation ADC (SAC) จากเว็บไซต์: http://eestud.kku.ac.th/~u4167536/adc.html

[7] บริษัท อีทีที จำกัด (2553) Product Interface Board จากเว็บไซต์: http://www.ett.co.th/product/InterfaceBoard/P-ET-A-00238.html

[8] บริษัท อีทีที จำกัด (2553) Product Interface Board จากเว็บไซต์: http://www.ett.co.th/product/InterfaceBoard/P-ET-A-00230/MAN_MINI_I_15_Board.pdf

[9] บริษัท อีทีที จำกัด (2553) Product AVR microcontroller จากเว็บไซต์: http://www.ett.co.th/prod2010/ET-AVR/man%20ET-AVRISP-mkII.pdf

[10] ครูวิทูลย์ งามขำ (2553) โรงเรียนพิบูลประชาสรรค์ วิชาการขียนโปรแกรม "Visual Basic 6.0" จากเว็บไซต์: http://www.pbps.ac.th/e_learning/vb6/contact.html

[11] แหล่งความรู้ สาระ เกี่ยวกับเทคโนโลยีและหุ่นยนต์ (2554) จากเว็บไซต์: http://ezybot.blogspot.com/2011/04/atmega-1-avr-studio-4.html

[12] ผศ.ดร.ปิยะ โควินท์ทวีวัฒน์ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (2554) บทที่ 5 "บูรณภาพของข้อมูล" จากเว็บไซต์: http://home.npru.ac.th/piya/RFID/file/Piya_Ch5.pdf

[13] Protocols จากเว็บไซต์: http://cptd.chandra.ac.th/selfstud/datacom/CAI/part3-8.htm

ภาคผนวก ก

GP2Y0A21YK0F

GP2Y0A21YK0F

Distance Measuring Sensor Unit
Measuring distance: 10 to 80 cm
Analog output type

■Description

GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit.
The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method.
This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Features

1. Distance measuring range : 10 to 80 cm
2. Analog output type
3. Package size : 29.5 × 13 × 13.5 mm
4. Consumption current : Typ. 30 mA
5. Supply voltage : 4.5 to 5.5 V

■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

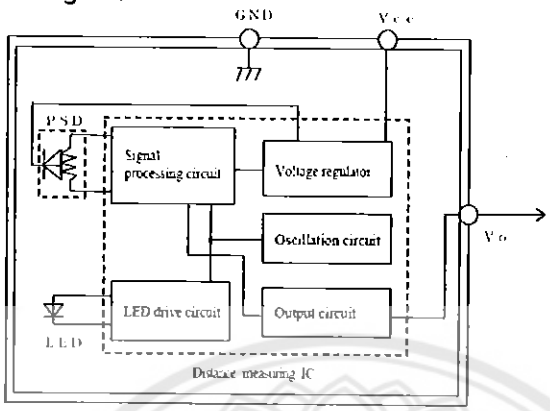
■Applications

1. Touch-less switch
   (Sanitary equipment, Control of illumination, etc.)
2. Robot cleaner
3. Sensor for energy saving
   (ATM, Copier, Vending machine)
4. Amusement equipment
   (Robot, Arcade game machine)

Notice The content of data sheet is subject to change without prior notice.
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

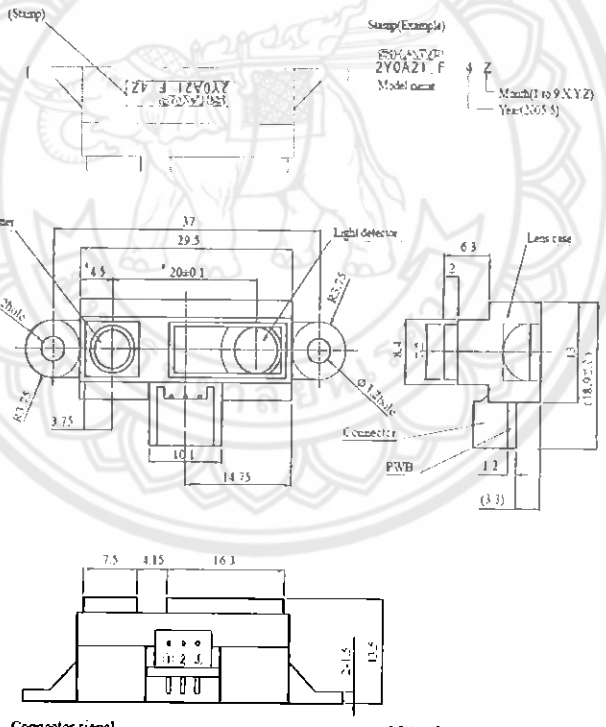
Sheet No. E4-A05201EN
Date Dec. 01. 2006
© SHARP Corporation

■Block diagram



■Outline Dimensions

(Unit : mm)



Connector signal

	Signal name
①	Vo
②	GND
③	Vcc

Connector :
J.S.T.TRADING COMPANY,LTD.
S3B-PH

Materials
Lens :Acrylic acid resin
(Visible light cut-off resin)
Case :Carbonic ABS
(Conductive resin)
PWB :Paper phenol

Note 1. The dimensions marked * are described the dimensions of lens center position.
Note 2. Unspecified tolerances shall be ± 0.3 mm.
Note 3. The dimensions in parenthesis are shown for reference.

Product mass : Approx. 3.6g

■Absolute Maximum Ratings (Ta=25℃,Vcc=5V)

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	-0.3 to +7	V
Output terminal voltage	V_O	-0.3 to V_{CC}+0.3	V
Operating temperature	T_{opr}	-10 to +60	℃
Storage temperature	T_{stg}	-40 to +70	℃

■Electro-optical Characteristics (Ta=25℃,Vcc=5V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Average supply current	I_{CC}	L=80cm (Note 1)	—	30	40	mA
Distance measuring	ΔL	(Note 1)	10	—	80	cm
Output voltage	V_O	L=80cm (Note 1)	0.25	0.4	0.55	V
Output voltage differential	ΔV_O	Output voltage differece between L=10cm and L=80cm (Note 1)	1.65	1.9	2.15	V

* L : Distance to reflective object

Note 1 : Using reflective object : White paper (Made by Kodak Co., Ltd. gray cards R-27, white face, reflectance: 90%)

■Recommended operating conditions

Parameter	Symbol	Rating	Unit
Supply voltage	V_{CC}	4.5 to 5.5	V

Fig. 1 Timing chart

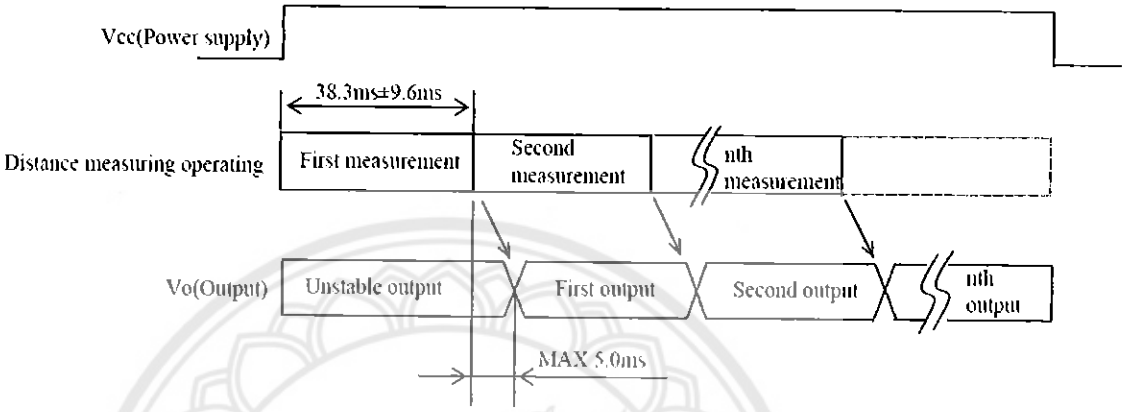
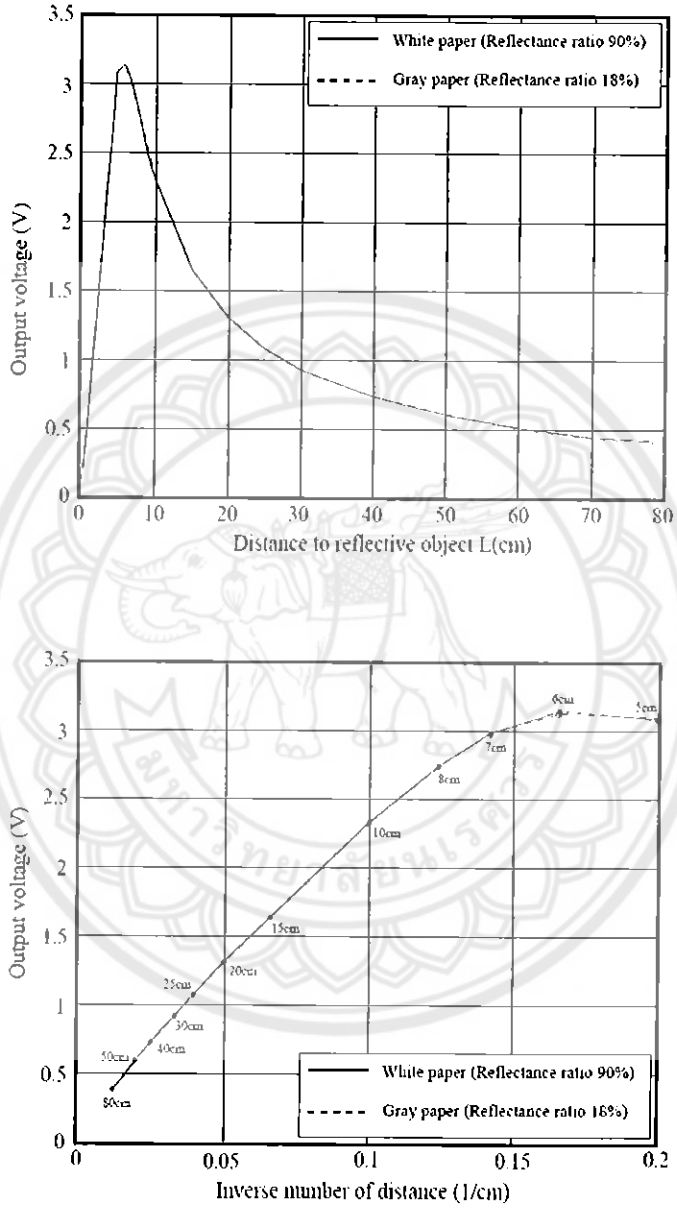


Fig. 2 Example of distance measuring characteristics(output)



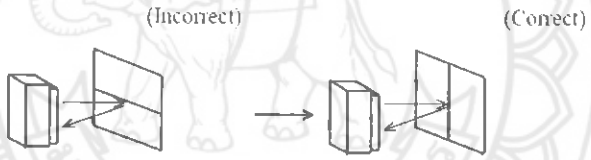
■Notes

●Advice for the optics

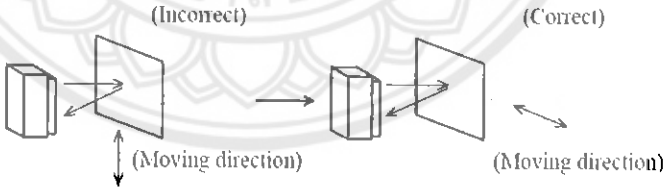
- The lens of this device needs to be kept clean. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on. Please confirm resistance to chemicals under the actual usage since this product has not been designed against washing.

●Advice for the characteristics

- In case that an optical filter is set in front of the emitter and detector portion, the optical filter which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda = 870 \pm 70$nm), shall be recommended to use. Both faces of the filter should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.
- In case that there is an object near to emitter side of the sensor between sensor and a detecting object, please use this device after confirming sufficiently that the characteristics of this sensor do not change by the object.
- When the detector is exposed to the direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector is not exposed to the direct light from such light source.
- Distance to a mirror reflector can not be sometimes measured exactly. In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line which material or color etc. are excessively different, in order to decrease deviation of measuring distance, it shall be recommended to set the sensor that the direction of boundary line and the line between emitter center and detector center are in parallel.



- In order to decrease deviation of measuring distance by moving direction of the reflective object, it shall be recommended to set the sensor that the moving direction of the object and the line between emitter center and detector center are vertical.



●Advice for the power supply

- In order to stabilize power supply line, we recommend to insert a by-pass capacitor of 10μF or more between Vcc and GND near this product.

●Notes on handling

- There are some possibilities that the internal components in the sensor may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the sensor package and also on the PCB while assembling this product.

