

## ดาต้าล็อกเกอร์และตัวประมวลผลการตัดสินใจ

Data Logger and Processing Unit

นายกฤษดา	วิวิตรกุล	รหัส	48361332
นายชัยเดช	ควรรคิด	รหัส	48361455
นายสุณวัฒน์	ชรรมานิมุข	รหัส	48361462

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 17 / ส.ค. 2553 / .....
เลขทะเบียน..... 15008645
เลขเรียกหนังสือ..... 2553
มหาวิทยาลัยย่นเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยย่นเรศวร

ปีการศึกษา 2551



## ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	คาคั่วล็อกเกอร์และตัวประมวลผลการตัดสินใจ		
ผู้ดำเนินโครงการ	นายณชดา	วิจิตรกุล	รหัส 48361332
	นายชัยเดช	ควรรคิด	รหัส 48361455
	นายฐณวัฒน์	ธรรมาภิมุข	รหัส 48361462
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร.ขงยุทธ ชนบดีเฉลิมรุ่ง		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยราชภัฏบรัง อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า  
คณะกรรมการสอบ โครงการวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ขงยุทธ ชนบดีเฉลิมรุ่ง)

.....กรรมการ

(ดร.อัครพันธ์ วงศ์กั้งแห)

.....กรรมการ

(ดร.ศุภวรรณ พลพิทักษ์ชัย)

หัวข้อโครงการ	คาน้ำต็อกเกอร์และตัวประมวลผลการตัดสินใจ		
ผู้ดำเนินโครงการ	นายกฤษดา	วิวิตรกุล	รหัส 48361332
	นายชัยเดช	ควรรคิด	รหัส 48361455
อาจารย์ที่ปรึกษา	นายธูณวัฒน์	ธรรมาภิมุข	รหัส 48361462
	ผู้ช่วยศาสตราจารย์ ดร.ขงยุทธ ชนบดีเฉลิมรุ่ง		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

### บทคัดย่อ

โครงการชิ้นนี้เกิดขึ้นเนื่องจาก ความต้องการระบบที่สามารถทำการเตือนภัยล่วงหน้าก่อนเกิดน้ำท่วมแบบฉบับพลัน ได้ ซึ่งในช่วงเวลาปัจจุบันนั้น สภาพอากาศได้แปรเปลี่ยนไปจากเดิมมาก เกิดฝนตกอย่างหนักในหลายพื้นที่ ทำให้มีการสูญเสียทั้งชีวิตและทรัพย์สินที่มีสาเหตุมาจากน้ำท่วมและดินถล่ม จึงเกิดแนวคิดที่จะสร้างระบบเตือนภัย โดยผ่านการเก็บข้อมูลและวิเคราะห์ข้อมูล เพื่อทำการส่งข้อมูลที่ได้รับไปทำการแจ้งเตือนภัยให้กับประชาชนที่อาศัยในบริเวณพื้นที่ที่อันตรายจากภัยพิบัติโดยใช้อุปกรณ์ที่ได้รับการประยุกต์ใช้งานในแบบต่างๆ ดังนั้นจึงได้มีการนำ ET-USB FLASH DRIVE มาใช้ในการเก็บข้อมูลที่ได้จากอินพุตที่ต้องการ เช่น จากวิทยุรับส่ง, จากสัญญาณอนาล็อกหรือดิจิตอล หรือแม้กระทั่งสัญญาณอินพุตที่อยู่ในรูปแบบของแรงดันหรือกระแส เพื่อเก็บข้อมูลและนำมาวิเคราะห์ข้อมูลในการแจ้งเตือนภัยธรรมชาติโดยอาศัยไมโครคอนโทรลเลอร์เป็นตัวประมวลผลและรับส่งสัญญาณ

<b>Project title</b>	Data Logger and Processing Unit		
<b>Name</b>	Mr. Gridsada	Vivitkul	ID. 48361332
	Mr. Chaidet	Kuankid	ID.48361455
	Mr. Thanawat	Tummapimuk	ID. 48361462
<b>Project advisor</b>	Asst. Prof. Dr. Yongyut Chonbodeechalermroong		
<b>Major</b>	Electrical Engineering.		
<b>Department</b>	Electrical and Computer Engineering.		
<b>Academic year</b>	2008		

### Abstract

The purpose of this project is to create a system enabling the operations to alert local people to the sudden flood. At present, many areas are affected by heavy rain due to weather change. As a result, floods and landslides have caused damage to people's lives as well as their property. Therefore, the concept of creating the alarming system by collecting data, analyzing the data and transferring the data to alert the people to the coming disaster is created. The data will be collected from many sources such as the movement of the earth and rainfall. These data will be stored in the Data Logger. After that, the Data Logger will categorize the data, record the time receiving the data and analyze the data. After processing the whole data, if there is any possibility of the flood, the Data Logger will transfer the data to warn the local people of a possibly dangerous flood as soon as possible. This will help mitigate the potential loss of life and property.

## กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. ขงยุทธ ชนบดีเฉลิมรุ่ง อาจารย์ที่ปรึกษาที่คอยให้คำปรึกษาและให้ความช่วยเหลือตลอดจนคำแนะนำต่างๆ ในการทำโครงการชิ้นนี้ สุดท้ายต้องขอขอบพระคุณอาจารย์ทุกท่านและเพื่อนๆ พี่ๆ ทุกคนที่ยังไม่ได้เอ่ยนามที่ให้คำแนะนำและให้การสนับสนุน ผู้จัดทำโครงการให้สามารถทำโครงการชิ้นนี้จนสำเร็จลุล่วงไปได้ด้วยดี

นายกฤษฎา วิวิตรกุล

นายชัยเดช ควรคิด

นายฐณวัฒน์ ธรรมากิมุข



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญรูป.....	จ
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตโครงการ.....	1
1.4 ขั้นตอนการดำเนินงานและแผนการดำเนินงานตลอดโครงการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ.....	2
1.6 งบประมาณ.....	3
บทที่ 2 หลักการและทฤษฎี	
2.1 การเชื่อมต่อกับ ET-USB FLASH DRIVER.....	4
2.2 การเชื่อมต่อกับไมโครคอนโทรลเลอร์.....	12
บทที่ 3 วิธีการออกแบบ	
3.1 ส่วนของภาคประมวลผลสัญญาณและเก็บข้อมูล.....	17
3.2 ส่วนของภาคส่งข้อมูลและรับข้อมูล.....	18
3.3 ลำดับการทำงานของอุปกรณ์ทั้งหมด.....	20
3.4 Flow Chart .....	21
3.5 แผนผังการทำงานของอุปกรณ์.....	22
3.6 ขั้นตอนการทำงานของระบบ.....	23
3.7 ลำดับการเชื่อมต่ออุปกรณ์.....	25
บทที่ 4 ผลการทดลอง	
4.1 ระบบการทำงาน.....	26
4.2 ผลการทดลอง.....	29

## สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการทดลอง	
5.1 สรุปผลการดำเนินงานโครงการ.....	33
5.2 ปัญหาและแนวทางแก้ไข.....	33
5.3 ข้อจำกัดของระบบ.....	33
5.4 ข้อเสนอแนะในการพัฒนาต่อไป.....	24



## สารบัญรูป

รูปที่	หน้า
2.1	ตำแหน่งขา DB9 Female (DCE) ของ ET-USB FLASH DRIVER.....6
2.2	ตำแหน่งขา DB9 male(DTE) ของ PC.....6
2.3	แสดงการต่อสายสัญญาณ RS232 ระหว่าง ET-USB FLASH DRIVER กับ PC แบบใช้ Handshake(CTS,RTS).....7
2.4	แสดงการต่อสายสัญญาณ RS232 ระหว่าง ET-USB FLASH DRIVER กับ MCU แบบไม่ใช้ Handshake(Jump ขา 7 CTS และขา 8 RTS เข้าด้วยกัน).....7
2.5	การ Set up โปรแกรมHyperTerminal ผ่านทางพอร์ต RS232.....8
2.6	การ Settings โดยเลือกที่ Tab settings แล้วเลือกที่ ASCII Setup.....9
2.7	การเซ็ทค่าในหน้าต่าง ASCII Setup.....9
2.8	ข้อความหลังจากทำการจ่ายไฟให้กับ ET-USB FLASH DRIVE .....10
2.9	แบบตัวถังของ ATmega8515L.....13
2.10	Pinout ATmega8515.....14
2.11	โครงสร้างของบอร์ด ATmega8515L.....14
2.12	แสดงโครงสร้างของบอร์ด ATmega8515L.....15
2.13	วงจร ET-MINI DS1307.....16
2.14	แสดงโครงสร้างของบอร์ด ET-MINI DS1307.....16
3.1	ET-USB FLASH DRIVE .....17
3.2	4-Channel Input Port.....18
3.3	ลักษณะโครงสร้างของแผงวงจร 4-Channel Input Port.....18
3.4	บอร์ดไมโครคอนโทรลเลอร์ ATmega8515L.....19
3.4	Block Diagram ของอุปกรณ์ทั้งหมด.....20
3.5	Flow Chart .....21
3.6	แผนผังการทำงานของอุปกรณ์.....22
3.7	ผังการต่ออุปกรณ์ทั้งหมด.....22
3.8	แผนผังการต่อวงจรภายในบอร์ด.....23
3.9	แผนผังการต่อวงจรออกไปยัง RS232.....24
3.10	การเชื่อมต่อระหว่าง Microcontroller กับ 4 Channel Input port.....25
3.11	การเชื่อมต่อระหว่าง Microcontroller กับ ET-USB Flash Drive.....25



## สารบัญรูป(ต่อ)

รูปที่	หน้า
3.12	การเชื่อมต่อและการส่งข้อมูลระหว่าง Microcontroller และ ET USB Flash Drive.....26
3.13	การเชื่อมต่อ Microcontroller กับ LCD เพื่อแสดงผลทางหน้าจอ.....26
3.14	ต่อ Adaptor เพื่อจ่ายไฟ DC 5V.....27
3.15	ต่อ Flash-Drive เข้ากับตัว Data-logger ทาง-USB port.....27
4.1	4-Channel Input Port.....28
4.2	บอร์ดไมโครคอนโทรลเลอร์ ATmega8515L.....29
4.3	Data Logger (ET USB Flash Drive).....29
4.4	การแสดงผลผ่านจอ LCD.....30
4.5	RS232 Output port.....30
4.6	การแสดงผลสถานะบนหน้าจอ LCD.....32
4.7	ข้อมูลที่ได้จาก Text file ในชื่อ DATALOG.txt ใน Data Logger.....33

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของโครงการ

ภัยธรรมชาติที่เกิดขึ้นในปัจจุบันนี้ มีแนวโน้มว่าโอกาสที่จะเกิดขึ้นโดยฉับพลันนั้นมีมากขึ้น เช่น ดินถล่ม น้ำหลาก เป็นต้น ทำให้มีผู้ได้รับความเสียหายจากภัยธรรมชาตินี้มากมายนับเนื่องจากไม่มีระบบเตือนภัยที่มีประสิทธิภาพที่จะสามารถแจ้งเตือนภัยล่วงหน้าได้ ดังนั้น จึงได้จัดทำโครงการนี้ขึ้นมาเพื่อเป็นการวิเคราะห์ข้อมูลที่ได้รับจากหลายๆทางไว้สำหรับวิเคราะห์ข้อมูลและสรุปผลแล้วทำการส่งข้อมูลที่วิเคราะห์นั้น ไปยังอุปกรณ์หรือเครื่องมือต่างๆ ที่ใช้ในการแจ้งเตือนภัยล่วงหน้าให้กับผู้คนที่อาศัยอยู่ในพื้นที่ที่เสี่ยงภัย เพื่อช่วยลดความเสียหายที่จะมาพร้อมกับภัยธรรมชาติ โครงการนี้จึงน่าจะช่วยลดความเสียหายที่จะเกิดขึ้นพร้อมกับภัยธรรมชาติในอนาคตได้

ดาต้าล็อกเกอร์เป็นอุปกรณ์ที่สามารถรับข้อมูลจากอินพุตที่อยู่ในรูปแบบที่หลากหลาย เช่น จากวิทยุรับส่ง, จากสัญญาณอนาล็อกหรือดิจิทัล หรือแม้กระทั่งสัญญาณอินพุตที่อยู่ในรูปแบบของแรงดันหรือกระแส เพื่อเก็บข้อมูลและนำมาวิเคราะห์ข้อมูล โครงการนี้จะมุ่งเน้นในการเก็บข้อมูลที่รับมาวิเคราะห์และสรุปผล โดยอาศัยไมโครคอนโทรลเลอร์เป็นตัวประมวลผลรับส่งสัญญาณ

### 1.2 วัตถุประสงค์ของโครงการ

- 1) เพื่อศึกษาการทำงานของดาต้าล็อกเกอร์และโปรแกรมที่สามารถนำมาประยุกต์ใช้งานได้
- 2) เพื่อศึกษาการรับส่งสัญญาณข้อมูลแบบอนาล็อกและแบบดิจิทัล
- 3) เพื่อศึกษาและพัฒนาอุปกรณ์ที่จะนำมาใช้งานในการรับส่งสัญญาณข้อมูลจากหลาย

รูปแบบ ซึ่งจะนำมาใช้ในการสรุปและวิเคราะห์ข้อมูล

### 1.3 ขอบเขตโครงการ

- 1) ศึกษาการรับส่งสัญญาณข้อมูลในรูปแบบอนาล็อกและแบบดิจิทัล
- 2) พัฒนาโปรแกรมที่ใช้ในดาต้าล็อกเกอร์และนำมาประยุกต์ต่อยอดในงานด้านอื่นๆ
- 3) สามารถจุและรับส่งข้อมูลได้ไม่ต่ำกว่า 1 GB. และสามารถประมวลผลได้

### 1.4 ขั้นตอนการดำเนินงานและแผนการดำเนินงานตลอดโครงการวิจัย

รายละเอียด	ปี 2551							ปี 2552		
	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1) รวบรวมข้อมูล	←→									
2) ศึกษาการทำงานของตัวเอง										
เครื่องรับและเครื่องส่งและโปรแกรมที่จะนำมาใช้งาน			←→							
3) จัดทำอุปกรณ์และทำการทดลองใช้งานอุปกรณ์								←→		
4) จัดทำรายงานและสรุปผลการทำงาน										←→

### 1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ

- 1) เข้าใจถึงหลักในการรับส่งสัญญาณในรูปแบบอนาล็อกและดิจิทัล
- 2) เข้าใจถึงหลักการเขียนโปรแกรมที่จะสามารถนำมาใช้งานในอนาคตได้
- 3) สามารถนำอุปกรณ์ของโครงการมาประยุกต์ใช้ในงานด้านอื่นๆ ได้
- 4) เพื่อเป็นประโยชน์ให้นิสิตที่จะใช้ศึกษาและต่อยอดโครงการนี้

**1.6 งบประมาณ**

1) ค่าเอกสารและค่าเช่าเล่มโครงการฉบับสมบูรณ์	500	บาท
2) ค่าอุปกรณ์ในการทำโครงการ	3000	บาท
3) ค่าหนังสือ	300	บาท
4) ค่าพิมพ์เอกสาร	200	บาท
รวมเป็นเงินทั้งหมด	<u>4000</u>	บาท

(สี่พันบาทถ้วน)



## บทที่ 2

# หลักการและทฤษฎี

จากแนวคิดที่จะทำการเก็บข้อมูลลงในคาร์ท้าลือกเกอร์โดยประกอบด้วยอุปกรณ์หลักๆ คือ คาร์ท้าลือกเกอร์ และ ตัวประมวลผลการตัดสินใจ โดยอุปกรณ์แต่ละตัวก็จะมีคุณสมบัติ หลักการทำงาน หรือแม้แต่การนำไปใช้งานที่แตกต่างกันไป ส่วนการเชื่อมต่อกับ ไมโครคอนโทรลเลอร์เพื่อที่จะเขียนโปรแกรมควบคุมลงไป ในคาร์ท้าลือกเกอร์ทำการเก็บข้อมูลที่ได้รับแล้วทำการประมวลผลออกไปยังภาคเอาต์พุตซึ่งสามารถประยุกต์ใช้งานได้หลากหลายรูปแบบ โดยมีหลักการและทฤษฎีพื้นฐานเกี่ยวกับคาร์ท้าลือกเกอร์ที่ควรจะต้องทราบดังนี้

ET-USB FLASH DRIVE คืออุปกรณ์ที่ใช้สำหรับอ่านเขียนไฟล์ข้อมูล,ลบไฟล์,สร้างไฟล์ และอื่นๆ ที่เก็บอยู่ใน FLASH DRIVE โดยใช้วิธีการส่ง Command ต่างๆ ผ่านทาง Serial Port ไปให้กับ ET-USB FLASH DRIVE เพื่อให้ติดต่อไปยังตัวเก็บข้อมูล Flash Drive โดยสามารถใช้งานร่วมกับ PC ผ่านทาง Port RS232 โดยใช้โปรแกรม Hyperterminal หรือ Procomm หรือ โปรแกรมอื่นๆที่สามารถสื่อสารผ่านทาง RS232ได้ เป็นตัวกลางในการรับและส่งคำสั่งของข้อมูลต่างๆ รวมทั้งแสดงผลการอ่านหรือการเขียนข้อมูลให้กับผู้ใช้ นอกจากนี้ก็ยังสามารถต่อใช้งานร่วมกับ ไมโครคอนโทรลเลอร์(MCU) แทน PC ก็ได้ โดยจะสื่อสารผ่านทาง UART Port ของ MCU

### 2.1 การเชื่อมต่อกับ ET-USB FLASH DRIVE

#### 2.1.1 คุณสมบัติของ ET-USB FLASH DRIVE

1. สามารถติดต่อกับตัวเก็บข้อมูล Flash Drive ที่มีโครงสร้างไฟล์แบบ FAT 12 ,FAT 16 หรือ FAT32 ได้
2. รองรับชื่อไฟล์ในรูปแบบ 8.3 คือชื่อไฟล์ไม่เกิน 8 ตัวอักษร นามสกุล 3 ตัวอักษร เช่น A1234567.txt
3. ในระบบ FAT 32 จะไม่รองรับชื่อไฟล์แบบยาว ถ้าชื่อไฟล์ยาวเกิน 8.3 จะแสดงชื่อไฟล์ให้เห็น เพียง 8.3
4. ควบคุมการอ่านเขียน Flash Drive โดยใช้การส่ง Command ผ่านทาง RS232
5. สามารถเลือก Baud Rate ในการติดต่อสื่อสารทาง RS232ได้
6. สามารถส่ง Command โดยใช้ PC หรือ MCU ได้
7. สามารถ สร้างและลบ ไฟล์ หรือ Directory ใน Flash Drive ได้

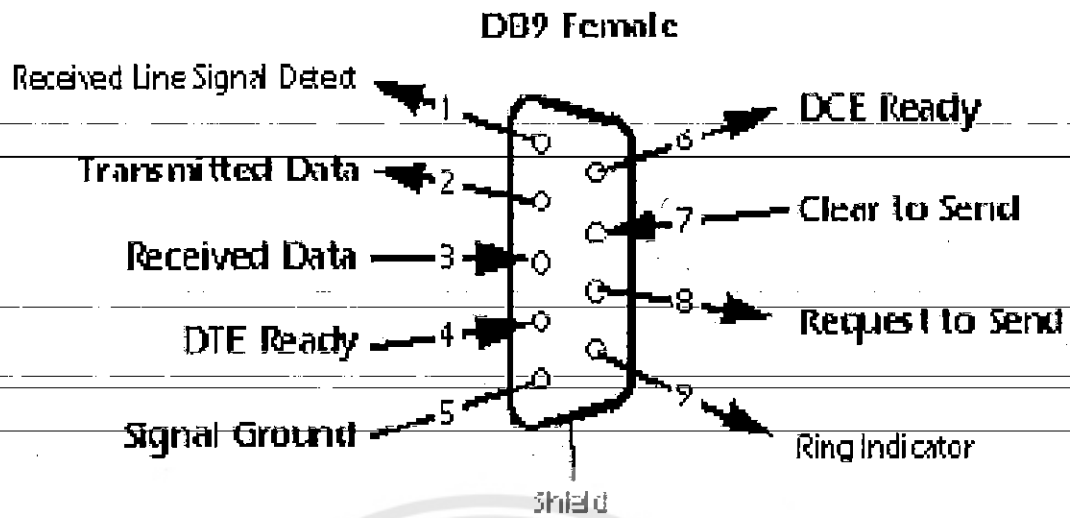
8. สามารถกำหนดจำนวน Byte ของข้อมูลที่จะทำการอ่านหรือเขียนจากไฟล์ที่อยู่ใน Flash Drive ได้
9. สามารถกำหนดตำแหน่งที่จะอ่านข้อมูลจากไฟล์ หรือเขียนข้อมูลลงไฟล์ ที่อยู่ใน Flash Drive ได้
10. สามารถอ่านข้อมูลออกมาทีละบิตทั้งไฟล์ จากไฟล์ที่อยู่ใน Flash Drive ได้
11. หลังจากปิด ไฟล์แล้ว สามารถเปิดไฟล์เดิมขึ้นมาทำการเขียนข้อมูลต่อจากของเดิม ได้ โดยข้อมูลเก่ายังอยู่
12. สามารถเปลี่ยนชื่อ ไฟล์หรือชื่อ Directory ใหม่ได้
13. สามารถเข้าไป อ่านเขียน สร้างหรือลบไฟล์ ที่อยู่ใน Directory ย่อยได้
14. สามารถเลือกรูปแบบการส่งคำสั่งได้ 2 แบบ คือ ส่งในรูปแบบอักขระ ASCII (Extended Mode) หรือส่งในรูปแบบ Hex เลขฐาน 16 (Short Mode)

### 2.1.2 การต่อใช้งาน ET-USB FLASH DRIVE

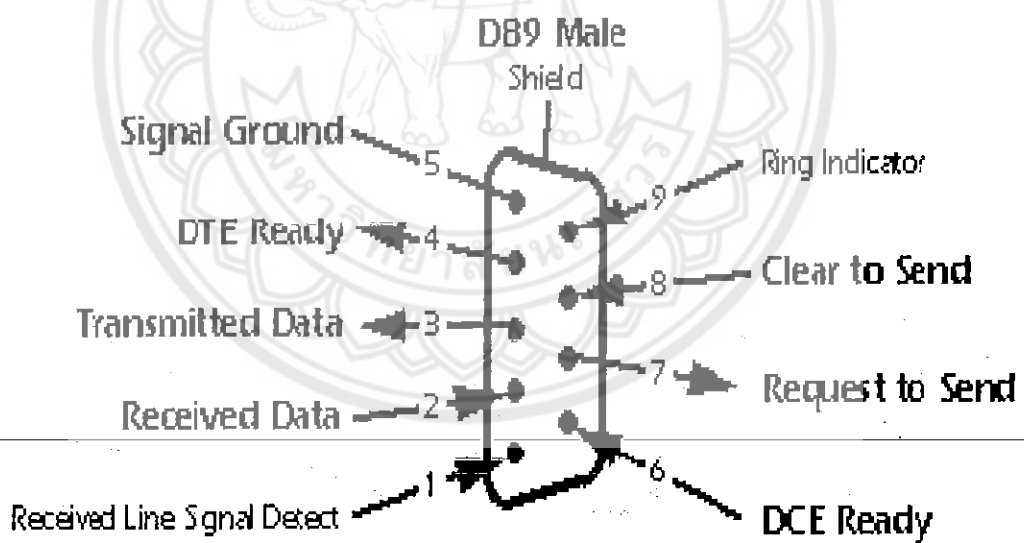
การใช้งาน ET-USB FLASH DRIVE จะใช้การ Interface ผ่านทาง Serial Port (RS232 หรือ Uart) โดยจะต้องกำหนดคุณสมบัติในการติดต่อสื่อสารทาง Serial Port ดังนี้

- Baud Rate จะต้องกำหนดเริ่มต้น default ไว้ที่ 9600 bit/s สามารถส่งคำสั่งเปลี่ยนแปลงได้ ในภายหลัง
- 8 Data bit , 1 stop bit และ No parity
- Flow Control : ให้กำหนดที่ Hardware ซึ่งก็คือ RTS/CTS จะต้องถูก Enable เพื่อใช้เป็น handshake
- ในกรณีที่ไม่ต้องการใช้ Handshake ใช้เพียงขา Rx(ขา3) และ Tx(ขา2) และ กราวด์(ขา5)

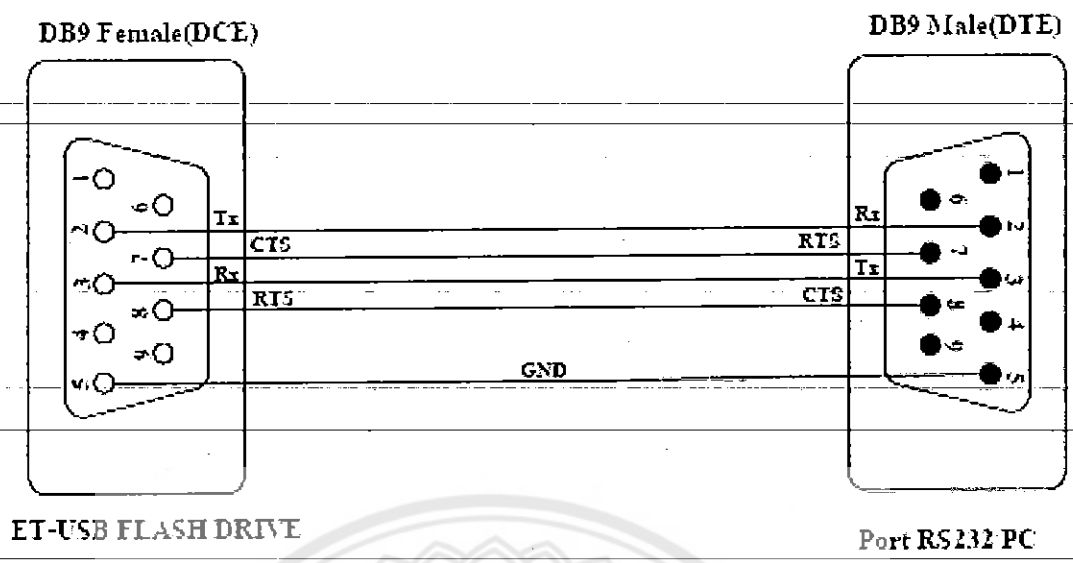
ในการติดต่อสื่อสารเท่านั้น ก็จะต้องทำการ Jump ขา-RTS(ขา-7) และ CTS (ขา-8) ที่ Port-DB9 ของ ET-USB FLASH DRIVE เข้าด้วยกัน จากนั้นก็ต่อขา Rx และ Tx ของ ET-USB FLASH DRIVE ไปยังขา Rx และ Tx ของอุปกรณ์ที่นำมาควบคุม โดยจะต้องต่อแบบไขว้เข้ากันคือ ต่อขา Rx เข้ากับ ขา Tx และต่อขา Tx เข้ากับ Rx ของอีกฝั่งหนึ่งส่วนกราวด์ให้ต่อเข้าด้วยกัน ดังแสดงในรูปที่ 2.4



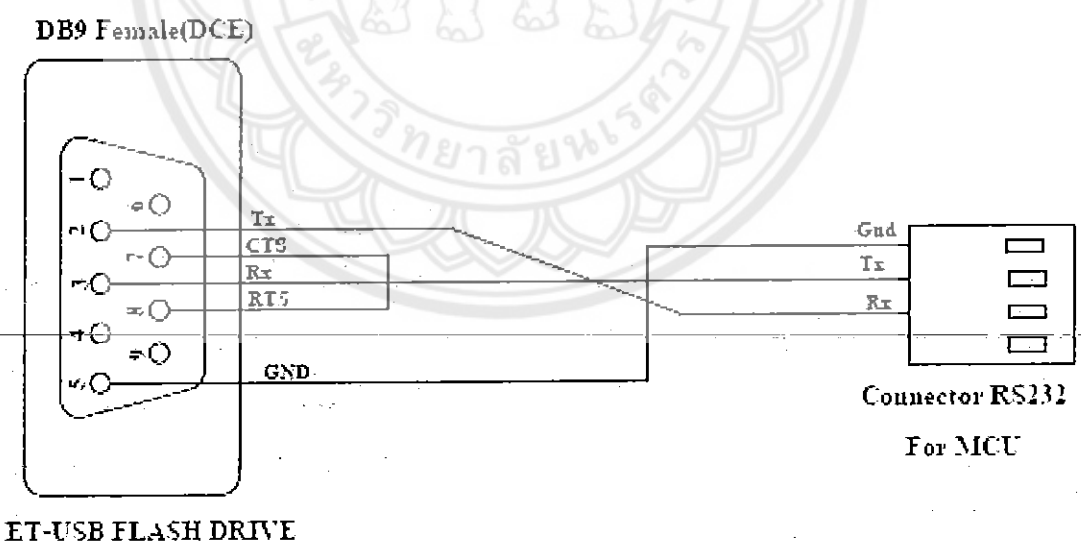
รูปที่ 2.1 ตำแหน่งขา DB9 Female (DCE) ของ ET-USB FLASH DRIVER



รูปที่ 2.2 ตำแหน่งขา DB9 male(DTE) ของ PC



รูปที่ 2.3 แสดงการต่อสายสัญญาณ RS232 ระหว่าง ET-USB FLASH DRIVER กับ PC แบบใช้ Handshake(CTS,RTS)

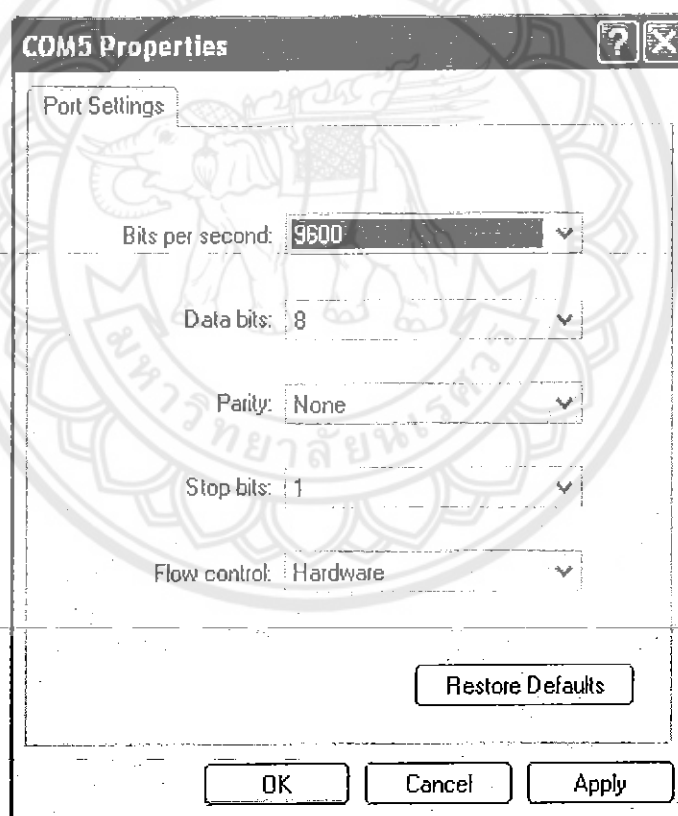


รูปที่ 2.4 แสดงการต่อสายสัญญาณ RS232 ระหว่าง ET-USB FLASH DRIVER กับ MCU แบบไม่ใช้ Handshake(Jump ขา 7 CTS และขา 8 RTS เข้าด้วยกัน)

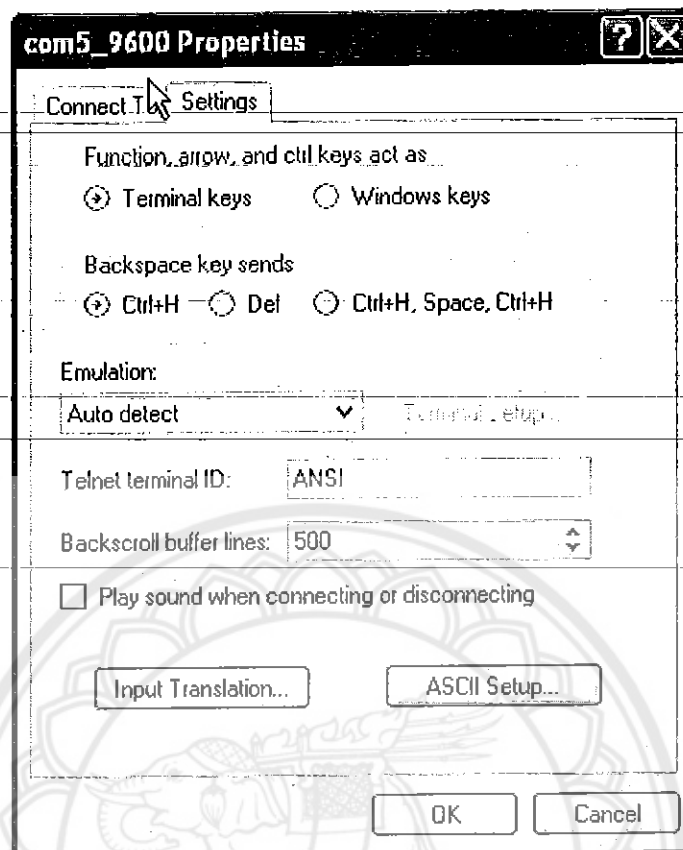


### 2.1.3 ขั้นตอนการทดสอบใช้งานกับ PC

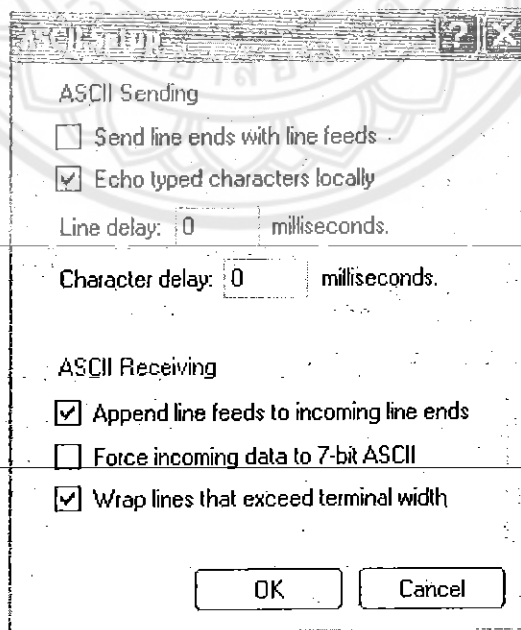
- 1) ต่อ Flash Drive เข้าที่ขั้วต่อ USB ของ ชุด ET-USB FLASH DRIVE
- 2) ต่อสาย RS232 จาก PC หรือ MCU เข้าที่ขั้วต่อ RS232 ของ ชุด ET-USB FLASH DRIVE
- 3) ถ้าส่งคำสั่งผ่านทาง PC ให้เปิดโปรแกรม HyperTerminal หรือโปรแกรมที่ใช้รับส่งข้อมูลผ่านทาง RS232 ขึ้นมารอไว้โดยให้ Set Up คุณสมบัติดังรูปที่ 2.5 จากนั้นให้กด Ok ก็จะได้หน้าจอ Hyperterminal ขึ้นมา จากนั้นให้คลิกที่ Icon Properties (-) จะได้นหน้าต่างออกมาดังในรูปที่ 2.6 ให้เลือกที่ TAB Setting และคลิกเลือกที่ปุ่ม ASCII Setup... จากนั้นก็จะได้นหน้าต่างในรูปที่ 3 ขึ้นมาให้ทำการกำหนดค่าตามในรูปที่ 2.7 จากนั้นให้กด Ok ในแต่ละหน้าต่างเป็นอันเรียบร้อยในการ Set การใช้งาน Hyperterminal



รูปที่ 2.5 การ Set up โปรแกรม HyperTerminal ผ่านทางพอร์ต RS232



รูปที่ 2.6 การ Settings โดยเลือกที่ Tab settings แล้วเลือกที่ ASCII Setup...



รูปที่ 2.7 การเซ็ทค่าในหน้าต่าง ASCII Setup

4.) จ่ายไฟ 7-12 VDC ให้กับชุด ET-USB FLASH DRIVE ให้สังเกตที่ LED Status สีเขียว จะติดค้างนั้นแสดงว่าตัว ET-USB FLASH DRIVE ถูกต่อเข้ากับ FLASH DRIVE และ อุปกรณ์ที่ใช้ สื่อสารทางค่าน RS232 เรียบร้อยพร้อมใช้งานแล้ว แต่ถ้า LED Status กระพริบสลับระหว่างสีเขียว และแดง แสดงว่า การเชื่อมต่อยังไม่สมบูรณ์ในขณะที่มีการอ่านเขียนข้อมูล LED สีเขียวจะ กระพริบ

5.) หลังจากจ่ายไฟเรียบร้อยแล้วให้รอกจนกว่าจะมีข้อความ :

```
Ver 09.55VDA PF On-Line:
Device Detected P2
No Upgrade
D:\>
```

### รูปที่ 2.8 ข้อความหลังจากทำการจ่ายไฟให้กับ ET-USB FLASH DRIVE แล้ว

ปรากฏขึ้นที่หน้าต่าง HyperTerminal และแสดง D:\> พร้อมทั้งจะรับคำสั่งต่างๆจากผู้ใช้เพื่อ ติดต่อใช้งาน USB FLASH DRIVE เมื่อพิมพ์คำสั่งเสร็จหรือพิมพ์คำสั่งผิดให้กด Enter เพื่อเริ่มต้น คำสั่งใหม่จะสังเกต D:\> จะขึ้นเสมอแสดงความพร้อมในการรับคำสั่ง โดยค่า default ของ ET-USB Flash Drive จะถูกกำหนดไว้ดังนี้คือ Baud Rate ในการสื่อสาร 9600 bit/s , รับคำสั่งในโหมด Extended Mode และ กำหนดให้มีการรับค่าหรือส่งผ่านค่าที่เป็นตัวเลขในแบบ Binary Mode (IPH) จากนั้นให้ลองทำการทดสอบการเขียนและอ่านไฟล์ ตามตัวอย่างในหัวข้อด้านล่าง

#### 2.1.4 ตัวอย่างการทดสอบ ET-USB FLASH DRIVE

ทดสอบกับโปรแกรม Hyper Terminal

1) ทดสอบการเขียนข้อมูลให้กับ ไฟล์ชื่อ test01.txt จำนวน 10 Byte

- ส่งคำสั่ง IPA เพื่อกำหนดรูปแบบการผ่านค่า จำนวน Byte ข้อมูลที่จะเขียน ให้กับ Monitor ใน ASCII Mode

- ส่งคำสั่ง OPW test01.txt เพื่อเปิดไฟล์สำหรับเขียน

- ส่ง คำสั่ง WRF 10 เพื่อเขียนไฟล์ โดยระบุจำนวน Byte ที่จะเขียน = 10 Byte แล้ว Enter

- ทำการเขียนไฟล์ 'abcdefghij' เมื่อครบ 10 Byte จะมี Response D:\> ส่งออกมาแสดงว่า เขียนข้อมูลครบแล้ว

- ส่งคำสั่ง CLF test01.txt เพื่อทำการปิดไฟล์ที่เขียน

D:\> [Prompt ใน Extended mode]

IPA ← [กำหนดรูปแบบการส่งจำนวน byte ที่จะเขียน ในแบบ ASCII

Mode]

D:\> [Response Prompt]

OPW test01.txt ↵ [ทำการ Open file ชื่อ test01.txt]

D:\> [Response Prompt]

WRF 10 ↵ [ส่งคำสั่งเขียน file โดยระบุจำนวน byte ที่จะเขียน 10 Byte]

abcdefghijkl [เขียน data 10 Byte]

D:\> [Response-Prompt จะแสดงอัตโนมัติเมื่อเขียนข้อมูลตัวที่ 10

เรียบร้อยแล้ว]

CLF test01.txt ↵ [ส่งคำสั่งปิด test01.txt ที่ได้เปิดเขียนไว้]

D:\> [Response-Prompt สิ้นสุดการเขียน file]

2) ทดสอบการอ่านข้อมูลจากไฟล์ชื่อ test01.txt ออกมาจำนวน 5 Byte ซึ่งค่าที่จะต้องอ่าน  
ได้คือ abcde

- ส่งคำสั่ง IPA เพื่อกำหนดรูปแบบการผ่านค่า จำนวน Byte ข้อมูลที่จะอ่าน ให้กับ Monitor ใน ASCII Mode
- ส่งคำสั่ง OPR test01.txt เพื่อเปิดไฟล์สำหรับอ่าน
- ส่ง คำสั่ง RDF 5 เพื่ออ่านไฟล์โดยระบุจำนวน Byte ที่จะอ่าน = 5 Byte แล้ว Enter
- ข้อมูลจะถูกอ่านออกมา 5 byte คือ abcde โดยข้อมูลที่อ่าน ได้นี้จะถูกนำด้วยค่า 0x0D

ส่งออกมาก่อนแล้วจึงตามด้วยข้อมูล 5 byte และปิดด้วยท้ายข้อมูลด้วย D:\>

D:\> [Prompt ใน Extended mode]

IPA ↵ [กำหนดรูปแบบการส่งจำนวน byte ที่จะอ่าน ในแบบ ASCII Mode]

D:\> [Response Prompt]

OPR test01.tx ↵ [ส่งคำสั่ง Open file ชื่อ test01.txt]

D:\> [Response Prompt]

RDF 5 ↵ [ส่งคำสั่งอ่าน file ที่เปิดอยู่ โดยระบุจำนวน byte ที่จะอ่าน 5 Byte]

↵ abcdeD:\> [ data ถูกอ่านออกมา 5 Byte โดยข้อมูลจะถูกนำด้วย 0x0D และ ปิดด้วย <prompt>]

↵ [ส่งคำสั่ง enter เพื่อรอรับคำสั่งต่อไป]

D:\> [Response <prompt>]

## 2.2 การเชื่อมต่อกับไมโครคอนโทรลเลอร์

### 2.2.1 คุณสมบัติของ ATmega8515L

#### • RISC Architecture

- 130 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

#### • Nonvolatile Program and Data Memories

- 8K Bytes of In-System Self-programmable Flash  
Endurance: 10,000 Write/Erase Cycles
- Optional Boot Code Section with Independent Lock bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
- 512 Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles
- 512 Bytes Internal SRAM
- Up to 64K Bytes Optional External Memory Space
- Programming Lock for Software Security

#### • Peripheral Features

- One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Three PWM Channels...
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

• **Special Microcontroller Features**

- Power-on-Reset and Programmable-Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Three Sleep Modes: Idle, Power-down and Standby

• **I/O and Packages**

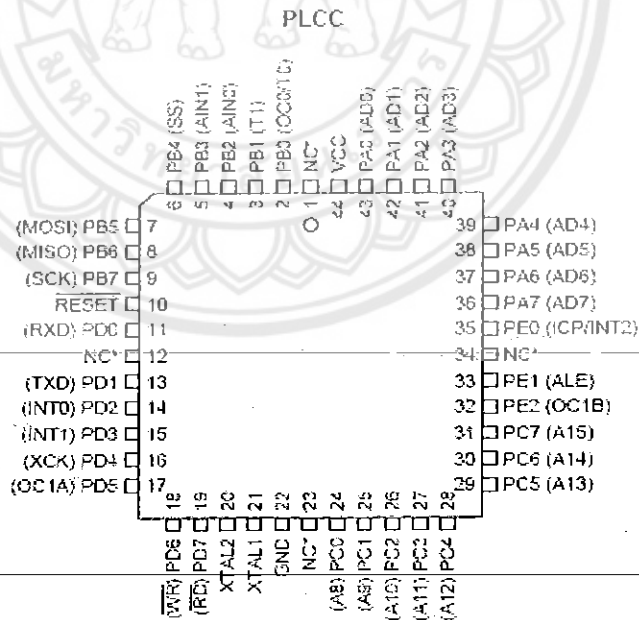
- 35 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF

• **Operating Voltages**

- 2.7 - 5.5V for ATmega8515L
- 4.5 - 5.5V for ATmega8515

• **Speed Grades**

- 0 - 8 MHz for ATmega8515L
- 0 - 16 MHz for ATmega8515



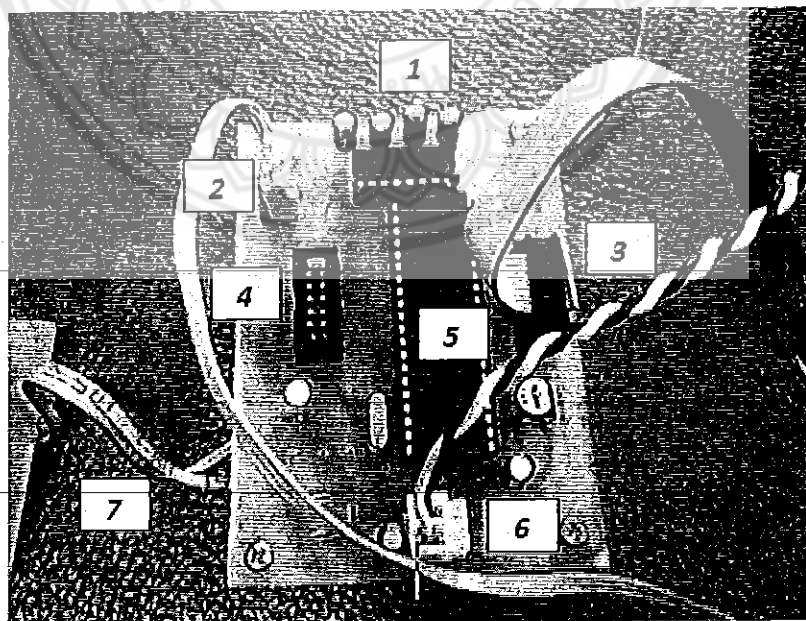
รูปที่ 2.9 แบบตัวถังของ ATmega8515L

PDIP

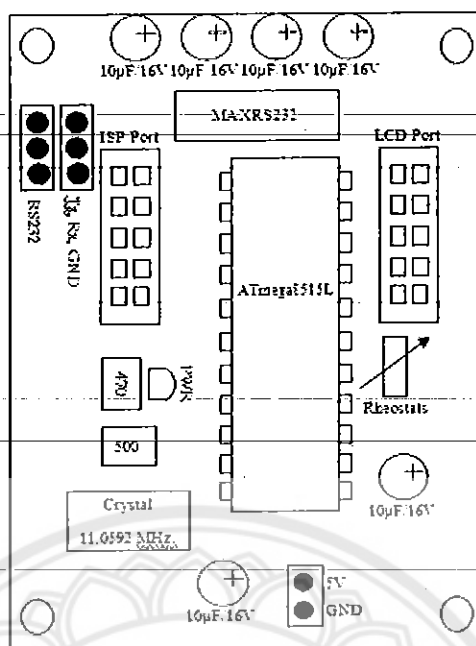
(OC0/T0) PB0	1	40	VCC
(T1) PB1	2	39	PA0 (AD0)
(AIN0) PB2	3	38	PA1 (AD1)
(AIN1) PB3	4	37	PA2 (AD2)
(SS) PB4	5	36	PA3 (AD3)
(MOSI) PB5	6	35	PA4 (AD4)
(MISO) PB6	7	34	PA5 (AD5)
(SCK) PB7	8	33	PA6 (AD6)
RESET	9	32	PA7 (AD7)
(RXD) PDC	10	31	PE0 (ICP/INT2)
(TDX) PD1	11	30	PE1 (ALE)
(INT0) PD2	12	29	PE2 (OC1B)
(INT1) PD3	13	28	PC7 (A15)
(XCK) PD4	14	27	PC6 (A14)
(OC1A) PD5	15	26	PC5 (A13)
(WR) PD6	16	25	PC4 (A12)
(RD) PD7	17	24	PC3 (A11)
XTAL2	18	23	PC2 (A10)
XTAL1	19	22	PC1 (A9)
GND	20	21	PC0 (A8)

รูปที่ 2.10 Pinout ATmega8515

### 2.2.2 โครงสร้างของบอร์ด ATmega8515L



รูปที่ 2.11 โครงสร้างของบอร์ด ATmega8515L



รูปที่ 2.12 แสดงโครงสร้างของบอร์ด ATmega8515L

#### ส่วนที่ 1 : IC เมอร์ MAX RS232

- ทำหน้าที่ขยายพอร์ตการใช้งานของ RS232 ให้เป็นแบบ Multichannel

#### ส่วนที่ 2 : RS232 Output port

- ในส่วนนี้ได้ทำการเชื่อมต่อสาย RS232 ออกเป็น 2 ช่องทาง คือ
  - ช่องทางที่ 1 ส่งออกไปยัง ET USB Flash Drive เพื่อทำการเก็บข้อมูล Input ที่ได้รับ
  - ช่องทางที่ 2 ส่งออกไปยัง ภาครับ ที่ต้องการจะทำการส่งข้อมูลออกไป เช่น ต่อกับบอร์ดไมโครคอนโทรลเลอร์ตัวอื่นๆ

#### ส่วนที่ 3 :LCD Output port

- เป็น port ที่ต่อเข้ากับ LCD ขนาด 16x4 แสดงผลข้อมูลที่ได้รับจาก Input ต่าง

#### ส่วนที่ 4 : AVR ISP port

- เป็น port ที่ใช้ในการ burn ข้อมูลลงไปใน Chipset ของ Chip ในตระกูล ATmega โดยใช้วิธีการ burn ข้อมูลแบบ ISP

#### ส่วนที่ 5 : ATmega8515L Chipset

- เป็น Chipset หลักที่ใช้ในบอร์ดนี้

#### ส่วนที่ 6 : แหล่งจ่ายไฟ

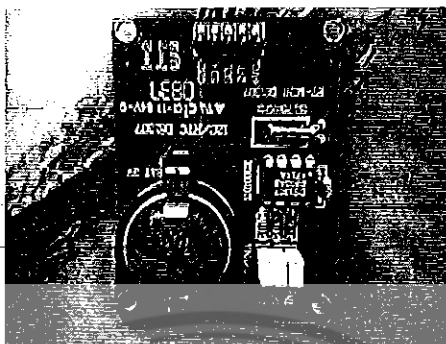
- สำหรับแหล่งจ่ายไฟของบอร์ดนี้สามารถต่อใช้งานได้ทั้งไฟกระแสตรงและกระแสสลับ โดยป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดัน 5 V ให้กับบอร์ด ซึ่งต่อกับขั้ว Connector แบบ CPA ขนาด 2 ขา



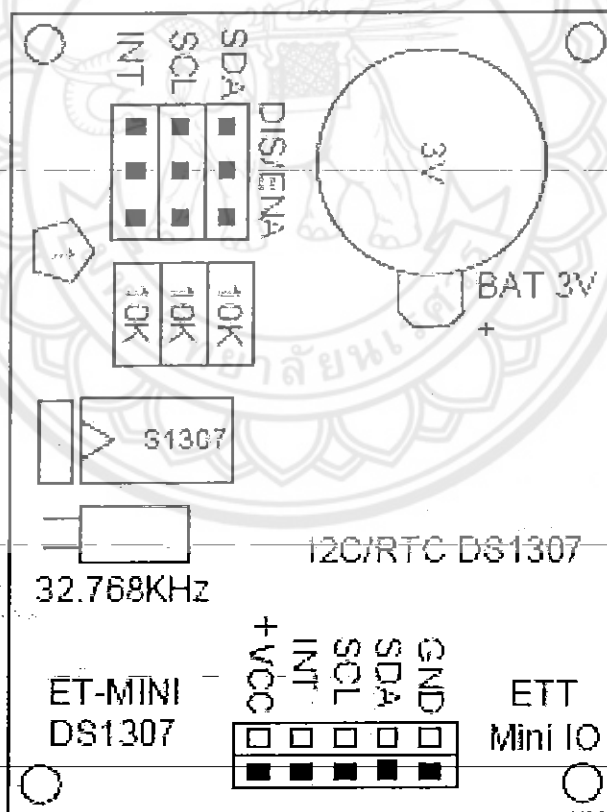
**ส่วนที่ 7 : Input port**

- ถูกต่อออกมาจาก IC เพื่อทำเป็นตัวรับ Input แบบ 4 Channel

**ส่วนที่ 8 : วงจร Clock**



**รูปที่ 2.13** วงจร ET-MINI DS1307



**รูปที่ 2.14** แสดง โครงสร้างของบอร์ด ET-MINI DS1307

- เป็นวงจรควบคุมวันและเวลาของบอร์ด ใช้ในการบันทึกข้อมูลลงไปใน Flash Drive
- สามารถตั้งค่าเวลา วัน/เดือน/ปี ได้ตรงตามเวลาที่แท้จริง

## บทที่ 3

# วิธีการออกแบบ

จากที่มาและหลักการต่างๆของการริเริ่มที่จะทำตัวเก็บข้อมูลแบบดาต้าล็อกเกอร์ เพื่อทำการเก็บข้อมูลที่ได้จากอินพุทที่สามารถประยุกต์ใช้งานได้หลายรูปแบบเช่น ตัวตรวจจับปริมาณน้ำฝน หรือ ตัวตรวจจับปริมาณความชื้นในอากาศ เป็นต้น นำมาวิเคราะห์ข้อมูลและสรุปผล เพื่อนำข้อมูลที่วิเคราะห์แล้วนั้น ส่งต่อไปยังภาคเอาต์พุทซึ่งสามารถนำข้อมูลที่ได้จากดาต้าล็อกเกอร์นำไปประยุกต์ใช้ในงานต่างๆ ในลำดับขั้นต่อไป ในส่วนของบทนี้จะกล่าวถึงวิธีการออกแบบและการจัดวางของอุปกรณ์ ET-USB FLASH DRIVE ซึ่งรายละเอียดต่างๆจะแสดงไว้ดังนี้

### การออกแบบและจัดวางอุปกรณ์

อุปกรณ์ที่ใช้ในการทดลอง ประกอบไปด้วยส่วนต่างๆ 2 ส่วนด้วยกัน คือ

#### 3.1 ส่วนของภาคประมวลผลสัญญาณและเก็บข้อมูล

ET-USB FLASH DRIVE นำมาใช้งานประยุกต์เป็นดาต้าล็อกเกอร์ สามารถใช้คำสั่งในการอ่าน, เขียนไฟล์ข้อมูล, ลบไฟล์, สร้างไฟล์ และคำสั่งอื่นๆ ในการรับข้อมูลมาจากอินพุท โดยสามารถต่อใช้งานกับเครื่องคอมพิวเตอร์ PC หรือ บอร์ดไมโครคอนโทรลเลอร์ ต่างๆ และยังสามารถนำมาประยุกต์ใช้งานแบบอื่นๆ ได้ เช่น งานระบบ GPS บันทึกข้อมูล, ระบบตรวจอากาศ, บันทึกข้อมูล, ฯลฯ โดยตัวอย่างของอุปกรณ์จะแสดงได้ดังรูปที่ 3.1

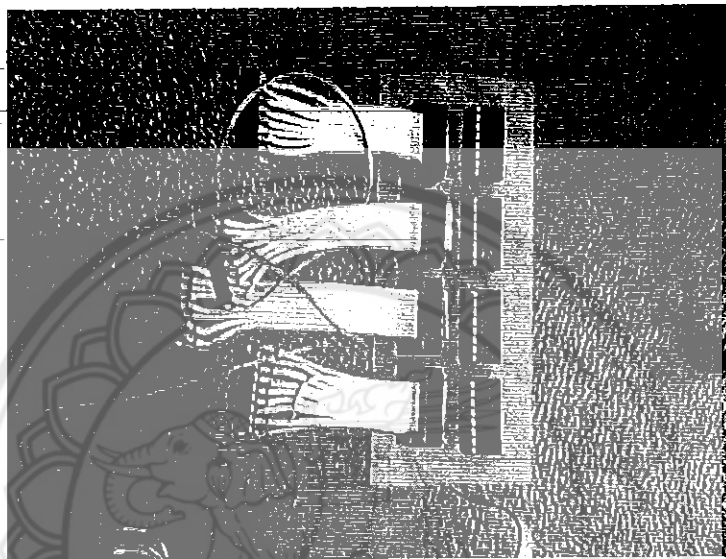


รูปที่ 3.1 ET-USB FLASH DRIVE

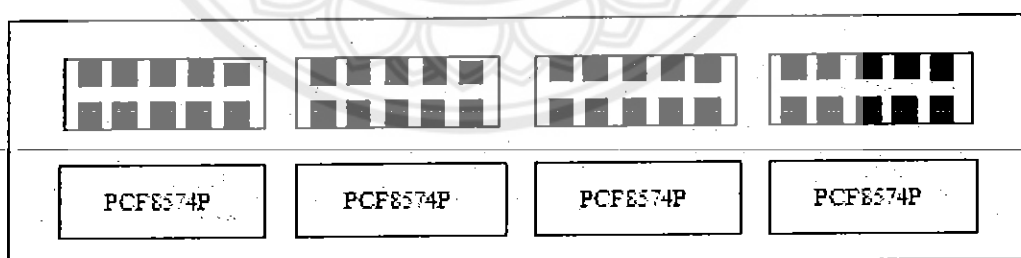
### 3.2 ส่วนของภาคส่งข้อมูลและรับข้อมูล

#### 3.2.1 ภาครับข้อมูล

ในส่วนของภาครับข้อมูล จะใช้ตัวรับสัญญาณ Input ที่จัดทำขึ้นมา กำหนดให้รับ Input ได้ 4 Channel โดยใช้ IC ของ Phillips เบอร์ PCF8574P เป็นตัวขยาย I/O port ขนาด 8 bit สำหรับ IC ใช้ทั้งหมด 4 port โดยต่อเข้ากับตัว test สัญญาณ Input สมมติ



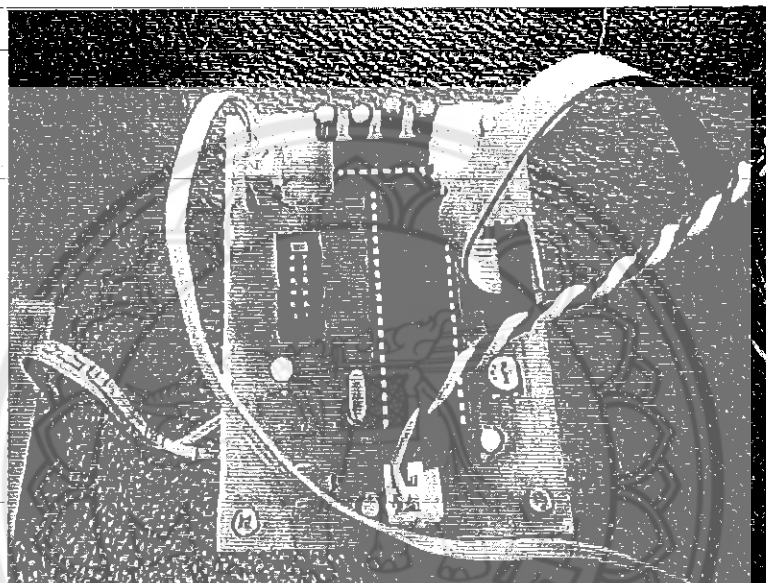
รูปที่ 3.2 4-Channel Input Port



รูปที่ 3.3 ลักษณะ โครงสร้างของแผงวงจร 4-Channel Input Port

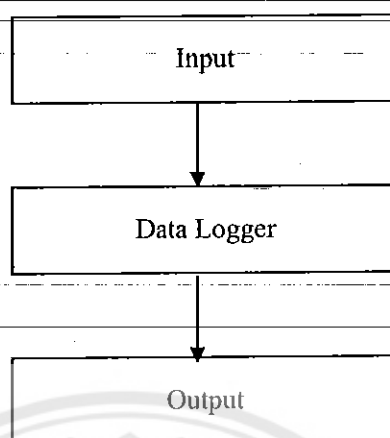
### 3.2.2 ภาคส่งข้อมูล

ในส่วนของภาคส่งข้อมูล จะใช้ ไมโครคอนโทรลเลอร์รุ่น ATmega8515L ที่จัดทำขึ้นมาเองเพื่อโครงการนี้โดยเฉพาะ มีหลักการทำงานดังนี้คือ เมื่อได้รับสัญญาณภาคอินพุตแล้ว ตัวของบอร์ดไมโครคอนโทรลเลอร์ จะทำการส่งสัญญาณอินพุตที่ได้รับออกไปยังภาคเอาต์พุตเพื่อส่งข้อมูลออกไปใช้งานต่อและจัดเก็บข้อมูลลงใน Data Logger อีกช่องทาง โดยผ่านทาง Port RS232 ที่ใช้ในการเชื่อมต่อ



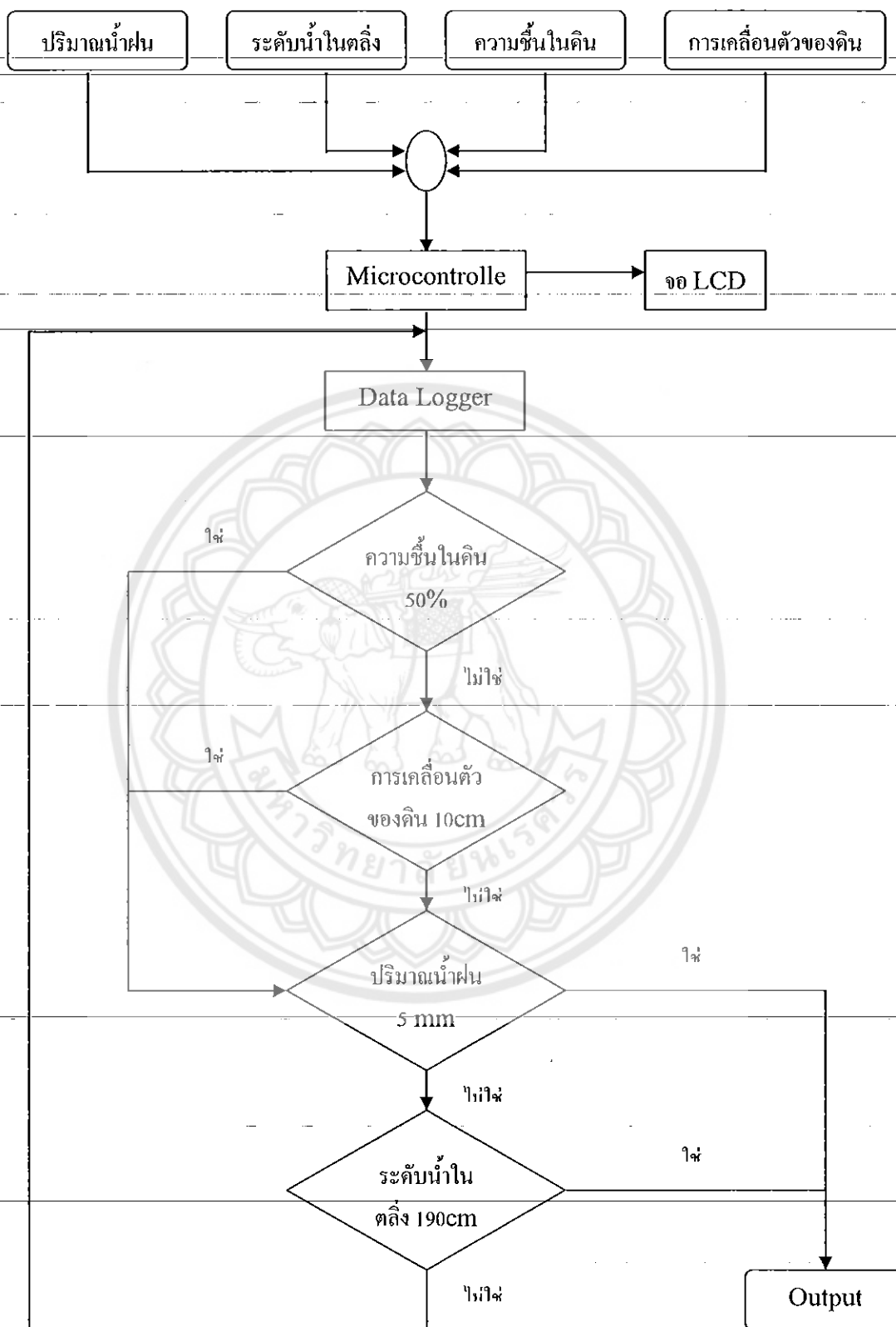
รูปที่ 3.3 บอร์ดไมโครคอนโทรลเลอร์ ATmega8515L

### 3.3 ลำดับการทำงานของอุปกรณ์ทั้งหมด



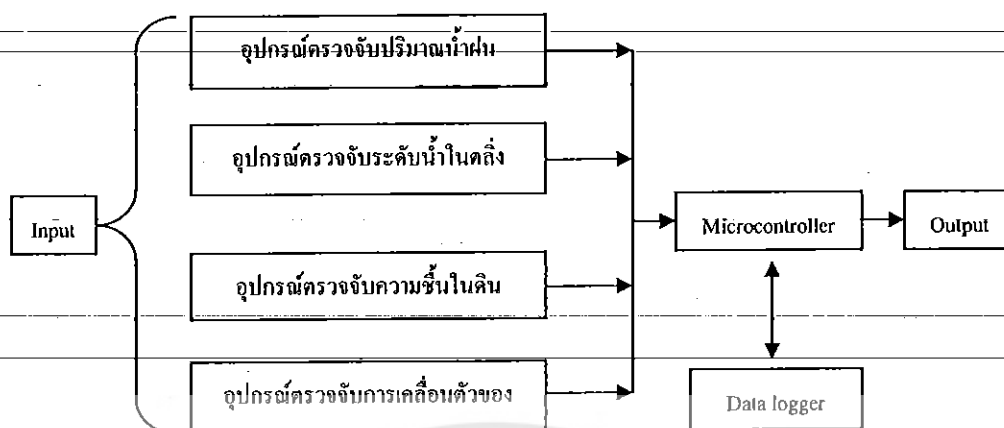
รูปที่ 3.4 Block Diagram ของอุปกรณ์ทั้งหมด

- ในบล็อกของอินพุตนั้น สามารถรับข้อมูลได้หลากหลายช่องทาง เช่น จากวิทยุรับส่ง, จากสัญญาณอนาล็อกหรือดิจิทัล หรือแม้กระทั่งสัญญาณอินพุตที่อยู่ในรูปแบบของแรงดันหรือกระแส โดยการรับสัญญาณเหล่านี้เข้ามา อาจจะทำช่องทางของสัญญาณแยกออกเป็นแต่ละช่องสัญญาณ เพื่อกระจายข้อมูลเข้าสู่คาตาด็อกเกอร์ในแต่ละช่องทาง หรือ อีกวิธีคือ การรวมสัญญาณอินพุตที่เข้ามาหลายช่องทาง นำมารวมกันให้ผ่านพอร์ตเดียวกัน โดยอาจจะใช้ตัวของ ไมโครคอนโทรลเลอร์เป็นตัวกลางคอยเชื่อมต่อข้อมูลหลายๆช่องทาง รวมไว้ในพอร์ตเข้าด้วยกันเพื่อส่งข้อมูลเข้าไปเก็บไว้ในคาตาด็อกเกอร์
- คาตาด็อกเกอร์จะทำการเก็บข้อมูลอินพุตไว้เป็นระยะๆ ขึ้นอยู่กับว่า จะทำการกำหนดช่วงเวลาในการบันทึกค่าตามช่วงเวลาที่เหมาะสม ซึ่งขึ้นอยู่กับความละเอียดของการใช้งานในแต่ละสถานการณ์ว่าต้องการข้อมูลที่มีความถี่มากน้อยเพียงใด ต้องคำนึงถึงเนื้อที่ในการเก็บข้อมูลที่เข้ามาในแต่ละคาบเวลา เพราะมีข้อมูลที่ไม่เกิดประโยชน์รวมอยู่ในการบันทึกค่าของอินพุต ดังนั้น ภาคเอาต์พุตมีอุปกรณ์ที่ประยุกต์ใช้งานได้หลากหลายรูปแบบ ตัวอย่างเช่น ตัวแจ้งเตือนภัยทางโทรศัพท์, ตัวประมวลผลการตัดสินใจ ฯลฯ แล้วแต่ว่าผู้ใช้จะกำหนดให้ภาคเอาต์พุตรับสัญญาณที่มาจากคาตาด็อกเกอร์แล้วนำข้อมูลนั้นๆ มาประยุกต์ใช้งานในด้านอื่นๆต่อไป