● **Presence of ODC etc.**

This product shall not contain the following materials.
And they are not used in the production process for this product.
Regulation substances : CFCs, Halon, Carbon tetrachloride, 1.1.1-Trichloroethane (Methylchloroform)

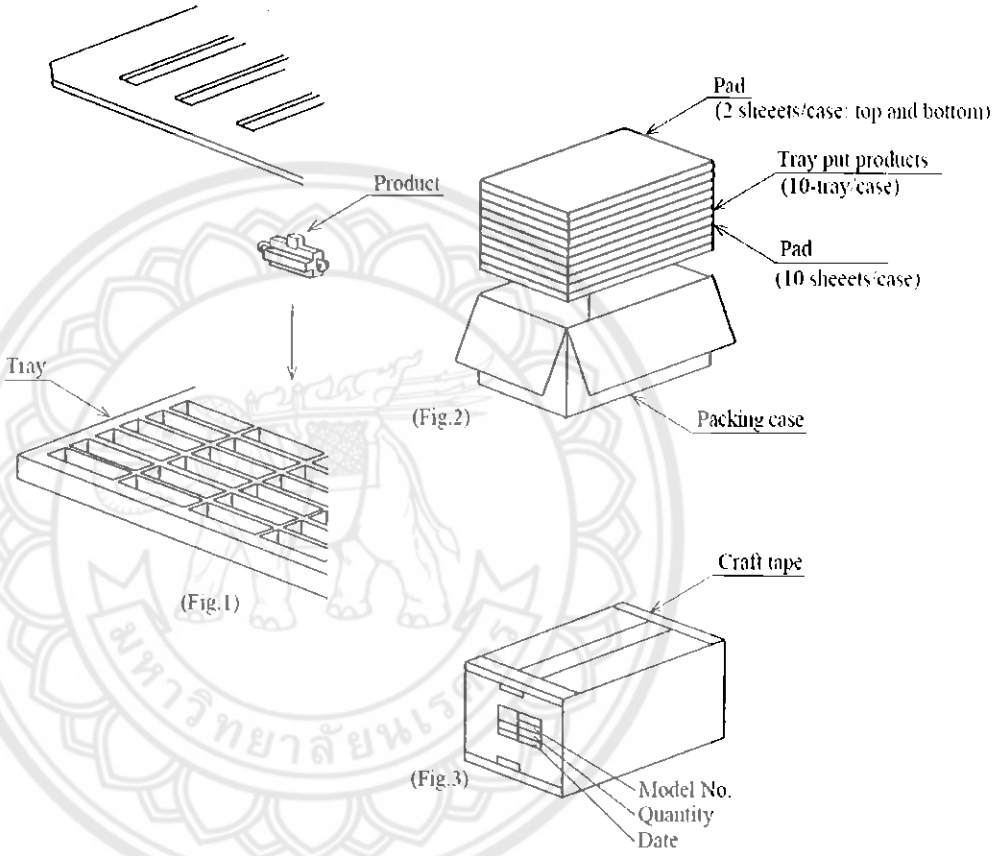
Specific brominated flame retardants such as the PBB and PBDE are not used in this product at all.

This product shall not contain the following materials banned in the RoHS Directive (2002/95/EC).

- Lead, Mercury, Cadmium, Hexavalent chromium, Polybrominated biphenyls (PBB), Polybrominated diphenyl ethers (PBDE).

■Package specification

Package composition



Packaging method

1.Put products of 100pcs. in tray. packing method is showed in the above fig.(Fig.1)
2.Put them(10-tray) in the packing box. Put pads on their top and bottom.
And put pads on each trays(Total 10 sheets) (Fig.2).
3.Seal the packing box with craft tape.
Print the model No.,quantity,inspection date (1000 pcs./a packing box)(Fig.3).

■Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:
- --- Personal computers
- --- Office automation equipment
- --- Telecommunication equipment [terminal]
- --- Test and measurement equipment
- --- Industrial control
- --- Audio visual equipment
- --- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection with equipment that requires higher reliability such as:
- --- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- --- Traffic signals
- --- Gas leakage sensor breakers
- --- Alarm equipment
- --- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:
- --- Space applications
- --- Telecommunication equipment [trunk lines]
- --- Nuclear power control equipment
- --- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

ภาคผนวก ข

คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-BASE MEGA 2560

ET-BASE MEGA1280/2560

ไมโครคอนโทรลเลอร์ตระกูล AVR ของ ATMEL เป็นไมโครคอนโทรลเลอร์อีกตระกูลหนึ่งซึ่งได้รับความนิยมอย่างแพร่หลายจากผู้ใช้งานทั่วไป ซึ่งทาง ATMEL เองก็ได้มีการปรับปรุง พัฒนาขีดความสามารถของ MCU เพื่อตอบสนองความต้องการใช้งานในลักษณะต่างๆ มีการผลิตชิพ MCU ออกมาจำหน่ายเป็นจำนวนมากมายหลายเบอร์ เพื่อให้ผู้ใช้งานสามารถเลือก MCU ไปประยุกต์ใช้งานให้เหมาะสมกับงานได้ง่ายและสะดวกมากยิ่งขึ้น

ATMEGA ก็เป็น MCU ตระกูล AVR ที่มีความโดดเด่นและมีขีดความสามารถสูงในระดับต้นๆของ MCU 8 บิต ซึ่งใน ATMEGA1280/2560 เองเป็น MCU ตระกูล AVR MEGA ที่มีระบบ Peripheral I/O ต่างๆรวบรวมไว้ภายใน MCU มากมายหลากหลาย สามารถโปรแกรมโหมดการทำงานของ Peripheral I/O ใน ลักษณะต่างๆได้หลากหลาย ทำให้ง่ายและสะดวกในการนำไปดัดแปลงใช้กับงานแบบต่างๆได้โดยง่าย ซึ่งการพัฒนาโปรแกรมของบอร์ดก็สามารถเลือกใช้รูปแบบในการพัฒนาโปรแกรมด้วยโปรแกรมภาษาต่างๆ ที่สนับสนุนการใช้งานกับ AVR MEGA ได้ทั่วไป ตามความเหมาะสม

โดยโครงสร้างของบอร์ดได้ออกแบบให้มีความสะดวกในการพัฒนาโปรแกรม และ ประยุกต์ใช้งานได้โดยสะดวก โดยมีพอร์ตสื่อสาร RS232 และ Micro-SD Card เป็นอุปกรณ์พื้นฐานภายในบอร์ต ส่วน GPIO ต่างๆจะออกแบบเป็น IDE Connector ไว้ให้เพื่อให้เกิดความสะดวกในการเชื่อมต่อออกไปใช้งาน โดยได้เพิ่มช่องทางในการพัฒนาโปรแกรมได้ทั้งการโปรแกรมผ่าน Bootloader หรือ ISP Programmer หรือ JTAG Interface สำหรับโปรแกรมและ Debug การทำงานได้อีกด้วย

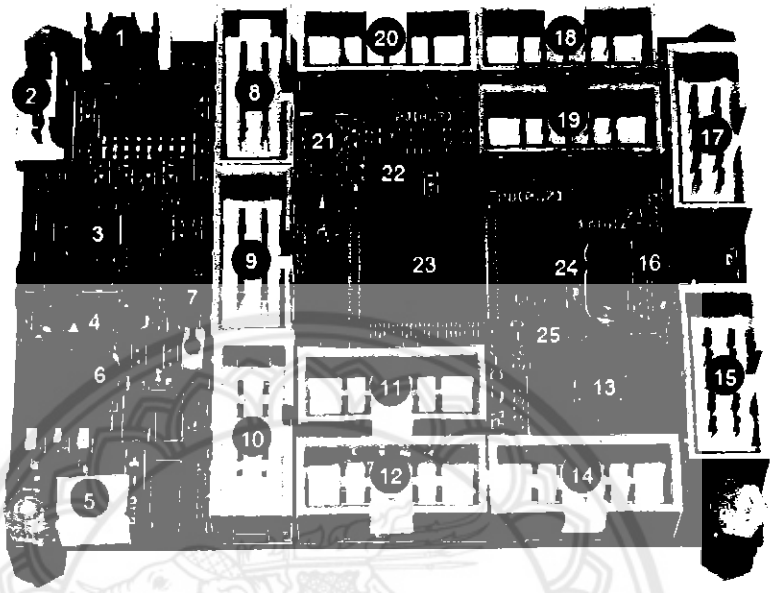
คุณสมบัติของ MCU ATMEGA1280/2560

- 128KB(ATMEGA1280,256K(ATMEGA2560) Flash / 8KB SRAM / 4KB EEPROM
- ทำงานที่แรงดัน 2.7V-5.5V (Run 16MHz ที่ 4.5V-5.5V, Run 8MHz ที่ 2.7-5.5V)
- มีวงจร Internal RC Clock 8MHz ภายใน
- มี 12 ช่อง 16Bit PWM
- มี 4 ช่อง USART
- มี 16 ช่อง 10Bit ADC(15KSPS)
- มี 86 Bit GPIO
- มี 2 ช่อง 8Bit Timer/Counter
- มี 4 ช่อง 16Bit Timer/Counter
- มี Real Time Counter
- มี 4 ช่อง 8Bit PWM
- มี 1 ช่อง I2C
- มี 1 ช่อง SPI
- มีระบบ JTAG(IEEE 1149.1 Compliant) สำหรับ Program และ Debug
- มี ISP (In System Programming) สำหรับ Program

คุณสมบัติของบอร์ด ET-BASE MEGA1280/2560

- ใช้ MCU ตระกูล MEGA AVR เบอร์ ATMEGA1280/2560 ของ ATMEL
- มีหน่วยความจำ Flash 128KB(ATMEGA1280,256K(ATMEGA2560), 8KB Boot loader, Static RAM 8KB และ EEPROM 4KB
- ใช้ Crystal 16.00 MHz
- มีวงจร Real Time Counter พร้อม XTAL ค่า 32.768KHz
- รองรับการโปรแกรมแบบ In-System Programming แบบ ISP
- มีวงจรเชื่อมต่อกับ AVR-JTAG ขนาด 10 Pin เพื่อทำการ Debug แบบ Real Time ได้
- Power Supply ใช้แรงดันไฟฟ้า +5VDC พร้อมวงจร Regulate +3V3/3A ภายในบอร์ด พร้อม Jumper เลือกระบบแหล่งจ่ายให้เป็น 3.3V หรือ 5V ได้ตามต้องการ
- มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card(Micro SD) เชื่อมต่อแบบ SPI จำนวน 1 ช่อง
- มีวงจรสื่อสาร RS232 โดยใช้ขั้วต่อแบบ 4-PIN มาตรฐาน ETT จำนวน 2 ช่อง
- มีวงจร LED แสดงสถานะเพื่อทดลอง Output(PG5) จำนวน 1 ชุด
- มี 83 Bit GPIO อิสระ สำหรับประยุกต์ใช้งานต่างๆ เช่น A/D, I2C, SPI, USART และ Input / Output แบบต่างๆ โดยมีการจัดสรรใช้งานภายในบอร์ดไว้แล้ว จำนวน 8 เส้นสัญญาณ คือ PB[0..3] สำหรับ micro SD Card, PD[2..3] และ PE[0..1] สำหรับ RS232 แต่สัญญาณทั้ง 8 เส้น ดังกล่าวยังมีการเชื่อมต่อสัญญาณออกมาไว้ที่ขั้วต่อ 10PIN IDE ของบอร์ดด้วย โดยใช้ขั้วต่อสัญญาณแบบ 10PIN IDE จำนวน 10 ชุด มีดังนี้
  - Header 10Pin IDE (PA[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PB[0..7]) สำหรับ GPIOและ PB[0..3] ถูกใช้สำหรับ SPI ในการเชื่อมต่อกับ micro SD Card
  - Header 10Pin IDE (PC[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PD[0..7]) สำหรับ GPIO(PD[2..3] ถูกใช้สำหรับ USART0)
  - Header 10Pin IDE (PE[0..7]) สำหรับ GPIO(PE[0..1] ถูกใช้สำหรับ USART1)
  - Header 10Pin IDE (PF[0..7]) สำหรับ GPIO(PF[4..7])ถูกใช้สำหรับ JTAG)
  - Header 10Pin IDE (PH[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PJ[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PK[0..7]) สำหรับ GPIO
  - Header 10Pin IDE (PL[0..7]) สำหรับ GPIO
  - Header 1x5Pin (PG[0..2]) สำหรับ GPIO

โครงสร้างบอร์ด ET-BASE MEGA1280/2560

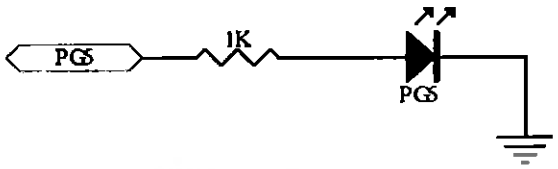


รูปแสดง โครงสร้างของบอร์ด ET-BASE MEGA1280/2560

- หมายเลข 1,2 คือ ขั้วต่อ USART0(RS232) และ USART1(RS232) สำหรับใช้งาน
- หมายเลข 3,4 คือ ช่องเสียบการ์ดหน่วยความจำ SD Card แบบ Micro-SD และ LED สถานะ
- หมายเลข 5,6 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ +5VDC และ LED Power
- หมายเลข 7 คือ Jumper สำหรับเลือกขนาดแรงดันทำงานของ MCU ระหว่าง 3.3V หรือ 5V
- หมายเลข 8,9,10,11 คือ ขั้วต่อ GPIO(PA[0..7]) ,(PK[0..7]),(PF[0..7]) และ GPIO(PE[0..7])
- หมายเลข 12 คือ ขั้วต่อ AVR-JTAG สำหรับ Debug แบบ Real Time
- หมายเลข 13 คือ ขั้วต่อ AVRISP สำหรับ Download โปรแกรม
- หมายเลข 14,15 คือ LED ขั้วต่อ GPIO(PH[0..7]) และ GPIO(PB[0..7])
- หมายเลข 16 คือ SW RESET
- หมายเลข 17,18,19,20 คือ ขั้วต่อ GPIO(PL[0..7]), (PC[0..7]),(PD[0..7]) และ GPIO(PJ[0..7])
- หมายเลข 21,22 คือ ขั้วต่อ GPIO(PG[0..2]) และ LED ใช้ทดสอบ Logic Output ของ PG5
- หมายเลข 23 คือ MCU เบอร์ ATMEGA1280 หรือ ATMEGA2560 (100Pin TQFP)
- หมายเลข 24 คือ Crystal ค่า 16.00 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU
- หมายเลข 25 คือ Crystal ค่า 32.768KHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU

การใช้งานวงจรขับ LED แสดงผล

LED แสดงผลของบอร์ด จะต่อวงจรแบบขับกระแส (Source Current) ทำงานด้วยโลจิก "1" และหยุดทำงานด้วยโลจิก "0" โดยควบคุมการทำงานจากขาสัญญาณ PG5 โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Output จากขาสัญญาณ PG5



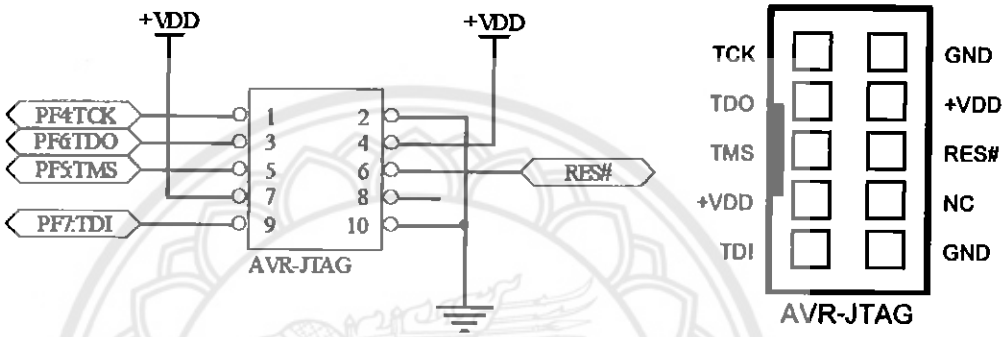
โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ PG[5] ทำหน้าที่เป็น GPIO Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ PG[5] ตามต้องการ ดังตัวอย่าง

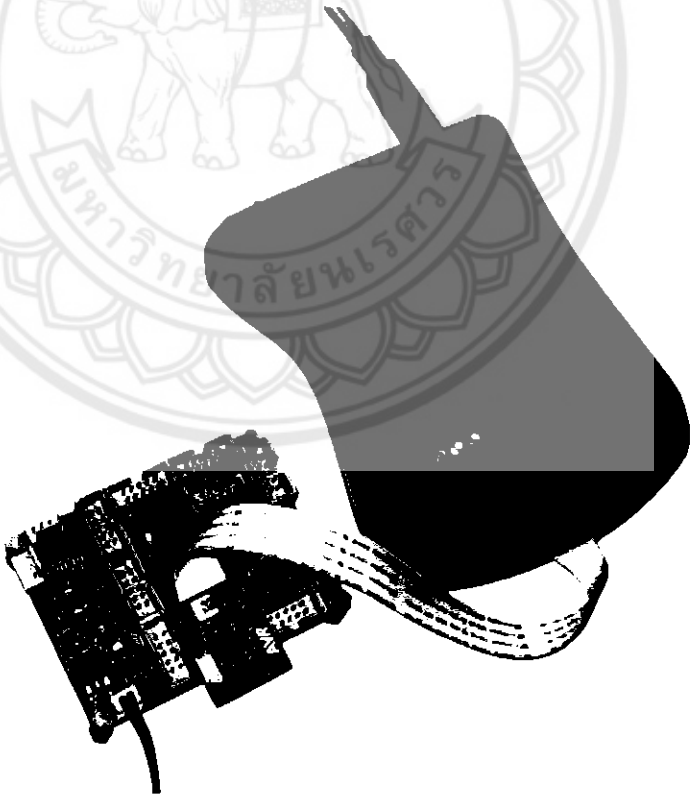
```
#define PORT_LED PORTG          // Port Drive LED = PG
#define DIR LED  DDRG           // Port Direction
#define LED 5                   // Pin Drive LED = PG5
.
.
.
int main(void)
{
  DIR LED |= (1<<LED);          // Pin Drive LED = Out
   .
   .
   .
  PORT LED &= ~(1<<LED);        // Pin LED = 0(OFF LED)
   .
   .
   .
  PORT LED |=  (1<<LED);        // Pin LED = 1(ON LED)
   .
   .
   .
  PORT_LED ^=  (1<<LED);        // Pin LED = Toggle
   .
   .
   .
}
```

ตัวอย่าง การกำหนดค่าการใช้งาน PG5 เป็น Output LED

การใช้งาน AVR-JTAG

AVR-JTAG จะเป็น Connector แบบ IDE 10 Pin สำหรับ สำหรับเชื่อมต่อกับเครื่อง โปรแกรม และ Debug ภายนอกที่ใช้มาตรฐาน AVR-JTAG ซึ่งรองรับการใช้งานร่วมกับ MCU เบอร์ ATMEGA1280/2560 เช่น AVR DRAGON หรือ AVR JTAGICE mkII หรือ เทียบเท่า โดยมีการจัดวงจรและสัญญาณตามแบบ AVR-JTAG ตามมาตรฐานของ ATMEL ไว้ดังนี้

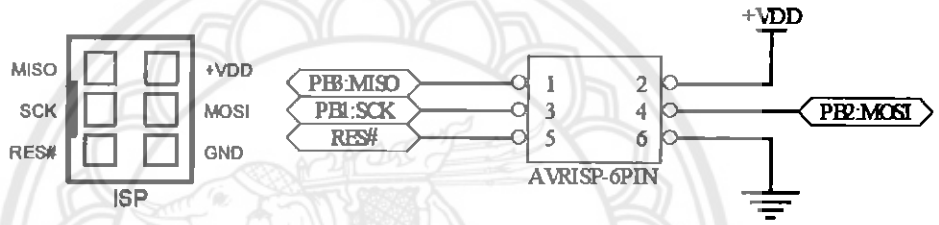


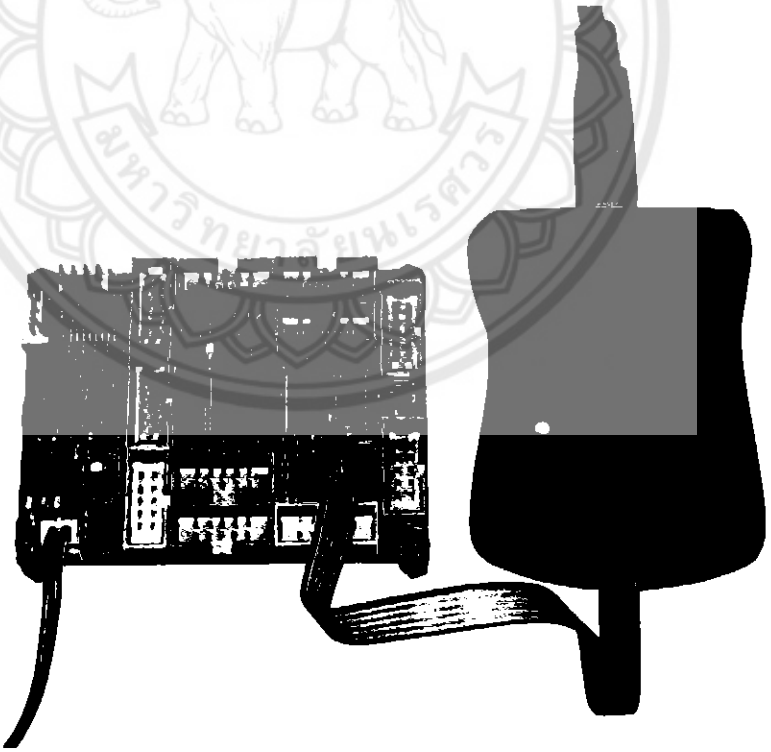


การใช้งาน ISP Program

ISP จะเป็น Connector แบบ IDE 6 Pin สำหรับ สำหรับเชื่อมต่อกับเครื่อง โปรแกรมภายนอกที่ใช้มาตรฐาน AVR ISP ซึ่งรองรับการใช้งานร่วมกับ MCU เบอร์ ATMEGA1280/2560 เช่น AVRISP mkII หรือเทียบเท่า เช่น ET-AVRISP mkII โดยมีการจัดวงจรและสัญญาณตามแบบ AVR ISP ตามมาตรฐานของ ATMEL ไว้ดังนี้

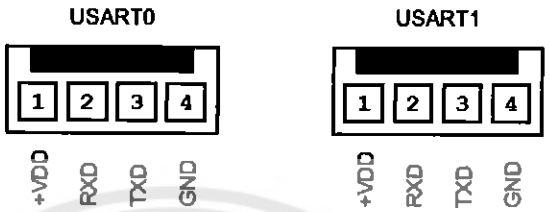
- PB1 = SCK
- PB2 = MOSI
- PB3 = MISO





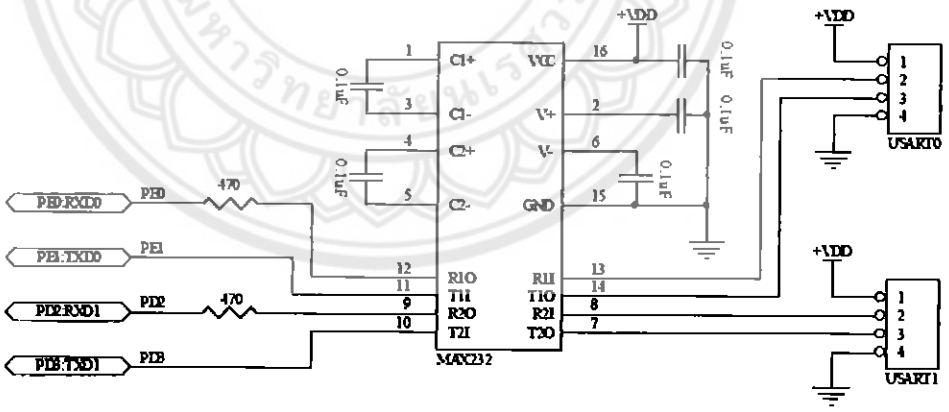
พอร์ต RS232

เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้ว โดยมีจำนวน 2 ช่อง ด้วยกันคือ USART0 และ USART1 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูลได้



- USART0 ใช้ขาสัญญาณจาก PE0(RXD) และ PE1(TXD)
- USART1 ใช้ขาสัญญาณจาก PD2(RXD) และ PD3(TXD)

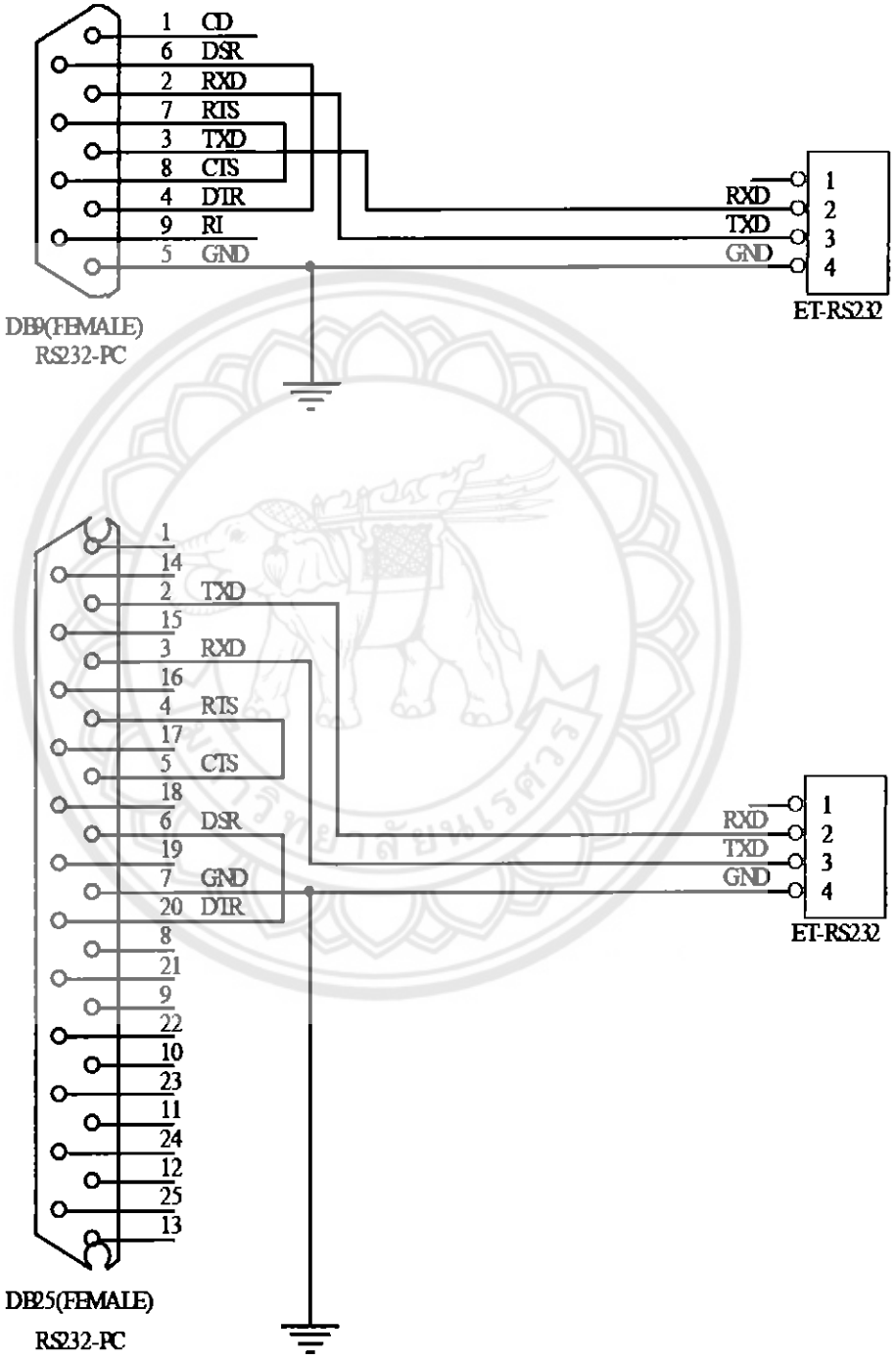
เนื่องจากระบบ Hardware USART ของ ATMEGA1280/2560 นั้นจะมี USART ไว้ใช้งานมากถึง 4 ชุด คือ PE0,PE1 และ PD2,PD3 และ PH0,PH1 และ PJ0,PJ1 โดยในการผลิตของบอร์ด ET-BASE MEGA1280/2560 จะจัดวงจร USART ร่วมกับวงจร Line Driver ของ RS232 ไว้ให้จำนวน 2 ช่อง คือ PE0,PE1 และ PD2,PD3 ดังวงจร



```
#include <avr/io.h>
#include <stdio.h>
#define F_CPU 16000000UL                    //16.0 MHz
#define BAUD 9600                           //9600 BPS
#define MYUBRR F_CPU/16/BAUD-1

void init serial(unsigned int ubrr);        //Initil UART
int   my putchar(char c, FILE *stream);
int   my_getchar(FILE *stream);

/* Retarget STDIO(putchar,getchar of printf) to My Function */
FILE uart_str = FDEV_SETUP_STREAM(my_putchar, my_getchar, _FDEV_SETUP_RW);

int main(void)
{
  init_serial(MYUBRR);                      //Initilial UART0 = 9600,N,8,1
  stdout = stdin = &uart str;

  printf("Hello World\n\r");
  while(1)
  {
    putchar(getchar());                     //Echo Receive Char
  }
}

/*****************/
/* Initial UART */
/*****************/
void init serial(unsigned int ubrr)
{
  UBRR0H = (unsigned char)(ubrr>>8);        //Set baud rate
  UBRR0L = (unsigned char)ubrr;
  UCSR0B = (1<<RXEN0)|(1<<TXEN0);           //Enable RX and TX
  UCSR0C = (1<<USBS0)|(3<<UCSZ00);          //8data, 2stop bit
}

/*****************************/
/* Write 1 Character To UART */
/*****************************/
int my_putchar(char c, FILE *stream)
{
  while (!(UCSR0A & (1<<UDRE0)));           // Wait TXD Buffer Empty
  UDR0 = c;
  return 0;
}

/*****************************/
/* Get 1 Character From UART */
/*****************************/
int my_getchar(FILE *stream)
{
  while(!(UCSR0A & (1<<RXC0)));             // Wait RXD Receive Data Ready
  return (UDR0);                            // Get Receice Data & Return
}
```

ตัวอย่างการใช้งาน USART0 รับส่งข้อมูล

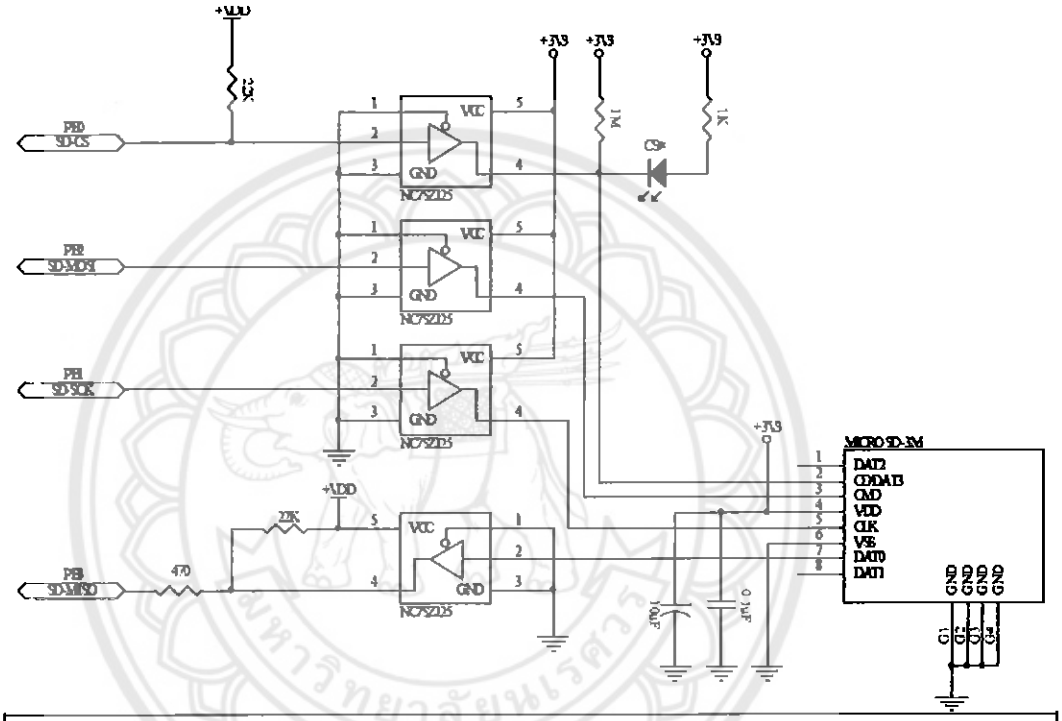
สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้วต่อ USART0 และ USART1 ของบอร์ด ET-BASE MEGA1280/2560 นั้น เป็นดังนี้



รูป แสดงวงจรสาย Cable สำหรับ RS232

การ์ดหน่วยความจำ SD Card แบบ Micro-SD

บอร์ด ET-BASE MEGA1280/2560 รองรับการเชื่อมต่อกับการ์ดหน่วยความจำ SD Card แบบ Micro-SD โดยใช้การเชื่อมต่อแบบ SPI โดยใช้ขาสัญญาณ PB[0..3] ในการเชื่อมต่อกับการ์ด ซึ่งในการติดต่อสั่งงาน การ์ดนั้น สามารถโปรแกรม Pin I/O ของ PB[0..3] ให้ทำงานในโหมด SPI โดยต้องกำหนดหน้าที่ของขาสัญญาณ PB[0..3] ของ MCU เป็นดังนี้

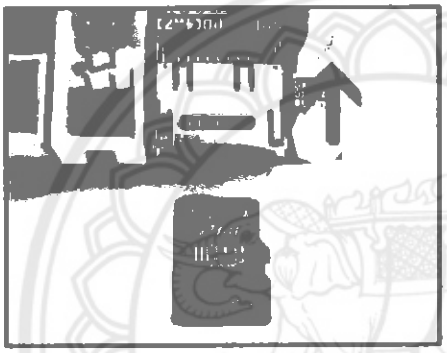



```
#define DDR_SPI DDRB
#define SD_CS   DDB0
#define SD_SCK  DDB1
#define SD_MOSI DDB2
.
.
// Set as master, clock and chip select output
DDR_SPI |= (1<<SD_MOSI)|(1<<SD_SCK)|(1<<SD_CS);  //MOSI,SCK,CS = Out
SPCR     = (1<<SPE)|(1<<MSTR);                   //SPI, Master

// SPI = Low Speed(Max. 400 kBit used in Card Initialization)
// SPI2X:SPR1:SPR0 = 0:1:1 (SCK = Fosc/128)
// SCK = Fosc  / 128
//     = 16MHz / 128
//     = 126KHz
SPSR &= ~(1 << SPI2X);                           // SPI2X = 0
SPCR |=  ((1 << SPR1) | (1 << SPR0));            // SPR1:SPR0 = 1:1
```

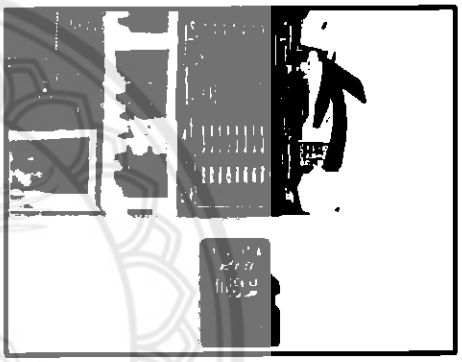
ตัวอย่าง การกำหนดค่า Pin สำหรับใช้งาน SD Card

การใส่การ์ดหน่วยความจำใน Socket

สำหรับ Socket สำหรับติดตั้งการ์ดหน่วยความจำของบอร์ด ET-BASE MEGA1280/2560 นั้น จะใช้ Socket หน่วยความจำแบบใส่การ์ดจากด้านบน แบบเดียวกันกับซิมการ์ดของโทรศัพท์มือถือ โดยเมื่อต้องการจะใส่หรือเอดการ์ดหน่วยความจำจะต้องทำการเปิดฝาครอบ Socket ออกเสียก่อน จากนั้นจึงจะสามารถใส่หรือเอดการ์ดหน่วยความจำได้ โดยการเปิด หรือ ปิด ฝาครอบ Socket จะใช้การกดเลื่อนฝาครอบเข้าหรือออก ซึ่งถ้าเลื่อนฝาครอบเข้าด้านในจะเป็นการเลื่อนเพื่อเปิดฝา แต่ถ้าเลื่อนออกด้านนอกจะเป็นการเลื่อนเพื่อล็อคฝาครอบดังรูป