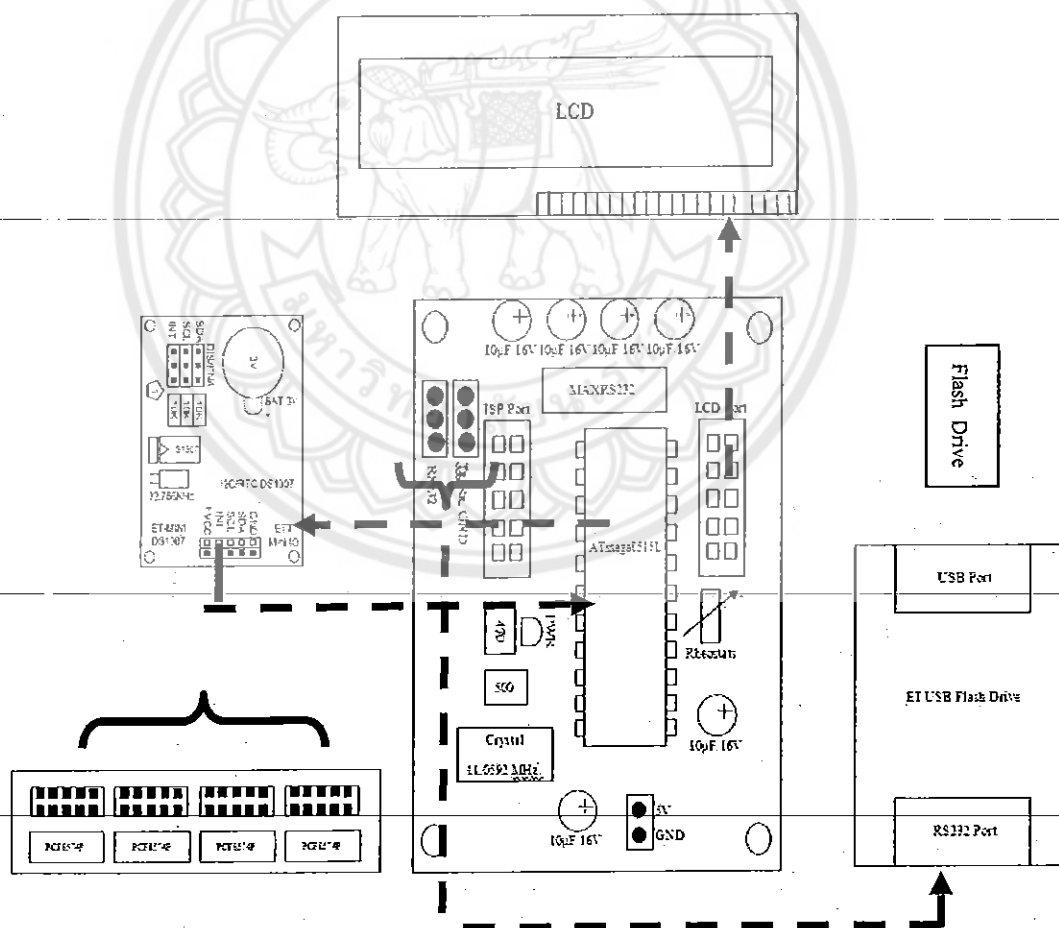


รูปที่ 3.5 Flow Chart

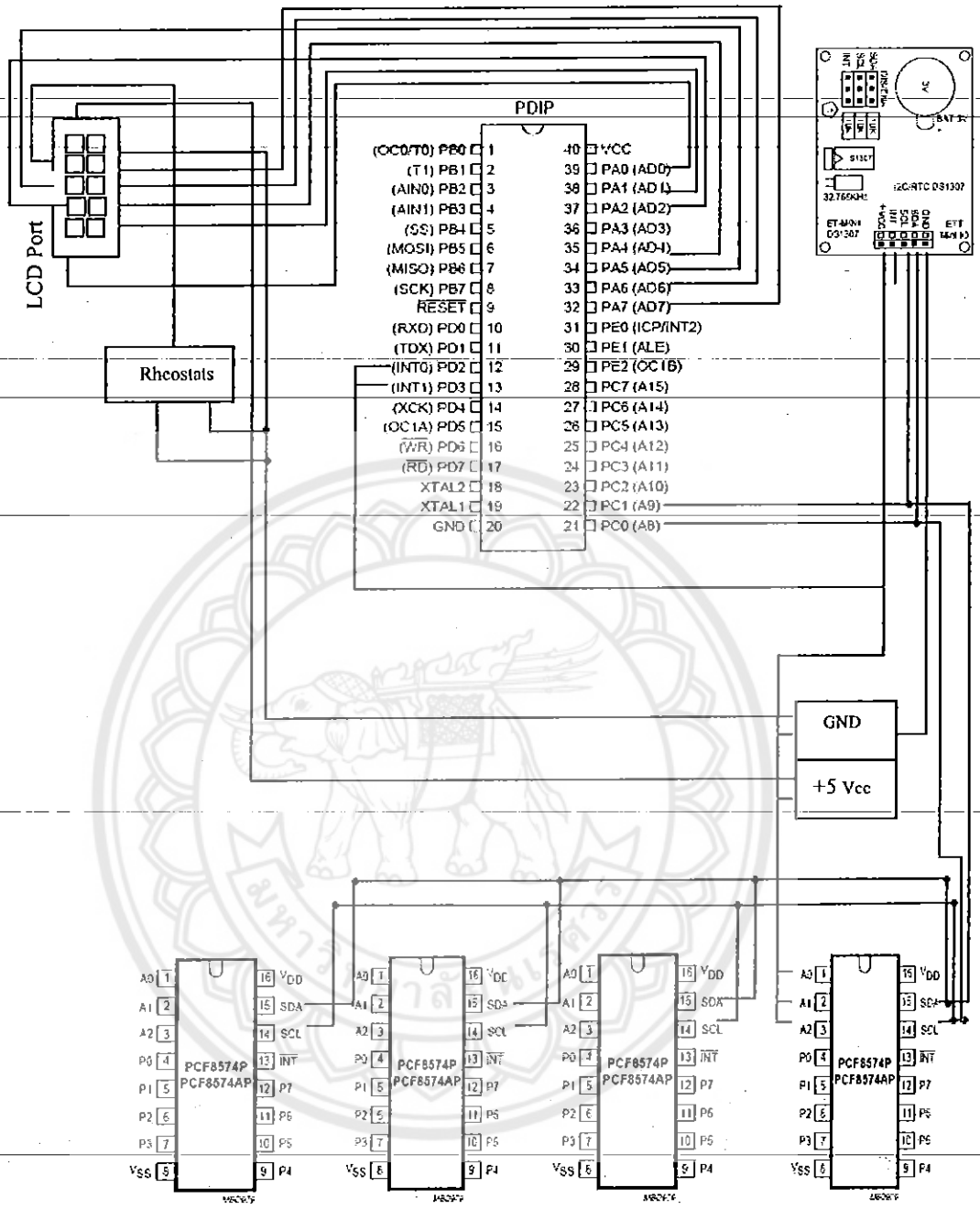
### 3.5 แผนผังการทำงานของอุปกรณ์



รูปที่ 3.6 แผนผังการทำงานของอุปกรณ์

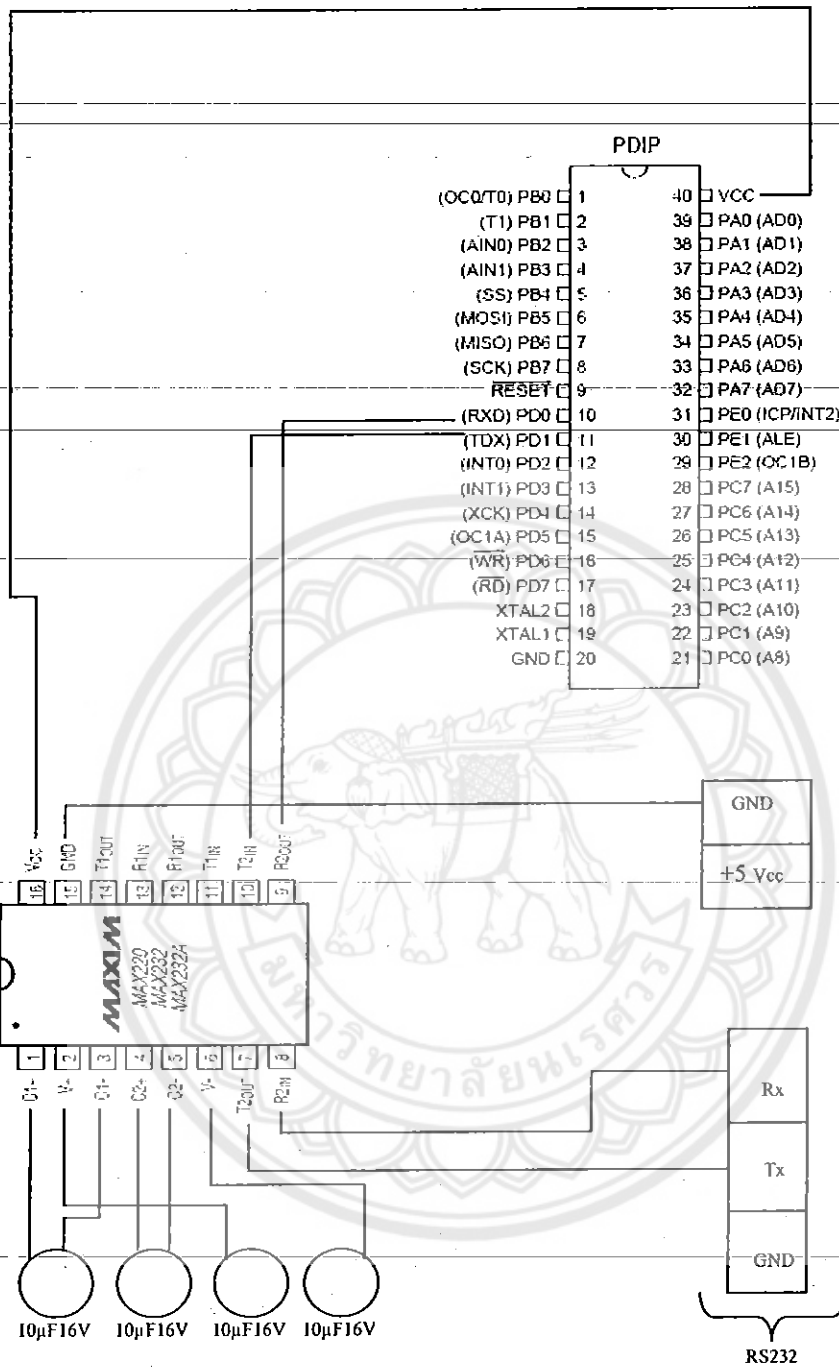


รูปที่ 3.7 ผังการต่ออุปกรณ์ทั้งหมด



รูปที่ 3.8 แผนผังการต่อวงจรภายในบอร์ด



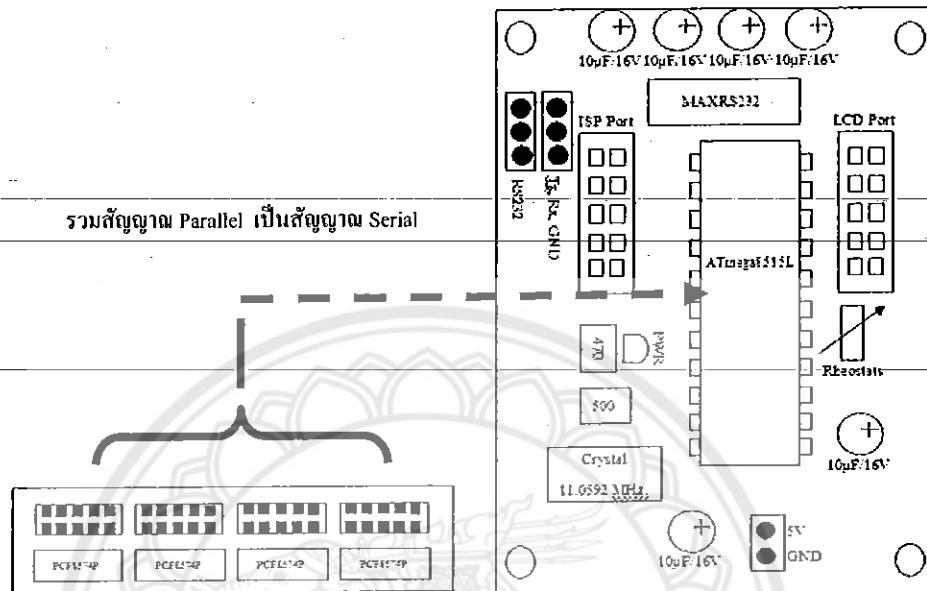


รูปที่ 3.9 แผนผังการต่อวงจรออกไปยัง RS232

### 3.6 ขั้นตอนการทำงานของระบบ

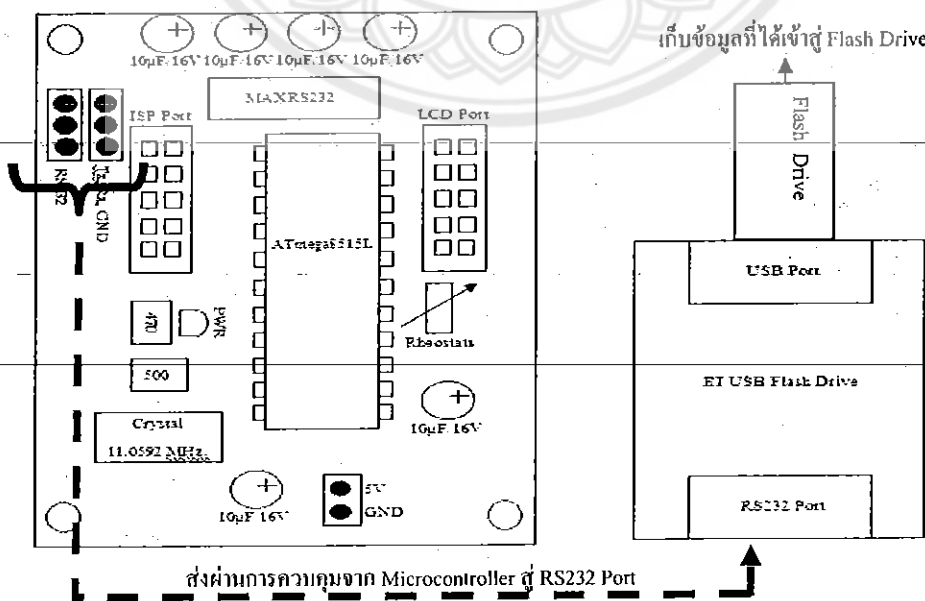
**ช่วงที่ 1** – เป็นขั้นตอนของการรับ Input ทั้ง 4 Channel เข้ามาจากอุปกรณ์ตรวจวัดที่อยู่ตามจุดวัด

ค่าต่างๆ โดยทำการแยกข้อมูลออกเป็นชุดๆ เพื่อให้ง่ายต่อการจัดเก็บข้อมูล



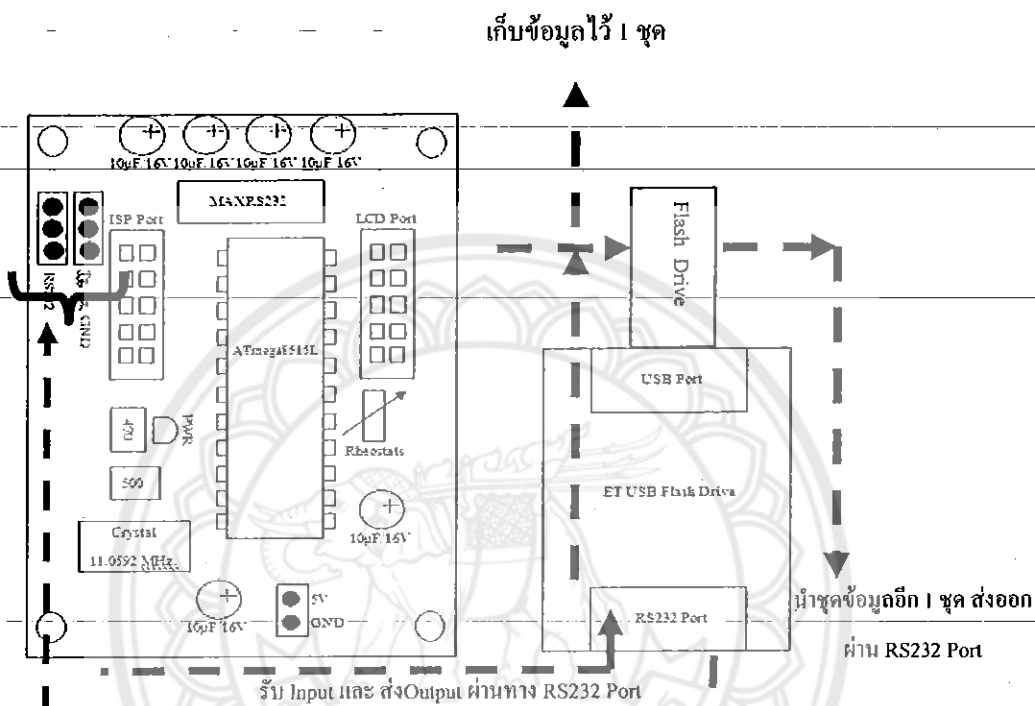
รูปที่ 3.10 การเชื่อมต่อระหว่าง Microcontroller กับ 4 Channel Input port

**ช่วงที่ 2** – เป็นขั้นตอนของการส่งข้อมูลผ่าน Microcontroller เพื่อให้ข้อมูลที่ได้รับ ถูกจัดเก็บไว้ใน Data Logger ใช้ในการส่งข้อมูลเมื่อถึงเกณฑ์ที่กำหนดไว้ ก็จะทำการส่งข้อมูลออกไปยังภาค Output ได้นำไปใช้งานต่อไป



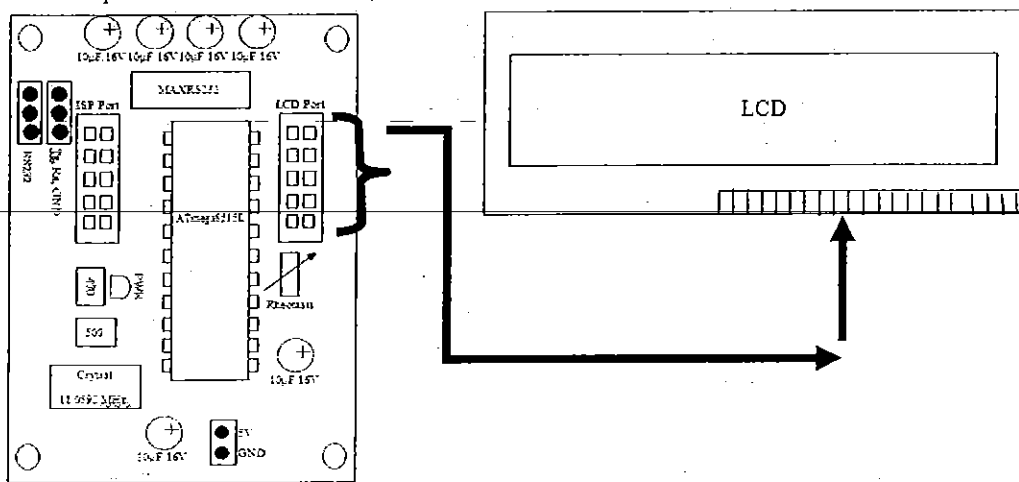
รูปที่ 3.11 การเชื่อมต่อระหว่าง Microcontroller กับ ET-USB Flash Drive

**ช่วงที่ 3** – เป็นขั้นตอนของการจัดเก็บข้อมูลลงใน Data Logger โดยทำการแยกเก็บข้อมูลไว้  
 อย่างละ 2 ชุด โดย ชุดแรกจะเก็บข้อมูลไว้ใน Data Logger จะทำการบันทึกเวลาที่ได้รับ  
 ข้อมูลจาก Input ไว้ ส่วนชุดที่ 2 จะทำการส่งข้อมูล Input ที่ได้รับ ออกไปยังภาค Output  
 โดยใช้เงื่อนไขของการตัดสินใจที่ได้กำหนดเอาไว้ใน Microcontroller



**รูปที่ 3.12** การเชื่อมต่อและการส่งข้อมูลระหว่าง Microcontroller และ ET USB Flash Drive

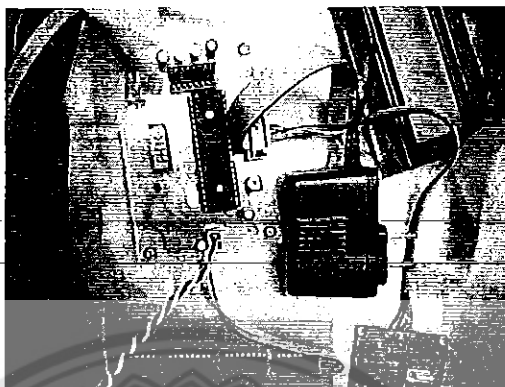
**ช่วงที่ 4** – เป็นขั้นตอนของภาค Output ที่จะทำการส่งข้อมูลออกไปยังอุปกรณ์ต่างๆที่เข้ามารองรับ  
 Input โดยจะมีการแสดงผลผ่านจอ-LCD



**รูปที่ 3.13** การเชื่อมต่อ Microcontroller กับ LCD เพื่อแสดงผลทางหน้าจอ

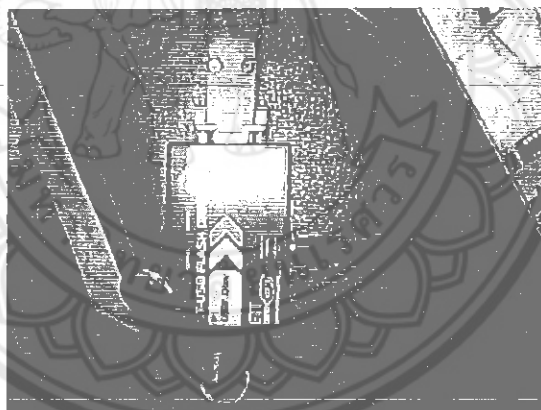
### 3.7 ลำดับการเชื่อมต่ออุปกรณ์

1. ต่อหม้อแปลง Adaptor เพื่อจ่ายไฟ DC 5 V เข้ากับบอร์ด Microcontroller



รูปที่ 3.14 ต่อ Adaptor เพื่อจ่ายไฟ DC 5V

2. ต่อ Flash Drive เข้ากับตัว Data logger ทาง USB port



รูปที่ 3.15 ต่อ Flash Drive เข้ากับตัว Data logger ทาง USB port

3. เชื่อมต่อตัว Data logger เข้ากับบอร์ด Microcontroller โดยผ่านทาง RS232 port
4. ต่อไฟเลี้ยงให้กับตัว Data logger จะสังเกตเห็นว่า หลอด LED จะกระพริบเป็นสีแดงและเขียวอย่างต่อเนื่องสักครู่ หลังจากหลอดนั้นเป็นไฟสีเขียวแล้วแสดงว่าการเชื่อมต่อระหว่าง Flash Drive และ Data logger สำเร็จ หลังจากนั้นให้สังเกตที่จอ LCD จะเห็นตัวแสดงสถานะขึ้นว่า Conn แสดงว่ากำลังเชื่อมต่อเพื่อรอรับข้อมูล

## บทที่ 4

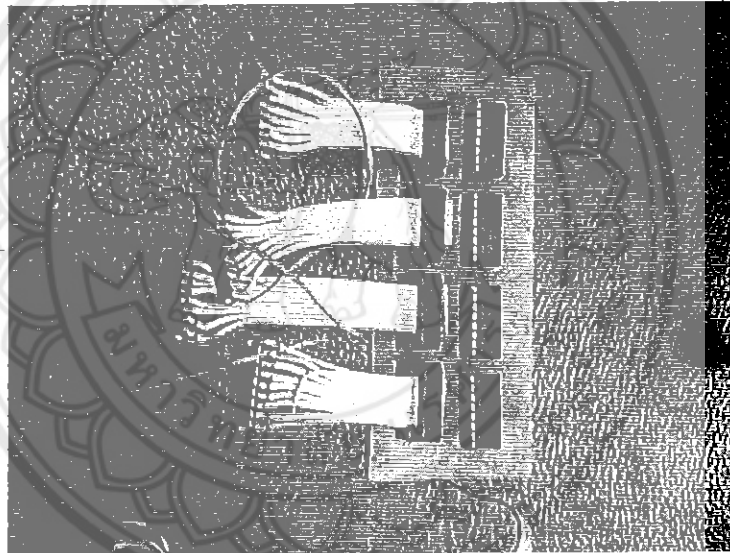
### ผลการทดลอง

#### 4.1 ระบบการทำงาน

มีขั้นตอนการทำงานแบ่งออกเป็น 4 ช่วง ดังที่ได้กล่าวในบทที่ผ่านมา ดังนี้

##### 4.1.1 ช่วงที่ 1 : 4-Channel Input

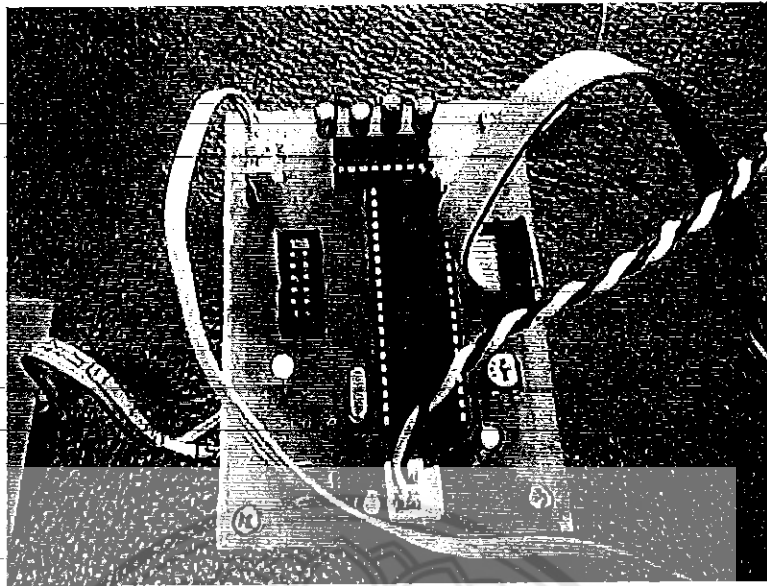
- ประกอบด้วย Port รองรับ Input ข้อมูลจาก Sensor ทั้ง 4 ชนิด โดยใช้ตัว test สมมติ สัญญาณ Input ขึ้นมา ใช้สายไฟสีแดงทำการจิ้มลงไปในแต่ละเส้นของสาย pair ขนาด 10 เส้น เพื่อวัดค่าแต่ละ bit ออกมาตามจำนวนที่ต้องการ



รูปที่ 4.1 4-Channel Input Port

##### 4.1.2 ช่วงที่ 2 : Microcontroller

- ใช้ Microcontroller ในการควบคุมข้อมูล Input ให้ทำการเก็บข้อมูลและบันทึกค่า พร้อมกับบันทึกเวลาที่เข้ามาของข้อมูลลงใน Data Logger โดยตัว Microcontroller จะทำการตัดสินใจในข้อมูลที่ได้รับ โดยใช้เกณฑ์การตัดสินใจที่ได้กำหนดเอาไว้ ถ้าข้อมูลที่ได้รับ มีเกณฑ์ที่ตรงกับโปรแกรมที่ได้ทำการเขียนไว้ Microcontroller ก็จะทำการส่งข้อมูลอีกชุดหนึ่ง ออกไปยังภาค Output ทันทีที่เกณฑ์การตัดสินใจถูกต้อง



รูปที่ 4.2 บอร์ดไมโครคอนโทรลเลอร์ ATmega8515L

#### 4.1.3 ช่วงที่ 3 : Data Logger

- ข้อมูลที่ได้รับจาก Input จะถูกเก็บเข้าไปใน Flash Drive ใว้อย่างละ 2 ชุด โดย 1 ชุด จะถูกเก็บไว้ใน Data Logger และข้อมูลอีก 1 ชุดจะถูกส่งไปยังภาค Output

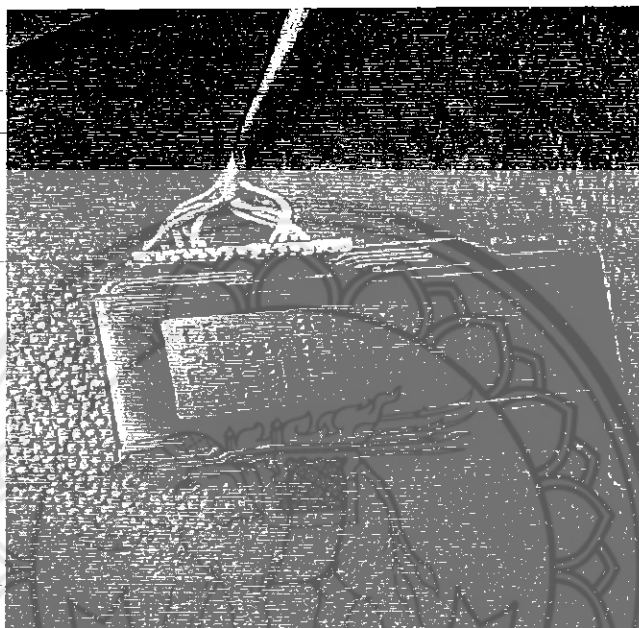


รูปที่ 4.3 Data Logger (ET USB Flash Drive)

#### 4.1.4 ช่วงที่ 4 : Output

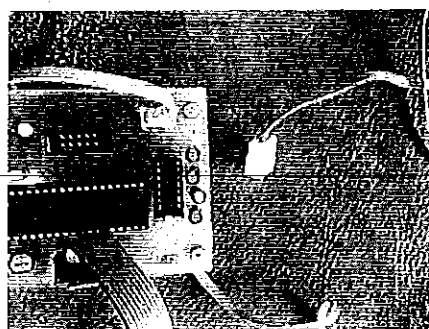
แบ่งออกเป็น 2 ช่วงย่อยๆ คือ

- LCD Output : จะทำการแสดงค่าออกทางจอ LCD. โดยมีการระบุวัน/เดือน/ปี และเวลา โดยใช้การเทียบเวลาจริง แสดงผลผ่านตัวเลขและตัวแปรที่กำหนดตามเกณฑ์ที่ใช้ในการตัดสินใจ



รูปที่ 4.4 การแสดงผลผ่านจอ LCD

- RS232 Output : เป็นการส่งชุดข้อมูลอีก 1 ชุดออกไปยังภาค Output ที่ต้องการรองรับข้อมูลจาก Data Logger โดยชุดข้อมูลที่ถูกส่งออกไปนั้น เป็นชุดข้อมูลที่กำหนด Protocol ให้เป็นเลขฐาน 10 ที่เข้าใจง่ายและนำไปใช้งานได้สะดวกโดยไม่ต้องผ่านการแปลงเลขฐานก่อนใช้งาน



รูปที่ 4.5 RS232 Output port

## 4.2 ผลการทดลอง

Input-port ของข้อมูล ได้ทำการรองรับตัว sensor ทั้ง 4 ชนิดไว้ มีดังนี้

### 4.2.1 Port 1 : อุปกรณ์ตรวจจับปริมาณน้ำฝน(Pulse Counter)

pin ที่ 1 และ pin ที่ 2 – ทำการรับสัญญาณการเคาะของอุปกรณ์ตรวจจับปริมาณน้ำฝน โดย

1 เคาะ = 0.2 mm. ให้เก็บไว้จนกว่าจะครบ 5 mm. จึงจะทำการส่งข้อมูลออกไปยังภาค Output

pin ที่ 3 และ pin ที่ 4 – เป็นตัวเช็คสัญญาณแบบ Digital ว่า มีสัญญาณเข้ามาให้แจ้งเตือนภัยหรือไม่

ในกรณีที่ Input port ที่ใช้เป็นตัวหลักใช้งานไม่ได้ โดย ให้ สัญญาณ

0 = ไม่อันตราย

1 = อันตราย

Pin 3 ใช้ตรวจเช็คสัญญาณของตัวตรวจจับความชื้นในดิน

Pin 4 ใช้ตรวจเช็คสัญญาณของตัวตรวจจับระดับน้ำในตลิ่ง

\*\*\* อีก 4 pin ที่เหลือ ปล่อยว่างเอาไว้

### 4.2.2 Port 2 : ตัวตรวจจับระดับน้ำในตลิ่ง (Analog to Digital) ความละเอียด 8 bit

- จะทำการแจ้งเตือนทุกครั้งที่มีข้อมูลเข้ามาทุกๆ 150 cm.

ระดับน้ำในตลิ่ง เมื่อมีข้อมูลเข้ามา 190 cm. ขึ้นไปแจ้งเตือน Low

210 cm. ขึ้นไปแจ้งเตือน Middle

230 cm. ขึ้นไปแจ้งเตือน High

### 4.2.3 Port ที่ 3 : ตัวตรวจจับความชื้นในดิน (Analog to Digital) ความละเอียด 8 bit

- ความชื้นของดิน เมื่อมีข้อมูลเข้ามา 50% ขึ้นไป ทำการแจ้งเตือนทันที

### 4.2.4 Port ที่ 4 : ตัวตรวจจับการเคลื่อนตัวของดิน แบบ Digital

- การเคลื่อนตัวของดิน เมื่อมีข้อมูลเข้ามามากกว่า 10 cm. ขึ้นไป ทำการแจ้งเตือนทันที



#### 4.2.5 การแสดงผลหน้าจอของ LCD

- บนหน้าจอ LCD จะมีการแสดงผลแต่ละบรรทัดดังนี้



รูปที่ 4.6 การแสดงสถานะบนหน้าจอ LCD

**บรรทัดที่ 1 :** จะแสดงผลเป็น วัน-เดือน-ปี ด้านหน้า และด้านหลังจะแสดงเวลาเมื่อเทียบกับเวลาปัจจุบันเป็น ชั่วโมง: นาที: วินาที

**บรรทัดที่ 2 :** จะแสดงสถานะของ Input port 1 ทั้งหมด โดย

S1[00.0] >> แสดงปริมาณน้ำฝนของ pin ที่ 1 และ pin ที่ 2 ที่รับเข้ามาจาก Sensor โดยจะแสดงผลของน้ำฝนออกมาทางช่อง [00.0] ครั้งละ 0.2 mm.

เมื่อครบ 5.0 mm จะทำการ save ค่าที่ได้ลงใน Flash-Drive แล้วเคลียร์ผลของค่าที่ได้รับ ให้กลับมาเป็น [00.0] อีกครั้ง

S2[0/1] >> เป็น pin ที่ 3 ใช้ในตรวจเช็คสัญญาณของตัวตรวจจับความชื้นในดิน

S3[0/1] >> เป็น pin ที่ 4 ใช้ในตรวจเช็คสัญญาณของตัวตรวจจับระดับน้ำในตลับ

\*\*\* ทั้ง pin 3 และ pin 4 เป็นการตรวจสอบสัญญาณแบบ Digital โดยใช้เกณฑ์

0 = ไม่อันตราย และ 1 = อันตราย >>> จะเป็นการเตือนภัยขั้นต้น

**บรรทัดที่ 3 :** จะแสดงสถานะของ Input port 2 และ 3 โดย

[2]=0/255[L/M/H] >>> ตัวตรวจจับระดับน้ำในตลิ่ง (Analog to Digital)

ความละเอียด 8 bit(Port 2)

โดยใช้การเตือนในช่อง [] ด้านหลัง 3 สถานะ เป็น Low[L], Middle[M] และ High[H]

[3]=0/255[H] >>> ตัวตรวจจับความชื้นในดิน (Analog to Digital)

ความละเอียด 8 bit(Port 3)

โดยใช้การเตือนในช่อง [] ด้านหลัง 1 สถานะ เป็น High[H]

**บรรทัดที่ 4 :** Input port 4 และ แสดงสถานการณ์ทำงานของอุปกรณ์ โดย

ด้านหน้าของบรรทัดที่ 4 – [4]=0/255[H] – ตัวตรวจจับการเคลื่อนตัวของดิน แบบ Digital(Port 4)

โดยใช้การเตือนในช่อง [] ด้านหลัง 1 สถานะ เป็น High[H]

ด้านหลังของบรรทัดที่ 4 – จะแสดงสถานการณ์ทำงานของอุปกรณ์ โดยใช้คำดังต่อไปนี้

Conn = Connect >>> ทำการรอเชื่อมต่อกับอุปกรณ์ ET USB Flash Drive

Stby = Stand by >>> เป็นสถานะนี้ได้ หลังจากที่เก็บข้อมูลบันทึกลงใน Flash Drive เรียบร้อยแล้ว  
หรืออยู่ในสถานะพร้อมทำงานในครั้งต่อไป

Save >>> จะเกิดสถานะนี้ขึ้นได้ ก็ต่อเมื่อมี Input เข้ามา จะถูกเก็บข้อมูลบันทึกลงใน  
Flash Drive

ข้อมูลที่ได้จาก Flash Drive จะอยู่ในรูปของ Text file ในชื่อ DATALOG.txt กำหนดเป็น Protocol  
ในลักษณะดังภาพ

```

DATALOG - Notepad
File Edit Format View Help
$A000B0C0D000E000F064!16/05/09 19:12:23
$A005B0C0D000E000F000!16/05/09 19:13:06
$A005B0C0D000E000F064!16/05/09 19:13:22
$A000B0C0D000E000F065!16/05/09 23:18:20
$A000B0C0D000E064F000!16/05/09 23:18:39
$A005B0C0D000E000F000!16/05/09 23:19:22
$A005B0C0D000E000F065!16/05/09 23:19:36
$A005B0C0D000E128F000!16/05/09 23:19:51
$A010B0C0D000E000F000!16/05/09 23:20:21
$A010B0C0D192E000F000!16/05/09 23:21:08
$A010B0C0D000E064F000!16/05/09 23:21:19
$A010B0C0D000E000F064!16/05/09 23:21:31
$A015B0C0D000E000F000!16/05/09 23:22:14
$A005B0C0D000E000F000!16/05/09 23:29:33

```

รูปที่ 4.7 ข้อมูลที่ได้จาก Text file ในชื่อ DATALOG.txt ใน Data Logger

โดย Protocol แต่ละตัวอักษร กำหนดไว้ดังนี้

A : ค่าที่ได้จากอุปกรณ์ตรวจจับปริมาณน้ำฝน(Pulse Counter) ทั้ง pin 1 และ pin 2

B : ค่าที่ได้จากสัญญาณของตัวตรวจจับความชื้นในดิน แบบ Digital (pin 3)

C : ค่าที่ได้จากสัญญาณของตัวตรวจจับระดับน้ำในตลิ่ง แบบ Digital (pin 4)

D : ค่าที่ได้จากตัวตรวจจับระดับน้ำในตลิ่ง (Analog to Digital) ความละเอียด 8 bit

E : ค่าที่ได้จากตัวตรวจจับความชื้นในดิน (A TO D) ความละเอียด 8 bit

F : ค่าที่ได้จากตัวตรวจจับการเคลื่อนตัวของดิน แบบ-Digital



## บทที่ 5

### สรุปผลการทดลอง

#### 5.1 สรุปผลการดำเนินงานโครงการงาน

จากโครงการงาน Data Logger และการคำนวณประมวลผลการตัดสินใจ ทำให้ได้ข้อสรุปจากผลการดำเนินงานดังนี้

1. สามารถนำการประมวลผลและการจัดเก็บข้อมูล ไปประยุกต์ใช้กับงานอื่นๆ เพื่อให้เกิดประโยชน์ในอนาคตได้
2. เกิดความรู้ความเข้าใจในการใช้งานอุปกรณ์ด้านอิเล็กทรอนิกส์ชนิดต่างๆ
3. เกิดความเข้าใจในการเขียนโปรแกรมเพื่อควบคุมการใช้งาน Microcontroller

#### 5.2 ปัญหาและแนวทางการแก้ไข

จากการทำโครงการงานการประมวลผลและส่งสัญญาณเตือนภัยทางโทรศัพท์มือถือพบปัญหาและอุปสรรคต่างๆ ดังนี้

1. การดำเนินงานมีความล่าช้า เนื่องจากขาดความรู้ความเข้าใจในการเลือกอุปกรณ์ โดย Chipset ที่ใช้ในงาน Microcontroller มีให้เลือกใช้ได้หลายรูปแบบ ทำให้ต้องศึกษาอย่างละเอียด ก่อนที่จะนำมาใช้งานได้จริง
2. การดำเนินงานในด้าน โปรแกรม บางครั้งเกิดปัญหาข้อผิดพลาดในด้านการพัฒนา โปรแกรมทำให้เกิดความล่าช้าในการดำเนินงาน แก้ไข โดยศึกษาด้วยตนเองและสอบถามผู้ที่เชี่ยวชาญในด้านนี้