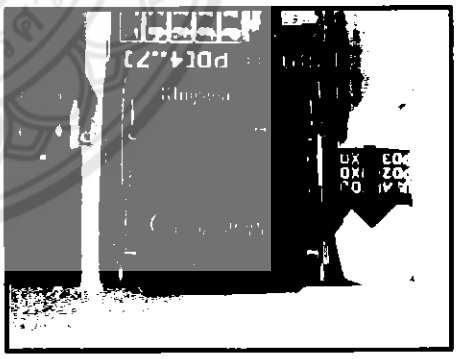


1. กดฝาครอบและเลื่อนเข้าเพื่อเปิดล็อค



2. เปิดฝาครอบ Socket ออก

3. ใสการ์ดหน่วยความจำใน Socket

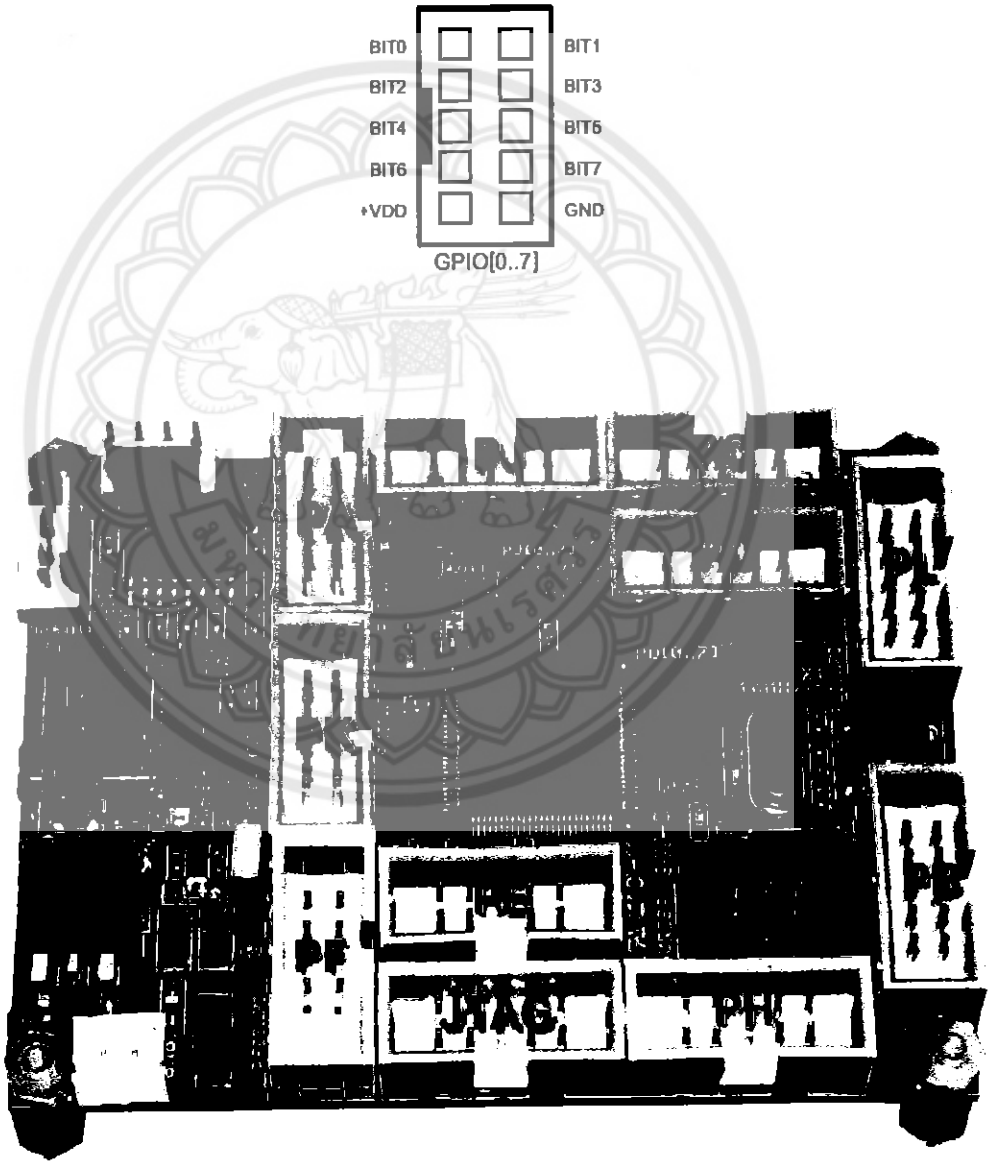


4. ปิดฝาครอบและกดเลื่อนออกเพื่อล็อค

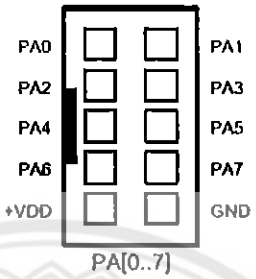
แสดงลำดับขั้นตอนการใส่การ์ดหน่วยความจำ

ขั้วต่อ Port I/O ต่าง ๆของบอร์ด

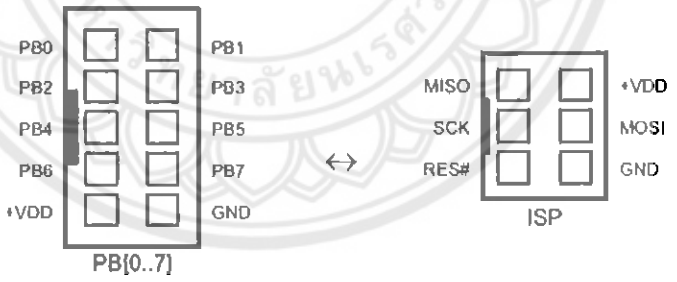
สำหรับขั้วต่อ Port I/O ของ MCU ของบอร์ด ET-BASE MEGA1280/2560 นั้น จะจัดเรียงออกมารอไว้ยังขั้วต่อแบบ IDE 10 Pin จำนวน 10 ชุดๆละ 8บิต สำหรับให้ผู้ใช้เลือกต่อออกไปใช้งานตามต้องการ โดยรูปแบบการจัดเรียงสัญญาณของแต่ละพอร์ต จะเรียงลำดับตำแหน่งบิต I/O เหมือนกัน โดยมีรายละเอียดของสัญญาณแต่ละชุดดังนี้



- ขั้วต่อ IDE 10 Pin ของ PA[0..7] โดย Port PA ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

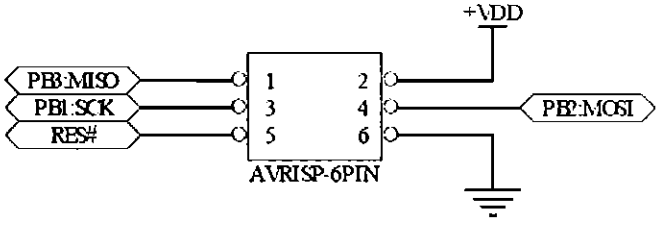


- ขั้วต่อ IDE 10 Pin ของ PB[0..7] โดย Port PB ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PB[1..3] ไปยังขั้วต่อ AVRISP และ PB[0..3] ก็จะถูกเชื่อมต่อเข้ากับวงจรของ Socket Micro SD-Card ด้วย ส่วน PB[4...7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ ซึ่งถ้ามีการต่อ PB[1..3] กับอุปกรณ์ภายนอกไว้ เมื่อต้องการโปรแกรมผ่าน AVRISP จะต้องปลดขาสัญญาณชุดนี้ให้เป็นอิสระจากอุปกรณ์ภายนอกด้วย ไม่เช่นนั้นอาจไม่สามารถโปรแกรมผ่าน AVRISP ได้ โดย PB[0..7] มีการจัดเรียงสัญญาณไว้ดังนี้

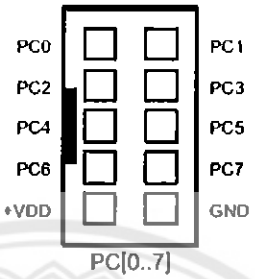


*****หมายเหตุ***** PB1...PB3 จะใช้งานกับกับ ISP Function ด้วย ซึ่งขาสัญญาณ PB1...PB3 ของบอร์ดจะถูกเชื่อมต่อไปยังขั้วต่อ AVRISP และ Micro SD-Card ด้วย โดยขาสัญญาณ PB1...PB3 เมื่อทำหน้าที่เป็น AVRISP จะมีหน้าที่ดังนี้

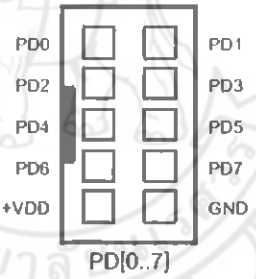
- PB1 = SCK
- PB2 = MOSI
- PB3 = MISO



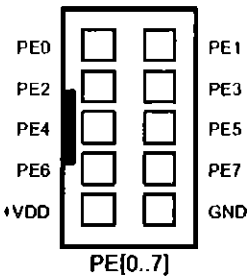
- ขั้วต่อ IDE 10 Pin ของ PC[0..7] โดย Port PC ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



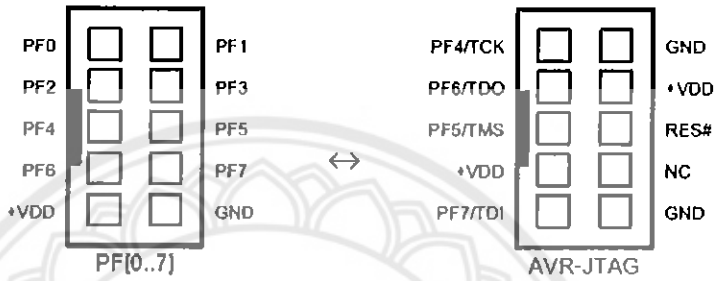
- ขั้วต่อ IDE 10 Pin ของ PD[0..7] โดย Port PD ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PD2...PD3 ไปยังวงจร USART1 ด้วย ส่วนสัญญาณ PD0...PD1 และ PD[4..7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



- ขั้วต่อ IDE 10 Pin ของ PE[0..7] โดย Port PE ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PE0...PE1 ไปยังวงจร USART0ด้วย ส่วน PE[2...7] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

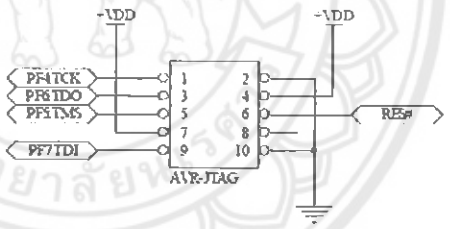


- ขั้วต่อ IDE 10 Pin ของ PF[0..7] โดย Port PF ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 นั้น จะมีการเชื่อมต่อ PF[4..7] ไปยังขั้วต่อสำหรับ AVR-JTAG ด้วย ส่วน PF[0...3] จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เอง ตามความต้องการ ซึ่งถ้าต้องการใช้งาน PF[4..7] ต้องไปสั่ง Disable Fuse Bit ของ JTAGEN ออกเสียก่อนด้วยเครื่องโปรแกรมทาง ISP Port โดย PF[0..7] มีการจัดเรียงสัญญาณไว้ดังนี้

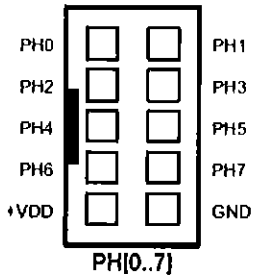


หมายเหตุ PF4...PF7 จะซ้อนกับกับ JTAG Function ด้วย ซึ่งขาสัญญาณ PF4...PF7 ของบอร์ดจะถูกเชื่อมต่อไปยังขั้วต่อ AVR-JTAG ด้วย ซึ่งถ้าผู้ใช้ได้เชื่อมต่อ AVR-JTAG ไว้ด้วย จะไม่สามารถใช้งานขาสัญญาณ PF4...PF7 ได้ โดยขาสัญญาณ PF4...PF7 เมื่อทำหน้าที่เป็น JTAG จะมีหน้าที่ดังนี้

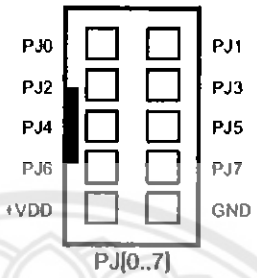
- PF4 = TCK JTAG
- PF5 = TMS JTAG
- PF6 = TDO JTAG
- PF7 = TDI JTAG



- ขั้วต่อ IDE 10 Pin ของ PH[0..7] โดย Port PH ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

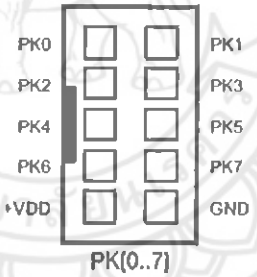


- ขั้วต่อ IDE 10 Pin ของ PJ[0..7] โดย Port PJ ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้



PJ[0..7]

- ขั้วต่อ IDE 10 Pin ของ PK[0..7] โดย Port PK ของ ATMEGA1280/2560 จะมีทั้งหมดจำนวน 8 บิต ซึ่งในบอร์ด ET-BASE MEGA1280/2560 เองนั้นสัญญาณทั้ง 8 เส้น จะยังคงอิสระปล่อยว่างไว้ให้ผู้ใช้นำไปต่อประยุกต์ใช้งานได้เองตามความต้องการ โดยมีการจัดเรียงสัญญาณไว้ดังนี้

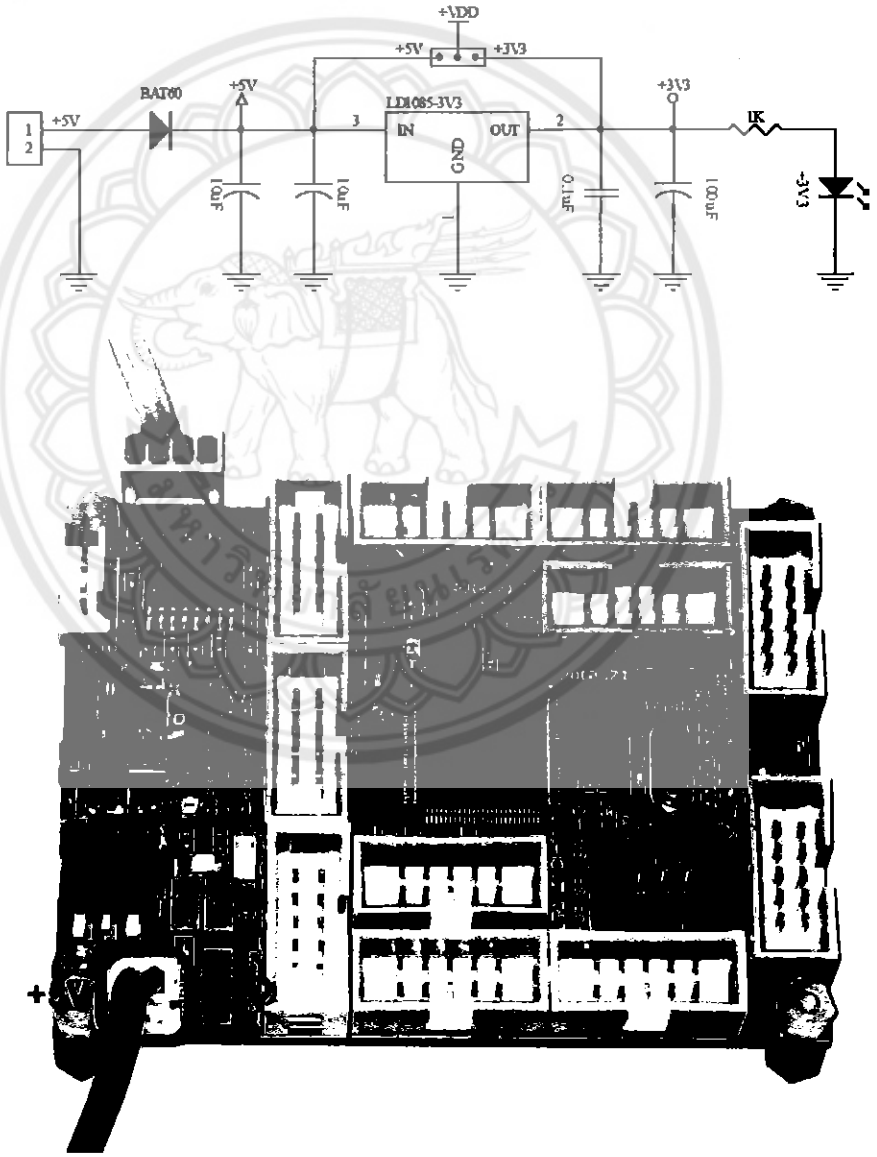


PK[0..7]