#### 5.3 ข้อจำกัดของระบบ

1. Microcontroller ที่ใช้ เป็นบอร์ดที่ทำขึ้นมาเพื่อโครงการงานนี้โดยเฉพาะ จะมีข้อจำกัดในการยืดหยุ่นของ Spec ถ้าหากมีการแก้ไข โปรแกรมในภายหลัง
2. ตัว Module ที่ใช้ทำ Data Logger มีปัญหาเกี่ยวกับ Flash Drive บางอัน ทำให้ไม่สามารถเก็บข้อมูลได้ตามต้องการ
3. อุปกรณ์ยังมีการ Delay ของข้อมูล ซึ่งการแก้ไขกับปัญหานี้ ต้องใช้ระยะเวลาที่ค่อนข้างนานพอสมควร

#### 5.4 ข้อเสนอแนะในการพัฒนาต่อไป

1. ควรมีการพัฒนาอุปกรณ์ที่ใช้อย่างสม่ำเสมอ เพื่อให้อุปกรณ์มีความสมบูรณ์และพร้อมใช้งานได้มากยิ่งขึ้น
2. ควรมีการตรวจเช็คอุปกรณ์อิเล็กทรอนิกส์อย่างสม่ำเสมอ เนื่องจากอุปกรณ์เหล่านี้มีโอกาสที่จะเกิดข้อผิดพลาดได้ง่าย ถ้าหากการดูแลรักษาไม่ดีเท่าที่ควร



## เอกสารอ้างอิง

- [1] คู่มือการใช้งาน manual-ET-USB20-Flash-Drive.pdf : <http://www.ett.co.th>
- [2] คู่มือการใช้งาน ATmega8515L.pdf : <http://www.datasheetcatalog.com/>
- [3] คู่มือการใช้งาน MAX220-MAX249.pdf : <http://datasheets.maxim-ic.com>
- [4] คู่มือการใช้งาน PCF8574P\_4.pdf : <http://www.chipsinfo.com/Philips/PCF8574P.htm>
- [5] คู่มือการใช้งาน คู่มือ LCD : <http://lcd-monitors.globalspec.com>
- [6] AVR ISP : เบื้องต้น : <http://www.electoday.com/bbs/archiver/?tid-180.html>
- [7] ประจัน พลังสันติกุล. ชัยวัฒน์ ลิมพรจิตรวิไล. *MCS-51 microcontroller experiment with Keil C51 compiler*. กรุงเทพมหานคร: บริษัท อิน โน โวทีฟ เอ็กเพอริเมนต์ จำกัด.2521



# ภาคผนวก ก.

## การลงโปรแกรม

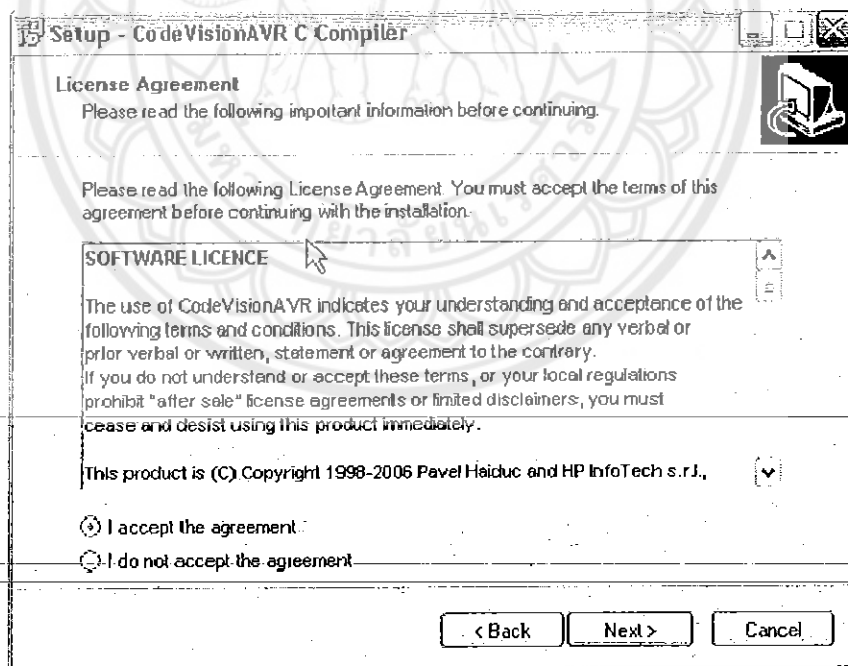
### 1. การติดตั้งโปรแกรม CodeVisionAVR

1). ติดตั้งโปรแกรม CodeVision AVR ลงบน PC โดยเปิดไฟล์ set up ตามภาพ



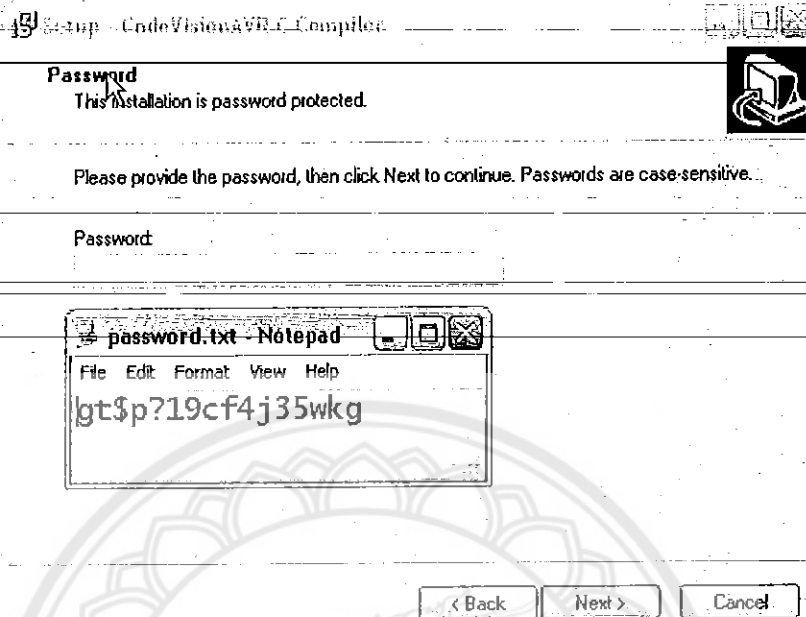
รูปที่ 1.1 setup.exe file

2). เลือกที่ช่อง "I accept the agreement" แล้วกด next



รูปที่ 1.2 Setup – CodeVisionAVR Compiler(1)

3). ทำการใส่ password ลงไป แล้วคลิกที่ next



รูปที่ 1.3 Setup – CodeVisionAVR Compiler(2)

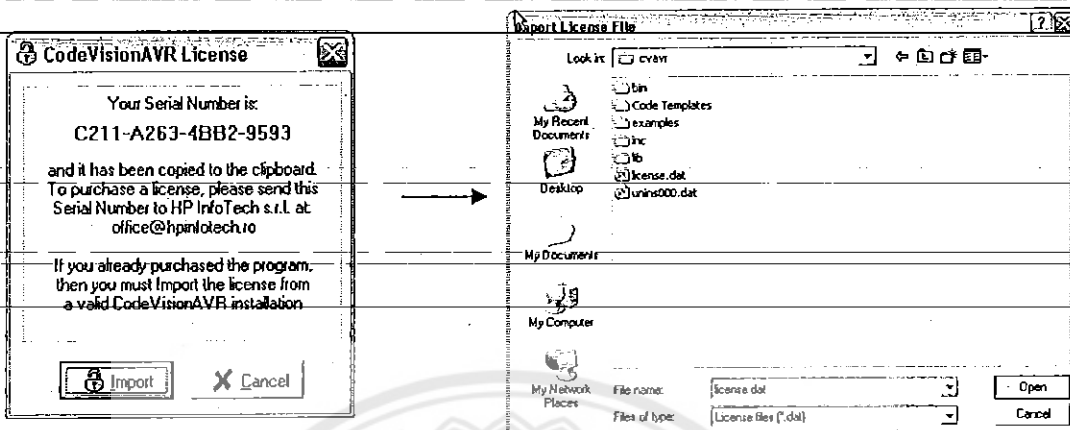
4). คลิก next ไปเรื่อยๆจนกว่าจะเจอหน้าต่างดังภาพ



รูปที่ 1.4 Setup – CodeVisionAVR Compiler(3)

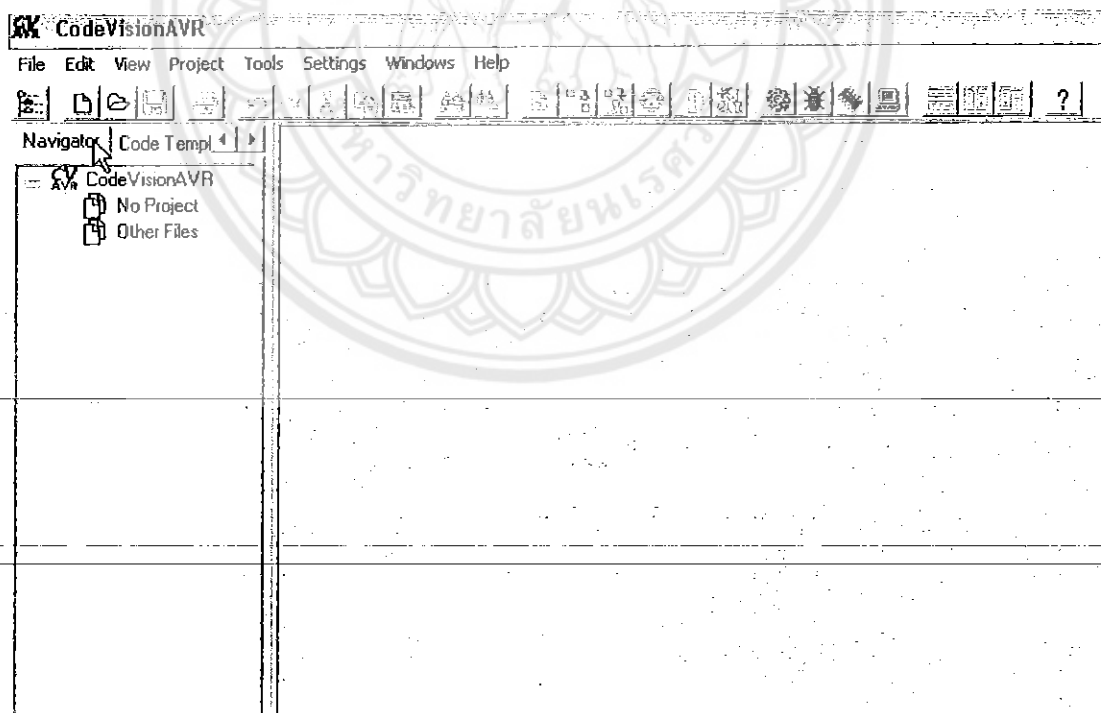


5). ทำการ Copy license.dat ไปไว้ที่ C:\Cvavr จากนั้นทำการ IMPORT ดังภาพด้านล่าง



รูปที่ 1.5 Setup -- CodeVisionAVR Compiler(4)

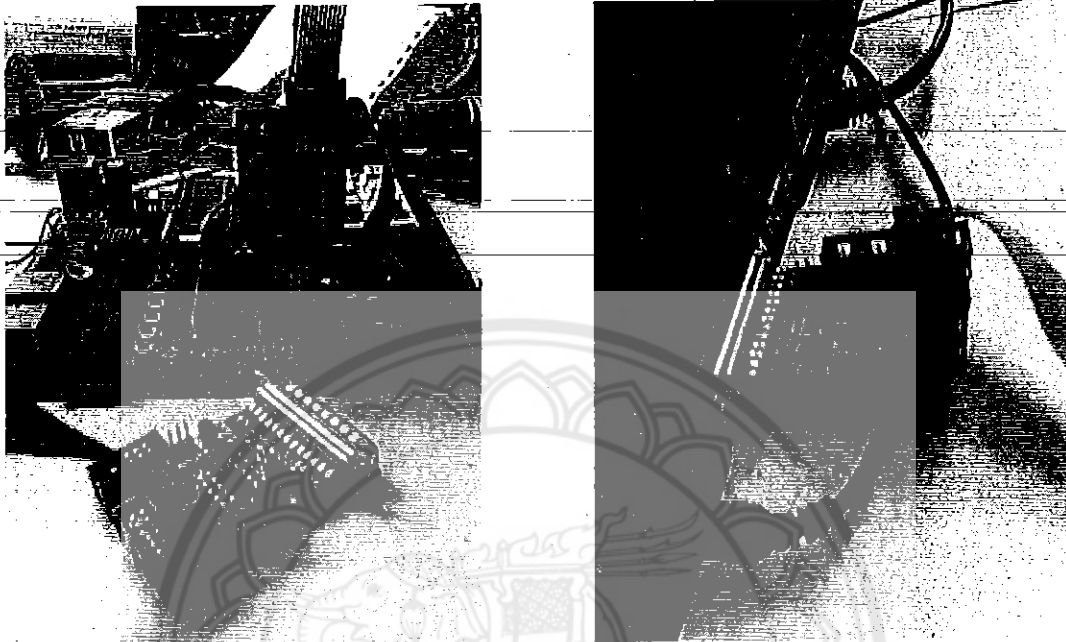
6). กด OPEN File license.dat จากนั้นกด OK



รูปที่ 1.6 Setup -- CodeVisionAVR Compiler(5)

## 2. การ Burn โปรแกรม

### 1). ต่ออุปกรณ์ตามภาพ



รูปที่ 2.1 Setup – AVR ISP Burn (1)

### 2). เปิดโปรแกรม CodeVisionAVR



HP InfoTech

# CodeVisionAVR

C Compiler, Integrated Development Environment  
Automatic Program Generator and In-System Programmer  
for the Atmel AVR Family of Microcontrollers

Version 1.24.8b Professional

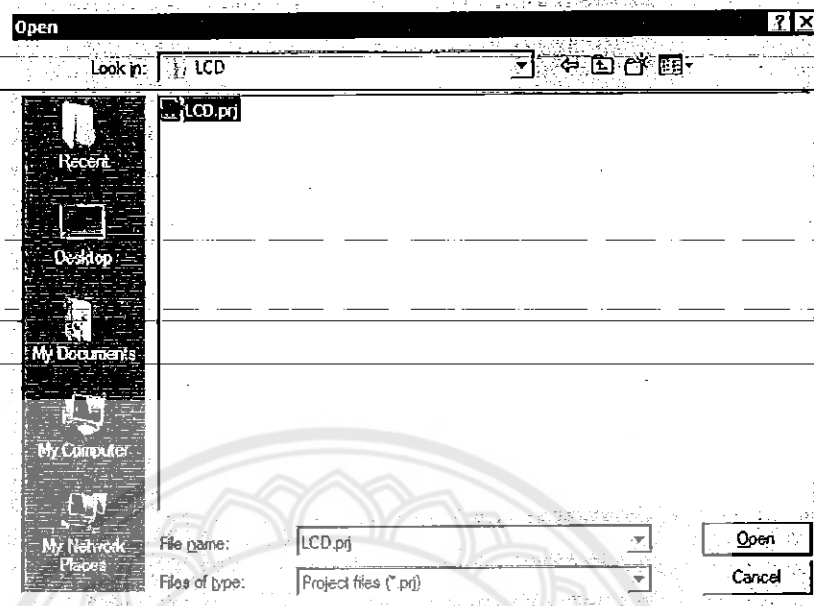
© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Licensed to: F4CG  
F4CG  
C211-A263-4882-9593

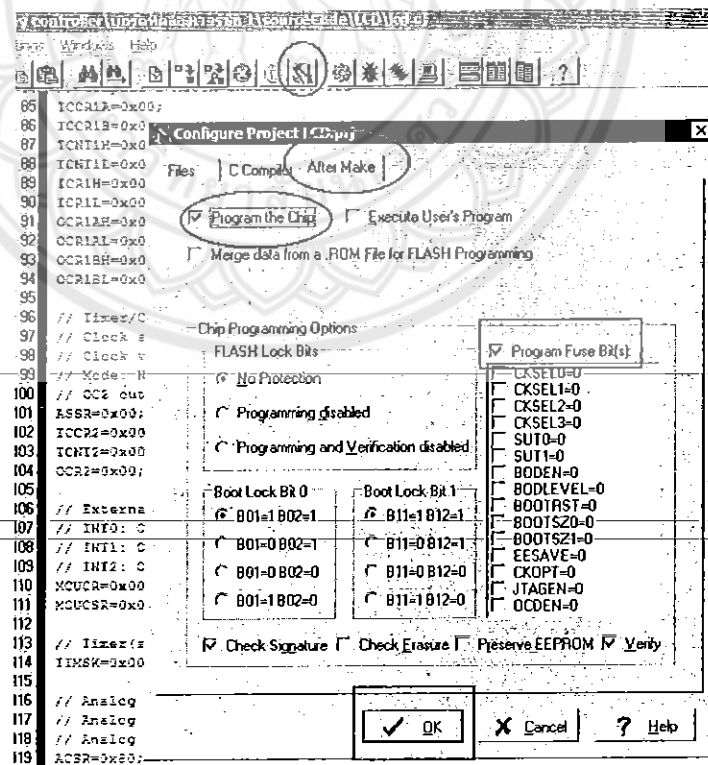
รูปที่ 2.2 Setup – AVR ISP Burn (2)

### 3). ทำการ Open Object ที่ได้สร้างเอาไว้



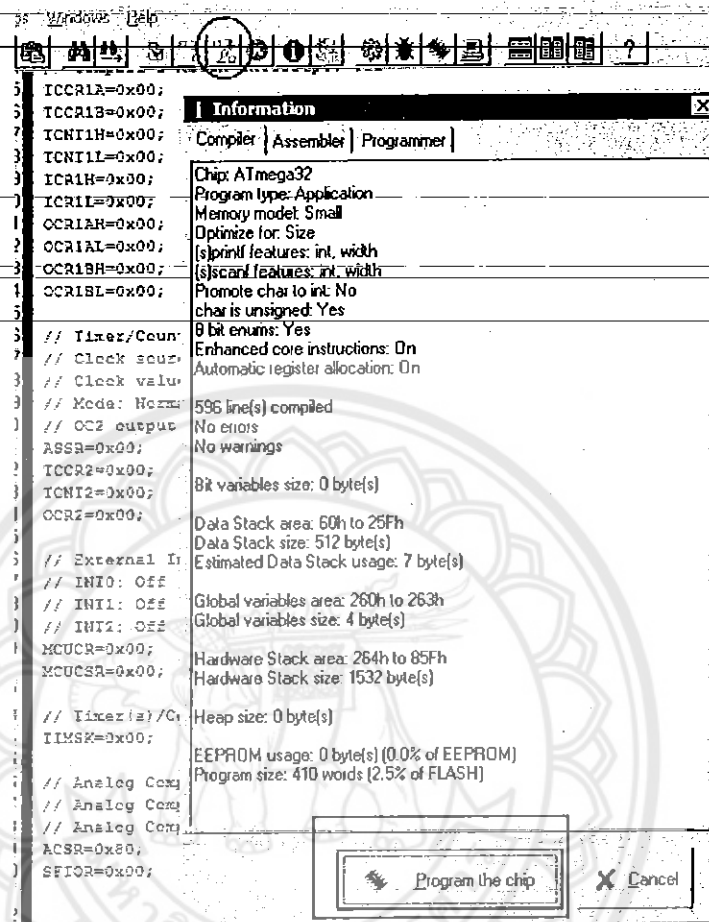
รูปที่ 2.3 Setup – AVR ISP Burn (3)

### 4). ทำการ set ค่าตามภาพ



รูปที่ 2.4 Setup – AVR ISP Burn (4)

5). ทำการ set ค่าตามภาพ จากนั้นจึงทำการ burn โปรแกรมที่เขียนได้ลงในบอร์ด



รูปที่ 2.5 Setup – AVR ISP Burn (5)

### **3. Data Sheet ATmega8515L**

#### **Features**

- High-performance, Low-power AVR® 8-bit Microcontroller
- RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
    - Up to 16 MIPS Throughput at 16 MHz
    - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-programmable Flash
    - Endurance: 10,000 Write/Erase Cycles
    - Optional Boot Code Section with Independent Lock bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
    - 512 Bytes EEPROM
      - Endurance: 100,000 Write/Erase Cycles
    - 512 Bytes Internal SRAM
    - Up to 64K Bytes Optional External Memory Space
    - Programming Lock for Software Security
- Peripheral Features
  - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Three PWM Channels
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator

- Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection

- Internal Calibrated RC Oscillator

- External and Internal Interrupt Sources

- Three Sleep Modes: Idle, Power-down and Standby

- I/O and Packages

- 35 Programmable I/O Lines

- 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MLF

- Operating Voltages

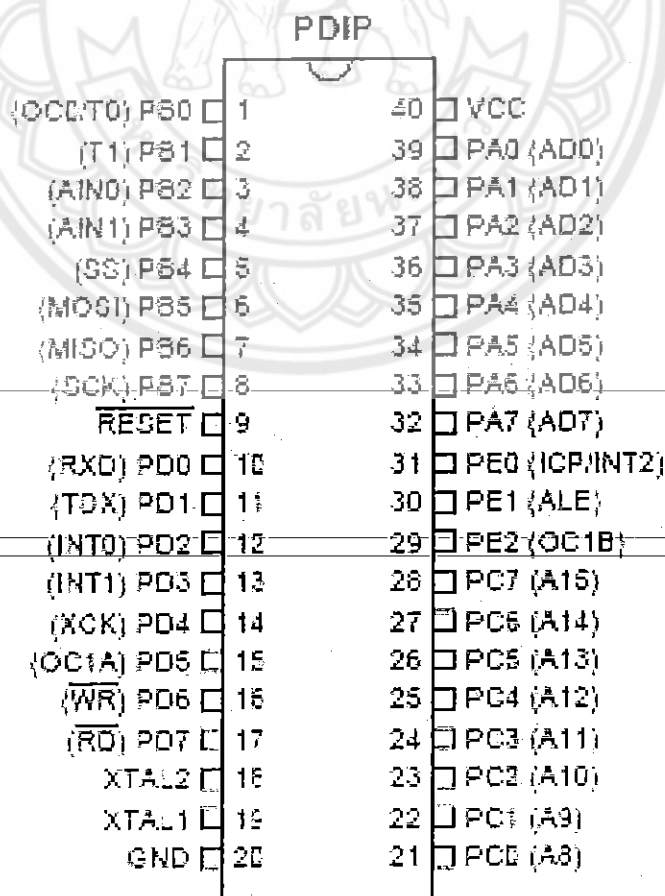
- 2.7 - 5.5V for ATmega8515L

- 4.5 - 5.5V for ATmega8515

- Speed Grades

- 0 - 8 MHz for ATmega8515L

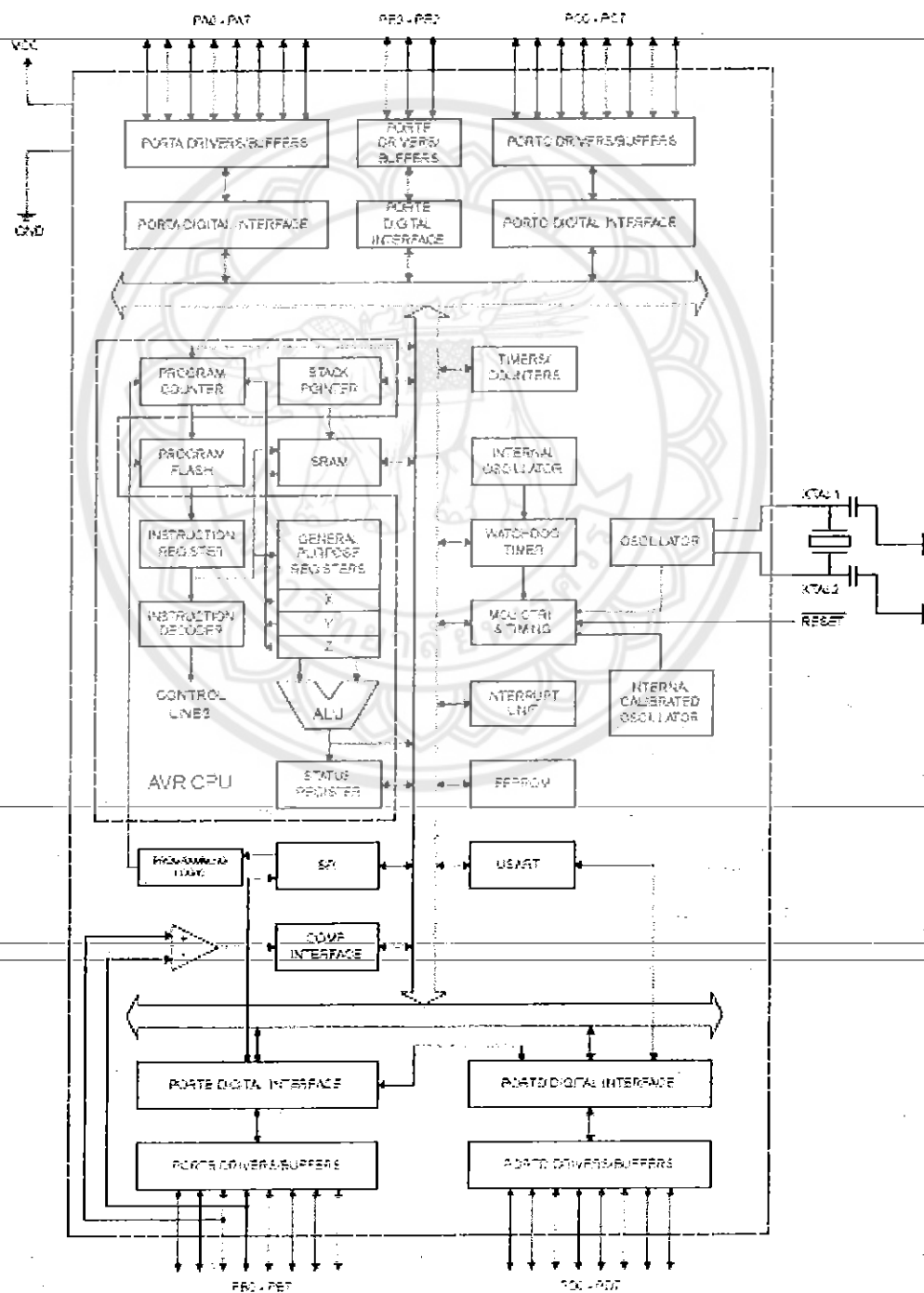
- 0 - 16 MHz for ATmega8515



## Overview

The ATmega8515 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8515 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clockcycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8515 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, an External memory interface, 35 general purpose I/O lines, 32 general purpose working registers, two flexible Timer/Counters with compare modes, Internal and External interrupts, a Serial Programmable USART, a programmable Watchdog Timer with internal Oscillator, a SPI serial port, and three software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and Interrupt system to continue functioning. The Power-down mode saves the Register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the Program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-programmable Flash on a monolithic chip, the Atmel ATmega8515 is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The ATmega8515 is supported with a full suite of program and system development tools including: C Compilers, Macro assemblers, Program debugger/simulators, In-circuit Emulators, and Evaluation kits.



**Disclaimer**

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

**AT90S4414/8515 and ATmega8515 Compatibility**

The ATmega8515 provides all the features of the AT90S4414/8515. In addition, several new features are added. The ATmega8515 is backward compatible with AT90S4414/8515 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S4414/8515 compatibility mode can be selected by programming the S8515C Fuse. ATmega8515 is 100% pin compatible with AT90S4414/8515, and can replace the AT90S4414/8515 on current printed circuit boards. However, the location of Fuse bits and the electrical characteristics differs between the two devices.

**AT90S4414/8515 Compatibility Mode**

Programming the S8515C Fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 52 for details.
- The double buffering of the USART Receive Registers is disabled. See “AVR USART vs. AVR UART – Compatibility” on page 135 for details.
- PORTE(2:1) will be set as output, and PORTE0 will be set as input

**Pin Descriptions**

VCC Digital supply voltage.

GND Ground.

**Port A (PA7..PA0)** Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the ATmega8515 as listed on page 66.

**Port B (PB7..PB0)** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the ATmega8515 as listed on page 66.

**Port C (PC7..PC0)** Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Port D (PD7..PD0)** Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega8515 as listed on page 71.

**Port E (PE2..PE0)** Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port E also serves the functions of various special features of the ATmega8515 as listed on page 73.

**RESET** Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 18 on page 45. Shorter pulses are not guaranteed to generate a reset.

**XTAL1** Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

**XTAL2** Output from the inverting Oscillator amplifier.

**Register summary**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
03F (03F)	GRES	-	T	H	G	V	N	Z	C	9
03E (03E)	SPH	SP1E	SP1A	SP13	SP12	SP11	SP10	SP9	SP8	11
03D (03D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	11
03C (03C)	Reserved									
039 (039)	GICR	INT1	INT0	INT2	-	-	-	INTL	INCE	25, 27
03A (03A)	GIFR	INTF1	INTF0	INTF2	-	-	-	-	-	26
038 (038)	TMSK	TQBF1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE1	OCIE0	32, 123
039 (039)	TIFR	TQBF1	COF1A	COF1B	-	ICF1	-	TOV0	OCVF0	32, 124
037 (037)	SPMR	SPME	RAMP2	-	RAMP2E	B3SET	PGART	PGBE3	SPMEN	168
036 (036)	EMUCSR	EMC	EM1	EM0	EM3	EM2	EM1	EM0	EM3	28, 41, 77
035 (035)	MCUCR	SR	SRW16	SE	SU1	ISC17	ISC10	ISC0	ISC0	28, 40, 75
034 (034)	MCUCSR	-	-	SM2	-	WDRF	DRRF	EXTRF	PORF	40, 48
033 (033)	TCOCR	FOCF	WGM00	COM00	COM03	WGM01	CS12	CS01	CS00	30
032 (032)	TCNT0	Timer Counter 0 (8 Bit)								32
031 (031)	OCRC	Timer Counter 0 Output Compare Register								32
030 (030)	QFOR	-	XMEK	XUM1	XUM0	XVMC	PU0	-	PSR10	30, 65, 95
02F (02F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	118
02E (02E)	TCCR1B	ICNF1	ICBF1	-	WGM13	WGM12	CS12	CS11	CS10	121
02C (02C)	TCNT1H	Timer Counter 1 - Counter Register High Byte								122
02C (02C)	TCNT1L	Timer Counter 1 - Counter Register Low Byte								122
02B (02B)	OCRAH	Timer Counter 1 - Output Compare Register A High Byte								122
02A (02A)	OCRAL	Timer Counter 1 - Output Compare Register A Low Byte								122
029 (029)	OCRH	Timer Counter 1 - Output Compare Register B High Byte								122
028 (028)	ORCL	Timer Counter 1 - Output Compare Register B Low Byte								122
027 (027)	Reserved									
026 (026)	Reserved									
025 (025)	ICR1H	Timer Counter 1 - Input Capture Register High Byte								123
024 (024)	ICR1L	Timer Counter 1 - Input Capture Register Low Byte								123
023 (023)	Reserved									
022 (022)	Reserved									
021 (021)	ADTCR	-	-	-	ADSC	ADIF	ADIF2	ADIF1	ADIF0	50
020 (020)	UBRRH	URSEL	-	-	-	-	UBRR[11:8]			167
01F (01F)	UCSRC	URSEL	UMSEL	UMIF	UMIF	USBS	UCSZ1	UCSZ0	UCPOL	165
01E (01E)	EEARH	-	-	-	-	-	-	-	EEAR3	15
01E (01E)	EEARL	EEPROM Address Register Low Byte								15
01D (01D)	EEDR	EEPROM Data Register								15
01C (01C)	EECR	-	-	-	-	EERE	EEMWE	EEMPE	EERE	15
01E (01E)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	74
01A (01A)	DDRA	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	74
019 (019)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	74
018 (018)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	74
017 (017)	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	74
016 (016)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	74
015 (015)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	74
014 (014)	DDRC	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	74
013 (013)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	74
012 (012)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	75
011 (011)	DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0	75
010 (010)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	75
00F (00F)	SPDR	SPI Data Register								151
00E (00E)	SPCR	SPF	WCOL	-	-	-	-	-	SP1X	151
00D (00D)	SPSR	SPE	SFE	CSR0	MSR	CPOL	CPHA	SPRI	SPRE	123
00C (00C)	UCR	USART I/O Data Register								153
00B (00B)	UCSRA	RXC	TXC	UDRE	FE	DRR	PE	U2X	UPSM	153
00A (00A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXBS	TXBS	154
009 (009)	UBRRL	USART Baud Rate Register Low Byte								157
008 (008)	ACSR	ACD	ACBG	ACC	AD1	AC1E	AC1C	AC1F	AC1D	152
007 (007)	PCRTE	-	-	-	-	-	PCRTD2	PCRTE1	PCRTE0	75
00E (00E)	DDRE	-	-	-	-	-	DDR3	DDR1	DDR0	75
00E (00E)	PINE	-	-	-	-	-	PINE2	PINE1	PINE0	75
004 (004)	OSCCAL	Oscillator Calibrator Register								35

3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.

**Instruction Set Summary**

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z, C, N, V, H	1
ADC	Rd, Rr	Add with Carry to Registers	Rd ← Rd + Rr + C	Z, C, N, V, H	1
ADDA	Rd, K	Add Immediate to Word	Rd ← Rd + Rn[Rd] + K	Z, C, N, V, H	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z, C, N, V, H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z, C, N, V, H	1
SBC	Rd, Rr	Subtract with Carry to Registers	Rd ← Rd - Rr - C	Z, C, N, V, H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z, C, N, V, H	1
SBIW	Rd, K	Subtract Immediate from Word	Rd ← Rd - Rn[Rd] + K	Z, C, N, V, H	2
AND	Rd, Rr	Logic AND Registers	Rd ← Rd & Rr	Z, N, V	1
ANDI	Rd, K	Logic AND Register and Constant	Rd ← Rd & K	Z, N, V	1
OR	Rd, Rr	Logic OR Registers	Rd ← Rd   Rr	Z, N, V	1
ORI	Rd, K	Logic OR Register and Constant	Rd ← Rd   K	Z, N, V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ^ Rr	Z, N, V	1
COM	Rd	One's Complement	Rd ← ~Rd	Z, C, N, V	1
NEG	Rd	Two's Complement	Rd ← ~Rd + 1	Z, C, N, V, H	1
SEB	Rd, K	Set Bits in Register	Rd ← Rd   K	Z, N, V	1
CEB	Rd, K	Clear Bits in Register	Rd ← Rd & (~K)	Z, N, V	1
INC	Rd	Increment	Rd ← Rd + 1	Z, N, V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z, N, V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd - Rd	Z, N, V	1
CLR	Rd	Clear Register	Rd ← Rd & 0	Z, N, V	1
SEI	Rd	Set Register	Rd ← 0xFF	None	1
MUL	Rd, Rr	Multiply Unsigned	R[RD] ← Rd * Rr	Z, C	2
MULS	Rd, Rr	Multiply Signed	R[RD] ← Rd * Rr	Z, C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R[RD] ← Rd * Rr	Z, C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R[RD] ← (Rd * Rr) << 1	Z, C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R[RD] ← (Rd * Rr) << 1	Z, C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R[RD] ← (Rd * Rr) << 1	Z, C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	r	Relative Jump	PC ← PC + r + 1	None	2
UJMP	r	Unconditional Jump to Z	PC ← Z	None	2
RJALL	r	Relative Jump to Call	PC ← PC + r + 1	None	2
RJALL	r	Relative Jump to Call	PC ← Z	None	2
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	None	4
CPSE	Rd, Rr	Compare, Skip if Equal	r (Rd = Rr) ? PC + 2 or 3	None	1/2/3
CP	Rd, Rr	Compare	Rd - Rr	Z, N, V, C, H	1
CPC	Rd, Rr	Compare with Carry	Rd - Rr - C	Z, N, V, C, H	1
CP	Rd, K	Compare Register with Immediate	Rd - K	Z, N, V, C, H	1
SBRC	R, b	Skip if Bit in Register Cleared	r (R[b] = 0) ? PC + 2 or 3	None	1/2/3
SBRD	R, b	Skip if Bit in Register Set	r (R[b] = 1) ? PC + 2 or 3	None	1/2/3
SBIC	R, b	Skip if Bit in I/O Register Cleared	r (I/O[b] = 0) ? PC + 2 or 3	None	1/2/3
SBIW	R, b	Skip if Bit in I/O Register Set	r (I/O[b] = 1) ? PC + 2 or 3	None	1/2/3
BRBS	r, b	Branch if Status Flag Set	r (SREG[b] = 1) ? then PC + 1	None	1/2
BRBC	r, b	Branch if Status Flag Cleared	r (SREG[b] = 0) ? then PC + 1	None	1/2
BRBE	r, b	Branch if Status Flag Equal	r (Z = 1) ? then PC + 1 - 1	None	1/2
BRNE	r, b	Branch if Not Equal	r (Z = 0) ? then PC + 1 - 1	None	1/2
BRCS	r, b	Branch if Carry Set	r (C = 1) ? then PC + 1 - 1	None	1/2
BRCC	r, b	Branch if Carry Cleared	r (C = 0) ? then PC + 1 - 1	None	1/2
BRSH	r, b	Branch if Greater or Higher	r (C = 0) ? then PC + 1 - 1	None	1/2
BRLO	r, b	Branch if Lower	r (C = 1) ? then PC + 1 - 1	None	1/2
BRPL	r, b	Branch if Plus	r (S = 1) ? then PC + 1 - 1	None	1/2
BRMPL	r, b	Branch if Minus	r (S = 0) ? then PC + 1 - 1	None	1/2
BRPL	r, b	Branch if Plus	r (S = 0) ? then PC + 1 - 1	None	1/2
BRGE	r, b	Branch if Greater or Equal Signed	r (S & V = 0) ? then PC + 1 - 1	None	1/2
BRLT	r, b	Branch if Less Than Signed	r (S & V = 1) ? then PC + 1 - 1	None	1/2
BRHI	r, b	Branch if Half Carry Flag Set	r (H = 1) ? then PC + 1 - 1	None	1/2
BRLO	r, b	Branch if Half Carry Flag Cleared	r (H = 0) ? then PC + 1 - 1	None	1/2
BRTC	r, b	Branch if T Flag Set	r (T = 1) ? then PC + 1 - 1	None	1/2
BRTO	r, b	Branch if T Flag Cleared	r (T = 0) ? then PC + 1 - 1	None	1/2
BRVS	r, b	Branch if Overflow Flag Set	r (V = 1) ? then PC + 1 - 1	None	1/2
BRVO	r, b	Branch if Overflow Flag Cleared	r (V = 0) ? then PC + 1 - 1	None	1/2
BRSE	r, b	Branch if Interrupt Enabled	r (I = 1) ? then PC + 1 - 1	None	1/2
BRSD	r, b	Branch if Interrupt Disabled	r (I = 0) ? then PC + 1 - 1	None	1/2

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd ← Rr; Rd ← Rr	None	1
LD	Rd, R	Load Immediate	Rd ← X	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X); X ← X + 1	None	2
LD	Rd, X-	Load Indirect and Pre-Dec.	X ← X - 1; Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y); Y ← Y + 1	None	2
LD	Rd, Y-	Load Indirect and Pre-Dec.	Y ← Y - 1; Rd ← (Y)	None	2
LDL	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	1
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z); Z ← Z + 1	None	2
LD	Rd, Z-	Load Indirect and Pre-Dec.	Z ← Z - 1; Rd ← (Z)	None	2
LDL	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	1
LDL	Rd, t	Load Direct from SRAM	Rd ← (t)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr; X ← X + 1	None	2
ST	X-, Rr	Store Indirect and Pre-Dec.	X ← X - 1; (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr; Y ← Y + 1	None	2
ST	Y-, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1; (Y) ← Rr	None	2
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr; Z ← Z + 1	None	2
ST	Z-, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1; (Z) ← Rr	None	2
STD	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STA	t, Rr	Store Direct to SRAM	(t) ← Rr	None	2
LFM		Load Program Memory	PC ← (Z)	None	3
LFM	Rd, Z	Load Program Memory	Rd ← (Z)	None	2
LFM	Rd, Z+	Load Program Memory and Post-Inc.	Rd ← (Z); Z ← Z + 1	None	3
SPM		Store Program Memory	(Z) ← Rr; Rr	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
BSF	R, b	Set Bit in I/O Register	(IOP, R) ← 1	None	2
BSF	R, b	Set Bit in I/O Register	(IOP, R) ← 0	None	2
BSL	Rd	Logical Shift Left	Rd[n+1] ← Rd[n]; Rd[n] ← 0	Z, C, N, V	1
BSR	Rd	Logical Shift Right	Rd[n] ← Rd[n+1]; Rd[7] ← 0	Z, C, N, V	1
RGL	Rd	Rotate Left Through Carry	Rd[0] ← C; Rd[n+1] ← Rd[0]; Rd[n] ← Rd[7]	Z, C, N, V	1
ROR	Rd	Rotate Right Through Carry	Rd[7] ← Rd[n]; Rd[n+1] ← Rd[0]	Z, C, N, V	1
ASR	Rd	Arithmetic Shift Right	Rd[n] ← Rd[n+1]; n=0, 1, 6	Z, C, N, V	1
SWAP	R	Swap Nibbles	Rd[3:0] ← Rd[7:4]; Rd[7:4] ← Rd[3:0]	None	1
BSET	R	Flag Set	SREG.S ← 1	SREG(S)	1
BCLR	R	Flag Clear	SREG.S ← 0	SREG(S)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr[b]	T	1
BLD	Rd, b	Bit Load from T to Register	Rd[b] ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
<b>MCU CONTROL INSTRUCTIONS</b>					

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>DATA TRANSFER INSTRUCTIONS</b>					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd ← 16B ← Rr ← 16B	None	1

### Ordering Information

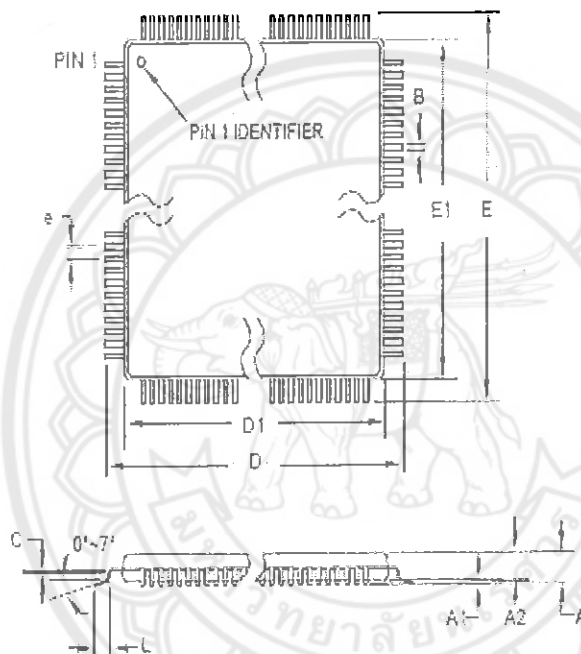
Speed (MHz)	Power Supply	Ordering Code	Package <sup>1)</sup>	Operation Range		
8	2.7 - 5.5V	ATmega8515L-8AC	44A	Commercial (0°C to 70°C)		
		ATmega8515L-8PC	40P6			
		ATmega8515L-8JC	44J			
		ATmega8515L-8MC	44M1			
		16	4.5 - 5.5V	ATmega8515L-8AI	44A	Industrial (-40°C to 85°C)
				ATmega8515L-8PI	40P6	
				ATmega8515L-8JI	44J	
				ATmega8515L-8MI	44M1	
16	4.5 - 5.5V	ATmega8515-16AC	44A	Commercial (0°C to 70°C)		
		ATmega8515-16PC	40P6			
		ATmega8515-16JC	44J			
		ATmega8515-16MC	44M1			
		16	4.5 - 5.5V	ATmega8515-16AI	44A	Industrial (-40°C to 85°C)
				ATmega8515-16PI	40P6	
				ATmega8515-16JI	44J	
				ATmega8515-16MI	44M1	

Note: 1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.

Package Type	
44A	44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
40P6	40-lead, 0.600" Wide, Plastic Dual In-line Package (PDIP)
44J	44-lead, Plastic J-Leaded Chip Carrier (PLCC)
44M1	44-pad, 7 x 7 x 1.0 mm body, lead pitch 0.50 mm, Micro Lead Frame Package (MLF)

Packaging Information

44A



COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	1.20	
A1	0.05	-	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	-	0.45	
C	0.09	-	0.20	
L	0.45	-	0.75	
e	0.80 TYP			

- Notes
1. This package conforms to JEDEC reference MS-026, Variation ACB
  2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold flash.
  3. Lead coplanarity is 0.10 mm maximum

0:5:2001



2325 Orchard Parkway  
San Jose, CA 95131

TITLE

44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

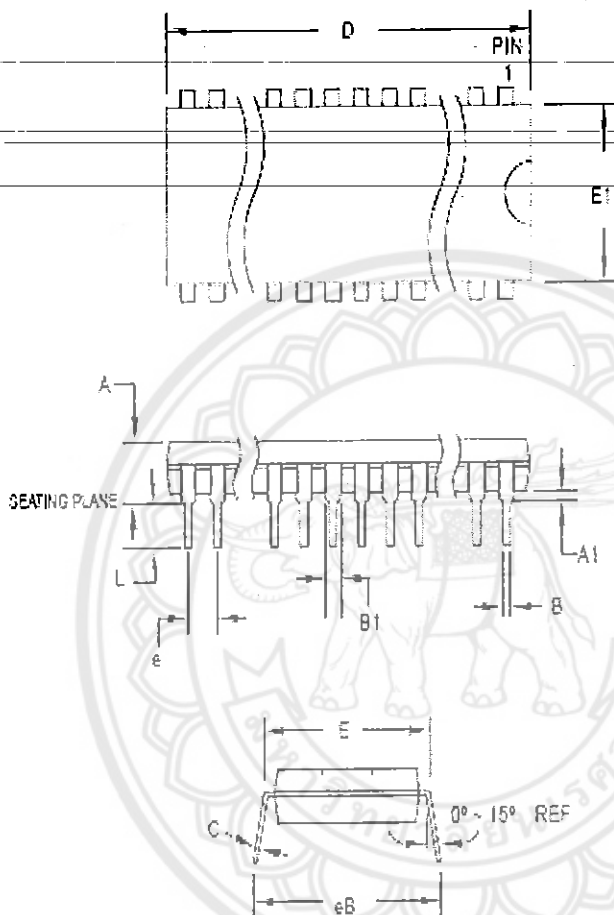
DRAWING NO.

44A

REV.

B

40P6



COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.626	
A1	0.381	-	-	
D	52.070	-	52.675	Note 2
E	15.240	-	15.875	
E1	13.452	-	13.970	Note 2
B	0.350	-	0.550	
B1	1.041	-	1.651	
L	3.048	-	3.556	
C	0.203	-	0.381	
eB	15.494	-	17.526	
e	2.540 TYP			

- Notes: 1. This package conforms to JEDEC reference MS-011, Variation AC.  
 2. Dimensions D and E1 do not include mold flash or protrusion. Mold flash or protrusion shall not exceed 0.25 mm (0.010").

09/28/01



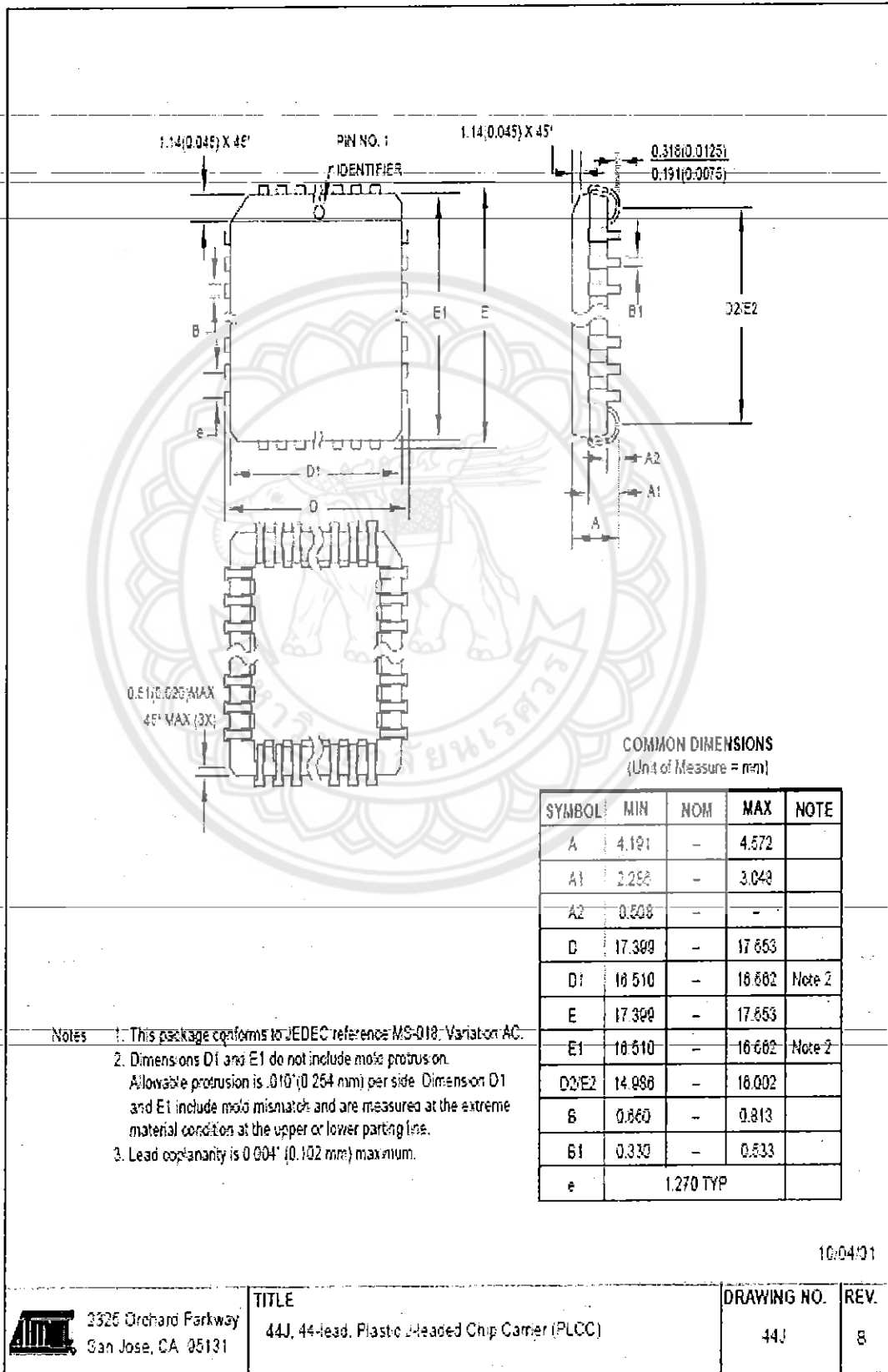
2325 Orchard Parkway  
San Jose, CA 95131

TITLE  
40P6, 40-Lead (0.600"/15.24 mm Wide) Plastic Dual  
Inline Package (PDIF)

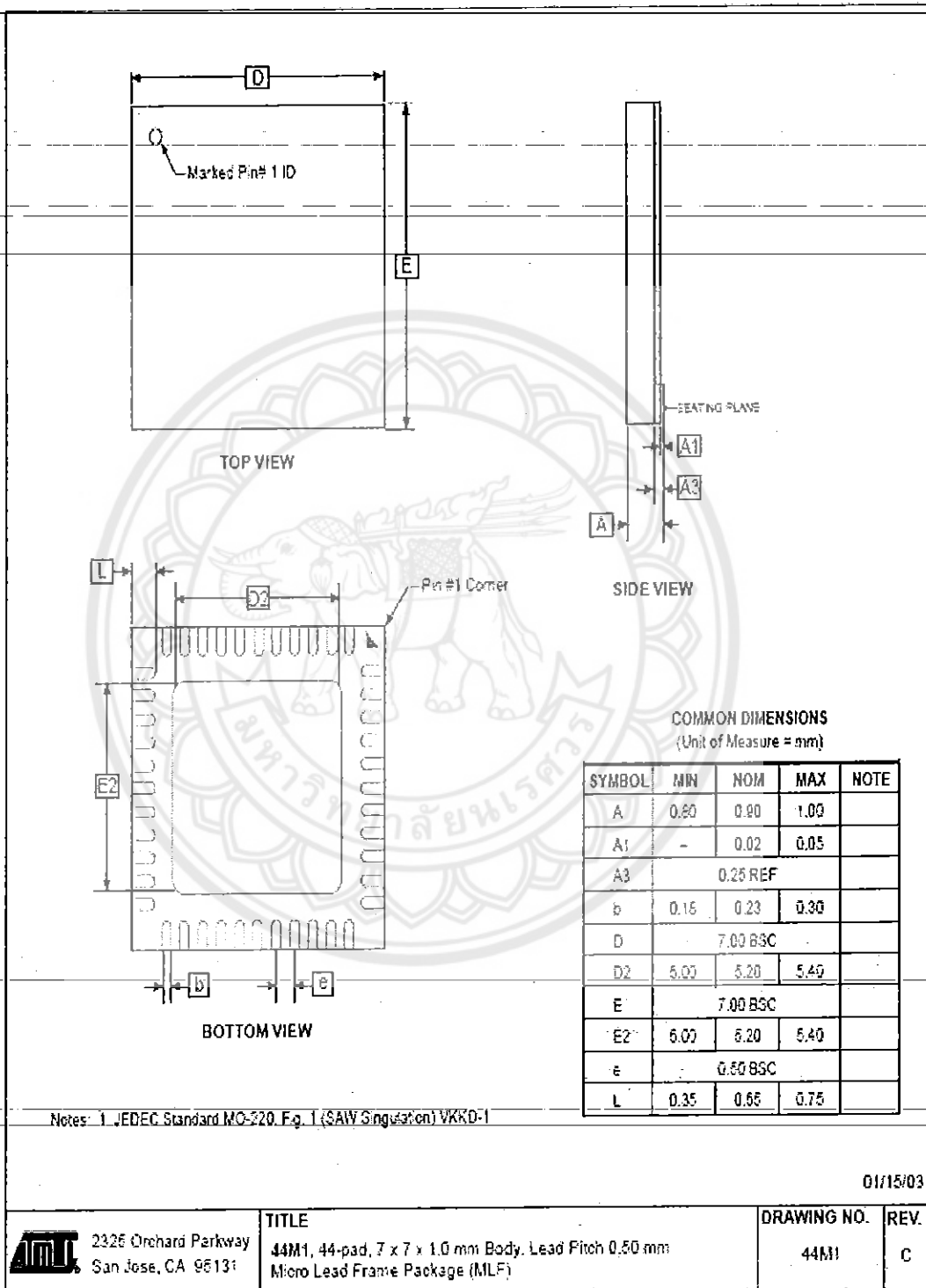
DRAWING NO.	REV.
40P6	B



44J



44M1



**Errata** The revision letter in this section refers to the revision of the ATmega8515 device.

**ATmega8515(L) Rev. B** There are no errata for this revision of ATmega8515.

## Datasheet Change Log for ATmega8515

---

Please note that the referring page numbers in this section are referring to this document.

---

The referring revision in this section are referring to the document revision.

---

Changes from Rev.

2512F-12/03 to Rev.

2512E-09/03

1. Updated "Calibrated Internal RC Oscillator" on page 38.

Changes from Rev.

2512D-02/03 to Rev.

2512E-09/03

1. Removed "Preliminary" from the datasheet.
2. Updated Table 18 on page 45 and "Absolute Maximum Ratings" and "DC Characteristics" in "Electrical Characteristics" on page 195.
3. Updated chapter "ATmega8515 Typical Characteristics" on page 205.

Changes from Rev.

2512C-10/02 to Rev.

2512D-02/03

1. Added "EEPROM Write During Power-down Sleep Mode" on page 22.
2. Improved the description in "Phase Correct PWM Mode" on page 87.
3. Corrected OCn waveforms in Figure 53 on page 110.
4. Added note under "Filling the Temporary Buffer (page loading)" on page 171 about writing to the EEPROM during an SPM page load.

---

5. Updated Table 93 on page 193.

6. Updated "Packaging Information" on page 12.

Changes from Rev.

2512B-09/02 to Rev.

2512C-10/02

1. Added "Using all Locations of External Memory Smaller than 64 KB" on page 30.

2. Removed all TBD.
3. Added description about calibration values for 2, 4, and 8 MHz.
4. Added variation in frequency of “External Clock” on page 39.
5. Added note about VBOT, Table 18 on page 45.
6. Updated about “Unconnected pins” on page 63.
7. Updated “16-bit Timer/Counter1” on page 96, Table 51 on page 118 and Table 52 on page 119.
8. Updated “Enter Programming Mode” on page 182, “Chip Erase” on page 182, Figure 77 on page 185, and Figure 78 on page 186.
9. Updated “Electrical Characteristics” on page 195, “External Clock Drive” on page 197, Table 96 on page 197 and Table 97 on page 198, “SPI Timing Characteristics” on page 198 and Table 98 on page 200.
10. Added “Errata” on page 16.

Changes from Rev.

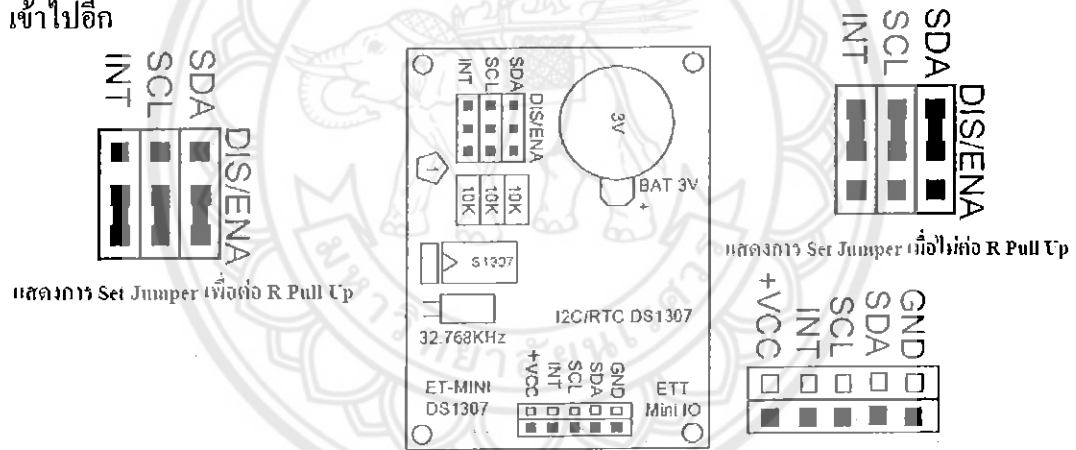
2512A-04/02 to Rev.

2512B-09/02

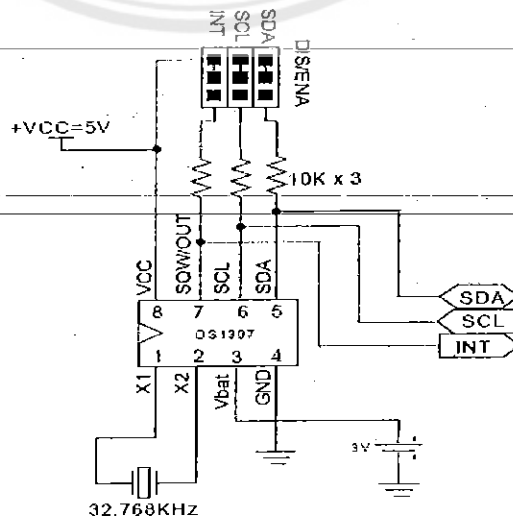
1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

**4. Data Sheet ET-MINI DS1307**

Module ET-MINI DS1307 เป็นชุด I C ประเภท Real Time Clock (RTC) ซึ่งทำหน้าที่เกี่ยวกับฐานเวลาในลักษณะของ นาฬิกา เวลา และปฏิทิน ในโมดูลนี้ จะมีขั้วต่อ 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 I C เบอร์นี้จะมี Control Byte อยู่ที่ "1101000x" นอกจากนี้ก็จะมีในส่วนของที่สำหรับใส่แบตเตอรี่เพื่อใช้ในการ Back Up ฐานเวลาให้นาฬิกายังคงเดินได้อย่างถูกต้อง เมื่อไม่มีการจ่ายไฟเลี้ยงให้กับ Module ก่อนการใช้งาน โมดูลนี้จะต้องทำการ Set Jumper SDA ,SCL และ INT มาทางด้าน ENA ดังรูป เพื่อเป็นการต่อ R Pull Up ให้กับขาทั้ง 3 ขาของ I C ถ้ามีการต่อ R Pull Up ใน Line SDA และ SCL จากภายนอกหรือจาก Module อื่นไว้แล้วให้ Set Jumper ทั้ง 3 มาทางด้าน DIS เพื่อไม่ต้องต่อ R Pull Up เข้าไปอีก



รูป โครงสร้าง ET-MINI DS1307 และตำแหน่งขา Port ใช้งาน



รูป วงจร ET-MINI DS1307

**5. Data Sheet PCF8574 Remote 8-bit I/O expander for I2C-bus**

## CONTENTS

## 1 FEATURES

## 2 GENERAL DESCRIPTION

## 3 ORDERING INFORMATION

## 4 BLOCK DIAGRAM

## 5 PINNING

## 5.1 DIP16 and SO16 packages

## 5.2 SSOP20 package

## 6 CHARACTERISTICS OF THE I2C-BUS

## 6.1 Bit transfer

## 6.2 Start and stop conditions

## 6.3 System configuration

## 6.4 Acknowledge

## 7 FUNCTIONAL DESCRIPTION

## 7.1 Addressing

## 7.2 Interrupt output

## 7.3 Quasi-bidirectional I/Os

## 8 LIMITING VALUES

## 9 HANDLING

## 10 DC CHARACTERISTICS

## 11 I2C-BUS TIMING CHARACTERISTICS

## 12 PACKAGE OUTLINES

## 13 SOLDERING

## 13.1 Introduction

## 13.2 Through-hole mount packages

## 13.2.1 Soldering by dipping or by solder wave

## 13.2.2 Manual soldering

## 13.3 Surface mount packages

## 13.3.1 Reflow soldering

## 13.3.2 Wave soldering 13.3.3 Manual soldering

### 13.4 Suitability of IC packages for wave, reflow and dipping soldering methods

## 14 DATA SHEET STATUS

## 15 DEFINITIONS

## 16 DISCLAIMERS

## 17 PURCHASE OF PHILIPS I2C COMPONENTS

Philips Semiconductors

Product specification

## Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

### 1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10  $\mu$ A maximum
- I<sup>2</sup>C-bus to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I<sup>2</sup>C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.



The device consists of an 8-bit quasi-bidirectional port and an I<sup>2</sup>C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line (INT) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I<sup>2</sup>C-bus. This means that the PCF8574 can remain a simple slave device.

### 2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I<sup>2</sup>C-bus).

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.10.

### 3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8574P; PCF8574AP	DIP16	plastic dual in-line package; 16 leads (300 mil)	SOT38-4
PCF8574T; PCF8574AT	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCF8574TS; PCF8574ATS	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 4 BLOCK DIAGRAM

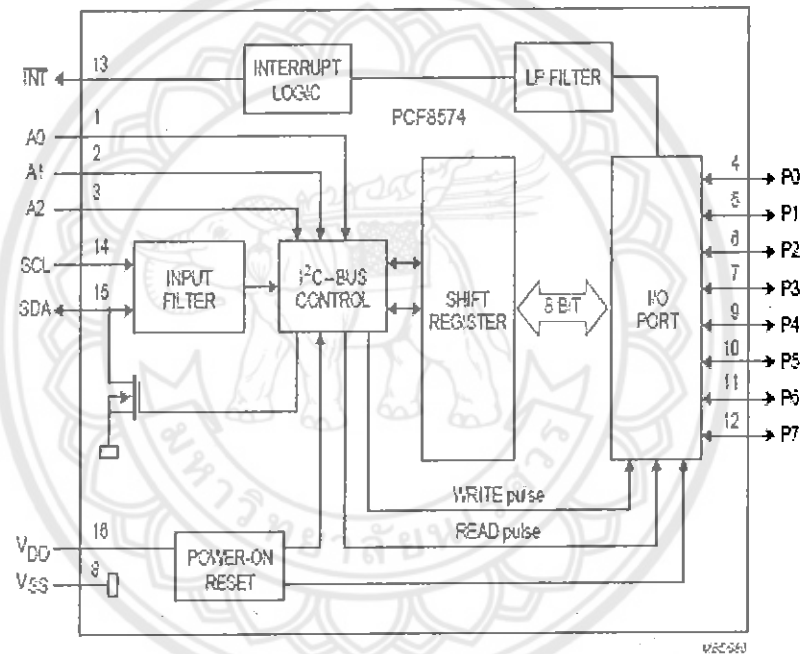


Fig.1—Block diagram (pin numbers apply to DIP-16 and SO-16 packages).



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 5 PINNING

## 5.1 DIP16 and SO16 packages

SYMBOL	PIN	DESCRIPTION
A0	1	address input 0
A1	2	address input 1
A2	3	address input 2
P0	4	quasi-bidirectional I/O 0
P1	5	quasi-bidirectional I/O 1
P2	6	quasi-bidirectional I/O 2
P3	7	quasi-bidirectional I/O 3
V <sub>SS</sub>	8	supply ground
P4	9	quasi-bidirectional I/O 4
P5	10	quasi-bidirectional I/O 5
P6	11	quasi-bidirectional I/O 6
P7	12	quasi-bidirectional I/O 7
INT	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V <sub>DD</sub>	16	supply voltage

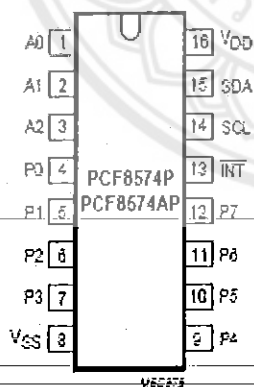


Fig.2 Pin configuration (DIP16).

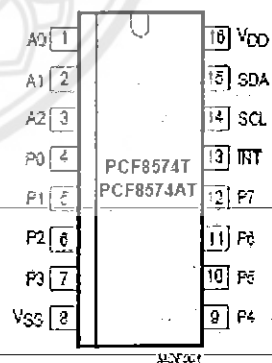


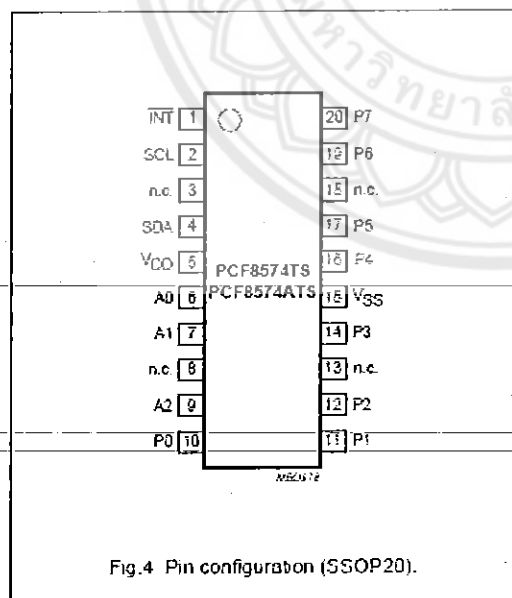
Fig.3 Pin configuration (SO16).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 5.2 SSOP20 package

SYMBOL	PIN	DESCRIPTION
INT	1	interrupt output (active LOW)
SCL	2	serial clock line
n.c.	3	not connected
SDA	4	serial data line
V <sub>DD</sub>	5	supply voltage
A0	6	address input 0
A1	7	address input 1
n.c.	8	not connected
A2	9	address input 2
P0	10	quasi-bidirectional I/O 0
P1	11	quasi-bidirectional I/O 1
P2	12	quasi-bidirectional I/O 2
n.c.	13	not connected
P3	14	quasi-bidirectional I/O 3
V <sub>SS</sub>	15	supply ground
P4	16	quasi-bidirectional I/O 4
P5	17	quasi-bidirectional I/O 5
n.c.	18	not connected
P6	19	quasi-bidirectional I/O 6
P7	20	quasi-bidirectional I/O 7



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

6 CHARACTERISTICS OF THE I<sup>2</sup>C-BUS

The I<sup>2</sup>C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

6.2 Start and stop conditions

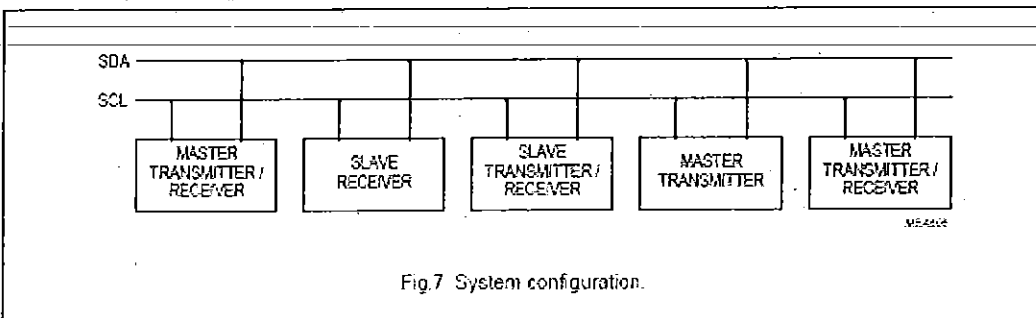
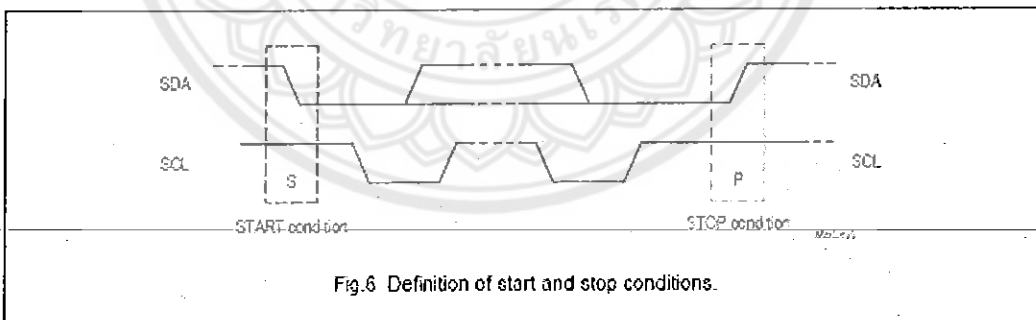
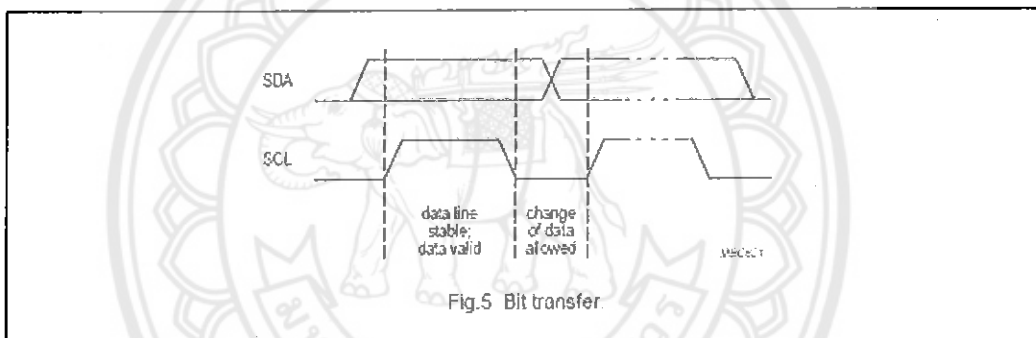
Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P) (see Fig.6).

6.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.5).

6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.7).