วงจรแหล่งจ่ายไฟ

วงจรแหล่งจ่ายไฟของบอร์ดใช้งานได้กับไฟ DC ขนาด +5V โดยใช้ขั้วต่อแบบ 2 Pin Block ป้องกันการเสียบสายกลับขั้ว พร้อมวงจร Regulate ขนาด +3V3/3A

โดยวงจรภาคแหล่งจ่ายไฟในส่วนที่เป็นวงจร Regulate ขนาด 3.3V นั้นจะจ่ายตรงไปให้กับ วงจร SD-Card ส่วนวงจรของ MPU และวงจร I/O ของบอร์ดนั้น จะสามารถเลือกขนาดแรงดันใช้งานได้จาก Jumper ว่าต้องการใช้งานเป็น 3V3 หรือ 5V ดังรูป



ตัวอย่างการพัฒนาโปรแกรมด้วย WinAVR ร่วมกับ AVR Studio4

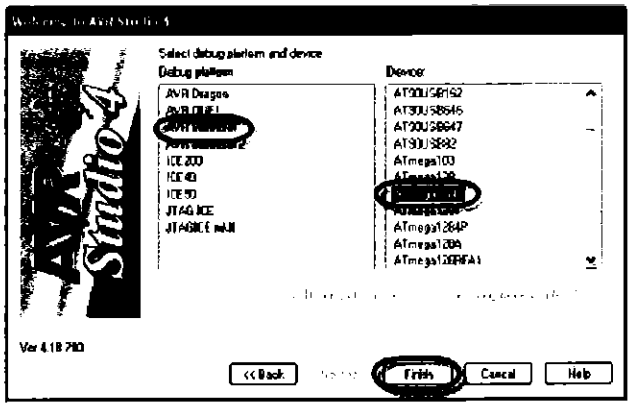
ในการพัฒนาโปรแกรมของ AVR เบอร์ ATMEGA1280/2560 นั้น ตามปรกติแล้วจะสามารถเลือกใช้ Compiler ต่างๆที่สนับสนุน MCU เบอร์นี้อยู่ได้ทั้งหมด แต่ในที่นี้จะขอแนะนำให้ใช้โปรแกรม AVR Studio4 ร่วมกับ WinAVR ซึ่งเป็นชุดโปรแกรมที่แจกจ่ายให้ใช้ได้ฟรีๆไม่เสียค่าใช้จ่าย มีการพัฒนาปรับปรุงความสามารถของโปรแกรมกันอย่างต่อเนื่องและมีผู้ใช้งานกันอย่างแพร่หลายทั่วโลก สามารถค้นหาตัวอย่างโปรแกรม และ Library ต่างๆที่ผู้ใช้ต่างๆจำนวนมากได้สร้างและเผยแพร่ไว้มาเป็นแนวทางในการศึกษาได้มากมายซึ่งปัจจุบัน (สิงหาคม 2553) ทาง ATMEL ได้ทำการปรับปรุงโปรแกรม AVR Studion4 เป็นรุ่น V4.18 แล้ว ส่วนโปรแกรม WinAVR สามารถ Download ได้จาก http://winavr.sourceforge.net/ จะปรับปรุงเป็น WINAVR-20100110 แล้ว ซึ่งผู้ใช้สามารถไป Download มาติดตั้งใช้งานได้ฟรี โดยในที่นี้จะขอแนะนำแนวทางการพัฒนาโปรแกรมแบบพอสังเขป เพื่อเป็นแนวทางสำหรับผู้เริ่มต้น ซึ่งรายละเอียดต่างๆสามารถศึกษาได้จากคู่มือของโปรแกรมได้ โดยแนวทางการพัฒนาโปรแกรมของ ATMEGA1280 และ ATMEGA2560 โดยใช้โปรแกรม AVR Studio4 ร่วมกับ WinAVR มีลำดับขั้นตอนดังนี้

1. สั่ง Run Program AVR Studio4 ซึ่งจะได้ผลดังรูป

2. สั่งสร้าง project ใหม่ โดยเลือกที่ project → New project จากนั้นเลือกกำหนดตัวเลือกต่างๆ ให้กับโปรแกรมดังนี้

   - Project type เลือกกำหนดเป็น AVR GCC
   - Location สำหรับบันทึก project ให้ระบุตำแหน่ง Folder ที่ต้องการใช้บันทึกไฟล์ และ Codeต่างของ project
   - Project name ให้กำหนดชื่อ project ตามต้องการในตัวอย่างกำหนดเป็น "demo" และให้เลือก Create initial file ไว้ด้วย ซึ่งเมื่อเราทำการกำหนดชื่อ project name เสร็จแล้ว โปรแกรมจะสร้างไฟล์ ที่มีชื่อเดียวกันกับ project name ให้เองโดยอัตโนมัติ ซึ่งถ้าต้องการกำหนดชื่อไฟล์เป็นชื่ออื่น ก็ไม่ต้องเลือก Create initial file

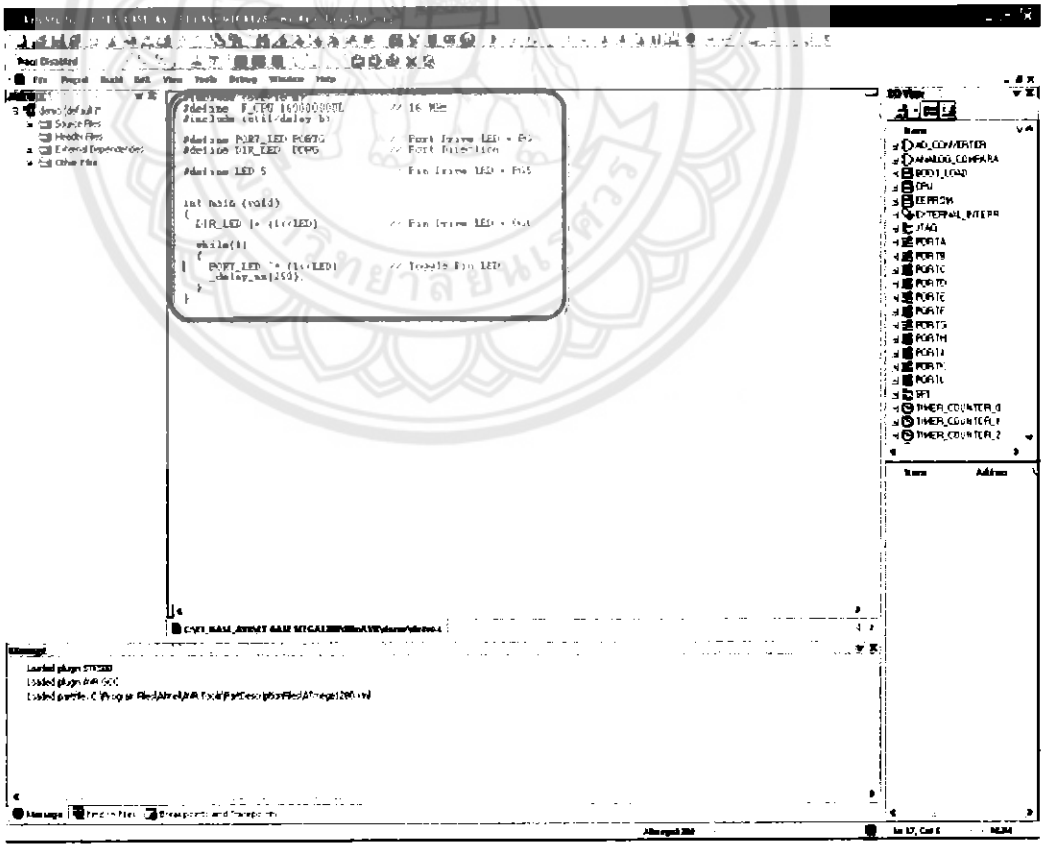
3. เมื่อกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมเรียบร้อยแล้ว ให้เลือกที่ Next แล้วกำหนดค่าใน Debug platform เป็น AVR Simulator และเลือก Device เป็น ATmega1280 ซึ่งเมื่อสร้าง project เสร็จโปรแกรมจะสร้างไฟล์ภาษาซีให้ โดยมีชื่อเดียวกับ project ไฟล์ ซึ่งในที่นี้จะเป็นไฟล์ชื่อ demo.c ให้เองโดยอัตโนมัติ เพียงแต่ไฟล์ดังกล่าวจะยังไม่มี code ใดๆบรรจุไว้ให้ เป็นเพียงหน้ากระดาษเปล่าๆ ซึ่งต้องรอให้เราเขียน code เพิ่มเข้าไปเอง ดังรูป



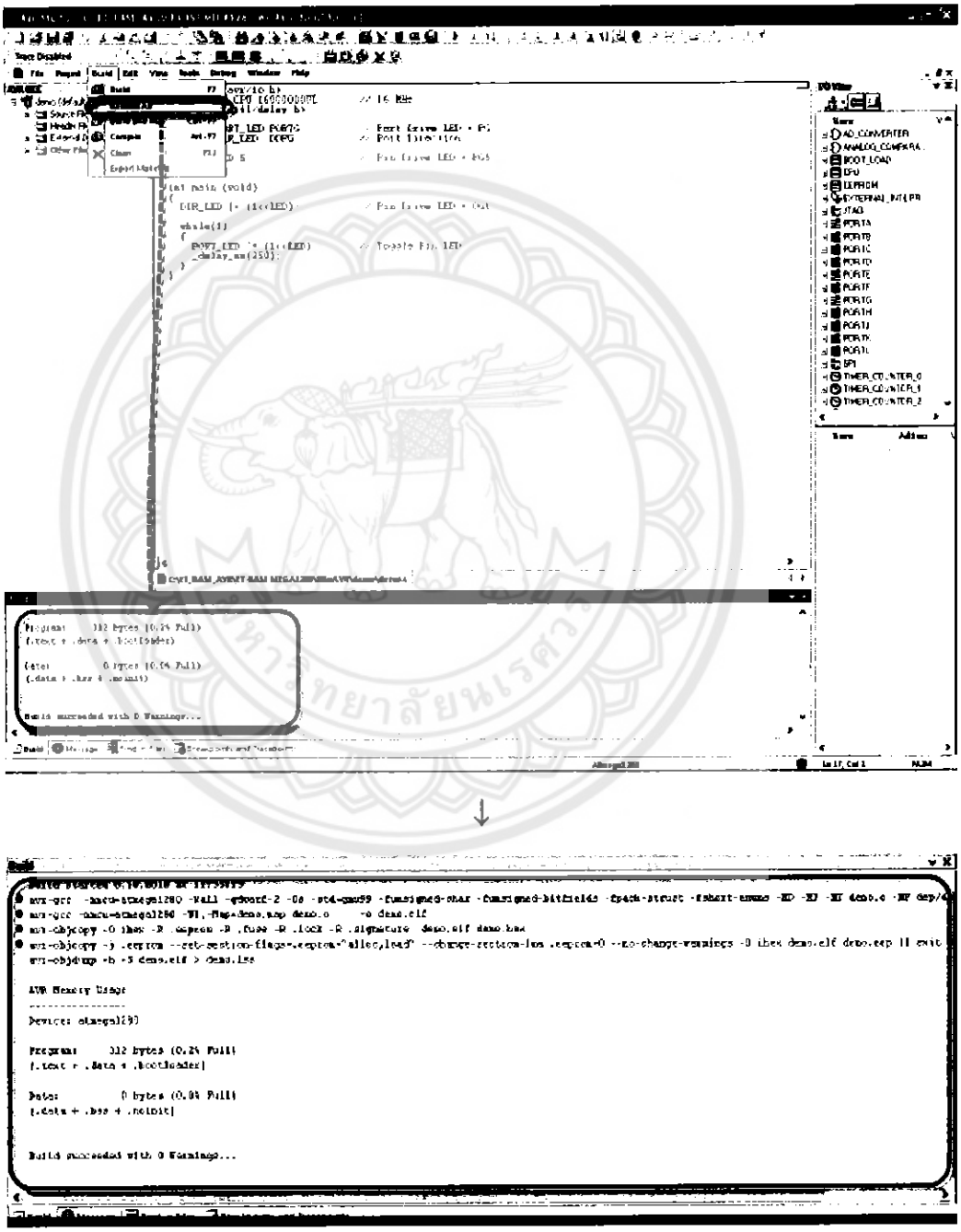
4. ให้พิมพ์คำสั่งของโปรแกรมสำหรับทดสอบการทำงาน ในหน้าต่าง Text Editor ของโปรแกรม โดยในที่นี้จะทดสอบด้วย Codeโปรแกรม สำหรับทำหน้าที่ทดสอบการทำงานของบอร์ดในเบื้องต้น โดยทำหน้าที่ ON/OFF หลอดแสดงผล LED ซึ่งต่อควบคุมจากขา PG5 ดังตัวอย่าง

```
#include <avr/io.h>
#define  F CPU 16000000UL                  // 16 MHz
#include <util/delay.h>

#define PORT LED PORTG                     // Port Drive LED = PG
#define DIR_LED  DDRG                      // Port Direction
#define LED 5                              // Pin Drive LED = PG5

int main (void)
{
  DIR_LED |= (1<<LED);                     // Pin Drive LED = Out

  while(1)
  {
    PORT_LED ^= (1<<LED);                  // Toggle Pin LED
    _delay_ms(250);
  }
}
```

↓

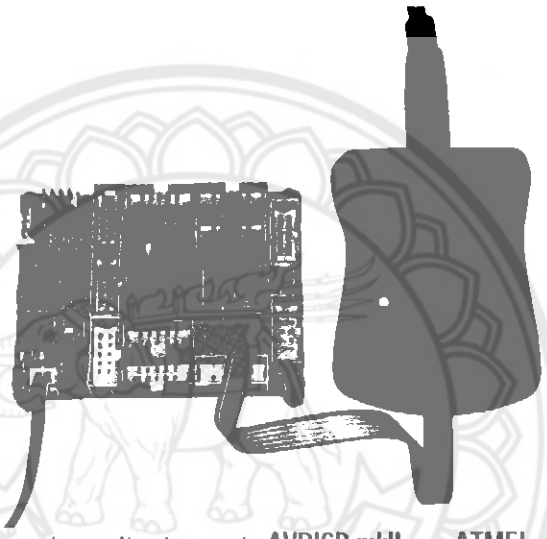


5. หลังจากพิมพ์ Code โปรแกรมเสร็จแล้วให้สั่งแปลโปรแกรม โดยเลือกที่ build →rebuild all ซึ่งถ้าทุกอย่างถูกต้อง ผลการแปลคำสั่งจะได้ผลลัพธ์เป็น "Build succeeded with 0 Warnings... " และจะรายงานผลการแปลพร้อมขนาดหน่วยความจำที่ใช้ไป และจะได้ Output เป็น HEX File ที่มีชื่อเดียวกันกับ project ที่สร้างไว้ โดยจะบรรจุอยู่ในDirectory ย่อยชื่อ default ดังรูป

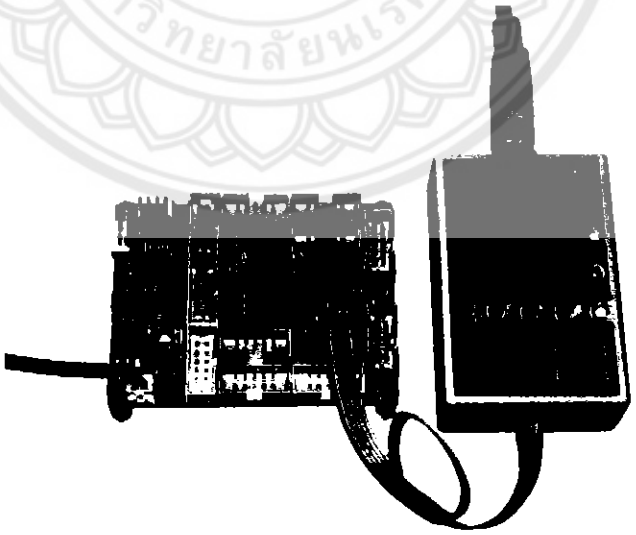


การโปรแกรม Hex File ให้บอร์ดโดยใช้ AVRISP mkII

ซึ่งเมื่อผ่านกระบวนการ Compile Code จนได้ Hex File มาแล้ว ขั้นตอนต่อไป จะเป็นขั้นตอนของการนำข้อมูล Code ใน Hex File เขียนเข้าไปในหน่วยความจำของ MCU ซึ่งขั้นตอนนี้สามารถทำได้หลายแนวทาง แต่ในที่นี้จะแนะนำให้ใช้เครื่องโปรแกรม AVRISP mkII หรือ อุปกรณ์อื่นที่มีความสามารถเทียบเท่ากัน เช่นเครื่องโปรแกรมรุ่น ET-AVRISP mkII ของ อีทีที โดยสามารถสั่งงานผ่านโปรแกรม AVR Studio4 ได้เลย ซึ่งในการเชื่อมต่อกับเครื่องโปรแกรมจะใช้ขั้วต่อISP ดังตัวอย่าง



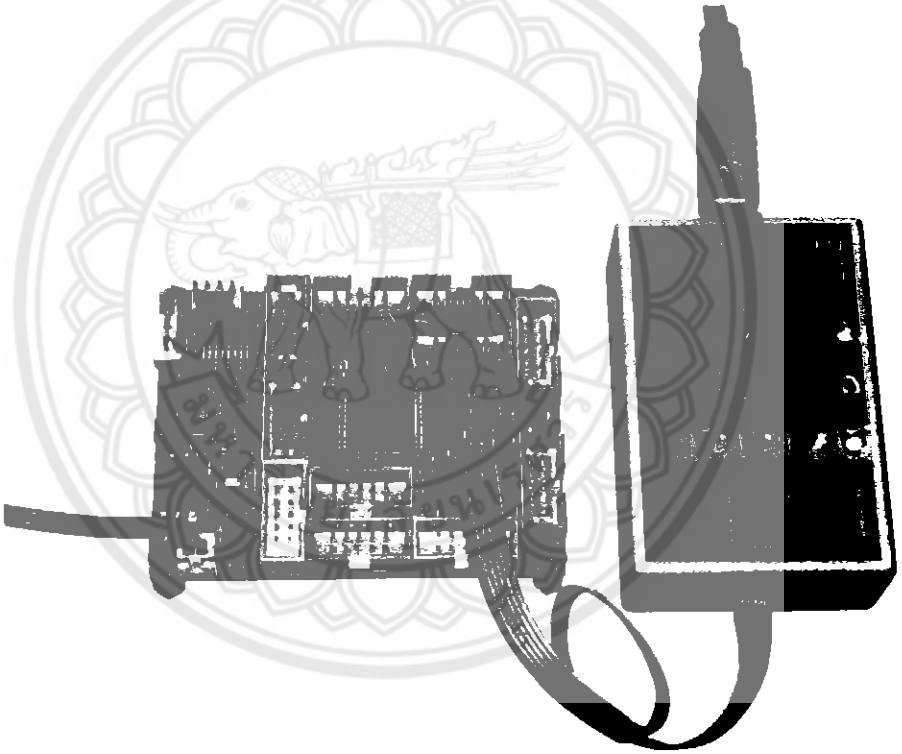
รูปแสดงตัวอย่างการต่อ AVRISP mkII ของ ATMEL



รูปแสดงตัวอย่างการต่อ ET-AVRISP mkII ของ อีทีที

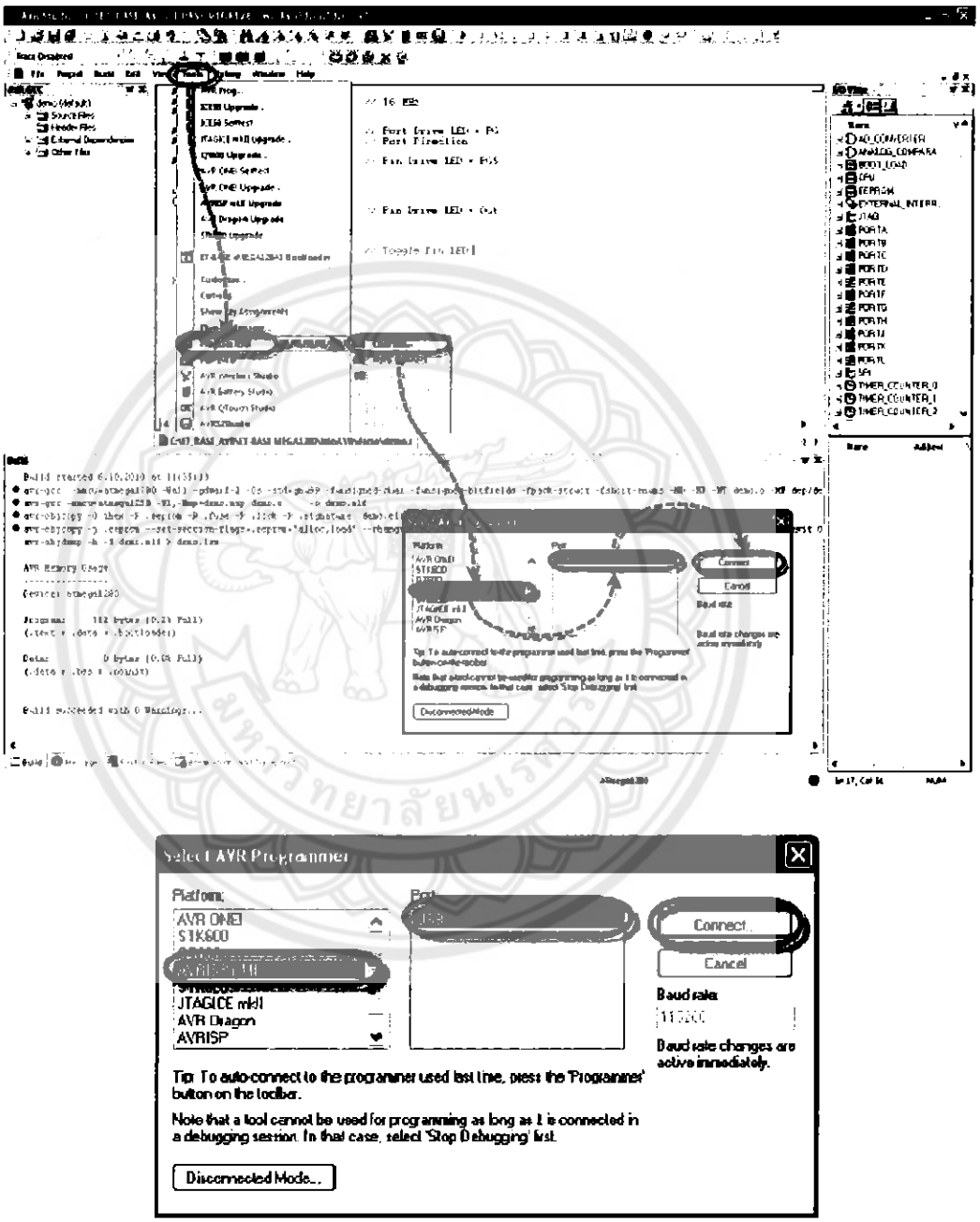
สำหรับลำดับขั้นตอนการโปรแกรม **Hex File** ด้วย **AVRISP mkII** มีดังนี้

1. จ่ายไฟให้บอร์ดโดยใช้ Adapter จ่ายไฟขนาด 5VDC โดยให้ระมัดระวัง และ ตรวจสอบขั้วของแหล่งจ่ายไฟให้ถูกต้องด้วย ซึ่งถ้าถูกต้องควรจะเห็น LED Power ติดสว่างให้เห็น
2. ต่อสายสัญญาณโปรแกรมของ ISP ระหว่างเครื่องโปรแกรม AVRISP mkII เข้ากับขั้วต่อ ISP ของบอร์ด ET-BASE MEGA1280/2560 โดยให้ตรวจตำแหน่งของขาสัญญาณให้ดี ระวังอย่าเสียบสายกลับด้าน ซึ่งถ้าเป็นเครื่องโปรแกรมและบอร์ดของอีทีที จะเลือกใช้ Connector IDE แบบ 6Pin ชนิดที่ป้องกันการเสียบสายกลับด้านเพื่อป้องกันไว้อยู่แล้ว ถ้าพบผิดความผิดปรกติเช่น LED Power ดับขณะเสียบสายให้รีบถอดสายออกและตรวจสอบสาเหตุความผิดพลาดทันที

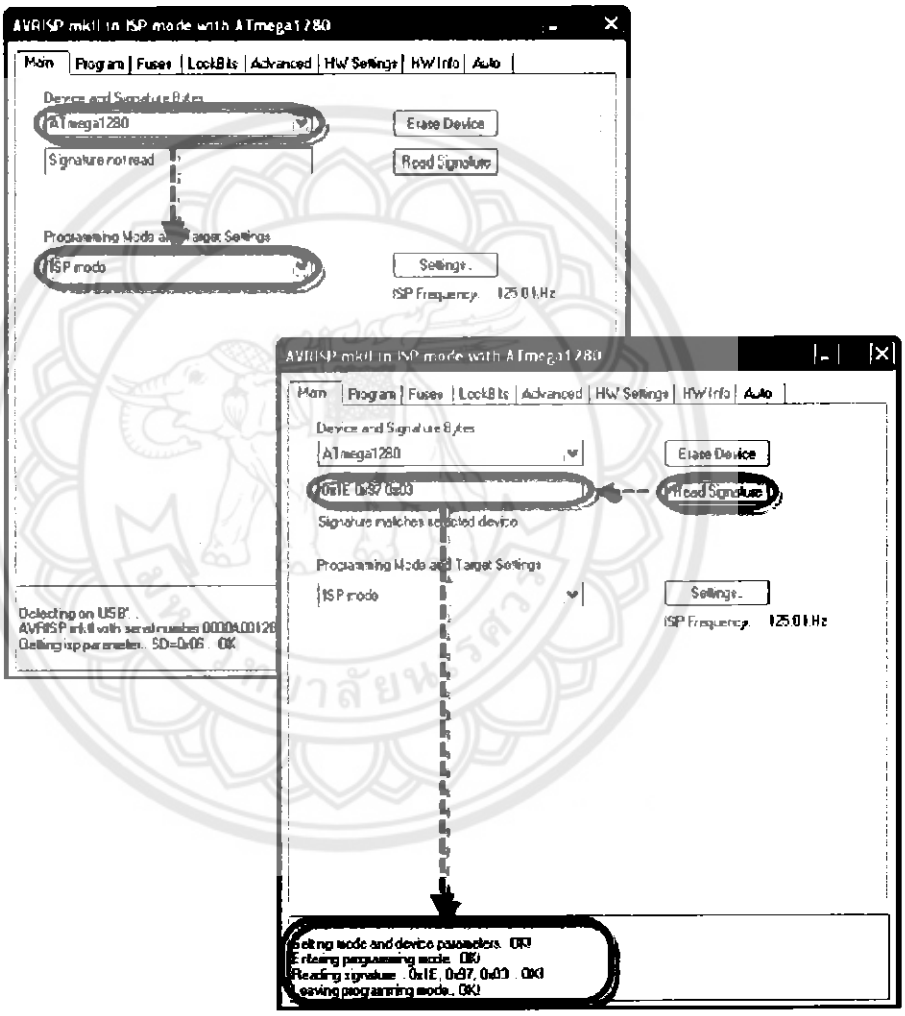


3. เสียบสาย USB ของเครื่อง AVRISP mkII เข้ากับเครื่องคอมพิวเตอร์ PC ซึ่งถ้าเป็นการใช้งานครั้งแรก Windows จะแจ้งว่าพบ new hardware และถามหาการติดตั้ง Driver ให้สั่งติดตั้ง Driver ให้เรียบร้อย โดยใน AVR Studio จะมีไฟล์ Driver ของ AVRISP mkII เตรียมไว้ให้ด้วยแล้ว โดยจะอยู่ใน C:\Program Files\Atmel\AVR Tools\usb ให้ทำการติดตั้ง Driver ให้เรียบร้อย (รายละเอียดศึกษาได้จากคู่มือการใช้งานของเครื่องโปรแกรม ET-AVRISP mkII)

4. เลือกคลิกเมาส์ที่ Tools → Program AVR → Connect.. → AVRISP mkII จากนั้นก็ให้เลือก port เป็น USB พร้อมกับเลือก Connect ดังรูป

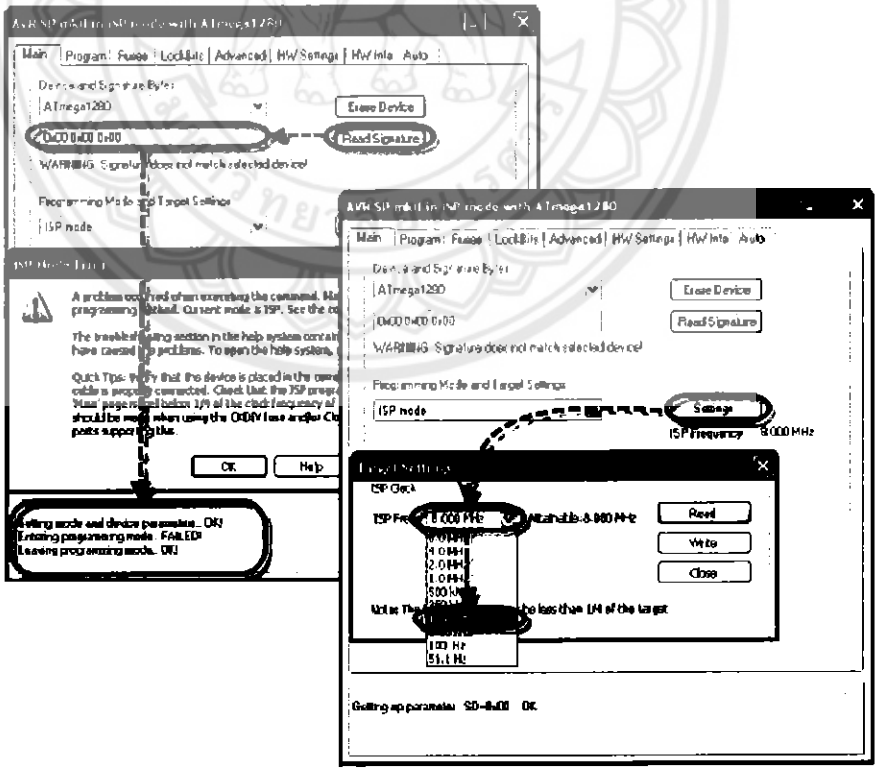


5. ถ้าทุกอย่างถูกต้องโปรแกรมจะเข้าสู่หน้าต่างโปรแกรมของ AVRISP mkII ให้ลองทดสอบการเชื่อมต่อระหว่าง ATMEGA1280/2560 กับ AVRISP mkII ดู ว่าสามารถสื่อสารกันได้เรียบร้อยหรือยัง โดยให้เลือกที่ tab ของ Main แล้วเลือกกำหนดเบอร์ MCU เป็น ATmega1280 หรือ ATmega2560 ให้ตรงกับเบอร์ที่ต่อไว้จริง พร้อมกับเลือกการเชื่อมต่อเป็น ISP mode แล้วลองเลือก Read Signature ดู ซึ่งถ้าทุกอย่างถูกต้องโปรแกรมควรต้องอ่านค่า Signature ของ ATMEGA1280 หรือ ATMEGA2560 ได้อย่างถูกต้อง ดังรูป

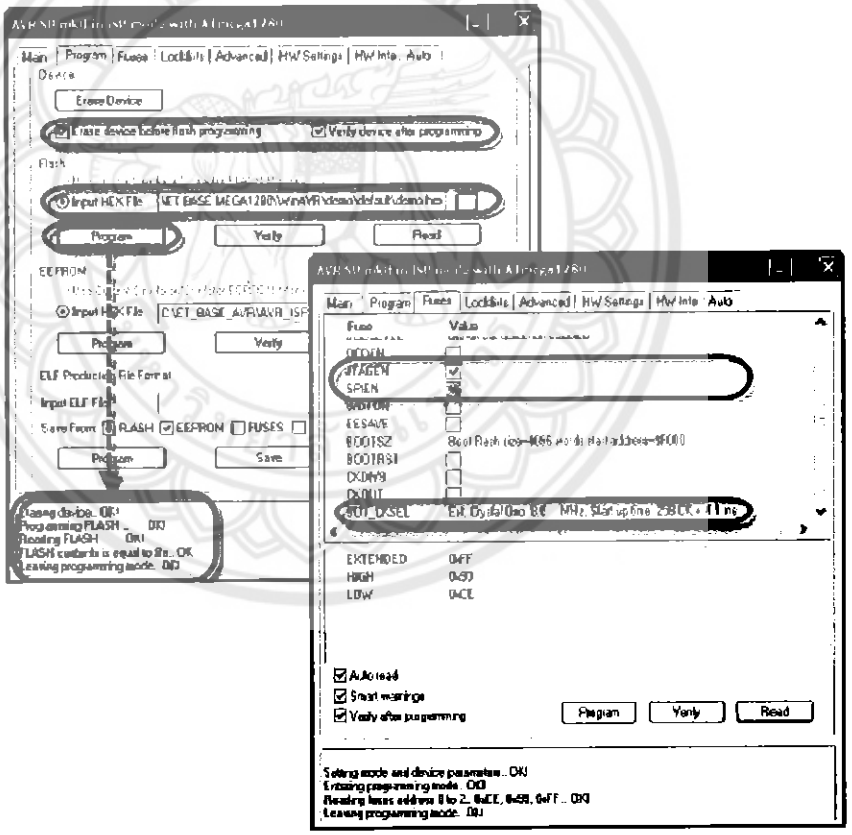


ในกรณีที่เกิดความผิดพลาดขึ้น ให้ลองตรวจสอบหาสาเหตุของความผิดพลาดและแก้ไขให้ถูกต้อง โดยมีแนวทางดังนี้

- การเชื่อมต่อสายระหว่างบอร์ด ET-BASE MEGA1280/2560 กับเครื่องโปรแกรม ถูกต้องหรือยัง บอร์ดอยู่ในสภาวะพร้อมที่จะทำงานหรือยัง
- ขาสัญญาณที่ทำหน้าที่เป็น ISP(PB1,PB2,PB3) มีการนำไปต่อใช้งานอย่างอื่นในขณะทำการโปรแกรมอยู่หรือไม่ ถ้ามีให้ปลดการเชื่อมต่อเหล่านั้นออกให้หมดเพื่อให้ขาสัญญาณดังกล่าวเป็นอิสระเสียก่อน
- กำหนดค่าความถี่ของ ISP Frequency ของเครื่องโปรแกรมไว้ สัมพันธ์สอดคล้องกันกับค่าความถี่ของ MCU ที่ทำงานใน Target Board หรือไม่ ซึ่งค่าความถี่ที่ถูกต้องของ ISP Frequency ต้องไม่เกิน 1/4 ของค่าความถี่ที่ MCU ทำงานอยู่ในขณะนั้น เช่น ถ้า MCU ทำงานที่ความถี่ 1MHz ค่าความถี่ของ ISP Frequency ต้องไม่เกิน 250KHz ตามปรกติถ้าเป็น MCU ตัวใหม่ๆจากโรงงานผลิต ค่าความถี่ของ MCU จากโรงงานจะถูกกำหนดให้ Run จากความถี่ 1MHz (Internal RC 8MHz / 8) ซึ่งถ้าไม่แน่ใจอาจทดลองปรับค่าความถี่ ISP Frequency ให้มีค่าต่ำๆดูก่อน เมื่อสามารถติดต่อสื่อสารกับ MCU ได้เรียบร้อยแล้วจึงค่อยเข้าไปตรวจสอบค่า Fuse Bit ของ MCU ที่เกี่ยวกับระบบสัญญาณนาฬิกา Clock ต่างๆ ในภายหลัง ดังรูป



6. เมื่อทุกอย่างถูกต้องให้เลือกไปที่ tab ของ Program พร้อมทั้งเลือก ตัวเลือกต่างๆดังนี้
   - Device ให้เลือก Erase device before flash programming และ Verify device after programming
   - Flash ให้เลือก Input HEX File ที่ต้องการจะโปรแกรมให้กับ MCU บนบอร์ต
   - Fuses และ Lock Bits สามารถเลือกกำหนด และสั่งโปรแกรมค่าได้ตามต้องการ ซึ่งก่อนจะสั่ง Program ค่าของ Fuse Bit ผู้ใช้ควรต้องศึกษารายละเอียดในการกำหนดค่าให้เข้าใจ ซึ่งจะต้องสัมพันธ์สอดคล้องกับความต้องการของระบบ Hardware ที่ใช้อยู่ด้วย ถ้ายังไม่แน่ใจในรายละเอียดไม่ควรไปสั่งโปรแกรมค่า ของ Fuse Bit เหล่านี้ เพราะถ้ามีการโปรแกรมค่าของ Fuse Bit ผิดไปอาจส่งผลให้ MCU ทำงานผิดพลาดไปจากความต้องการ ซึ่งในเบื้องต้น ค่า Fuse Bit และ Lock Bit แนะนำให้ข้ามไปก่อนยังไม่ต้องสั่ง โปรแกรมค่าทั้ง สองนี้ ให้จัดการเฉพาะส่วนของการโปรแกรม Flash ด้วย Hex ดังรูป



ซึ่งหลังจากสั่งโปรแกรมเสร็จ MCU จะเริ่มต้นทำงานตามคำสั่งใน Hex ที่ได้สั่งโปรแกรมไปแล้วทันที ถ้าใช้เครื่องโปรแกรม ET-AVRISP mkII แต่ถ้าเป็นเครื่องโปรแกรมรุ่นอื่นอาจต้องทำการกดสวิตช์รีเซ็ตที่บอร์ดก่อน 1 ครั้ง เพื่อสั่งให้ MCU เริ่มต้นทำงานหลังจากโปรแกรมเสร็จ