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

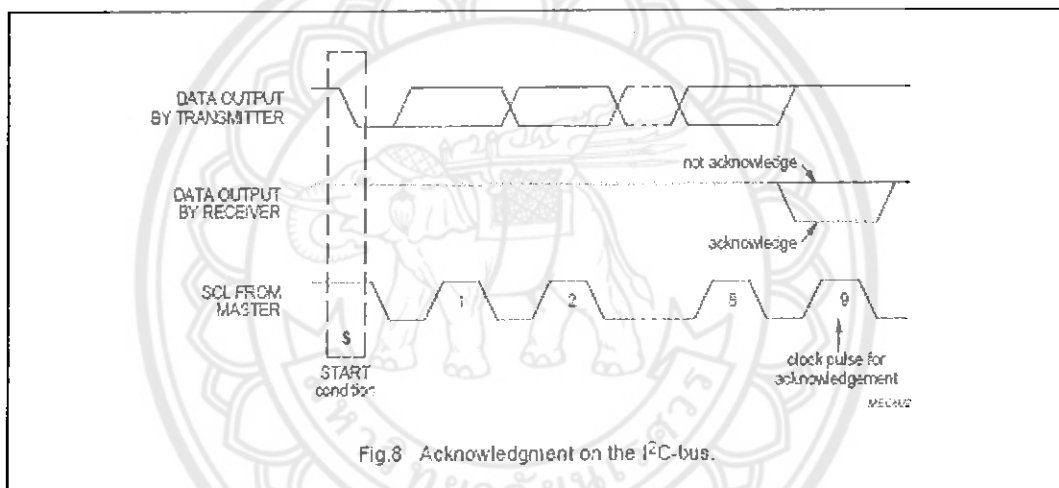
## 6.4 Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit (see Fig.8). The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception

of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account.

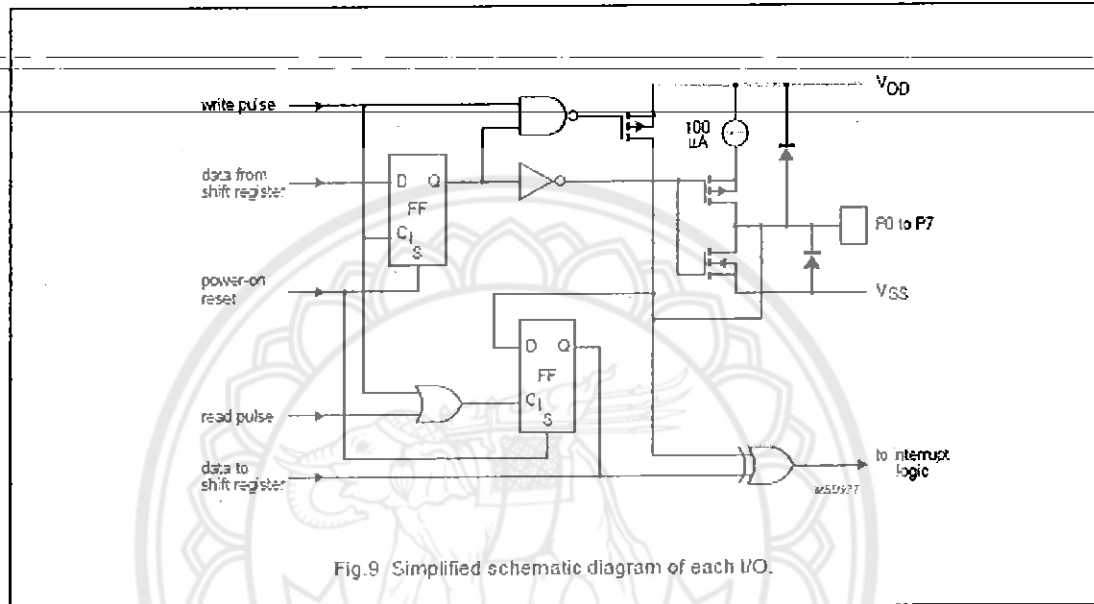
A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

Fig.8 Acknowledgment on the I<sup>2</sup>C-bus.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

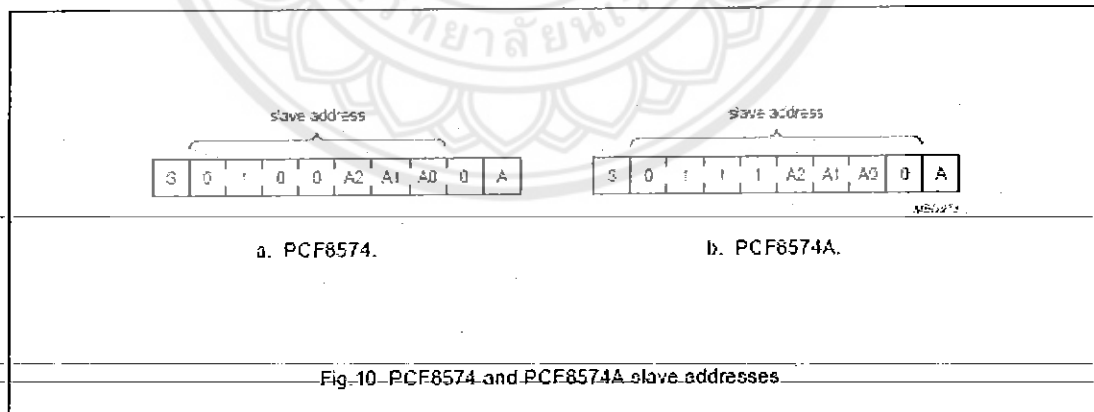
PCF8574

7 FUNCTIONAL DESCRIPTION

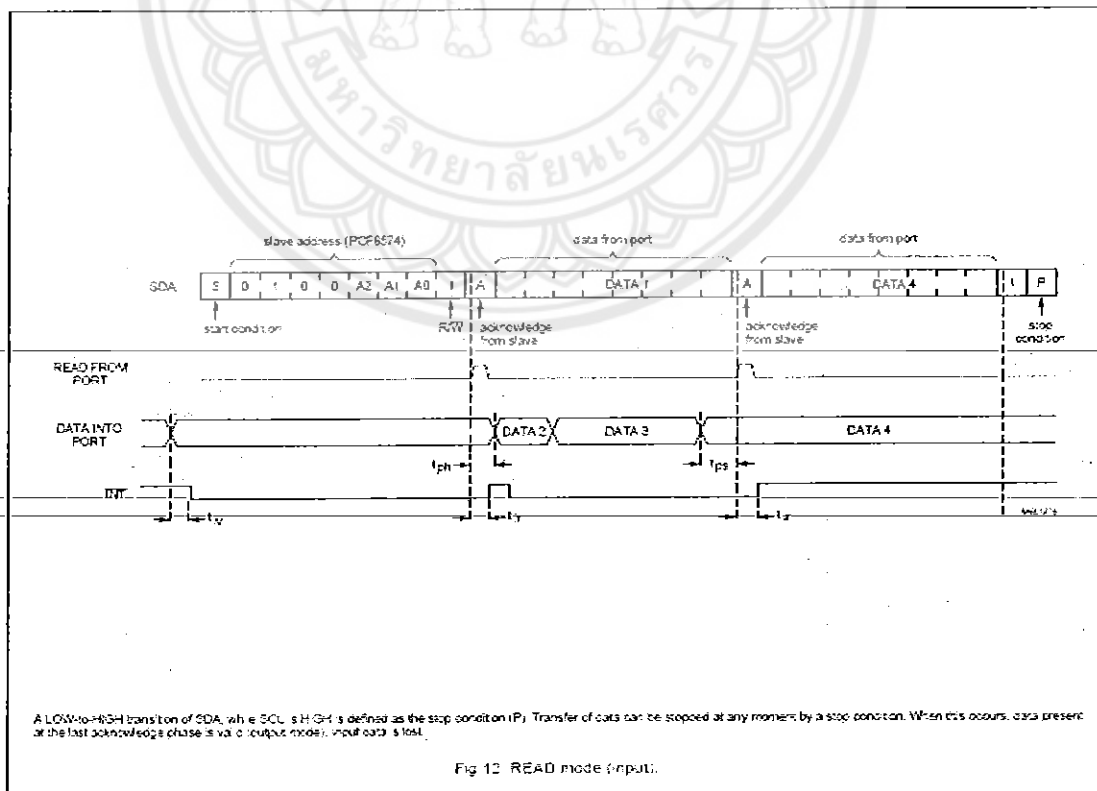
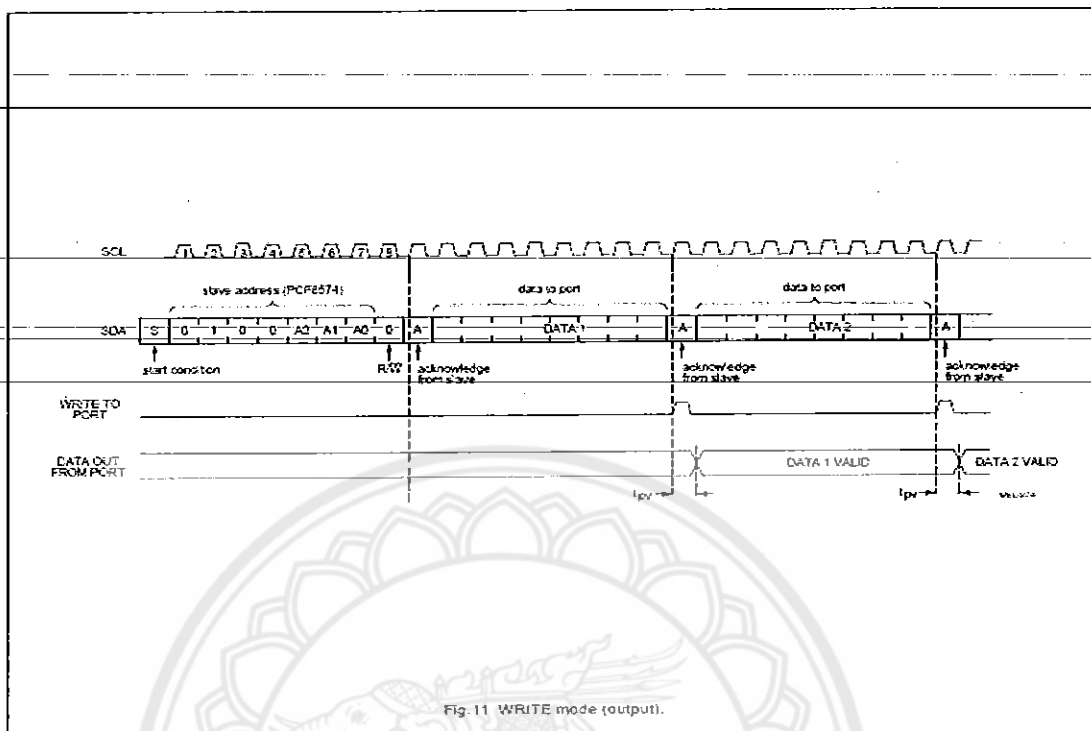


7.1 Addressing

For addressing see Figs 10, 11 and 12.



Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.12). Output data is transmitted to the port by the WRITE mode (see Fig.11).



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

7.2 Interrupt output

The PCF8574 provides an open-drain output ( $\overline{\text{INT}}$ ) which can be fed to a corresponding input of the microcontroller (see Figs 13 and 14). This gives these chips a type of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time  $t_v$  the signal  $\overline{\text{INT}}$  is valid.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.

Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal
- In the WRITE mode at the acknowledge bit after the HIGH-to-LOW transition of the SCL signal

- Interrupts which occur during the acknowledge clock pulse may be lost (or very short) due to the resetting of the interrupt during this pulse.

Each change of the I/Os after resetting will be detected and, after the next rising clock edge, will be transmitted as  $\overline{\text{INT}}$ . Reading from or writing to another device does not affect the interrupt circuit.

7.3 Quasi-bidirectional I/Os

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction (see Fig.15). At power-on the I/Os are HIGH. In this mode only a current source to  $V_{DD}$  is active. An additional strong pull-up to  $V_{DD}$  allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs.

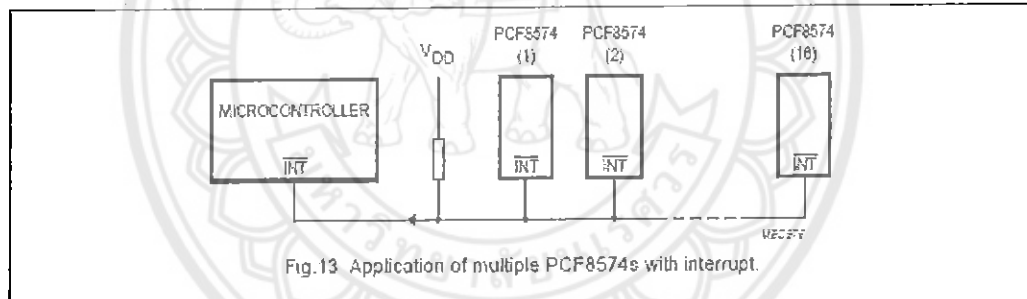


Fig. 13 Application of multiple PCF8574s with interrupt.

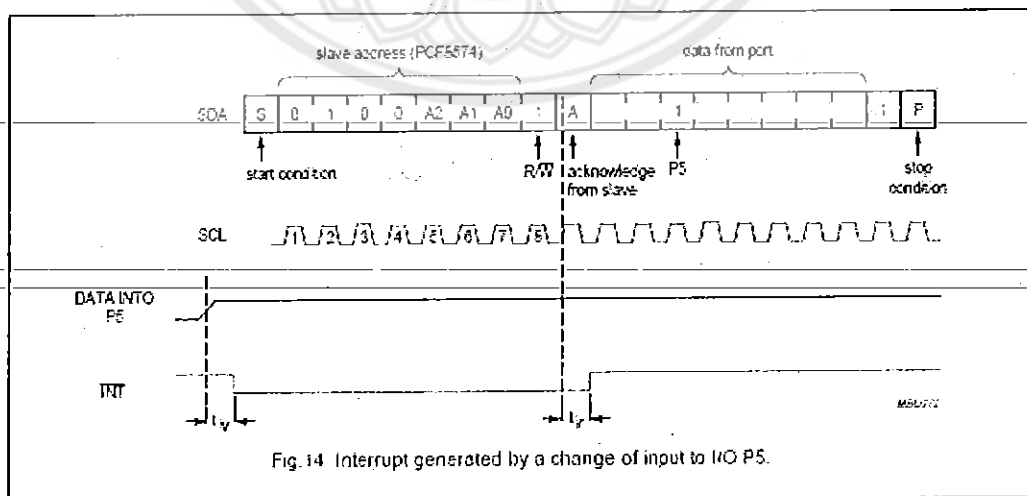


Fig. 14 Interrupt generated by a change of input to I/O P5.

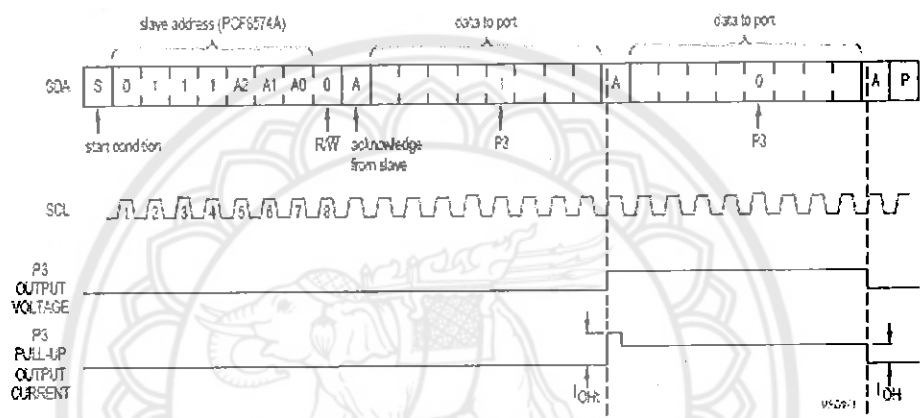


Fig. 15. Transient pull-up current  $I_{OH}$  while P3 changes from LOW-to-HIGH and back to LOW.



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 8 LIMITING VALUES

In accordance with the Absolute Maximum Rating-System (IEC 60134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V <sub>DD</sub>	supply voltage	-0.5	+7.0	V
V <sub>I</sub>	input voltage	V <sub>SS</sub> - 0.5	V <sub>DD</sub> + 0.5	V
I <sub>I</sub>	DC input current	-	±20	mA
I <sub>O</sub>	DC output current	-	±25	mA
I <sub>DD</sub>	supply current	-	±100	mA
I <sub>SS</sub>	supply current	-	±100	mA
P <sub>tot</sub>	total power dissipation	-	400	mW
P <sub>O</sub>	power dissipation per output	-	100	mW
T <sub>stg</sub>	storage temperature	-65	+150	°C
T <sub>amb</sub>	ambient temperature	-40	+85	°C

## 9 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However it is good practice to take normal precautions appropriate to handling MOS devices (see "Handling MOS devices").

## 10 DC CHARACTERISTICS

V<sub>DD</sub> = 2.5 to 6 V; V<sub>SS</sub> = 0 V; T<sub>amb</sub> = -40 to +85 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>Supply</b>						
V <sub>DD</sub>	supply voltage		2.5	-	6.0	V
I <sub>DD</sub>	supply current	operating mode; V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub> ; f <sub>SCL</sub> = 100 kHz	-	40	100	µA
I <sub>SS</sub>	standby current	standby mode; V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub>	-	2.5	10	µA
V <sub>POR</sub>	Power-on reset voltage	V <sub>DD</sub> = 6 V; no load; V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub> ; note 1	-	1.3	2.4	V
<b>Input-SCL: input/output-SDA</b>						
V <sub>IL</sub>	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
V <sub>IH</sub>	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
I <sub>OL</sub>	LOW level output current	V <sub>OL</sub> = 0.4 V	3	-	-	mA
I <sub>L</sub>	leakage current	V <sub>I</sub> = V <sub>DD</sub> or V <sub>SS</sub>	-1	-	+1	µA
C <sub>i</sub>	input capacitance	V <sub>I</sub> = V <sub>SS</sub>	-	-	7	pF

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
<b>I/Os</b>						
$V_{L}$	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
$V_{H}$	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
$I_{IH(max)}$	maximum allowed input current through protection diode	$V_I \geq V_{DD}$ or $V_I \leq V_{SS}$	-	-	±400	µA
$I_{OL}$	LOW level output current	$V_{OL} = 1\text{ V}$ ; $V_{DD} = 5\text{ V}$	10	25	-	mA
$I_{OH}$	HIGH level output current	$V_{OH} = V_{SS}$	30	-	300	µA
$I_{OHt}$	transient pull-up current	HIGH during acknowledge (see Fig.15); $V_{OH} = V_{SS}$ ; $V_{DD} = 2.5\text{ V}$	-	-1	-	mA
$C_i$	input capacitance		-	-	10	pF
$C_o$	output capacitance		-	-	10	pF
<b>Port timing; <math>C_L \leq 100\text{ pF}</math> (see Figs 11 and 12)</b>						
$t_{pw}$	output data valid		-	-	4	µs
$t_{su}$	input data set-up time		0	-	-	µs
$t_h$	input data hold time		4	-	-	µs
<b>Interrupt INT (see Fig. 14)</b>						
$I_{OL}$	LOW level output current	$V_{OL} = 0.4\text{ V}$	1.6	-	-	mA
$I_L$	leakage current	$V_I = V_{DD}$ or $V_{SS}$	-1	-	+1	µA
<b>TIMING: <math>C_L \leq 100\text{ pF}</math></b>						
$t_v$	input data valid time		-	-	4	µs
$t_r$	reset delay time		-	-	4	µs
<b>Select inputs A0 to A2</b>						
$V_{L}$	LOW level input voltage		-0.5	-	+0.3V <sub>DD</sub>	V
$V_{H}$	HIGH level input voltage		0.7V <sub>DD</sub>	-	V <sub>DD</sub> + 0.5	V
$I_{L}$	input leakage current	pin at V <sub>DD</sub> or V <sub>SS</sub>	-250	-	+250	nA

## Note

- The Power-on reset circuit resets the I<sup>2</sup>C-bus logic at  $V_{DD} < V_{POR}$  and sets all I/Os to logic 1 (with current source to V<sub>DD</sub>).

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

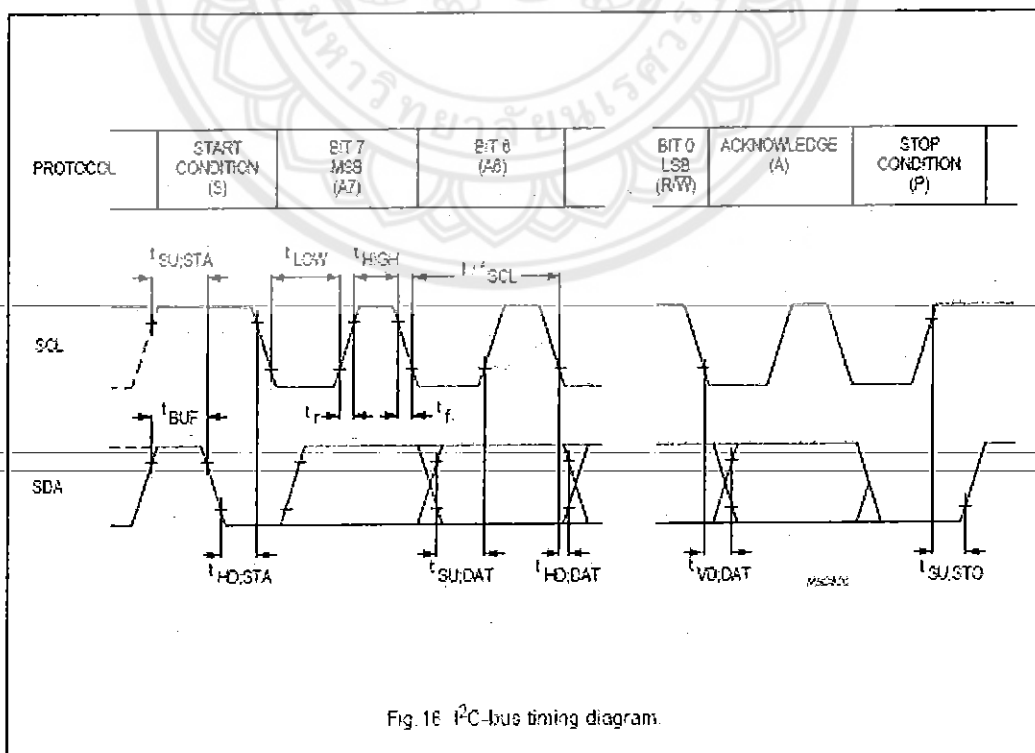
PCF8574

11 I<sup>2</sup>C-BUS TIMING CHARACTERISTICS

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I <sup>2</sup> C-bus timing (see Fig.16; note 1)					
f <sub>SCL</sub>	SCL clock frequency	—	—	100	kHz
t <sub>SW</sub>	tolerable spike width on bus	—	—	100	ns
t <sub>BUF</sub>	bus free time	4.7	—	—	μs
t <sub>SU,STA</sub>	START condition set-up time	4.7	—	—	μs
t <sub>HO,STA</sub>	START condition hold time	4.0	—	—	μs
t <sub>LOW</sub>	SCL LOW time	4.7	—	—	μs
t <sub>HIGH</sub>	SCL HIGH time	4.0	—	—	μs
t <sub>r</sub>	SCL and SDA rise time	—	—	1.0	μs
t <sub>f</sub>	SCL and SDA fall time	—	—	0.3	μs
t <sub>SU,DAT</sub>	data set-up time	250	—	—	ns
t <sub>HO,DAT</sub>	data hold time	0	—	—	ns
t <sub>VO,DAT</sub>	SCL LOW to data out valid	—	—	3.4	μs
t <sub>SU,STO</sub>	STOP condition set-up time	4.0	—	—	μs

## Note

- All the timing values are valid within the operating supply voltage and ambient temperature range and refer to V<sub>IL</sub> and V<sub>IH</sub> with an input voltage swing of V<sub>SS</sub> to V<sub>DD</sub>.

Fig. 16 I<sup>2</sup>C-bus timing diagram.

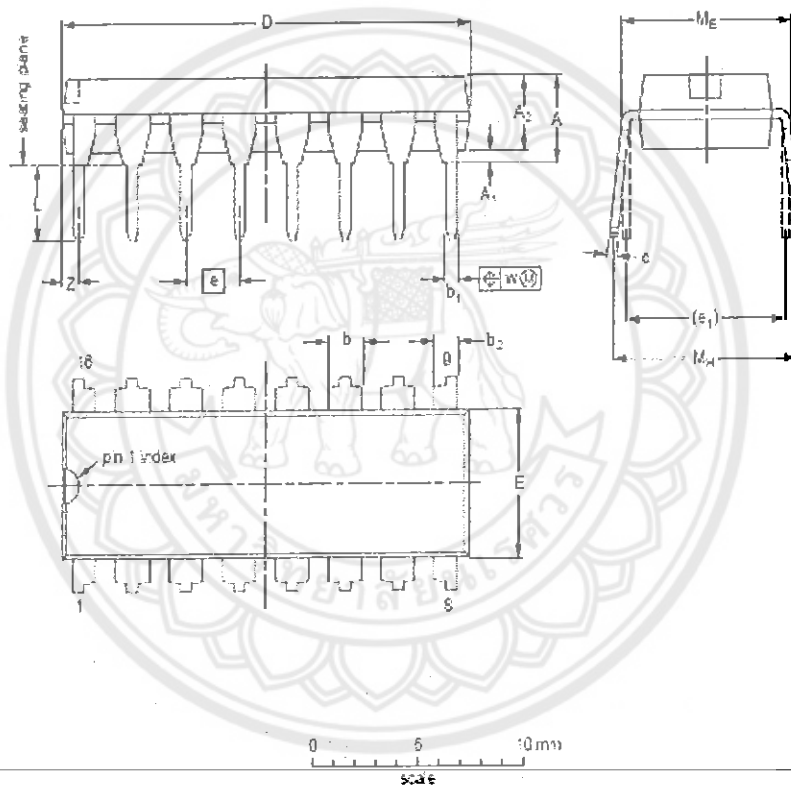
Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

12 PACKAGE OUTLINES

DIP16: plastic dual-in-line package; 16 leads (300 mil)

SOT38-4



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A <sub>1</sub> min.	A <sub>2</sub> max.	b	b <sub>1</sub>	b <sub>2</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	e <sub>1</sub>	L	M <sub>E</sub>	M <sub>H</sub>	w	Z <sup>(1)</sup> max.
mm	4.2	0.51	3.2	1.73 1.30	0.53 0.35	1.25 0.55	0.35 0.23	19.50 12.55	6.46 6.20	2.54	7.62	3.80 3.05	8.25 7.60	10.0 8.3	0.254	0.76
inches	0.17	0.020	0.13	0.068 0.051	0.021 0.015	0.049 0.033	0.014 0.009	0.77 0.73	0.26 0.24	0.10	0.30	0.14 0.12	0.32 0.31	0.39 0.33	0.01	0.030

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

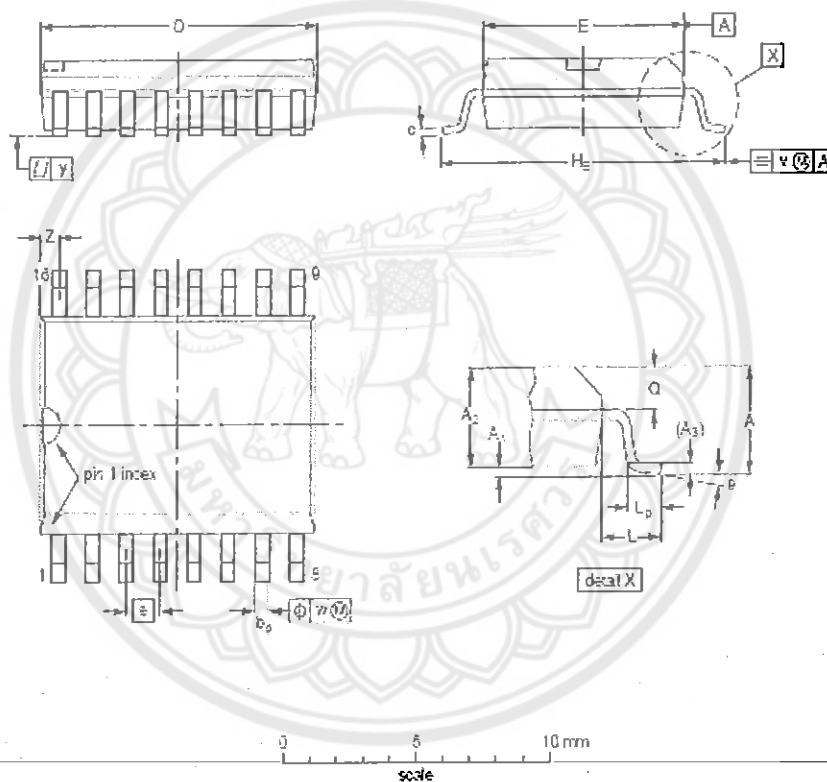
OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT38-4					93-11-F 98-01-14

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

SOT16: plastic small outline package; 16 leads; body width 7.5 mm

SOT162-1



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	b <sub>p</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	H <sub>E</sub>	L	L <sub>P</sub>	Q	v	w	y	z <sup>(1)</sup>	α
mm	2.65	-0.20 -0.10	-2.45 -2.25	0.25	-0.42 -0.36	-0.32 -0.23	-10.5 -10.1	-7.8 -7.4	1.27	-10.65 -10.00	1.4	-1.1 -0.4	-1.1 -1.0	0.25	0.25	0.1	0.9 0.4	5°
inches	0.10	0.012 0.004	0.036 0.089	0.01	0.016 0.014	0.013 0.009	0.41 0.40	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.015	0.043 0.039	0.01	0.01	0.034	0.035 0.016	5°

Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

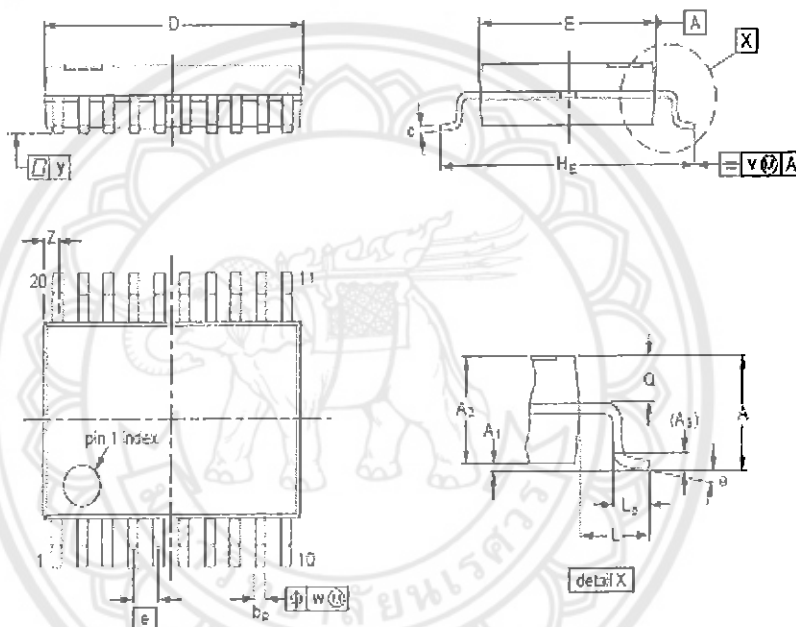
OUTLINE VERSION	REFERENCES			EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ		
SOT162-1	075E03	MS-012			07-06-22 02-12-27

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm

SOT266-1



DIMENSIONS [mm are the original dimensions]

UNIT	A <sub>max</sub>	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	b <sub>P</sub>	c	D <sup>(1)</sup>	E <sup>(1)</sup>	e	H <sub>E</sub>	L	L <sub>P</sub>	Q	v	w	y	Z <sup>(1)</sup>	θ
mm	1.5	0.15 0	1.4 1.2	0.25	0.32 0.20	0.20 0.13	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1.0	0.75 0.45	0.65 0.45	0.2	0.13	0.1	0.49 0.18	10° 0°

Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT266-1		MO-152				<del>92-02-25</del> 92-12-27

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

**13 SOLDERING****13.1 Introduction**

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our "Data Handbook IC26: Integrated Circuit Packages" (document order number 9398 652 90011).

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mount components are mixed on one printed-circuit board. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

**13.2 Through-hole mount packages****13.2.1 SOLDERING BY DIPPING OR BY SOLDER WAVE**

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joints for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ( $T_{stgmax}$ ). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

**13.2.2 MANUAL SOLDERING**

Apply the soldering iron (24 V or less) to the lead(s) of the package, either below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.

**13.3 Surface mount packages****13.3.1 REFLOW SOLDERING**

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stenciling or pressure-syringe dispensing before package placement.

Several methods exist for reflowing, for example, convection or convection/infrared heating in a conveyor type oven. Throughout times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferably be kept below 220 °C for thick/large packages, and below 235 °C for small/thin packages.

**13.3.2 WAVE SOLDERING**

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is preferred to be parallel to the transport direction of the printed-circuit board;
  - smaller than 1.27 mm, the footprint longitudinal axis must be parallel to the transport direction of the printed-circuit board.

The footprint must incorporate solder thieves at the downstream end.

- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time is 4 seconds at 250 °C.

A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

**13.3.3 MANUAL SOLDERING**

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 13.4 Suitability of IC packages for wave, reflow and dipping soldering methods

MOUNTING	PACKAGE <sup>(1)</sup>	SOLDERING METHOD		
		WAVE	REFLOW <sup>(2)</sup>	DIPPING
Through-hole mount	DBS, DIP, HDIP, SDIP, SIL	suitable <sup>(3)</sup>	–	suitable
Surface mount	BGA, LBGA, LFBGA, SQFP, TFBGA, VFBGA	not suitable	suitable	–
	HBCC, HBGA, HLOFP, HSQFP, HSOP, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable <sup>(4)</sup>	suitable	–
	PLCC <sup>(5)</sup> , SQ, SOJ	suitable	suitable	–
	LQFP, QFP, TQFP	not recommended <sup>(5/6)</sup>	suitable	–
	SSOP, TSSOP, VSO	not recommended <sup>(7)</sup>	suitable	–

## Notes

- For more detailed information on the BGA packages refer to the '(LF)BGA Application Note' (AN01026); order a copy from your Philips Semiconductors sales office.
- All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the 'Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods'.
- For SDIP packages, the longitudinal axis must be parallel to the transport direction of the printed-circuit board.
- These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.
- If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
- Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
- Wave soldering is suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.



Remote 8-bit I/O expander for I<sup>2</sup>C-bus

PCF8574

## 14 DATA SHEET STATUS

LEVEL	DATA SHEET STATUS <sup>(1)</sup>	PRODUCT STATUS <sup>(2),3)</sup>	DEFINITION
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

## Notes

1. Please consult the most recently issued data sheet before initiating or completing a design.
2. The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL <http://www.semiconductors.philips.com>.
3. For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

## 15 DEFINITIONS

**Short-form specification** — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

**Limiting values definition** — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

**Application information** — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

## 16 DISCLAIMERS

**Life support applications** — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

**Right to make changes** — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

## +5V-Powered, Multichannel RS-232 Drivers/Receivers

### General Description

The MAX220-MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where  $\pm 12V$  is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5 $\mu$ W. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

### Next-Generation Device Features

- ◆ For Low-Voltage, Integrated ESD Applications  
MAX3222E/MAX3232E/MAX3237E/MAX3241E/  
MAX3246E: +3.0V to +5.5V, Low-Power, Up to  
1Mbps, True RS-232 Transceivers Using Four  
0.1 $\mu$ F External Capacitors (MAX3246E Available  
in a UCSP™ Package)
- ◆ For Low-Cost Applications  
MAX221E:  $\pm 15kV$  ESD-Protected, +5V, 1 $\mu$ A,  
Single RS-232 Transceiver with AutoShutdown™

### Applications

Portable Computers  
Low-Power Modems  
Interface Translation  
Battery-Powered RS-232 Systems  
Multidrop RS-232 Networks

### Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Ordering information continued at end of data sheet.  
\*Contact factory for dice specifications.

### Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value ( $\mu$ F)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.047/0.33	No	—	120	Ultra-low-power, industry-standard pinout.
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✓	120	Available in SO
MAX226 (MAX206)	+5	5/3	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to -13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	—	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1468
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1468 and 1469
MAX239 (MAX209)	+5 and +7.5 to -13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-drain detect or similitude control
MAX244	+5	2/0	4	1.0	No	—	120	High slew rate
MAX245	+5	2/0	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	2/0	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	2/0	0	—	Yes	✓	120	High slew rate, int. caps, no operating modes
MAX248	+5	3/6	4	1.0	Yes	✓	120	High slew rate, selective half-amp enables
MAX249	+5	2/0	4	1.0	Yes	✓	120	Available in quad flatpack package

## ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage ( $V_{CC}$ )	-0.3V to +6V	18-Pin Plastic DIP (derate 11.1mW/°C above +70°C)	889mW
$V_+$ (Note 1)	( $V_{CC} - 0.3V$ ) to +14V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW
$V_-$ (Note 1)	+0.3V to +14V	18-Pin Narrow SO (derate 8.70mW/°C above +70°C)	606mW
Input Voltages		16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
$T_{IN}$	-0.3V to ( $V_{CC} - 0.3V$ )	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
$R_{IN}$ (Except MAX220)	±30V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
$R_{IN}$ (MAX220)	±25V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
$T_{OUT}$ (Except MAX220) (Note 2)	±15V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
$T_{OUT}$ (MAX220)	±13.2V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
Output Voltages		Operating Temperature Ranges	
$T_{OUT}$	±15V	MAX2_AC, MAX2_C	0°C to +70°C
$R_{OUT}$	-0.3V to ( $V_{CC} + 0.3V$ )	MAX2_AE, MAX2_E	-40°C to +85°C
Driver/Receiver Output Short-Circuited to GND	Continuous	MAX2_AM, MAX2_M	-55°C to +125°C
Continuous Power Dissipation ( $T_A = +70°C$ )		Storage Temperature Range	-65°C to +160°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	Lead Temperature (soldering, 10s) (Note 3)	+300°C

**Note 1:** For the MAX220,  $V_+$  and  $V_-$  can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

**Note 2:** Input voltage measured with  $T_{OUT}$  in high-impedance state, SHDN or  $V_{CC} = 0V$ .

**Note 3:** Maximum reflow temperature for the MAX233A is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

( $V_{CC} = +5V \pm 10%$ ,  $C_1 - C_4 = 0.1\mu F$ , MAX220,  $C_1 = 0.047\mu F$ ,  $C_2 - C_4 = 0.33\mu F$ ,  $T_A = T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
<b>RS-232 TRANSMITTERS</b>						
Output Voltage Swing	All transmitter outputs loaded with 3k $\Omega$ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: $V_{CC} = 5.0V$		2.4			
Logic Pullup/Inpurt Current	All except MAX220, normal operation			5	40	$\mu A$
	SHDN = 0V, MAX222/MAX242 shutdown, MAX220			±0.01	±1	
Output Leakage Current	$V_{CC} = 5.5V$ , SHDN = 0V, $V_{OUT} = \pm 15V$ , MAX222/MAX242			±0.01	±10	$\mu A$
	$V_{CC} = \text{SHDN} = 0V$	$V_{OUT} = \pm 15V$		±0.01	±10	
		MAX220, $V_{OUT} = \pm 12V$			±25	
Data Rate				200	118	kbps
Transmitter Output Resistance	$V_{CC} = V_+ = V_- = 0V$ , $V_{OUT} = \pm 2V$		300	10M		$\Omega$
Output Short-Circuit Current	$V_{OUT} = 0V$	$V_{OUT} = 0V$	±7	±22		mA
		MAX220			±60	
<b>RS-232 RECEIVERS</b>						
RS-232 Input Voltage Operating Range					±30	V
	MAX220				±25	
RS-232 Input Threshold Low	$V_{CC} = 5V$	All except MAX243 R2 <sub>N</sub>	0.8	1.3		V
		MAX243 R2 <sub>N</sub> (Note 4)	-3			
RS-232 Input Threshold High	$V_{CC} = 5V$	All except MAX243 R2 <sub>N</sub>		1.8	2.4	V
		MAX243 R2 <sub>N</sub> (Note 4)		-0.5	-0.1	

### ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

( $V_{CC} = +5V \pm 10\%$ ,  $C_1$ – $C_4 = 0.1\mu F$ , MAX220,  $C_1 = 0.047\mu F$ ,  $C_2$ – $C_4 = 0.33\mu F$ ,  $T_A = T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Hysteresis	All except MAX220/MAX243, $V_{CC} = 5V$ , no hysteresis in SHDN		0.2	0.5	1.0	V
	MAX220		0.3			
	MAX243		1			
RS-232 Input Resistance	$T_A = +25^\circ C$ (MAX220)		3	5	7	k $\Omega$
			3	5	7	
TTL/CMOS Output Voltage Low	$I_{OUT} = 3.2mA$		0.2		0.4	V
	$I_{OUT} = 1.6mA$ (MAX220)		0.4			
TTL/CMOS Output Voltage High	$I_{OUT} = -1.0mA$		3.5	$V_{CC} - 0.2$		V
TTL/CMOS Output Short-Circuit Current	Sourcing $V_{OUT} = GND$		-2	-10		mA
	Sinking $V_{OUT} = V_{CC}$		10	30		
TTL/CMOS Output Leakage Current	SHDN = $V_{CC}$ or EN = $V_{CC}$ (SHDN = 0V for MAX222), $0V \leq V_{OUT} \leq V_{CC}$		$\pm 0.05$		$\pm 10$	$\mu A$
EN Input Threshold Low	MAX242		1.4		0.8	V
EN Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5	5.5		V
V <sub>CC</sub> Supply Current (SHDN = $V_{CC}$ ). Figures 5, 6, 11, 19	No load	MAX220	0.5		2	$\mu A$
		MAX222/MAX232A/MAX233A/ MAX242/MAX243	4		10	
	3k $\Omega$ load both inputs	MAX220	12			
		MAX222/MAX232A/MAX233A/ MAX242/MAX243	15			
Shutdown Supply Current	MAX222/ MAX242	$T_A = +25^\circ C$	0.1		10	$\mu A$
		$T_A = 0^\circ C$ to $+70^\circ C$	2		50	
		$T_A = -40^\circ C$ to $+85^\circ C$	2		50	
		$T_A = -55^\circ C$ to $+125^\circ C$	35		100	
SHDN Input Leakage Current	MAX222/MAX242				$\pm 1$	$\mu A$
SHDN Threshold Low	MAX222/MAX242		1.4		0.8	V
SHDN Threshold High	MAX222/MAX242		2.0	1.4		V
Transition Slew Rate	$C_L = 50pF$ to 2500pF, $R_L = 5k\Omega$ to 7k $\Omega$ , $V_{CC} = 5V$ , $T_A = +25^\circ C$ , measured from +3V to -3V or -3V	MAX222/MAX232A/MAX233/ MAX242/MAX243	6	12	30	V/ $\mu s$
		MAX220	1.5	3	30.0	
Transmitter Propagation Delay T <sub>LL</sub> to RS-232 (Normal Operation) Figure 1	T <sub>P,LT</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	1.3		3.5	$\mu s$
		MAX220	4		10	
	T <sub>P,HT</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	1.5		3.5	
		MAX220	5		10	

Note 4: MAX243  $R_{2OUT}$  is guaranteed to be low when  $R_{2IN}$  is  $\geq 0V$  or is floating.

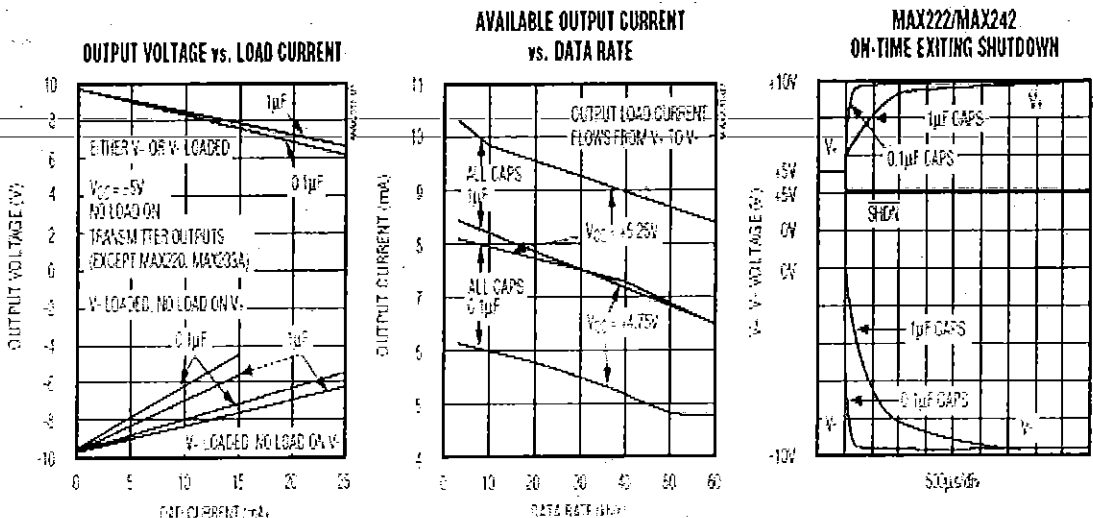
**ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)**

(V<sub>CC</sub> = +5V ± 10%, C<sub>1</sub>-C<sub>4</sub> = 0.1μF, MAX220, C<sub>1</sub> = 0.047μF, C<sub>2</sub>-C<sub>4</sub> = 0.33μF, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Receiver Propagation Delay RS-232 to TLL (Normal Operation), Figure 2	t <sub>PLR</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.5	1		μs
		MAX220	0.6	3		
	t <sub>PLH</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	0.6	1		
		MAX220	0.8	3		
Receiver Propagation Delay RS-232 to TLL (Shutdown), Figure 2	t <sub>PHL</sub>	MAX242	0.5	10		μs
	t <sub>PHL</sub>	MAX242	2.5	10		
Receiver-Output Enable Time, Figure 3	t <sub>ER</sub>	MAX242	125	500		ns
Receiver-Output Disable Time, Figure 3	t <sub>DR</sub>	MAX242	160	500		ns
Transmitter-Output Enable Time (SHDN Goes High), Figure 4	t <sub>ET</sub>	MAX222/MAX242, 0.1μF caps (includes charge-pump start-up)	250			μs
Transmitter-Output Disable Time (SHDN Goes Low), Figure 4	t <sub>DT</sub>	MAX222/MAX242, 0.1μF caps	600			ns
Transmitter + to - Propagation Delay Difference (Normal Operation)	t <sub>PHL</sub> - t <sub>PLH</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	300			ns
		MAX220	2000			
Receiver + to - Propagation Delay Difference (Normal Operation)	t <sub>PLR</sub> - t <sub>PLH</sub>	MAX222/MAX232A/MAX233/ MAX242/MAX243	100			ns
		MAX220	225			

**Typical Operating Characteristics**

**MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243**



## ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

$V_{CC}$ .....	-0.3V to +6V	20-Pin Wide SO (derate 10.00mW/°C above +70°C).....	800mW
$V_+$ .....	( $V_{CC} - 0.3V$ ) to +14V	24-Pin Wide SO (derate 11.76mW/°C above +70°C).....	941mW
$V_-$ .....	+0.3V to -14V	28-Pin Wide SO (derate 12.50mW/°C above +70°C).....	1W
Input Voltages		44-Pin Plastic FP (derate 11.11mW/°C above +70°C).....	889mW
$T_{IN}$ .....	-0.3V to ( $V_{CC} + 0.3V$ )	14-Pin CERDIP (derate 9.09mW/°C above +70°C).....	727mW
$P_{IN}$ .....	±30V	16-Pin CERDIP (derate 10.00mW/°C above +70°C).....	800mW
Output Voltages		20-Pin CERDIP (derate 11.11mW/°C above +70°C).....	889mW
$V_{OUT}$ .....	( $V_+ + 0.3V$ ) to ( $V_- - 0.3V$ )	24-Pin Narrow CERDIP	
$P_{OUT}$ .....	-0.3V to ( $V_{CC} + 0.3V$ )	(derate 12.50mW/°C above +70°C).....	1W
Short-Circuit Duration: $I_{OUT}$	Continuous	24-Pin Sidebrazed (derate 20.0mW/°C above +70°C).....	1.6W
Continuous Power Dissipation ( $T_A = +70^\circ\text{C}$ )		28-Pin SSOP (derate 9.52mW/°C above +70°C).....	762mW
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C).....	800mW	Operating Temperature Ranges	
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C).....	842mW	MAX2 __ C .....	0°C to +70°C
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C).....	889mW	MAX2 __ E .....	-40°C to +85°C
24-Pin Narrow Plastic DIP		MAX2 __ M .....	-55°C to +125°C
(derate 13.33mW/°C above +70°C).....	1.07W	Storage Temperature Range .....	-65°C to +160°C
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C).....	500mW	Lead Temperature (soldering, 10s) (Note 4) .....	+300°C
16-Pin Wide SO (derate 9.52mW/°C above +70°C).....	762mW		

**Note 4:** Maximum reflow temperature for the MAX233/MAX235 is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at those or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241,  $V_{CC} = +5V \pm 10\%$ ; MAX233/MAX235,  $V_{CC} = 5V \pm 5\%$ , C1–C4 = 1.0 $\mu\text{F}$ ; MAX231/MAX239,  $V_{CC} = 5V \pm 10\%$ ;  $V_+ = 7.5V$  to 13.2V;  $T_A = T_{MIN}$  to  $T_{MAX}$ ; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3k $\Omega$ to ground	±5.0	±7.3		V
$V_{CC}$ Power-Supply Current	No load, $T_A = +25^\circ\text{C}$	MAX232/233	5	10	mA
		MAX223/230/234–239/240/241	7	15	
		MAX231/239	0.4	1	
$V_+$ Power-Supply Current		MAX231	1.8	5	mA
		MAX239	5	15	
Shutdown Supply Current	$T_A = +25^\circ\text{C}$	MAX223	15	50	$\mu\text{A}$
		MAX230/235/236/240/241	1	10	
Input Logic Threshold Low	$T_{IN}$ : EN, SHDN (MAX233); EN, SHDN (MAX230/235–241)			0.8	V
Input Logic Threshold High	$T_{IN}$	2.0			V
	EN, SHDN (MAX223); EN, SHDN (MAX230/235/236/240/241)	2.4			
Logic Pullup Current	$T_{IN} = 0V$		15	200	$\mu\text{A}$
Receiver Input Voltage Operating Range		-30		+30	V

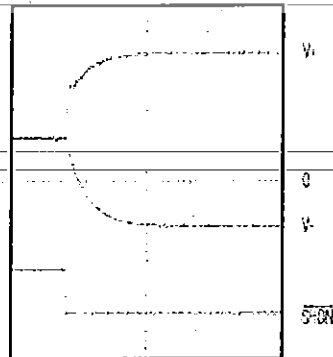
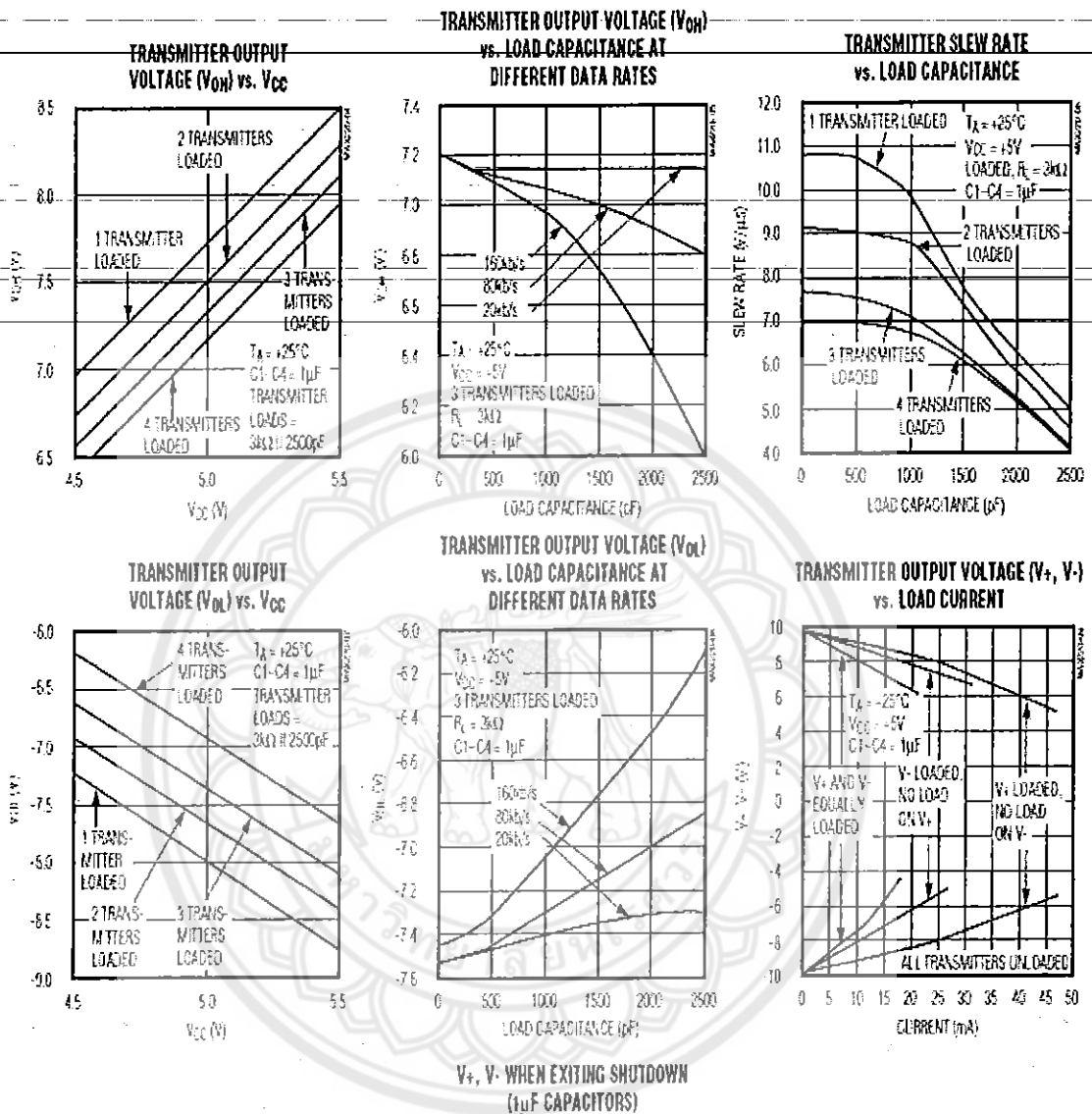
## ELECTRICAL CHARACTERISTICS—MAX223/MAX230—MAX241 (continued)

(MAX223/230/232/234/236/237/238/240/241,  $V_{CC} = +5V \pm 10\%$ ; MAX233/MAX235,  $V_{CC} = 5V \pm 5\%$ ,  $C_1$ - $C_4 = 1.0\mu F$ ; MAX231/MAX239,  $V_{CC} = 5V \pm 10\%$ ;  $V_{+} = 7.5V$  to  $13.2V$ ;  $T_A = T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	$T_A = +25^\circ C$ , $V_{CC} = 5V$	Normal operation $\overline{SHDN} = 5V$ (MAX223) $\overline{SHDN} = 0V$ (MAX235/236/240/241)	0.8	1.2		V
		Shutdown (MAX223) $\overline{SHDN} = 0V$ , $EN = 5V$ ( $R_{4IN}$ , $R_{5IN}$ )	0.6	1.5		
RS-232 Input Threshold High	$T_A = +25^\circ C$ , $V_{CC} = 5V$	Normal operation $\overline{SHDN} = 5V$ (MAX223) $\overline{SHDN} = 0V$ (MAX235/236/240/241)		1.7	2.4	V
		Shutdown (MAX223) $\overline{SHDN} = 0V$ , $EN = 5V$ ( $R_{4IN}$ , $R_{5IN}$ )		1.5	2.4	
RS-232 Input Hysteresis	$V_{CC} = 5V$ , no hysteresis in shutdown		0.2	0.5	1.0	V
RS-232 Input Resistance	$T_A = +25^\circ C$ , $V_{CC} = 5V$		3	5	7	k $\Omega$
TTL/CMOS Output Voltage Low	$I_{OUT} = 1.6mA$ (MAX231/232/233, $I_{OUT} = 3.2mA$ )				0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = -1mA$		3.5	$V_{CC} - 0.4$		V
TTL/CMOS Output Leakage Current	$0V \leq R_{OUT} \leq V_{CC}$ ; $EN = 0V$ (MAX223); $EN = V_{CC}$ (MAX235-241)			0.05	$\pm 10$	$\mu A$
Receiver Output Enable Time	Normal operation	MAX223	600		ns	
		MAX235/236/239/240/241	400			
Receiver Output Disable Time	Normal operation	MAX223	900		ns	
		MAX235/236/239/240/241	250			
Propagation Delay	RS-232 IN to TTL/CMOS OUT, $C_L = 150pF$	Normal operation		0.5	10	$\mu s$
		$\overline{SHDN} = 0V$ (MAX223)	$t_{PHLS}$	4	40	
			$t_{PLHS}$	6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234-241, $T_A = +25^\circ C$ , $V_{CC} = 5V$ , $R_L = 3k\Omega$ to $7k\Omega$ , $C_L = 50pF$ to $2500pF$ , measured from $+3V$ to $-3V$ or $-3V$ to $+3V$		3	5.1	30	V/ $\mu s$
	MAX231/MAX232/MAX233, $T_A = +25^\circ C$ , $V_{CC} = 5V$ , $R_L = 3k\Omega$ to $7k\Omega$ , $C_L = 50pF$ to $2500pF$ , measured from $-3V$ to $-3V$ or $-3V$ to $+3V$			4	30	
Transmitter Output Resistance	$V_{CC} = V_{-} = V_{+} = 0V$ , $V_{OUT} = \pm 2V$		300		$\Omega$	
Transmitter Output Short-Circuit Current			$\pm 10$		mA	

## Typical Operating Characteristics

### MAX223/MAX230-MAX241



\*SHUTDOWN POLARITY IS REVERSED FOR NON-MAX241 PARTS



## +5V-Powered, Multichannel RS-232 Drivers/Receivers

### ABSOLUTE MAXIMUM RATINGS—MAX225/MAX244—MAX249

Supply Voltage ( $V_{CC}$ )	-0.3V to +6V	Continuous Power Dissipation ( $T_A = +70^\circ\text{C}$ )	
Input Voltages		28-Pin Wide SO (derate 12.50mW/°C above +70°C)	1W
$T_{IN}$ , ENA, ENB, ENR, ENT, ENRA,		40-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	0.611W
ENRB, ENTA, ENTB	-0.3V to ( $V_{CC} + 0.3$ V)	44-Pin PLCC (derate 13.33mW/°C above +70°C)	1.07W
$R_{IN}$	$\pm 25$ V	Operating Temperature Ranges	
$T_{OUT}$ (Note 5)	$\pm 15$ V	MAX225C, MAX24C	0°C to +70°C
$R_{OUT}$	-0.3V to ( $V_{CC} + 0.3$ V)	MAX225E, MAX24E	-40°C to +85°C
Short Circuit (one output at a time)		Storage Temperature Range	-65°C to +160°C
$T_{OUT}$ to GND	Continuous	Lead Temperature (soldering, 10s) (Note 6)	+300°C
$R_{OUT}$ to GND	Continuous		

**Note 5:** Input voltage measured with transmitter output in a high-impedance state, shutdown, or  $V_{CC} = 0$ V.

**Note 6:** Maximum reflow temperature for the MAX225/MAX245/MAX246/MAX247 is +225°C.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### ELECTRICAL CHARACTERISTICS—MAX225/MAX244—MAX249

(MAX225,  $V_{CC} = 5.0$ V  $\pm 5\%$ ; MAX244—MAX249,  $V_{CC} = +5.0$ V  $\pm 10\%$ , external capacitors C1—C4 = 1 $\mu$ F;  $T_A = T_{MIN}$  to  $T_{MAX}$ ; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	
<b>RS-232 TRANSMITTERS</b>						
Input Logic Threshold Low			1.4	0.8	V	
Input Logic Threshold High		2	1.4		V	
Logic Fullup/Input Current	Tables 1a-1d	Normal operation		10	50	$\mu$ A
		Shutdown		$\pm 0.01$	$\pm 1$	
Data Rate	Tables 1a-1d, normal operation		120	64	kbps	
Output Voltage Swing	All transmitter outputs loaded with 3k $\Omega$ to GND	$\pm 5$	$\pm 7.5$		V	
Output Leakage Current (Shutdown)	Tables 1a-1d	ENA, ENB, ENT, ENTA, ENTB = $V_{CC}$ , $V_{OUT} = \pm 15$ V		$\pm 0.01$	$\pm 25$	$\mu$ A
		$V_{CC} = 0$ V, $V_{OUT} = \pm 15$ V		$\pm 0.01$	$\pm 25$	
Transmitter Output Resistance	$V_{CC} = V_+ = V_- = 0$ V, $V_{OUT} = \pm 2$ V (Note 7)	300	10M		$\Omega$	
Output Short-Circuit Current	$V_{OUT} = 0$ V	$\pm 7$	$\pm 30$		mA	
<b>RS-232 RECEIVERS</b>						
RS-232 Input Voltage Operating Range				$\pm 25$	V	
RS-232 Input Threshold Low	$V_{CC} = 5$ V	0.8	1.3		V	
RS-232 Input Threshold High	$V_{CC} = 5$ V		1.8	2.4	V	
RS-232 Input Hysteresis	$V_{CC} = 5$ V	0.2	0.5	1.0	V	
RS-232 Input Resistance		3	5	7	k $\Omega$	
TTL/CMOS Output Voltage Low	$I_{OUT} = 3.2$ mA		0.2	0.4	V	
TTL/CMOS Output Voltage High	$I_{OUT} = -1.0$ mA	3.5	$V_{CC} - 0.2$		V	
TTL/CMOS Output Short-Circuit Current	Sourcing $V_{OUT} = \text{GND}$	-2	-10		mA	
	Sinking $V_{OUT} = V_{CC}$	10	30			
TTL/CMOS Output Leakage Current	Normal operation, outputs disabled, Tables 1a-1d, $0$ V $\leq V_{OUT} \leq V_{CC}$ , ENR $_i = V_{CC}$		$\pm 0.05$	$\pm 0.10$	$\mu$ A	

## +5V-Powered, Multichannel RS-232 Drivers/Receivers

### ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249 (continued)

(MAX225,  $V_{CC} = 5.0V \pm 5\%$ ; MAX244–MAX249,  $V_{CC} = +5.0V \pm 10\%$ , external capacitors C1–C4 = 1 $\mu$ F;  $T_A = T_{MIN}$  to  $T_{MAX}$ ; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
<b>POWER SUPPLY AND CONTROL LOGIC</b>					
Operating Supply Voltage	MAX225	4.75		5.25	V
	MAX244–MAX249	4.5		5.5	
$V_{CC}$ Supply Current (Normal Operation)	No load	MAX225	10	20	mA
		MAX244–MAX249	11	30	
	3k $\Omega$ loads on all outputs	MAX225	40		
		MAX244–MAX249	57		
Shutdown Supply Current	$T_A = +25^\circ\text{C}$		8	25	$\mu$ A
	$T_A = T_{MIN}$ to $T_{MAX}$			50	
Control Input	Leakage current			$\pm 1$	$\mu$ A
	Threshold low		1.4	0.8	V
	Threshold high	2.4	1.4		
<b>AC CHARACTERISTICS</b>					
Transition Slaw Rate	$C_L = 60\text{pF}$ to 2500pF, $R_L = 3\text{k}\Omega$ to 7k $\Omega$ , $V_{CC} = 5V$ , $T_A = +25^\circ\text{C}$ , measured from +3V to -3V or -3V to +3V	5	10	30	V/ $\mu$ s
Transmitter Propagation Delay TLL to RS-232 (Normal Operation), Figure 1	$t_{PHLT}$		1.3	3.5	$\mu$ s
	$t_{PLHT}$		1.5	3.5	
Receiver Propagation Delay TLL to RS-232 (Normal Operation), Figure 2	$t_{PHLR}$		0.6	1.5	$\mu$ s
	$t_{PLHR}$		0.6	1.5	
Receiver Propagation Delay TLL to RS-232 (Low-Power Mode), Figure 2	$t_{PHLS}$		0.6	10	$\mu$ s
	$t_{PLHS}$		3.0	10	
Transmitter + to - Propagation Delay Difference (Normal Operation)	$t_{PHLT} - t_{PLHT}$		350		ns
Receiver + to - Propagation Delay Difference (Normal Operation)	$t_{PHLR} - t_{PLHR}$		350		ns
Receiver-Output Enable Time, Figure 3	$t_{ER}$		100	500	ns
Receiver-Output Disable Time, Figure 3	$t_{DR}$		100	500	ns
Transmitter Enable Time	$t_{ET}$	MAX246–MAX249 (excludes charge-pump startup)	5		$\mu$ s
		MAX225/MAX245–MAX249 (includes charge-pump startup)	10		ms
Transmitter Disable Time, Figure 4	$t_{DT}$		100		ns

**Note 7:** The 300 $\Omega$  minimum specification complies with EIA/TIA-232E, but the actual resistance when in shutdown mode or  $V_{CC} = 0V$  is 10M $\Omega$  as is implied by the leakage specification.

Typical Operating Characteristics

MAX225/MAX244-MAX249

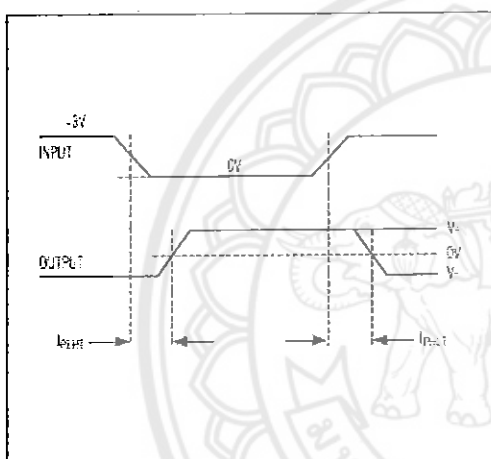
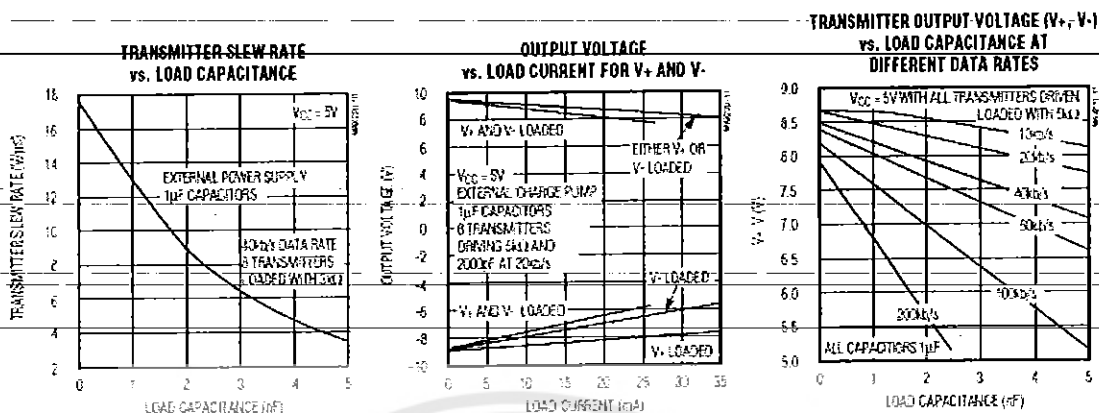


Figure 1 Transmitter Propagation-Delay Timing

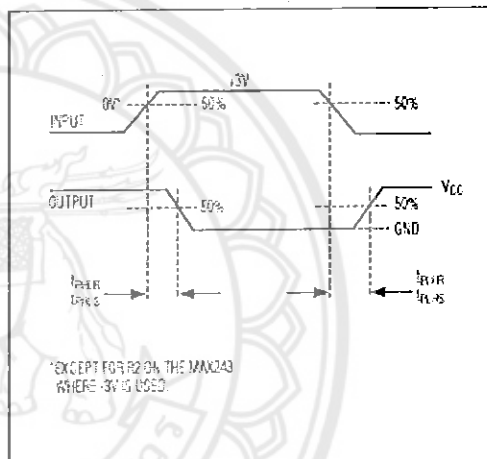


Figure 2 Receiver Propagation-Delay Timing

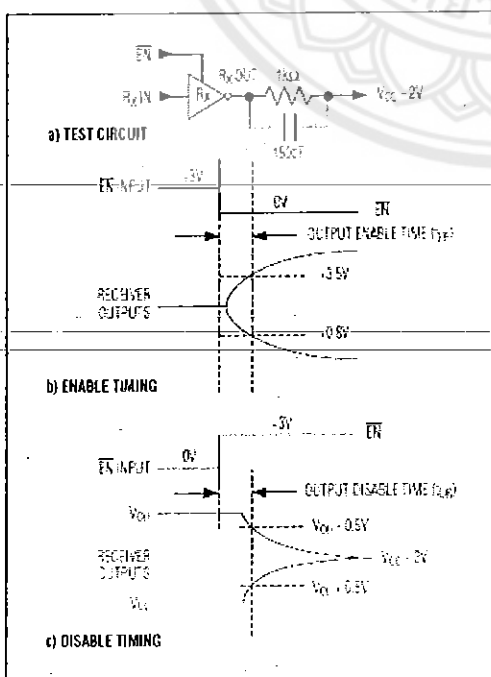


Figure 3 Receiver-Output Enable and Disable Timing

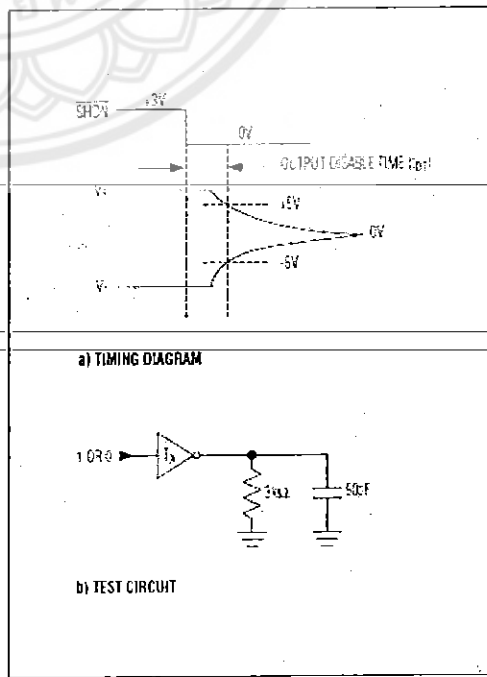


Figure 4 Transmitter-Output Disable Timing

MAX220-MAX249

Table 1d. MAX247/MAX248/MAX249 Control Pin Configurations

ENTA	ENTB	ENRA	ENRB	OPERATION STATUS	TRANSMITTERS			RECEIVERS	
					MAX247	TA1-TA4	TB1-TB4	RA1-RA4	RB1-RB5
					MAX248	TA1-TA4	TB1-TB4	RA1-RA4	RB1-RB4
					MAX249	TA1-TA3	TB1-TB3	RA1-RA5	RB1-RB5
0	0	0	0	Normal Operation		All Active	All Active	All Active	All Active
0	0	0	1	Normal Operation		All Active	All Active	All Active	All 3-State, except RB5 stays active on MAX247
0	0	1	0	Normal Operation		All Active	All Active	All 3-State	All Active
0	0	1	1	Normal Operation		All Active	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
0	1	0	0	Normal Operation		All Active	All 3-State	All Active	All Active
0	1	0	1	Normal Operation		All Active	All 3-State	All Active	All 3-State, except RB5 stays active on MAX247
0	1	1	0	Normal Operation		All Active	All 3-State	All 3-State	All Active
0	1	1	1	Normal Operation		All Active	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247
1	0	0	0	Normal Operation		All 3-State	All Active	All Active	All Active
1	0	0	1	Normal Operation		All 3-State	All Active	All Active	All 3-State, except RB5 stays active on MAX247
1	0	1	0	Normal Operation		All 3-State	All Active	All 3-State	All Active
1	0	1	1	Normal Operation		All 3-State	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
1	1	0	0	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	Low-Power Receive Mode
1	1	0	1	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	All 3-State, except RB5 stays active on MAX247
1	1	1	0	Shutdown		All 3-State	All 3-State	All 3-State	Low-Power Receive Mode
1	1	1	1	Shutdown		All 3-State	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247

MAX220-MAX249

## Detailed Description

The MAX220-MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

### Dual Charge-Pump Voltage Converter

The MAX220-MAX249 have two internal charge-pumps that convert +5V to  $\pm 10V$  (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see the *Typical Operating Characteristics* section), except on the MAX225 and MAX245-MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum  $\pm 5V$  EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245-MAX249, avoid using V+ and V- to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the SHDN pin must be tied to VCC. This is because V+ is internally connected to VCC in shutdown mode.

### RS-232 Drivers

The typical driver output voltage swing is  $\pm 8V$  when loaded with a nominal 5k $\Omega$  RS-232 receiver and VCC = -5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28 specification, which calls for  $\pm 5V$  minimum driver output levels under worst-case conditions. These include a minimum 3k $\Omega$  load, VCC = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400k $\Omega$  input pullup resistors to VCC are built in (except for the MAX220). The pullup resistors force the outputs of unused drivers low because all drivers invert. The internal input output resistors typically source 12 $\mu A$  except in shutdown mode where the pullups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25 $\mu A$ )—when in shutdown

mode, in three-state mode, or when device power is removed. Outputs can be driven to  $\pm 15V$ . The power-supply current typically drops to 8 $\mu A$  in shutdown mode. The MAX220 does not have pullup resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or VCC.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239-MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1 $\mu A$  with the driver output pulled to ground. The driver output leakage remains less than 1 $\mu A$ , even if the transmitter output is backdriven between 0V and (VCC + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1k $\Omega$  series impedance. The transmitter is also zener clamped to approximately VCC + 6V, with a series impedance of 1k $\Omega$ .

The driver output slew rate is limited to less than 30V/ $\mu s$  as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/ $\mu s$  unloaded and 10V/ $\mu s$  loaded with 3 $\Omega$  and 2500pF.

### RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28-eves.

The receiver inputs withstand an input overvoltage up to  $\pm 25V$  and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	SHDN	EN	EN(R)	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

nominal 5k $\Omega$  values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.

#### Low-Power Receive Mode

The low-power receive mode feature of the MAX223, MAX242, and MAX245-MAX249 puts the IC into shutdown mode but still allows it to receive information. This is important for applications where systems are periodically awakened to look for activity. Using low-power receive mode, the system can still receive a signal that will activate it on command and prepare it for communication at faster data rates. This operation conserves system power.

#### Negative Threshold—MAX243

The MAX243 is pin compatible with the MAX232A, differing only in that RS-232 cable fault protection is removed on one of the two receiver inputs. This means that control lines such as CTS and RTS can either be driven or left floating without interrupting communication. Different cables are not needed to interface with different pieces of equipment.

The input threshold of the receiver without cable fault protection is -0.8V rather than +1.4V. Its output goes positive only if the input is connected to a control line that is actively driven negative. If not driven, it defaults to the 0 or "OK to send" state. Normally, the MAX243's other receiver (+1.4V threshold) is used for the data line (TD or RD), while the negative-threshold receiver is connected to the control line (DTR, DTS, CTS, RTS, etc.).

Other members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA-232E specifications. This means a receiver output goes high whenever its input is driven negative, left floating, or shorted to ground. The high output tells the serial communications IC to stop sending data. To avoid this, the control lines must either be driven or connected with jumpers to an appropriate positive voltage level.

#### Shutdown—MAX222-MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown.

On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5 $\mu$ s for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ( $\overline{\text{EN}}$  for the MAX242 and  $\overline{\text{EN}}$  for the MAX223) that allows receiver output control independent of  $\overline{\text{SHDN}}$  ( $\overline{\text{SHDN}}$  for MAX241). With all other devices,  $\overline{\text{SHDN}}$  ( $\overline{\text{SHDN}}$  for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25 $\mu$ A and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the  $\overline{\text{ENR}}$  input. On the MAX245, eight of the receiver outputs are controlled by the  $\overline{\text{ENR}}$  input, while the remaining two receivers (RA5 and RB5) are always active. RA1-RA4 and RB1-RB4 are put in a three-state mode when  $\overline{\text{ENR}}$  is a logic high.

#### Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245-MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

**MAX220-MAX249**

Tables 1a-1d define the control states. The MAX244 has no control pins and is not included in these tables.

The MAX246 has ten receivers and eight drivers with two control pins, each controlling one side of the device. A logic high at the A-side control input ( $\overline{ENA}$ ) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input ( $\overline{ENB}$ ) causes the four B-side drivers and receivers to go into a three-state mode. As in the MAX245, one A-side and one B-side receiver (RA5 and RB5) remain active at all times. The entire device is put into shutdown mode when both the A and B sides are disabled ( $\overline{ENA} = \overline{ENB} = +5V$ ).

The MAX247 provides nine receivers and eight drivers with four control pins. The  $\overline{ENRA}$  and  $\overline{ENRB}$  receiver enable inputs each control four receiver outputs. The  $\overline{ENTA}$  and  $\overline{ENTB}$  transmitter enable inputs each control four drivers. The ninth receiver (RB5) is always active. The device enters shutdown mode with a logic high on both  $\overline{ENTA}$  and  $\overline{ENTB}$ .

The MAX248 provides eight receivers and eight drivers with four control pins. The  $\overline{ENRA}$  and  $\overline{ENRB}$  receiver enable inputs each control four receiver outputs. The  $\overline{ENTA}$  and  $\overline{ENTB}$  transmitter enable inputs control four drivers each. This part does not have an always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both  $\overline{ENTA}$  and  $\overline{ENTB}$ .

The MAX249 provides ten receivers and six drivers with four control pins. The  $\overline{ENRA}$  and  $\overline{ENRB}$  receiver enable inputs each control five receiver outputs. The  $\overline{ENTA}$  and  $\overline{ENTB}$  transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both  $\overline{ENTA}$  and  $\overline{ENTB}$ . In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kb/s.

**Applications Information**

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, VCC should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.

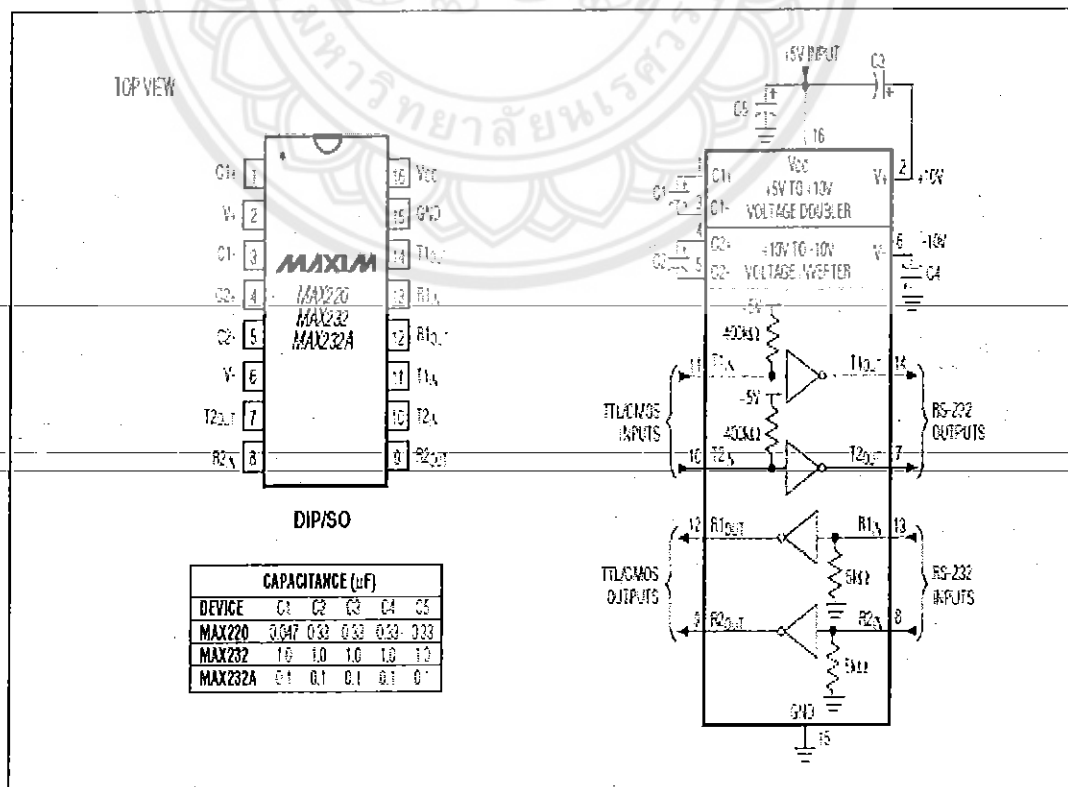


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

ภาคผนวก ข.  
Code ของโปรแกรม

/\*\*\*\*\*\*

This program was produced by the  
CodeWizardAVR V1.24.8b Professional

Automatic Program Generator

© Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 7/5/2009

Author : F4CG

Company : F4CG

Comments:

Chip type : ATmega8515L

Program type : Application

Clock frequency : 20.000000 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 128

\*\*\*\*\*/

```
#include <mega8515.h> //ประกาศใช้งาน Library ATmega8515
#include <delay.h> //ประกาศใช้งาน Library delay หน่วงเวลา
#define ADDR_1 0b01000000 //กำหนด ID ติดต่อ I2C ตัวที่ 1
#define ADDR_2 0b01000010 //กำหนด ID ติดต่อ I2C ตัวที่ 2
```



```

#define ADDR_3 0b01000100 //กำหนด ID ติดต่อ I2C ตัวที่ 3
#define ADDR_4 0b01000110 //กำหนด ID ติดต่อ I2C ตัวที่ 4

char ii,DATAOUT[41]; //ประกาศตัวแปร ii,DATAOUT
int cnt_timer=0; //ประกาศตัวแปร cnt_timer
char temp[20]; //ประกาศตัวแปร temp
int Dataout1; //ประกาศตัวแปร Dataout1
char read1,read2,read3,read4; //ประกาศตัวแปร read1,read2,read3,read4
char status_save=0; //ประกาศตัวแปร status_save
int cnt_day,icnt_day; //ประกาศตัวแปร cnt_day,icnt_day
unsigned char cnt_min=0,cnt_sec=0,cnt_hr=0,icnt_hr; //ประกาศตัวแปร
cnt_min=0,cnt_sec=0,cnt_hr=0,icnt_hr

char warning1=0; //ประกาศตัวแปร warning1
char warning2=0; //ประกาศตัวแปร warning2
char warning3=0; //ประกาศตัวแปร warning3
char warning4=0; //ประกาศตัวแปร warning4

char sign1,sign2,sign3; //ประกาศตัวแปร sign1,sign2,sign3

char inbit1=0; //ประกาศตัวแปร inbit1
int cnt_sign1,cnt_sign2,cnt_sign3,scnt_sign1=0; //ประกาศตัวแปร
cnt_sign1,cnt_sign2,cnt_sign3,scnt_sign1
//I2C-Bus-functions
#asm
.equ __i2c_port=0x15 ;PORTC //กำหนด ให้พอร์ต C ทำหน้าที่ติดต่อแบบ I2C
.equ __sda_bit=0 //กำหนดให้พอร์ต C บิต 0 เป็น SDA
.equ __scl_bit=1 //กำหนดให้พอร์ต C บิต 1 เป็น SCL
#endasm
#include <i2c.h> //ประกาศใช้งาน library I2C

```

```

// DS1307 Real Time Clock functions
#include <ds1307.h> //ประกาศใช้งาน library DS1307

// Alphanumeric LCD Module functions
#include <lcd.h> //ประกาศใช้งาน library LCD

.equ _lcd_port=0x1B;PORTA //กำหนดให้พอร์ต-A ทำหน้าที่ติดกับจอ-LCD

#include <stdio.h> //ประกาศใช้งาน library stdio
#include <string.h> //ประกาศใช้งาน library string

// Declare your global variables here
unsigned char hour,minute,second;
unsigned char year,month,date;

int PCF8574_read(char chip) //ฟังก์ชันอ่านข้อมูลจาก PC8574
{
    int indata; //ประกาศตัวแปร indata
    //printf("start i2c\r\n");
    i2c_start(); //printf("write i2c\r\n"); //เริ่มต้นติดต่อ I2C
    i2c_write(chip+1); //printf("read\r\n"); //เขียนข้อมูลลงไปใน chip I2C
    indata = i2c_read(0); //เอาข้อมูลที่อ่านได้มาเก็บใน indata
    i2c_stop(); //printf("stop i2c\r\n"); //หยุดการติดต่อ I2C
    delay_ms(50); // หน่วงเวลา 50 ms
    return(indata); //กลับไป indata
}

```

---

```
// Timer 0 overflow interrupt service routine
```

---

```
interrupt [TIM0_OVF] void timer0_ovf_isr(void) //ฟังก์ชัน Interrupt Timer1
```

```
{
```

```
// Place your code here
```

---

```
cnt_timer++; //ทำการบวก cnt_timer ทีละ 1 ไปเรื่อยๆ
```

```
if (cnt_timer > 180) //ถ้า cnt_timer มีค่ามากกว่า 180
```

```
{
```

```
cnt_timer=0; //เคลียร์ค่าในตัวแปร cnt_timer
```

```
cnt_sec++; //ทำการบวก cnt_sec ทีละ 1 ไปเรื่อยๆ
```

```
//sprintf(temp,"Time= %i:%i:%i %i Days
```

```
",(int)cnt_hr,(int)cnt_min,(int)cnt_sec,cnt_day);
```

```
//lcd_gotoxy(5,3); lcd_puts(temp);
```

```
if (status_save) //ถ้า status_save มีค่าเท่ากับ 1
```

```
{
```

```
lcd_gotoxy(16,3); sprintf(temp,"Save"); lcd_puts(temp); ส่งคำว่า save ออก LCD
```

```
}
```

---

```
if (cnt_sec >= 60) //ถ้า cnt_sec มีค่ามากกว่าหรือเท่ากับ 60
```

```
{
```

```
cnt_sec=0; //เคลียร์ค่าในตัวแปร cnt_sec
```

```
cnt_min++; //ทำการบวก cnt_min ทีละ 1 ไปเรื่อยๆ
```

```
if (cnt_min >= 60) //ถ้า cnt_min มีค่ามากกว่าหรือเท่ากับ 60
```

```
{
```

```

cnt_min=0; //เคลียร์ค่าในตัวแปร cnt_min
cnt_hr++; //ทำการบวก cnt_hr ทีละ 1 ไปเรื่อยๆ
icnt_hr++; //ทำการบวก icnt_hr ทีละ 1 ไปเรื่อยๆ
}

if(cnt_hr >=24) //ถ้า มีค่าเท่ากับ cnt_hr มีค่ามากกว่าหรือ เท่ากับ 24
{
cnt_day++; //ทำการบวก icnt_day ทีละ 1 ไปเรื่อยๆ
icnt_day++; //ทำการบวก icnt_day ทีละ 1 ไปเรื่อยๆ
}
}
}

}

void SaveDataToUSB(void) //ฟังก์ชัน save ข้อมูลลงไปใน flash drive
{
printf("ECS\r"); //while(UDR != '>'); //ส่งอักขระ ECS เพื่อติดต่อกับ โมดูล flash drive

printf("IPA\r"); //while(UDR != '>'); //ส่งอักขระ ECS เพื่อติดต่อกับ โมดูล flash drive

printf("OPW datalog.txt\r"); //while(!end); //สร้าง ไฟล์ และเปิด datalog.txt เพื่อทำการเขียน
ข้อมูล
delay_ms(2000); // หน่วงเวลา 2000 ms

```

```

printf("WRF 41\r");//while(!end); //จำนวนอักขระที่จะเขียน คือ 41 อักขระ
delay_ms(2000); // หน่วงเวลา 2000 ms
for(ii=0;ii<41;ii++) //ส่งอักขระ จำนวน 41 อักขระ
{
    putchar(DATAOUT[ii]);
    delay_ms(100); // หน่วงเวลา 100 ms
}

```

```

printf("\r"); //เว้นบรรทัด
printf("\r"); //เว้นบรรทัด
delay_ms(1000); // หน่วงเวลา 1000 ms
printf("CLF datalog.txt\r"); //เปิด ไฟล์ datalog.txt เมื่อเขียนข้อมูลเสร็จ
printf("\r"); //เว้นบรรทัด
printf("\r"); //while(UDR != '>');//เว้นบรรทัด

status_save=0; //กำหนด status_save เท่ากับ 0
scnt_sign1=0; //เคลียร์ค่าในตัวแปร scnt_sign1
lcd_gotoxy(16,3); sprintf(temp,"Stby"); lcd_puts(temp); //ส่งคำว่า Stby ออกไปที่ LCD แถว
ที่ 3 ตำแหน่งที่ 16
delay_ms(1000); // หน่วงเวลา 1000 ms

```

```

void main(void) //ฟังก์ชันหลัก
{
    // Declare your local variables here
    // unsigned char hour,minute,second;
    // unsigned char year,month,date;
    // Declare your local variables here

```

```
int top = 0;
```

---

```
// Input/Output Ports initialization //กำหนดค่าเริ่มต้น พอร์ต A
```

```
// Port A initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTA=0x00; //กำหนดค่าเริ่มต้น พอร์ต A เท่ากับ 0x00 ในเลขฐาน 16
```

```
DDRA=0x00; //กำหนดพอร์ต A ทั้งหมดทำหน้าที่เป็นอินพุต
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00; //กำหนดค่าเริ่มต้น พอร์ต B เท่ากับ 0x00 ในเลขฐาน 16
```

```
DDRB=0x00; //กำหนดพอร์ต B ทั้งหมดทำหน้าที่เป็นอินพุต
```

```
// Port C initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTC=0x00; //กำหนดค่าเริ่มต้น พอร์ต C เท่ากับ 0x00 ในเลขฐาน 16
```

```
DDRC=0x0f; //กำหนดพอร์ต C ทั้งหมดทำหน้าที่ บิต 7-4 เป็นเอาต์พุต บิต 3-0
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTD=0x00; //กำหนดค่าเริ่มต้น พอร์ต D เท่ากับ 0x00 ในเลขฐาน 16
```

```
DDRD=0x00; //กำหนดพอร์ต D ทั้งหมดทำหน้าที่เป็นอินพุต
```

```
// Port E initialization
```

```
// Func2=In Func1=In Func0=In
```

```
// State2=T State1=T State0=T
```

```
PORTE=0x00; //กำหนดค่าเริ่มต้น พอร์ต E เท่ากับ 0x00 ในเลขฐาน 16
```

DDRE=0x00; //กำหนดพอร์ต E ทั้งหมดทำหน้าที่เป็นอินพุต

// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped

// Mode: Normal top=FFh

// OC0 output: Disconnected

TCCR0=0x04;

TCNT0=0x00;

OCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: Timer 1 Stopped

// Mode: Normal top=FFFFh

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer 1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0x00;

TCCR1B=0x00;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

```
OCR1BH=0x00;
```

```
OCR1BL=0x00;
```

---

```
// External Interrupt(s) initialization
```

```
// INT0: Off
```

```
// INT1: Off
```

```
// INT2: Off
```

---

```
MCUCR=0x00;
```

```
EMUCR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x02;
```

```
// USART initialization //เซตอัตราบิตเรท ในการส่งผ่านข้อมูลผ่านพอร์ตอนุกรม ที่ 9600b/s
```

```
// Communication Parameters: 8 Data, 1 Stop, No Parity
```

```
// USART Receiver: On
```

```
// USART Transmitter: On
```

```
// USART Mode: Asynchronous
```

```
// USART Baud rate: 9600
```

```
UCSRA=0x00;
```

```
UCSRB=0x18;
```

```
UCSRC=0x86;
```

---

```
UBRRH=0x00;
```

```
UBRRL=0x47;
```

---

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
// I2C Bus initialization
```



```
i2c_init(); //ประกาศฟังก์ชันเริ่มต้น I2C
```

```
// DS1307 Real Time Clock initialization
```

```
// Square wave output on pin SQW/OUT: Off
```

```
// SQW/OUT pin state: 0
```

```
rtc_init(0,0,0); //เซตค่าเริ่มต้น ของ DS1307 ให้ SQW/OUT: ไม่ทำงาน และ ขา SQW/OUT มีสถานะเท่ากับ 0
```

```
rtc_set_date(16,5,9); //ฟังก์ชันเซตวันที่
```

```
rtc_set_time(19,10,59); //ฟังก์ชันเซตเวลา
```

```
// LCD module initialization
```

```
lcd_init(20); //ฟังก์ชัน กำหนดให้ LCD แสดงผล 20 อักขระ
```

```
#asm("sei")
```

```
//printf("hello 9600\n");
```

```
sprintf(temp," Waiting ! USB "); delay_ms(200); //ส่งคำว่า "Waiting ! USB" ออกทางหน้าจอ
```

```
LCD บรรทัดที่ 1 ตำแหน่ง 0
```

```
lcd_gotoxy(0,1); lcd_puts(temp);
```

```
lcd_gotoxy(16,3); sprintf(temp,"Conn"); lcd_puts(temp); //ส่งคำว่า "Conn" ออกทางหน้าจอ
```

```
LCD บรรทัดที่ 3 ตำแหน่ง 16
```

```
Dataout1=0; //เคลียร์ค่าในตัวแปร Dataout1
```

```
//read=0;
```

```
read1=0; //เคลียร์ค่าในตัวแปร read1
```

```

read2=0; //เคลียร์ค่าในตัวแปร read2
read3=0; //เคลียร์ค่าในตัวแปร read3
read4=0; //เคลียร์ค่าในตัวแปร read4

```

```

while (1)

```

```

{
    rtc_get_date(&date,&month,&year); //อ่านค่าวัน เดือน ปี มาจาก IC DS1307
    rtc_get_time(&hour,&minute,&second); //อ่านค่า ชั่วโมง นาที วินาที มาจาก IC DS1307
    lcd_gotoxy(0,0); sprintf(temp,"%2u-%2u-200%u
    %2u:%2u:%2u",date,month,year,hour,minute,second); lcd_puts(temp); //ส่งค่า วัน เดือน ปี ชั่วโมง
    นาที วินาที ออกทางหน้าจอ LCD บรรทัดที่ 0 ตำแหน่ง 0
    delay_ms(100);

    warning1='0'; //เคลียร์ค่าในตัวแปร warning1
    warning2='0'; //เคลียร์ค่าในตัวแปร warning2
    warning3='0'; //เคลียร์ค่าในตัวแปร warning3
    warning4='0'; //เคลียร์ค่าในตัวแปร warning4

```

```

    read1 = PCF8574_read(ADDR_1); //อ่านค่ามาจาก PCF8574 ตัวที่ 1 มาเก็บไว้ในตัวแปร
    read1

```

```

    sign1 = ((read1 & 0x01) | ((read1 & 0x02) >> 1));

```

```

    if ((sign1==1) && (inbit1==0)) //ฟังก์ชันเปรียบเทียบเงื่อนไข ถ้าเป็นจริง คือ sign1==1 แอนด์
    กับ (inbit1==0) มีค่าเป็น 1

```

```

    {

```

```

        cnt_sign1 = cnt_sign1+2; //cnt_sign1 บวกค่าเพิ่มไปเรื่อยๆครั้งละ 2

```

```
inbit1=1;
```

```
scnt_sign1 = scnt_sign1+2; /scent_sign1 บวกค่าเพิ่มไปเรื่อยๆครั้งละ 2
```

```
if (scnt_sign1 == 50) //ถ้า scnt_sign1 == 50 จริง
```

```
{
```

```
    //Dataout1==50;
```

```
    status_save=1; //ส่งข้อมูลไป save flash drive
```

```
    if (top > 500) { ถ้า top มีค่ามากกว่า 500
```

```
        top = 0; //เคลียร์ค่า top
```

```
    }
```

```
    else //ถ้าไม่
```

```
    {
```

```
        top = top+5; //topบวก 5
```

```
    }
```

```
    //cnt_sign1=0;
```

```
    }
```

```
    else
```

```
    { status_save=0; //หยุดส่งข้อมูลไป save flash drive
```

```
    }
```

```
}
```

```
if (sign1 == 0) inbit1=0;
```

```
sign2 = (read1 & 0x80) >> 7; //sign2 มีค่าเท่ากับค่า read1 ที่อ่านมาได้ แอนด์กับ 0x80 เลื่อน  
บิตข้อมูลไปทางขวา 7 บิต
```

```
if (sign2 == 1) //ถ้า sign2 == 1
```

```
{
```

```
    cnt_sign2=1; // cnt_sign2 มีค่าเท่ากับ 1
```

```
Dataout1=1; // Dataout1 มีค่าเท่ากับ 1
status_save=1; //ส่งข้อมูลไป save flash drive
```

```
}
```

```
sign3=(read1 & 0x04) >> 2; //sign3 มีค่าเท่ากับค่า read1 ที่อ่านมาได้ แอนด์กับ 0x84 เลื่อน
บิตข้อมูลไปทางขวา 2 บิต
```

```
if(sign3 == 1) // ถ้า sign3 == 1
{
    cnt_sign3=1; // cnt_sign3 มีค่าเท่ากับ 1
    Dataout1=2; // Dataout1 มีค่าเท่ากับ 2
    status_save=1; //ส่งข้อมูลไป save flash drive
}
```

```
read2 = PCF8574_read(ADDR_2); //อ่านค่ามาจาก PCF8574 ตัวที่ 2 มาเก็บไว้ในตัวแปร
read2
read3 = PCF8574_read(ADDR_3); //อ่านค่ามาจาก PCF8574 ตัวที่ 3 มาเก็บไว้ในตัวแปร
read3
read4 = PCF8574_read(ADDR_4); //อ่านค่ามาจาก PCF8574 ตัวที่ 4 มาเก็บไว้ในตัวแปร
read4
```

```
//=====cal=====
```

```
//icnt_hr=1;
```

```
/* if (icnt_hr >=1)
```

```
{
```

```
        if (scnt_sign1 == 50) {status_save=1;scnt_sign1=0;}
        // else    if (scnt_sign1 == 200) {warning1 = 'M'; status_save=1;}
        //else    if (scnt_sign1 == 100) {warning1 = 'L'; status_save=1;}

        // else    if (scnt_sign1 == 50) {warning1 = 'L'; status_save=1;scnt_sign1=0; }

    }*/
```

```
/*if (icnt_day >=3)
{
    if (scnt_sign1 >= 200) {warning1 = 'H'; status_save=1;}
    else    if (scnt_sign1 >= 185) {warning1 = 'M'; status_save=1;}
    else    if (scnt_sign1 >= 170) {warning1 = 'L'; status_save=1;}
    icnt_day =0;
    cnt_day =0;
    scnt_sign1=0;
}
else if (icnt_day ==1)
{
```

```
        if (scnt_sign1 >= 100) {warning1 = 'H'; status_save=1;}
        else    if (scnt_sign1 >= 85) {warning1 = 'M'; status_save=1;}
        else    if (scnt_sign1 >= 70) {warning1 = 'L'; status_save=1;}
```

```
*/
```

```
if (read2 > 230) {warning2 = 'H'; status_save=1;} //ถ้าค่าใน read2 มากกว่า 230 แล้ว
warning2 เก็บค่าอักขระ 'H' ส่งข้อมูลไป save flash drive
```

```
else if (read2 > 210) {warning2 = 'M'; status_save=1;} //ถ้าค่าใน read2 มากกว่า 210 แล้ว
warning2 เก็บค่าอักขระ 'M' ส่งข้อมูลไป save flash drive
```

```
else if (read2 > 190) {warning2 = 'L'; status_save=1;} //ถ้าค่าใน read2 มากกว่า 190 แล้ว
warning2 เก็บค่าอักขระ 'L' ส่งข้อมูลไป save flash drive
```

```
if (read3 > 50) {warning3 = 'H'; status_save=1;} //ถ้าค่าใน read3 มากกว่า 50 แล้ว warning3
เก็บค่าอักขระ 'H' ส่งข้อมูลไป save flash drive
```

```
if (read4 > 10) {warning4 = 'H'; status_save=1;} //ถ้าค่าใน read4 มากกว่า 10 แล้ว warning4
เก็บค่าอักขระ 'H' ส่งข้อมูลไป save flash drive
```

```
//top
```

```
sprintf(temp, "S1[%i%i.%i] S2[%i]
```

```
S3[%i]", scnt_sign1/100, (scnt_sign1%100)/10, scnt_sign1%10, cnt_sign2, cnt_sign3); //ส่งค่า
scnt_sign1, cnt_sign2, cnt_sign3 ออกทางหน้าจอ LCD บรรทัดที่ 1 ตำแหน่ง 0
```

```
lcd_gotoxy(0,1); lcd_puts(temp);
```

```
sprintf(temp, "[2]=%i[%e] ", (int)read2, warning2); //ส่งค่า read2, warning2 ออกทางหน้าจอ
```

```
LCD บรรทัดที่ 2 ตำแหน่ง 0
```

```
lcd_gotoxy(0,2); lcd_puts(temp);
```

```

    sprintf(temp, "[3]=%i[%c] ", (int)read3, warning3); //ส่งค่า read3, warning3 ออกทางหน้าจอ
LCD บรรทัดที่ 2 ตำแหน่ง 10
    lcd_gotoxy(10,2); lcd_puts(temp);

```

```

    sprintf(temp, "[4]=%i[%c] ", (int)read4, warning4); //ส่งค่า read4, warning4 ออกทางหน้าจอ
LCD บรรทัดที่ 3 ตำแหน่ง 10

```

```

    lcd_gotoxy(0,3); lcd_puts(temp);

// ===== save data to memory =====

if (status_save == 1) //ถ้า status_save เท่ากับ 1
{

// $A000B0C0D000E000F000!00/00/00 00/00/00
DATAOUT[0]='$'; //ส่งอักขระ $ ไป save ใน Flash Drive
DATAOUT[1]='A'; //ส่งอักขระ A ไป save ใน Flash Drive
//DATAOUT[2]= (cnt_sign1/100)+'0';
//DATAOUT[3]= (cnt_sign1%100)/10+'0';
//DATAOUT[4]= (cnt_sign1%10)+'0';
DATAOUT[2]= (top/100)+'0'; //ส่งค่า top/100)+'0' ไป save ใน Flash Drive
DATAOUT[3]= (top%100)/10+'0'; //ส่งค่า (top%100)/10+'0' ไป save ใน Flash Drive
DATAOUT[4]= (top%10)+'0'; //ส่งค่า (top%10)+'0' ไป save ใน Flash Drive
DATAOUT[5]='B'; //ส่งอักขระ B ไป save ใน Flash Drive
DATAOUT[6]=cnt_sign2+'0'; //ส่งค่า cnt_sign2+'0' ไป save ใน Flash Drive
DATAOUT[7]='C'; //ส่งอักขระ C ไป save ใน Flash Drive
DATAOUT[8]=cnt_sign3+'0'; //ส่งค่า cnt_sign3+'0' ไป save ใน Flash Drive
DATAOUT[9]='D'; //ส่งอักขระ D ไป save ใน Flash Drive
DATAOUT[10]= (read2/100)+'0'; //ส่งค่า (read2/100)+'0' ไป save ใน Flash Drive

```

DATAOUT[11]=(read2%100)/10+'0'; //ส่งค่า (read2%100)/10+'0' ไป save ใน Flash

Drive

DATAOUT[12]=(read2%10)+'0'; //ส่งค่า (read2%10)+'0' ไป save ใน Flash Drive

DATAOUT[13]='E'; //ส่งอักขระ E ไป save ใน Flash Drive

DATAOUT[14]=(read3/100)+'0'; //ส่งค่า (read3/100)+'0' ไป save ใน Flash Drive

DATAOUT[15]=(read3%100)/10+'0'; //ส่งค่า (read3%100)/10+'0' ไป save ใน Flash

Drive

DATAOUT[16]=(read3%10)+'0'; //ส่งค่า (read3%10)+'0'; ไป save ใน Flash Drive

DATAOUT[17]='F'; //ส่งอักขระ F ไป save ใน Flash Drive

DATAOUT[18]=(read4/100)+'0'; //ส่งค่า (read4/100)+'0' ไป save ใน Flash Drive

DATAOUT[19]=(read4%100)/10+'0'; //ส่งค่า (read4%100)/10+'0' ไป save ใน Flash

Drive

DATAOUT[20]=(read4%10)+'0'; //ส่งค่า (read4%10)+'0' ไป save ใน Flash Drive

DATAOUT[21]='!'; //ส่งอักขระ ! ไป save ใน Flash Drive

DATAOUT[22]=(date%100)/10+'0'; //ส่งค่า (date%100)/10+'0' ไป save ใน Flash Drive

DATAOUT[23]=(date%10)+'0'; //ส่งค่า (date%10)+'0' ไป save ใน Flash Drive

DATAOUT[24]=': '; //ส่งอักขระ : ไป save ใน Flash Drive

DATAOUT[25]=(month%100)/10+'0'; //ส่งค่า (month%100)/10+'0' ไป save ใน Flash

Drive

DATAOUT[26]=(month%10)+'0'; //ส่งค่า (month%10)+'0' ไป save ใน Flash Drive

DATAOUT[27]=': '; //ส่งอักขระ : ไป save ใน Flash Drive

DATAOUT[28]=(year%100)/10+'0'; //ส่งค่า (year%100)/10+'0' ไป save ใน Flash Drive

DATAOUT[29]=(year%10)+'0'; //ส่งค่า (year%10)+'0' ไป save ใน Flash Drive

DATAOUT[30]=': '; //ส่งอักขระ : ไป save ใน Flash Drive

DATAOUT[31]=(hour%100)/10+'0'; //ส่งค่า (hour%100)/10+'0' ไป save ใน Flash Drive

DATAOUT[32]=(hour%10)+'0'; //ส่งค่า (hour%10)+'0' ไป save ใน Flash Drive

DATAOUT[33]=': '; //ส่งอักขระ : ไป save ใน Flash Drive

DATAOUT[34]=(minute%100)/10+'0'; //ส่งค่า (hour%100)/10+'0' ไป save ใน Flash Drive



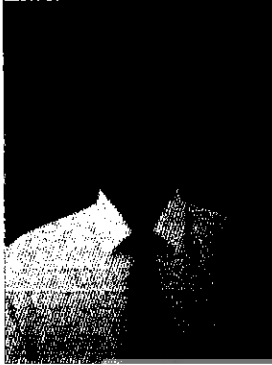
```
DATAOUT[35]= (minute%10)+'0'; //ส่งค่า (hour%100)/10+'0' ไป save ใน Flash Drive
DATAOUT[36]= ':'; //ส่งอักขระ : ไป save ใน Flash Drive
DATAOUT[37]= (second%100)/10+'0'; //ส่งค่า (second%100)/10+'0' ไป save ใน Flash
Drive
DATAOUT[38]= (second%10)+'0'; //ส่งค่า (second%10)+'0' ไป save ใน Flash Drive

DATAOUT[39]= '\r'; //เว้นบรรทัด
DATAOUT[40]= '\n';
SaveDataToUSB(); //เรียกฟังก์ชันส่งข้อมูลไป save ใน Flash Drive

}
cnt_sign3=0; //เคลียร์ค่าใน cnt_sign3
cnt_sign2=0; //เคลียร์ค่าใน cnt_sign2
}
}
```



## ประวัติผู้เขียนโครงการ



ชื่อ นายชัยเดช ควรกิต

ภูมิลำเนา 152 หมู่ 1 ต.จีวงาม อ.เมือง จ.อุดรธานี 53000

### ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนแสงวิทยา
- จบระดับมัธยมศึกษาจากโรงเรียนอุดรดิตต์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : [konlovecreed@hotmail.com](mailto:konlovecreed@hotmail.com)



ชื่อ นายธวัชวัฒน์ ธรรมากิมุข

ภูมิลำเนา 83/69 หมู่ 11 ต.สะเตียง อ.เมือง จ.เพชรบูรณ์ 67000

### ประวัติการศึกษา

- จบระดับประถมศึกษาจาก โรงเรียนเซนต์โยเซฟศรีเพชรบูรณ์
- จบระดับมัธยมศึกษาจาก โรงเรียนเพชรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : [thanawat\\_mu@hotmail.com](mailto:thanawat_mu@hotmail.com)



ชื่อ นายกฤษดา วิวิตรกุล

ภูมิลำเนา 19/9 หมู่ 2 ต.ท่าทอง อ.เมือง จ.พิษณุโลก 65000

### ประวัติการศึกษา

- จบระดับประถมศึกษาจาก โรงเรียนโรจนวิทย์
- จบระดับมัธยมศึกษาจาก โรงเรียนจุฬาลักษณ์มหาวิทยาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร