



รายงานการวิจัย

เรื่อง

เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกลี

อ.เมือง จ.ขอนแก่น

Automatic water system pump Mechine

Ban Nong Hua Wua Amphoe Muang Khonkaen Province

มหาวิทยาลัยราชภัฏมหาสารคาม

RAJABHAT MAHASARAKHAM UNIVERSITY

กิตติพงษ์ ชินสุข

สำนักวิทยบริการฯ มหาวิทยาลัยราชภัฏมหาสารคาม	
ได้รับ.....	
วันลงทะเบียน.....	16 พ.ค. 2560
เลขทะเบียน.....	249940
เลขเรียกหนังสือ.....	ศก. 621.2570113 2554

มหาวิทยาลัยราชภัฏมหาสารคาม

2557

ลิขสิทธิ์ของมหาวิทยาลัยราชภัฏมหาสารคาม

(งานวิจัยนี้ได้รับทุนอุดหนุนจากสถาบันวิจัยและพัฒนา มหาวิทยาลัยราชภัฏมหาสารคาม ปีงบประมาณ 2554)

หัวข้อวิจัย เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น  
ผู้ดำเนินการวิจัย กิตติพงษ์ ชินสุข

หน่วยงาน กลุ่มโปรแกรมวิชาคอมพิวเตอร์และเทคโนโลยี คณะวิทยาศาสตร์และเทคโนโลยี  
มหาวิทยาลัยราชภัฏมหาสารคาม

ปี พ.ศ. 2557

### บทคัดย่อ

การวิจัยครั้งนี้วัตถุประสงค์เพื่อออกแบบและสร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ให้สามารถใช้เครื่องควบคุมดังกล่าวได้ตามวัตถุประสงค์การวิจัย โดยดำเนินการวิจัยกับชาวบ้าน บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ผู้จัดทำได้ทำการติดตั้งเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ ทำงานตามเงื่อนไขระดับน้ำที่กำหนดไว้ เพื่อรักษาให้น้ำคงที่ติดจำนวน 1 เครื่องติดไว้ที่โรงเครื่องปั้มน้ำหมู่บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น การทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ จะใช้ไมโครคอนโทรลเลอร์ ตระกูล AVR ATTINY2313 เป็นตัวประมวลผล โดยหลักการทำงานมีดังนี้ เริ่มต้นจากภาคจ่ายไฟเลี้ยง 12 โวลต์ จะได้ไฟเลี้ยง 5 โวลต์ใช้เลี้ยง ไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 วงจรอุลตราโซนิกตรวจวัดระดับน้ำ (Ultrasonic Level Water Sensor) และใช้ไฟเลี้ยง 12 โวลต์ เลี้ยงรีเลย์ตัวที่ 1 จะควบคุมการทำงานโดยใช้คำสั่งจากไมโครคอนโทรลเลอร์ เมื่อวัดระดับน้ำอยู่ระดับใกล้หมด แล้วจะสั่งให้รีเลย์ตัวที่ 1 ทำงาน แล้วตัด-ต่อไฟ 220 โวลต์ เอซี จ่ายให้กับรีเลย์ตัวที่ 2,3 ทำหน้าที่คล้ายแมกเนติก แต่มีขนาดเล็กกว่า ทำงานเพื่อต่อไฟให้กับปั้มน้ำ พร้อมมีไฟแสดงสถานะการทำงานของปั้ม ถ้าปั้มทำงานไฟสีแดงจะติด และถ้าปั้มไม่ทำงานไฟสีแดงจะดับ โดยกลุ่มตัวอย่างเป็นชาวบ้านที่เพาะเห็ด จำนวน 30 คน สถิติที่ใช้ค่าเฉลี่ย ผลการวิจัยพบว่าเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ตรงตามวัตถุประสงค์การวิจัย เครื่องสามารถปิด-เปิด น้ำที่ตั้งไว้ได้

**Research Title** Automatic water system pumps Machine Ban Nong Hua Wua  
Amphoe Muang Khonkaen Province

**Researcher** Kittipong Chinsook

**Organization** Computer and Technology Science and Technology  
Rajabhat Maha Sarakham University

**Year** 2014

### ABSTRACT

This research aims to design and build automated pump control system. The village Ban NongHuaWua Amphoe Muang Khonkaen Province. Such control can be used for the purposes of research. The research was setup with the locals. The village Ban NongHuaWua Amphoe Muang Khonkaen Province. The report is for the setup automatic water pump control system. Working conditions, water levels are defined. To maintain a constant water next to the number one machine installed at a water pump in the village of Ban NongHuaWua Amphoe Muang KhonKaen Province. The function of the automatic water pump controller. Uses a microcontroller AVR ATTINY2313 processors. The principle is as follows: Starting from the 12 volt power supply can supply 5 volts Party. Microcontroller AVR ATTINY2313 integrated ultrasonic level measurement (Ultrasonic Level Water Sensor) and 12-volt power feed relay 1 is controlled by commands from the microcontroller. When the water level is low. It will instruct the relay to first, and then cut - 220-volt AC power supply to the relay 2,3 acts like a magnetic. But on a smaller scale Work to connect power to the water pump. A power functionality of the pump. If the pump runs the red light is on. If the pump is not running to be light is off. The sample only locals cultivation statistics, the average number of 30 people. The results showed that the water pump automatic control system. The village Ban Nong Hua Wua Amphoe Muang Khonkaen Province. Objective research the machine can - off Water has set

---

## กิตติกรรมประกาศ

การจัดทำโครงการเครื่องควบคุมอุณหภูมิและความชื้นในโรงเพาะเห็ด บ้านโนนสูง อำเภอ  
ยางตลาด จังหวัดกาฬสินธุ์ ได้รับทุนอุดหนุนจากสถาบันวิจัยและพัฒนา มหาวิทยาลัยราชภัฏ  
มหาสารคาม ผู้จัดทำขอขอบพระคุณอาจารย์ รองศาสตราจารย์ ดร.มนตรี ศิริปรัชญานันท์ ผู้ช่วย  
ศาสตราจารย์ ดร.สมศักดิ์ อรรคทิมากุล และอาจารย์โกศิน สนวนานนท์ ที่ให้คำแนะนำชี้แนวทางที่เป็น  
ประโยชน์แก่ผู้จัดทำและขอขอบคุณท่านอาจารย์วิทยา ชำนาญไพร อาจารย์มหาวิทยาลัยเทคโนโลยี  
ราชมงคลอีสาน (วิทยาเขตขอนแก่น) ที่ช่วยแนะนำชี้แนะแนวทางโครงสร้าง

สุดท้ายนี้ขอขอบพระคุณท่านบิดามารดา และญาติพี่น้องทุกคน ที่ให้การสนับสนุนในการ  
จัดทำโครงการนี้มาด้วยดีโดยตลอด จนทำให้โครงการนี้สำเร็จลุล่วงไปด้วยดี



กิตติพงษ์ ชินสุข

มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ .....	ง
สารบัญตาราง .....	ฉ
สารบัญภาพ .....	ช
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญ .....	1
1.2 วัตถุประสงค์ของโครงการ .....	2
1.3 ขอบเขตของโครงการ .....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับจากโครงการ .....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง .....	4
2.1 ไมโครคอนโทรลเลอร์ AVR mega32 .....	4
2.2 ไอซีฐานนาฬิกา DS1370 .....	9
2.3 ไอซีวัดอุณหภูมิ DS1820 .....	11
2.4 สเต็ปป์มอเตอร์ .....	16
2.5 จอแสดงผล LCD .....	18
2.5 ไอซีเปรียบเทียบแรงดัน .....	23
บทที่ 3 วิธีการดำเนินการวิจัย .....	26
3.1 การศึกษาและรวบรวมข้อมูล .....	27
3.2 การออกแบบและสร้างเครื่อง .....	28
บทที่ 4 ผลการวิจัย .....	17
4.1 ผลของโครงการ .....	18
4.2 ผลของการทดสอบ .....	19

## สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ .....	34
5.1 สรุปผลการวิจัย .....	34
5.2 ปัญหาและแนวทางการแก้ไข .....	34
5.3 ข้อเสนอแนะ .....	35
บรรณานุกรม .....	36
ภาคผนวก .....	27
ภาคผนวก ก รูปวงจรเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ .....	28
ภาคผนวก ข โปรแกรมควบคุม .....	31
ภาคผนวก ค DATA SHEET .....	36
ประวัติผู้วิจัย .....	76

## สารบัญตาราง

	หน้า
2.1 คำสั่ง DS1820 .....	14
2.2 รูปแบบ(pattern) คำสั่ง .....	15
4.1 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำต่ำสุดแล้วค่อยๆเพิ่มขึ้น .....	32
4.2 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำสูงสุดแล้วค่อยๆ ลดระดับลง .....	33



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## สารบัญภาพ

	หน้า
2.1 ไมโครโปรเซสเซอร์ .....	4
2.2 การจัดเรียงขาของไมโครคอนโทรลเลอร์ ATmega32 .....	7
2.3 การรับส่งข้อมูลผ่านบัส I2C .....	10
2.4 โครงสร้างภายนอกและตำแหน่งขา IC DS1307 .....	10
2.5 ไอซีวัดอุณหภูมิ DS1820 .....	12
2.6 สัญญาณ RESET .....	12
2.7 สัญญาณ READ/WRITE .....	13
2.8 โครงสร้างของสแต็ปมอเตอร์ .....	16
2.9 สแต็ปมอเตอร์ ชนิดมีสาย 6 เส้น .....	17
2.10 สแต็ปมอเตอร์ชนิดมีสาย 5 เส้น .....	17
2.11 การบิดตัวของผลึกเหลวชนิด TN เป็นมุม 90 องศา .....	18
2.12 ส่วนประกอบของตัวแสดงผลแบบผลึกเหลว .....	18
2.13 การแสดงผลของแสดงผลแบบผลึกเหลว .....	19
2.14 ส่วนประกอบมอดูลตัวแสดงผลแบบผลึกเหลว .....	19
2.15 ตัวแสดงผลแบบผลึกเหลวชนิดสะท้อนกลับ .....	20
2.16 ตัวแสดงผลแบบผลึกเหลวชนิดส่งผ่าน .....	20
2.17 ตัวแสดงผลแบบผลึกเหลวชนิดสะท้อนกลับและส่งผ่าน .....	21
2.18 โครงสร้างภายในมอดูลตัวแสดงผลแบบผลึกเหลว .....	21
2.19 การเชื่อมต่อมอดูลตัวแสดงผลกับไมโครคอนโทรลเลอร์โดยตรง .....	19
2.20 การเชื่อมต่อมอดูลตัวแสดงผลกับไมโครคอนโทรลเลอร์โดยอาศัยชิปควบคุม .....	24
2.21 วงจรภายในของ ไอซีเบอร์ LM311 .....	23
2.22 โครงสร้างภายใน LM311 .....	24
2.23 ต่อวงจรต้องการลอจิก 0 .....	24
2.24 ต่อวงจรต้องการลอจิก 1 .....	24
2.25 การเซตสโตรบ (ขา 6) .....	25
3.1 แผนผังขั้นตอนการดำเนินโครงการ .....	26
3.2 แผนผังการรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ .....	27
3.3 บล็อกไดอะแกรมการทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ .....	28



## สารบัญภาพ(ต่อ)

	หน้า
3.4 วงจรควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 .....	29
3.5 วงจรตรวจวัดระดับน้ำ .....	29
3.6 วงจรควบคุมปั้มน้ำ .....	30
3.7 วงจรภาคจ่ายไฟ .....	30
4.1 ผลของโครงการ .....	30



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## บทที่ 1

### บทนำ

#### ความเป็นมาและความสำคัญ

ในปัจจุบันระบบควบคุมอัตโนมัติด้วยระบบคอมพิวเตอร์เป็นระบบที่มีบทบาทและความสำคัญเป็นอย่างมากในภาคอุตสาหกรรม ซึ่งระบบการควบคุมอัตโนมัตินั้นสามารถทำงานได้โดยการทำงานที่ได้ออกแบบไว้แล้วนำมาใช้ควบคุมเครื่องจักร แทนแรงงานมนุษย์ในส่วนของกระบวนการผลิตต่างๆของภาคอุตสาหกรรม ระบบสามารถลดเวลาในการทำงานและทำหน้าที่ได้หลากหลายรูปแบบในระบบอุตสาหกรรม ตัวอย่างเช่น วิศวกรสามารถโปรแกรมให้ระบบสามารถเปิด-ปิดการทำงานของเครื่องจักรที่มีความเสี่ยงและอันตราย, ควบคุมการเปิด-ปิดมอเตอร์ปั้มน้ำแบบอัตโนมัติ, งานตรวจวัดระดับสารเคมีหรือของเหลวต่างๆทางอุตสาหกรรม ,งานควบคุมการเปิด-ปิดวาล์วน้ำขนาดใหญ่ทางอุตสาหกรรม หรือประตูน้ำตามเขื่อนที่มีขนาดใหญ่ ฯลฯ ซึ่งงานที่กล่าวมาทั้งหมดนี้ถ้าเราเลือกใช้แรงงานมนุษย์ทั้งหมดในการควบคุมอาจต้องใช้เวลาในการทำงานที่ยาวนานและใช้งบประมาณในการดำเนินงานสูงขึ้นที่เกิดความจำเป็น ทั้งนี้เนื่องจากการ ปฏิบัติงานของมนุษย์ในภาคอุตสาหกรรมยังมีความเสี่ยงหรือโอกาสที่อาจก่อเกิดความผิดพลาดและอันตรายขึ้นได้ในอัตราที่สูง เพื่อเป็นการพัฒนาและเพิ่มศักยภาพการผลิตรวมถึงคุณภาพชีวิตของแรงงานให้กับภาคอุตสาหกรรม ระบบควบคุมอัตโนมัติด้วยคอมพิวเตอร์จึงมีการนำมาติดตั้งใช้งานกันอย่างแพร่หลาย ถือได้ว่ามีความสำคัญเป็นอย่างยิ่งต่อกระบวนการผลิตทางอุตสาหกรรม เนื่องจากการควบคุมอัตโนมัติด้วยระบบคอมพิวเตอร์ สามารถทำให้เกิดการปฏิบัติงานที่ถูกต้อง แม่นยำและลดความเสี่ยงในการทำงานรวมถึงต้นทุนการผลิตในงานอุตสาหกรรมได้ในระยะยาว

บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ยังมีการใช้เครื่องปั้มน้ำแบบลักษณะใช้ลูกลอยหรือใช้คนเพื่อจ่ายน้ำให้กับหมู่บ้าน เพื่อเป็นการแก้ปัญหาดังกล่าวคณะผู้วิจัยทำได้คิดค้นทำโครงการเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ ขึ้นมาเพื่อช่วยอำนวยความสะดวกการปั้มน้ำเก็บไว้ใช้ภายในหมู่บ้านโดย การทำงานโดยใช้คำสั่งจากไมโครคอนโทรลเลอร์ เมื่อวัดระดับน้ำอยู่ระดับใกล้หมด จะต่อปั้มน้ำอัตโนมัติเข้ามาเก็บ ซึ่งช่วยลดภาระการทำงานของชาวบ้านและทำให้ความเป็นอยู่ของชาวบ้านดีขึ้น ลดภาระค่าใช้จ่ายในการจ้างคนดูแล

## วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและสร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี

อ.เมือง จ.ขอนแก่น

## ขอบเขตการวิจัย

1. สร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ 1 เครื่อง
2. เครื่องควบคุมระบบปั้มน้ำอัตโนมัติสามารถทำงานได้อย่างอิสระต่อกันโดยมีรายละเอียด

ดังนี้

2.1 ควบคุมอุณหภูมิและความชื้นตามต้องการและอุณหภูมิที่ตั้งค่าไว้ 1 ช่องสัญญาณ

ก. เครื่องสามารถเปิด-ปิดที่ตั้งไว้ได้

ข. อ่านอุณหภูมิ ณ เวลานั้นๆ

ค. โหมดตั้งค่าเปิด-ปิดตามอุณหภูมิกับเปิด-ปิดธรรมดา แยกกันอย่างอิสระ

2.2 ควบคุมเครื่องใช้ไฟฟ้าให้เปิด-ปิดตามต้องการ 2 ช่องสัญญาณ

3. สามารถควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าที่ใช้กระแสสูงสุดไม่น้อยกว่า 1 แอมแปร์ ต่อช่องควบคุม

4. มีปุ่มควบคุม 4 ปุ่มกด โดยมีรายละเอียดดังนี้

4.1 ปุ่มที่ 1-3 ใช้สั่งการเปิด-ปิดเครื่องใช้ไฟฟ้าตามปกติ

4.2 ปุ่มที่ 4 เป็นปุ่มที่ใช้เข้าหรือออกจากโหมด (Mode) เพื่อเรียกดูการตั้งค่าโดยใช้ปุ่มที่ 1 เป็นตัวเลื่อนขึ้นหรือเพิ่มค่า ใช้ปุ่มที่ 2 เป็นตัวเลื่อนลงหรือลดค่า และปุ่มที่ 3 ใช้เป็นปุ่มตกลง (Enter)

5. ใช้เวลา 5-10 วินาทีในการประมวลผล

## นิยามศัพท์เฉพาะ

เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ หมายถึง เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ

บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น

รีเลย์ หมายถึง อุปกรณ์อิเล็กทรอนิกส์ ที่ทำหน้าที่ตัดต่อวงจรคล้ายกับสวิตช์ รีเลย์ใช้ในการควบคุมวงจรไฟฟ้าจึงมีผลต่อการ เปิด-ปิดของโซลินอยด์วาล์ว

โซลินอยด์วาล์ว หมายถึง อุปกรณ์ที่ใช้ไฟฟ้าในการควบคุมเพื่อเปิด-ปิดน้ำ

สเต็ปปีงมอเตอร์ หมายถึง มอเตอร์ที่ขับเคลื่อนด้วยสัญญาณพัลส์ ลักษณะการขับเคลื่อน หมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ที่สร้างขึ้นนำไปใช้งานจริง ในชีวิตประจำวันได้อย่างมีประสิทธิภาพ
2. ผลของโครงการสามารถใช้เป็นแนวทางในการพัฒนาให้สามารถประยุกต์ใช้กับงานอื่นๆได้



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## บทที่ 2

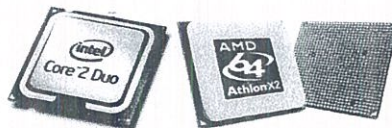
### แนวคิด ทฤษฎีและเอกสารงานวิจัยที่เกี่ยวข้อง

เอกสารที่เกี่ยวข้องในการดำเนินการออกแบบและสร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ได้ศึกษาเอกสารที่เกี่ยวข้อง สามารถแบ่งออกได้ดังนี้

1. ไมโครคอนโทรลเลอร์ AVR mega32
2. ไอซีฐานนาฬิกา DS1370
3. ไอซีวัดอุณหภูมิ DS1820
4. สเต็ปปีงมอเตอร์
5. จอแสดงผล LCD
6. ไอซีเปรียบเทียบแรงดัน

#### 1. ไมโครคอนโทรลเลอร์ AVR mega32

พัฒนาการด้านอิเล็กทรอนิกส์มีการเปลี่ยนแปลงไปอย่างรวดเร็ว เราสามารถสร้างทรานซิสเตอร์ทำจากชิ้นซิลิคอนเล็กๆ ได้โดยกินพลังงานไฟฟ้าต่ำลงและผลิตได้จำนวนมากในราคาถูก เราสามารถสร้างทรานซิสเตอร์จำนวนหลายแสนตัวบรรจุบนชิ้นซิลิคอนเล็กๆ เป็นวงจรรวมที่เรียกว่า ไมโครชิพ (Microchip) และใช้ไมโครชิพเป็นชิ้นส่วนหลักที่ประกอบอยู่ในคอมพิวเตอร์ ทำให้ขนาดของคอมพิวเตอร์เล็กลง ไมโครชิพที่มีขนาดเล็กนี้สามารถทำงานได้หลายหน้าที่เช่น ทำหน้าที่เป็นหน่วยความจำสำหรับเก็บข้อมูล ทำหน้าที่เป็นหน่วยควบคุมอุปกรณ์รับเข้าและส่งออก หรือทำหน้าที่เป็นหน่วยประมวลผลกลางที่เรียกว่า ไมโครโปรเซสเซอร์ ดังรูปที่ 2.1 ซึ่งเป็นหน่วยหลักในการคิดคำนวณवलกลบคูลนหกร การดำเนินการทางตรรกะ ไมโครโปรเซสเซอร์ บางทีก็เรียกว่าซีพียู (Central Processing Unit: CPU) การใช้งานไมโครโปรเซสเซอร์ โดยทั่วไปจะมีการเชื่อมต่อกับหน่วยความจำที่เป็น แรม รอม และอุปกรณ์ภายนอกที่เป็นอินพุต-เอาต์พุต ซึ่งการที่เรานำไมโครโปรเซสเซอร์มาเป็นตัวประมวลผลกลางมีหน่วยความจำแรม พอร์ต อินพุตและเอาต์พุตเราเรียกว่าไมโครคอมพิวเตอร์ (Microcomputer)



รูปที่ 2.1 ไมโครโปรเซสเซอร์

ที่มา : google , 2557

การนำไมโครคอมพิวเตอร์มาใช้งานในลักษณะการควบคุมขนาดเล็กเข้า ระบบควบคุม เครื่องปรับอากาศในบ้าน ระบบปิดเปิดไฟในบ้านอัตโนมัติ หรือระบบปิดเปิดประตูบ้าน เป็นต้น อาจไม่คุ้มกับการลงทุนเนื่องจากไมโครคอมพิวเตอร์มีขนาดใหญ่ มีประสิทธิภาพเกินความจำเป็นและราคาแพง ดังนั้นการพัฒนาด้านเทคโนโลยีในการสร้างไอซี จึงมีองค์ประกอบเกือบทุกอย่างของไมโครคอมพิวเตอร์ที่เรียกว่าไมโครคอนโทรลเลอร์ (Microcontroller)

ส่วนประกอบของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์มีส่วนประกอบหลักภายในอยู่ 6 ส่วนด้วยกัน คือ

ส่วนประมวลผล (CPU)

ส่วนความจำแรม (RAM)

ส่วนความจำออม (EEROM/EPROM/ROM)

- พรอม (Programmable Read-Only Memory: PROM) เป็นหน่วยความจำที่สามารถบันทึกด้วยเครื่องบันทึกพิเศษได้หนึ่งครั้ง และไม่สามารถลบหรือแก้ไขได้

- อีพรอม (Erasable PROM: EPROM) เป็นหน่วยความจำที่สามารถบันทึกโดยใช้เครื่องมือพิเศษ และสามารถลบข้อมูลโดยใช้แสงอัลตราไวโอเล็ต

- อีอีพรอม (Electrically Erasable PROM: EEPROM) เป็นหน่วยความจำที่ผู้ใช้สามารถลบข้อมูลได้ โดยอาศัยการกระตุ้นด้วยแรงดันไฟฟ้าที่สูงกว่าปกติจะทำให้ข้อมูลที่อยู่ภายในสูญหายไป และสามารถบันทึกคำสั่งหรือโปรแกรมใหม่ได้

ส่วนเชื่อมต่อภายนอก (Input / Output)

ส่วนจับเวลาและตัวนับ (Timer-Controller)

ส่วนควบคุมจังหวะการทำงาน (Interrupt Controller)

จากส่วนประกอบทั้งหมดนั้นจะเห็นว่า ไมโครคอนโทรลเลอร์มีความเป็นคอมพิวเตอร์ที่สมบูรณ์แบบในตัวเอง เนื่องจากมันสามารถรับค่าเข้าไปคำนวณและแสดงผลลัพธ์ออกมาได้สิ่งที่ไมโครคอนโทรลเลอร์แตกต่างจากคอมพิวเตอร์ที่เราเห็นอยู่ในท้องตลาด คือการที่อินพุตและเอาต์พุตของไมโครคอนโทรลเลอร์เป็นสัญญาณไฟฟ้าดิจิทัล ไม่ใช่คีย์บอร์ดหรือจอคอมพิวเตอร์อย่างที่เราคุ้นเคยกัน ปัจจุบันการใช้งานควบคุมขนาดเล็ก เช่น เครื่องปรับอากาศ หม้อหุงข้าวอัจฉริยะ เครื่องซักผ้า เครื่องเล่นซีดีและเครื่องใช้ไฟฟ้าแบบอื่นๆ จึงนิยมนำไมโครคอนโทรลเลอร์มาใช้งานเพราะมีขนาดเล็ก อุปกรณ์ที่จะนำมาต่อรวมมีน้อยและเหมาะสมสำหรับใช้งานในการคำนวณที่ไม่ซับซ้อนมากนัก

ข้อควรพิจารณาในการเลือกใช้ไมโครคอนโทรลเลอร์และไมโครโปรเซสเซอร์

ไมโครคอนโทรลเลอร์

- ระบบควบคุมขนาด 8 บิต

- มีความต้องการเชื่อมต่ออุปกรณ์ภายนอกมากนัก

### ไมโครโปรเซสเซอร์

- ใช้งานกับข้อมูลที่มีขนาดมากกว่า 8 บิตตลอดเวลา
- ต้องการความเร็วในการทำงานสูง
- ติดต่อกับหน่วยความจำภายนอกจำนวนมาก
- มีความต้องการในการเชื่อมต่ออุปกรณ์ภายนอกจำนวนมาก

ไมโครคอนโทรลเลอร์เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่ยรวมหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณไว้ด้วยกัน ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุมได้อย่างอิสระ ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่ขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

คุณสมบัติพื้นฐานของ Atmega32

#### สถาปัตยกรรมขั้นสูงแบบ RISC

- ชุดคำสั่งภาษาแอสเซมบลี 131 คำสั่งซึ่งส่วนใหญ่ทำงานที่ 1 รอบสัญญาณนาฬิกา (clock cycle)
- รีจิสเตอร์ ขนาด 8 บิต 32 ตัว
- ความเร็วในการประมวลผล 16 ล้านคำสั่งต่อวินาที (Million Instructions Per Second: MIPS) ที่สัญญาณนาฬิกา 16 เมกะเฮิร์ตซ์ (MHz)

#### หน่วยความจำ

- แบบ Flash ขนาด 16 กิโลไบต์สามารถเขียนลบโปรแกรมได้ 10,000 ครั้ง
- แบบ EEPROM ขนาด 512 ไบต์ สามารถเขียน/ลบข้อมูลได้ 100,000

ครั้ง

- แบบ SRAM ขนาด 1 กิโลไบต์ (Kbytes)

#### อินเตอร์เฟซ Joint Test Action Grope: JTAG

- ใช้ในการตรวจสอบวงจรแบบ Boundary scan
- ใช้ในการโปรแกรมชิพ flash และ EEPROM
- ใช้ในการตรวจสอบสถานะของชิพ

#### ความเร็วสัญญาณนาฬิกา

- 0 ถึง 16 MHz
- ไฟเลี้ยง

- ระหว่าง 4.5 ถึง 5.5 VDC

#### การรองรับอุปกรณ์ต่อพ่วง

- พอร์ตอินพุตและเอาต์พุตขนาด 8 บิต จำนวน 4 พอร์ต
- ตัวจับเวลาและตัวนับ ขนาด 8 บิต 2 ตัวกับขนาด 16 บิต 1 ตัว
- ระบบการเปลี่ยนสัญญาณอนาล็อกเป็นดิจิตอลขนาด 10 บิตจำนวน 8 ช่อง
- ตัวเปรียบเทียบสัญญาณอนาล็อกภายในชิพ อื่น ๆ
- ระบบตรวจจับการทำงานผิดพลาดของซีพียู (watchdog timer)
- ระบบการขัดจังหวะจากภายในและภายนอก
- โหมดการอนุรักษ์พลังงาน 6 โหมด ได้แก่ Idle, ADC noise reduction, Power Save, Power down, Standby และ Extended standby
- ระบบการรีเซ็ตไมโครคอนโทรลเลอร์แบบอัตโนมัติเมื่อเริ่มจ่าย

กระแสไฟฟ้า

#### รายละเอียดขาสัญญาณ

ไมโครคอนโทรลเลอร์ ATmega32 มีการจัดเรียงขาตั้งรูปด้านล่าง

(XCK/TD) PB0	1	40	PA0 (ADC0)
(T1) PB1	2	39	PA1 (ADC1)
(INT2/AIND) PB2	3	38	PA2 (ADC2)
(OC0/AIN1) PB3	4	37	PA3 (ADC3)
(SS) PB4	5	36	PA4 (ADC4)
(MOSI) PB5	6	35	PA5 (ADC5)
(MISO) PB6	7	34	PA6 (ADC6)
(SCK) PB7	8	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TDO)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP) PD6	20	21	PD7 (OC2)

รูปที่ 2.2 การจัดเรียงขาของไมโครคอนโทรลเลอร์ ATmega32

#### ไฟเลี้ยงและแรงดันอ้างอิง

- VCC เป็นขากำจ่ายไฟให้กับตัวไมโครคอนโทรลเลอร์
- GND เป็นขากราวด์
- AREF เป็นขาแรงดันอ้างอิงที่ใช้งานในส่วนของวงจรแปลงสัญญาณ

อนาล็อกเป็นดิจิตอล ปกติจะต่อกับขา VCC



- AVCC ใช้จ่ายไฟให้กับวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลมักต่อเข้ากับขา VCC

#### พอร์ต A (PA0 - PA7)

- ทำหน้าที่เป็นพอร์ตรับข้อมูลสัญญาณอนาล็อกแปลงเป็นสัญญาณดิจิตอล
- เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตต่อกับ pull-up resistor ภายในซึ่งแยกจากกัน เพื่อตั้งแรงดันของลอจิก 1 ให้เท่ากับ 5 โวลต์ สามารถรับกระแส sink 20 มิลลิแอมแปร์ (mA)

#### พอร์ต B (PB0 - PB7)

- เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตต่อกับ pull-up resistor ภายในซึ่งแยกจากกัน สามารถรับกระแส sink 20 mA
- สามารถใช้งานพิเศษตามความต้องการของ ATmega 32

#### พอร์ต C (PC0 - PC7)

- เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตต่อกับ Pull-up resistor ภายในซึ่งแยกจากกัน สามารถรับกระแส sink 20 mA
- สามารถทำหน้าที่เป็น JTAG interface ได้

#### พอร์ต D (PD0 - PD7)

- เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตต่อกับ pull-up resistor ภายในซึ่งแยกจากกัน สามารถรับกระแสไหลเข้า 20 mA
- สามารถใช้งานพิเศษตามความต้องการของ ATmega 32
- กลุ่มขาสำหรับติดต่อกับอุปกรณ์ภายนอกแบบอนุกรม (SPI – Serial Peripheral Interface pin group)

Peripheral Interface pin group)

- SCK เป็นสัญญาณนาฬิกาของ SPI
- MISO เป็นสัญญาณอินพุตของ SPI
- MOSI เป็นสัญญาณเอาต์พุตของ SPI
- SS ใช้เลือกสัญญาณของ SPI ในกรณีที่มีอุปกรณ์ที่ต่อพ่วงหลาย Slave

กลุ่มขาเกี่ยวกับ JTAG หรือ IEEE 1149.1) ขาเหล่านี้ใช้สำหรับตรวจสอบวงจร (Debug) และสามารถใช้ในการโปรแกรมชิปได้

#### กลุ่มขาเกี่ยวกับตัวจับเวลา และตัวนับ (Timer / Counter pin group)

- T0 (Timer/Counter0 External Counter Input)
- T1 (Timer/Counter1 External Counter Input)

นอกจากนี้หากไม่ต้องการใช้สัญญาณนาฬิกาจากตัวไมโครคอนโทรลเลอร์  
เรายังสามารถป้อนสัญญาณนาฬิกาภายนอกผ่านทางขาต่อไปนี้ได้

- TOSC1 (Timer Oscillator Pin 1)
- TOSC2 (Timer Oscillator Pin 2)

#### กลุ่มขาสำหรับรับสัญญาณขัดจังหวะจากภายนอก

- INT0 (External Interrupt 0 Input)
- INT1 (External Interrupt 1 Input)
- INT2 (External Interrupt 2 Input)

การเปลี่ยนแปลงระดับแรงดันที่ขาเหล่านี้จะมีผลทำให้โปรแกรมกระโดดไป  
ทำงานในส่วนของชุดคำสั่งให้บริการการขัดจังหวะ (Interrupt service routine)

#### กลุ่มขาสำหรับติดต่อกับพอร์ตอนุกรม (USART pin group)

- TXD (USART Output Pin) ขาส่งข้อมูลออก
- RXD (USART Input Pin) ขารับข้อมูลเข้า
- XCK (USART External Clock Input / Output) ขาสัญญาณนาฬิกา

ใช้สำหรับโหมด synchronous

## 2. ไอซีฐานนาฬิกา DS1307

ระบบฐานเวลา เป็นสิ่งสำคัญที่สามารถนำไปใช้ในอุปกรณ์อิเล็กทรอนิกส์ได้หลากหลาย ภายใน  
ไมโครคอนโทรลเลอร์เองก็มีไทมเมอร์เพื่อใช้ในการจับเวลา หรือนำไปใช้เป็นฐานเวลาจริงได้เช่นกัน แต่  
เนื่องจากไมโครคอนโทรลเลอร์สามารถทำงานได้ต่อเมื่อมีไฟเลี้ยงเท่านั้น ดังนั้นการใช้ไทมเมอร์ของ  
ไมโครคอนโทรลเลอร์ สร้างฐานเวลาจริงจึงไม่เหมาะสมในบางแอปพลิเคชัน DS1307 เป็น IC ฐานเวลา  
ของดัลลัสเซมิคอนดักเตอร์ (Dallas Semiconductor) มีบัสรับส่งข้อมูลแบบ I2C ซึ่งเป็นแบบ 2 wire  
สามารถสื่อสารได้ 2 ทิศทาง (bi-direction bus) ฐานเวลาของ DS1307 นั้นสามารถเก็บข้อมูล วินาที,  
นาฬิกา, ชั่วโมง, วัน, วันที่, เดือน และปี ได้ ระบบเวลาสามารถทำงานโหมดรูปแบบ 24 ชั่วโมง หรือ 12  
ชั่วโมง AM/PM ก็ได้ ภายมีระบบตรวจจับแหล่งจ่ายไฟ โดยถ้าแหล่งจ่ายไฟหลักถูกตัดไป DS1307  
สามารถสวิตช์ไปใช้ไฟจากแบตเตอรี่ และทำงานต่อไป โดยที่ยังสามารถรักษาข้อมูลไว้ได้ โครงสร้างมีขา  
ทั้งหมด 8 ขา

### 2.2.1 รับส่งข้อมูลแบบ I2C นั้นมีข้อกำหนดอยู่ 2 ประการด้วยกันคือ

การรับส่งข้อมูลจะเริ่มขึ้นได้เมื่อบัสมีสถานะว่างเท่านั้น ในช่วงที่ทำการรับส่งข้อมูลอยู่  
สายสัญญาณ SDA ต้องไม่เปลี่ยนสถานะในช่วงที่ SCL มีสถานะเป็นลอจิก "1" ถ้า SDA มีการ  
เปลี่ยนสถานะในช่วงที่ SCL เป็นลอจิก "1" จะถือว่าเป็นสัญญาณควบคุมการรับส่งข้อมูล

### 2.2.1.1 สถานะของการรับส่งข้อมูลแบบ I2C

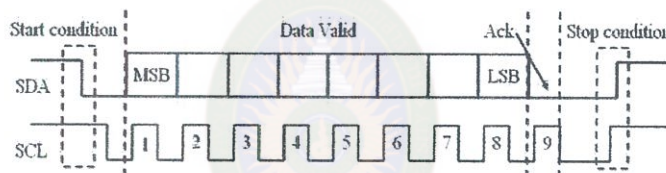
สถานะของการรับส่งข้อมูลแบบ I2C สามารถแบ่งออกได้เป็น 4 สถานะด้วยกันดังแสดงในรูปที่ 2-3 และมีรายละเอียดดังนี้

ก) สถานะว่าง (Bus not busy): สัญญาณ SDA และ SCL มีระดับสัญญาณเป็น High

ข) เริ่มส่งข้อมูล (Start data transfer): มีการเปลี่ยนระดับสัญญาณของ SDA จาก High เป็น Low ในขณะที่ SCL มีระดับสัญญาณเป็น High ค้างไว้

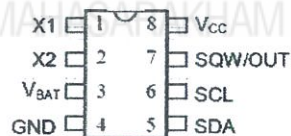
ค) หยุดส่งข้อมูล (Stop data transfer): มีการเปลี่ยนระดับสัญญาณของ SDA จาก Low เป็น High ในขณะที่ SCL มีระดับสัญญาณเป็น High ค้างไว้

ง) รับส่งข้อมูล (Data valid): มีการรับส่งข้อมูลผ่านสายสัญญาณ SDA โดยข้อมูลแต่ละบิตจะถูกส่งในช่วงที่ SCL มีระดับเป็น High โดยในช่วงที่ SCL มีสถานะเป็น High อยู่ นั้น SDA จะต้องไม่เกิดการเปลี่ยนระดับสัญญาณ



รูปที่ 2.3 การรับส่งข้อมูลผ่านบัส I2C

### 2.2.2 โครงสร้างภายนอกและรายละเอียดของขาสัญญาณ



รูปที่ 2.4 โครงสร้างภายนอกและตำแหน่งขา IC DS1307

จากรูปที่ 2.4 เป็นโครงสร้างภายนอกและตำแหน่งขา IC DS1307 โดยมีรายละเอียดดังต่อไปนี้

- VCC: ใช้ต่อไฟเลี้ยง +5V
- GND: ใช้ต่อกราวด์
- VBAT: ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงาน ในกรณีที่ไม่มีไฟเลี้ยงจ่าย
- SDA: ขารับส่งข้อมูลด้วยระบบบัส I2C
- SCL: ขาสัญญาณนาฬิกาสำหรับการรับส่งข้อมูลด้วยระบบบัส I2C

- SQW/OUT: ขาเอาต์พุตสัญญาณ Square Wave สามารถเลือกความถี่ได้
- X1, X2: ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อสร้างฐานเวลาจริงให้กับ IC

### 2.3 ไอซีวัดอุณหภูมิ DS1820

DS1820 จะให้สัญญาณเอาต์พุตออกมาเป็นแบบดิจิตอล และยังสามารถที่จะทำการโปรแกรมเข้าไปยังหน่วยความจำและควบคุมฟังก์ชันภายในไอซีได้ ซึ่งมีหน่วยความจำรวมภายในขนาด 64 บิต แบบเลเซอร์รอม ดังนั้นจึงสามารถที่จะทำการอ่านและเขียนข้อมูลต่างๆ เกี่ยวกับหน้าที่การทำงานในการตรวจวัดอุณหภูมิได้อย่างมากมายตามการประมวลผลของไมโครโปรเซสเซอร์ นอกจากนั้นแล้วยังสามารถติดตั้ง DS 1820 เพื่อการตรวจวัดอุณหภูมิได้ในหลายลักษณะและหลายสถานที่ตำแหน่งการติดตั้งที่มีความแตกต่างอย่างมากกับอุปกรณ์ทั่วไปไม่ว่าจะเป็นการติดตั้งภายในอาคาร, อุปกรณ์เครื่องใช้ต่างๆ หรือ ภายในเครื่องจักร และเอาต์พุตที่เป็นอนุกรมตัวเลขของ DS 1820 สามารถต่อเอาต์พุตบนสายสัญญาณเพียงเส้นเดียวได้หลายๆ ชุดโดยไม่สับสนข้อมูลซึ่งกันและกัน ในรูปที่ 2.5 แสดงรูปร่างลักษณะตัวถังและการจัดขาใช้งานของ DS 1820

#### 2.3.1 คุณสมบัติเด่นของ DS 1820

2.3.1.1 อินเตอร์เฟซสัญญาณผ่านขาเอาต์พุตเพียงพอร์ตเดียวแบบ 1 สายข้อมูล (1 - Wire TM)

2.3.1.2 ขยายจุดตรวจจับอุณหภูมิได้หลายๆ จุดบนสายข้อมูลเพียง 1 สายข้อมูล

2.3.1.3 ไม่ต้องใช้อุปกรณ์ภายนอกมาต่อร่วม

2.3.1.4 สามารถควบคุมการทำงานเพาเวอร์ออนได้ผ่านทางสายข้อมูล

2.3.1.5 เพาเวอร์ขณะสแตนด์บายเป็นศูนย์

2.3.1.6 ย่านการวัดอุณหภูมิตั้งแต่ -55 องศาเซลเซียสถึง +125 องศาเซลเซียสที่ 0.5 องศาต่อสแต็ป หรือตั้งแต่ย่าน -67 องศาฟาเรนไฮต์ ถึง +257 องศาฟาเรนไฮต์ที่ 0.9 องศาฟาเรนไฮต์ต่อสแต็ป

2.3.1.7 อุณหภูมิจะถูกอ่านออกมาเป็นค่าทางดิจิตอล 9 บิต

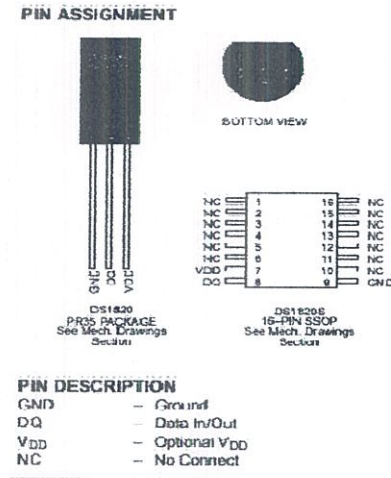
2.3.1.8 อัตราความเร็วในการแปลงจากอุณหภูมิมาเป็นค่าตัวเลขทางดิจิตอลเท่ากับ 200 มิลลิวินาที

2.3.1.9 ผู้ใช้งานสามารถกำหนดการเซตค่าเตือนย่านอุณหภูมิได้ในแบบ non - volatile

2.3.1.10 การเตือนย่านอุณหภูมินั้นสามารถกำหนดรหัสผ่านการสั่งการและแอดเดรสของอุปกรณ์ได้จากภายนอกพื้นที่ตรวจวัดอุณหภูมิผ่านทางโปรแกรมภายนอก

2.3.1.11 เหมาะกับการประยุกต์ใช้งานตรวจวัดอุณหภูมิและติดตั้งไว้ในอุปกรณ์

ควบคุมเทอร์โมสแตติก ระบบโรงงานอุตสาหกรรม ผลิตภัณฑ์ เทอร์โมมิเตอร์ หรือระบบอื่นๆ ที่มีส่วนตรวจจับอุณหภูมิทำงานร่วมอยู่

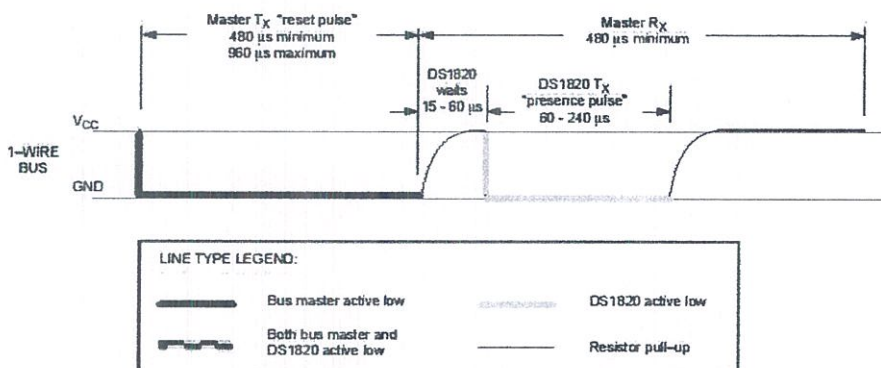


### รูปที่ 2.5 ไอซีวัดอุณหภูมิ DS1820

#### 2.3.2 สัญญาณต่างๆ

##### 2.3.2.1 สัญญาณ RESET

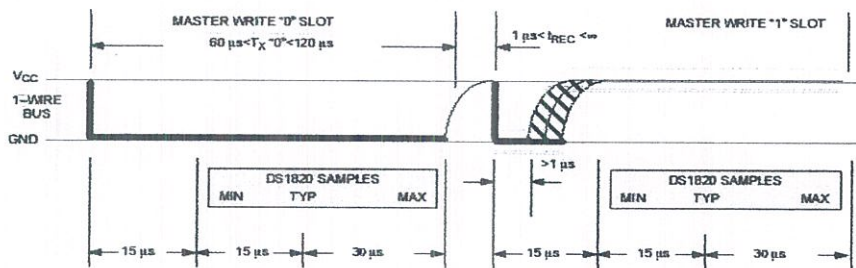
ในการเริ่มติดต่อกับ DS1820 จะต้องเริ่มด้วยสัญญาณ RESET ก่อนทุกครั้ง จากรูปเส้นสีดำคือสัญญาณ RESET จากตัว MCU ส่วนสีเทาคือ สัญญาณที่ DS1820 จะ RESPONSE กลับมา ความกว้างของสัญญาณนี้ต้องแม่นยำทีเดียวต้องปรับเวลาหน่วง หลายครั้งเพราะภาษา C มักจะมีปัญหาเรื่องความแม่นยำของเวลาหน่วงที่เป็นช่วงสั้นมากๆ ดังรูปที่ 2-6



### รูปที่ 2.6 สัญญาณ RESET

### 2.3.2.2 สัญญาณ READ/WRITE

รูปแบบ Pattern ช่องสัญญาณ READ/WRITE จะเป็นดังรูปที่ 2.7 สังเกต MIN, MAX time ของ DS1820 SAMPLES นั่นคือเวลาที่ DS1820 ต้องใช้ในการอ่านข้อมูลบิตที่ส่งให้ เหมือนเดิมระวังเรื่อง timing ให้มากๆ



รูปที่ 2.7 สัญญาณ READ/WRITE

คำสั่งเพิ่มเติมใน DS1820 ก็มีดังตารางที่ 2.2 ข้างล่าง แต่ที่จะใช้กันก็คือ คำสั่ง 44h กับ BEh ซึ่งเป็นคำสั่งให้ Convert T และ Read Scratchpad ยังมี command อื่นๆ หลาย command ในระบบ 1 wire

## ตารางที่ 2.1 คำสั่งใน DS1820

INSTRUCTION	DESCRIPTION	PRO TOC OL	1-WIRE BUS AFTER ISSUING PROTOCOL	NOTE S
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	<read temperature busy status>	1
MEMORY COMMANDS				
Read Scratchpad	Reads bytes from scratchpad and reads CRC byte.	BEh	<read data up to 9 bytes>	
Write Scratchpad	Writes bytes into scratchpad at addresses 2 and 3 (TH and TL temperature triggers).	4Eh	<write data into 2 bytes at addr. 2 and addr. 3>	
Copy Scratchpad	Copies scratchpad into nonvolatile memory (addresses 2 and 3 only).	48h	<read copy status>	2
Recall E2	Recalls values stored in nonvolatile memory into scratchpad (temperature triggers).	B8h	<read temperature busy status>	
Read Power Supply	Signals the mode of DS1820 power supply to the master.	B4h	<read supply status>	

รูปแบบ Pattern คำสั่งที่จะส่งไปยัง DS1820

รูปแบบการรับส่งจะเป็นดังตารางตารางที่ 2.3 ข้างล่าง จะเป็นรูปแบบที่ง่ายที่สุด อธิบายเพิ่มเติมใน 1 wire component ทุกตัวจะมีรหัส ในการติดต่อทุกครั้งต้องอ้างรหัสของอุปกรณ์ ตัวนั้นๆ ซึ่งจะมีรหัสไม่เหมือนกันในแต่ละตัว แต่เนื่องจากว่าในตอนนี้อยู่เพียงตัวเดียว ไม่จำเป็นต้องอ้างรหัสนั้น ก็ใช้คำสั่ง CCh เพื่อข้ามขั้นตอนการอ้างรหัสของอุปกรณ์ที่จะติดต่อไปได้

ตารางที่ 2.2 รูปแบบ (pattern) คำสั่ง

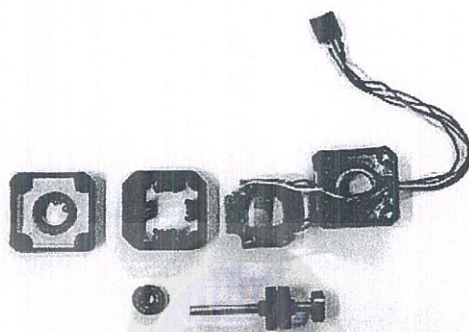
MASTER MODE	DATA (LSB FIRST)	COMMENTS
TX	Reset	Reset pulse (480–960 ms).
RX	Presence	Presence pulse.
TX	55h	Issue “Match ROM” command.
TX	<64-bit ROM code>	Issue address for DS1820.
TX	44h	Issue “Convert T” command.
TX	<I/O LINE HIGH>	I/O line is held high for at least 500 ms by bus master to allow conversion to complete.
TX	Reset	Reset pulse.
RX	Presence	Presence pulse.
TX	55h	Issue “Match ROM” command. Issue address for DS1820.
TX	<64-bit ROM code>	Issue address for DS1820.
RX	<9 data bytes>	Read entire scratchpad plus CRC; the master now recalculates the CRC of the eight data bytes received



## 2.4 สเต็ปป์มอเตอร์ (Stepping Motor)

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเคลื่อนด้วยพัลส์ ลักษณะการขับเคลื่อน จะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่อง แต่มีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1, 1.5, 1.8 หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์ลักษณะที่ นำมอเตอร์ไปใช้ จะเป็นงานที่ต้องการตำแหน่งแม่นยำ เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ (PRINTER) ระบบขับเคลื่อนหัวอ่านในเครื่องอ่านบันทึกเทป ระบบขับเคลื่อนตำแหน่งของปากกาใน X-Y PLOTTER เป็นต้น

### 2.4.1 โครงสร้างของสเต็ปป์มอเตอร์



รูปที่ 2.8 โครงสร้างของสเต็ปป์มอเตอร์

ที่มา : google , 2557

### 2.4.2 สเต็ปป์มอเตอร์ที่พบในปัจจุบันมี 3 ลักษณะดังนี้

#### 2.4.2.1 แบบแม่เหล็กถาวร(PERMANENT MAGNET\_PM)

สเต็ปป์มอเตอร์แบบ PM จะมีสเตเตอร์ (STATOR) ที่พันขดลวดไว้หลายๆ โพล โดยมีโรเตอร์ (ROTOR) เป็นรูปทรง กระจบอกฟันเลื่อย และโรเตอร์ทำด้วยแม่เหล็กถาวร เพื่อป้องกันกระแสตรงให้กับขดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าผลัดต่อโรเตอร์ ทำให้มอเตอร์หมุนมอเตอร์แบบ PM จะเกิดแรงดูดยึดให้โรเตอร์หยุดอยู่กับที่ แม้จะไม่ได้ป้อนไฟเข้าขดลวด

2.4.2.2 แบบแปรค่ารีลักแตนซ์ (VARIABLE RELUCTANCE- VR) สเต็ปป์มอเตอร์แบบ VR จะมีการหมุนโรเตอร์ได้อย่างอิสระ แม้จะไม่ได้จ่ายไฟให้โรเตอร์ทำจากสารเฟอร์โรแมกเนติกกำลังอ่อน มีลักษณะเป็นฟันเลื่อย รูปทรงกระจบอกโดยจะมีความสัมพันธ์ โดยตรงกับจำนวนโพลในสเตเตอร์ แรงบิดที่เกิดขึ้นจะไปหมุนโรเตอร์ ไปในเส้นทางของอำนาจแม่เหล็กที่มีค่ารีลักแตนซ์ต่ำที่สุด ตำแหน่งที่จะเกิดแน่นอนและมีเสถียรภาพแต่จะเกิดขึ้นได้หลายๆ จุดดังนั้นเมื่อป้อนไฟเข้าขดลวดต่างๆ ในมอเตอร์แตกต่างกันไป ก็ทำให้มอเตอร์ หมุนไปตำแหน่งต่างๆ กันโรเตอร์ของ VR จะมีความเฉื่อยของโรเตอร์น้อยจึงมีความเร็วรอบสูงกว่ามอเตอร์แบบ PM

2.4.2.3 แบบผสม(HYBRID-H) สเต็ปป์มอเตอร์แบบ H จะเป็นลูกผสมของ VR กับ PM โดยจะมีสเตเตอร์คล้ายกับที่ใช้ใน VR โรเตอร์มีหมวกหุ้ม ปลายซึ่งมีลักษณะของสารแม่เหล็กที่มีกำลังสูง โดยการควบคุมขนาดรูปร่างของหมวกแม่เหล็กอย่างดีทำให้ได้มุม การหมุนและครั้งน้อยและแม่นยำ ข้อดีก็คือ ให้แรงบิดสูงและมีขนาดกะทัดรัด และให้แรงฉุดยึดโรเตอร์นิ่งกับที่ตอนไม่จ่ายไฟ

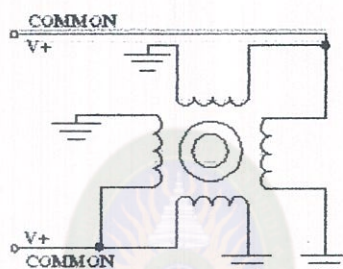
2.4.3 SP MOTOR แบบ PM จะมีอยู่ 2 ชนิด

2.4.3.1 ชนิดที่เป็น COMMON ภายนอก SP MOTOR แบบนี้มีสายอยู่ 6 เส้นดังรูปที่

2.9 คือ

-สายที่เป็น COMMON 2 เส้น

-สายที่เป็น GROUND 4 เส้น



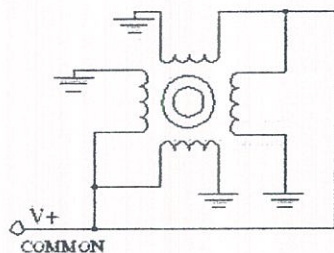
รูปที่ 2.9 สเต็ปป์มอเตอร์ ชนิดมีสาย 6 เส้น

2.4.3.2 ชนิดที่เป็น COMMON ภายใน SP MOTOR แบบนี้มีสายอยู่ 5 เส้นดังรูปที่

2.10 คือ

-สายที่เป็น COMMON 1 เส้น

-สายที่เป็น GROUND 4 เส้น

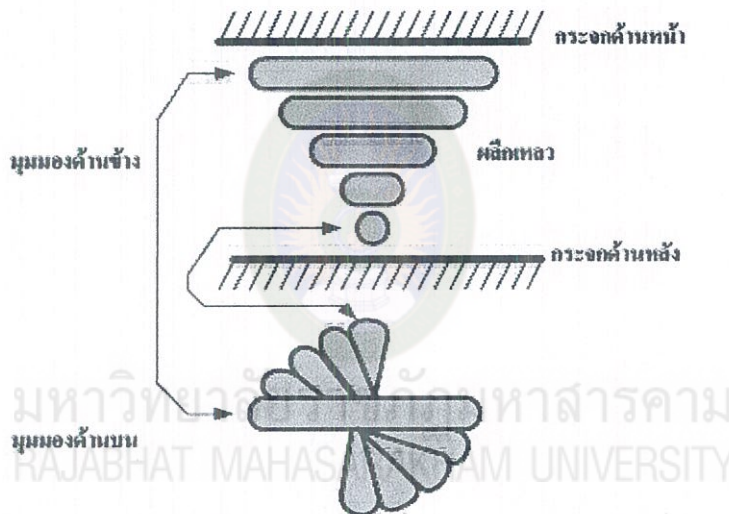


รูปที่ 2.10 สเต็ปป์มอเตอร์ชนิดมีสาย 5 เส้น

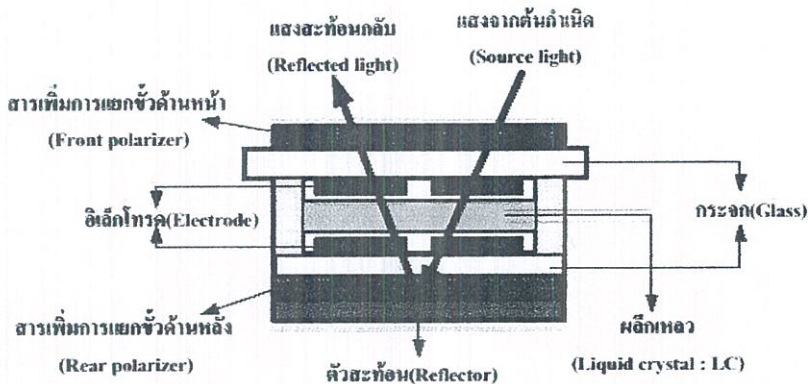
2.5 จอแสดงผล LCD(Liquid Crystal Display: LCD)

2.5.1 การควบคุมตัวแสดงผลแบบผลึกเหลว

ปัจจุบันมีการนำตัวแสดงผลแบบผลึกเหลวมาใช้แสดงผลในอุปกรณ์อิเล็กทรอนิกส์ แทนตัวแสดงผลแบบ 7 ส่วน อย่างกว้างขวางเนื่องด้วยตัวแสดงผลแบบผลึกเหลวมีข้อดีกว่าตัวแสดงผลแบบ 7 ส่วนหลายด้าน เช่น สามารถแสดงข้อความและรูปภาพ ใช้กำลังงานไฟฟ้าน้อยกว่า เป็นต้น ตัวแสดงผลแบบผลึกเหลวจะประกอบด้วยจุดเล็กๆหลายๆจุดซึ่งจุดเล็กๆเหล่านี้จะประกอบกันเป็นตัวอักษรหรือรูปตามต้องการได้ ภายในจุดเล็กๆจะบรรจุด้วยผลึกเหลวซึ่งสามารถบิดตัวเมื่อเกิดสนามไฟฟ้าในขณะป้อนแรงดันไฟฟ้า ผลึกเหลวที่นิยมใช้คือผลึกเหลวชนิดทวิสตีนีเมติก ดังแสดงรูป การบิดตัวของผลึกเหลวชนิด TN ไป 90 องศา และแสดงส่วนประกอบของตัวแสดงผลแบบผลึกเหลวดังในรูปที่ 2.11



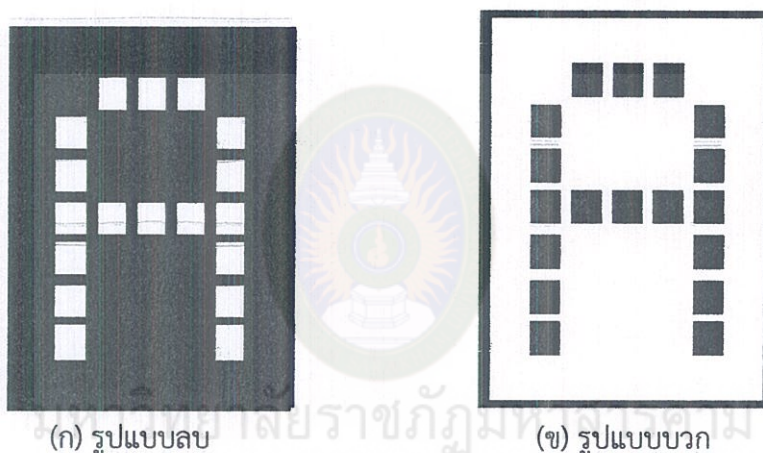
รูปที่ 2.11 การบิดตัวของผลึกเหลวชนิด TN เป็นมุม 90 องศา



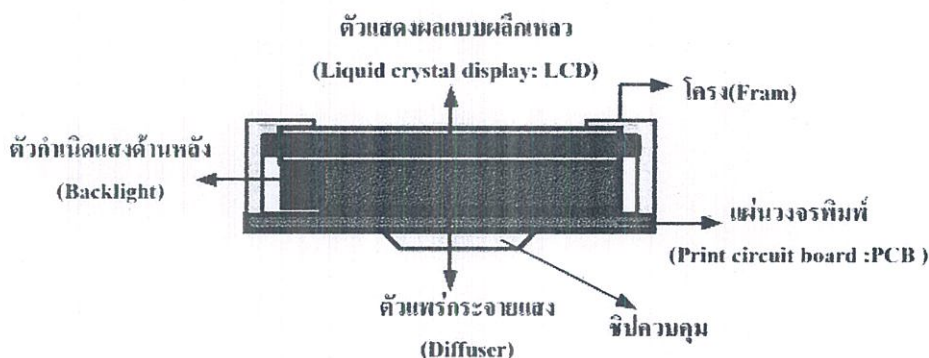
รูปที่ 2.12 ส่วนประกอบของตัวแสดงผลแบบผลึกเหลว

จากรูปที่ 2.12 แสดงให้เห็นส่วนประกอบของตัวแสดงผลแบบผลึกเหลว เมื่อไม่มีการป้อนแรงดันไฟฟ้าให้กับอิเล็กโทรดแสงจากภายนอกจะสามารถผ่านตัวแสดงผลแบบผลึกเหลวไปได้ และสะท้อนกลับออกมาทำให้สามารถมองเห็นเป็นแสงสว่างแต่เมื่อป้อนแรงดันไฟฟ้าให้กับอิเล็กโทรดสารผลึกเหลวที่บรรจุอยู่ภายในจะเกิดการบิดตัวในทิศทางเดียวกับสนามไฟฟ้าเป็นผลให้แสงจากภายนอกไม่สามารถทะลุผ่านไปได้เราจึงมองเห็นเป็นความมืดหรือสีดำ แบบวิธีของการแสดงผลแบ่งออกได้เป็น 2 แบบวิธีใหญ่ๆคือ รูปแบบลบดังในรูปที่ 2.13 (ก) และ รูปแบบบวกดังในรูปที่ 2.13 (ข)

การควบคุมการแสดงผลของตัวแสดงผลจำเป็นต้องใช้วงจรที่ซับซ้อน ดังนั้นทางบริษัทผู้ผลิตจึงได้ผลิตตัวแสดงผลในลักษณะมอดูลออกมา กล่าวคือมีชิพสำหรับควบคุมการแสดงผลแทน



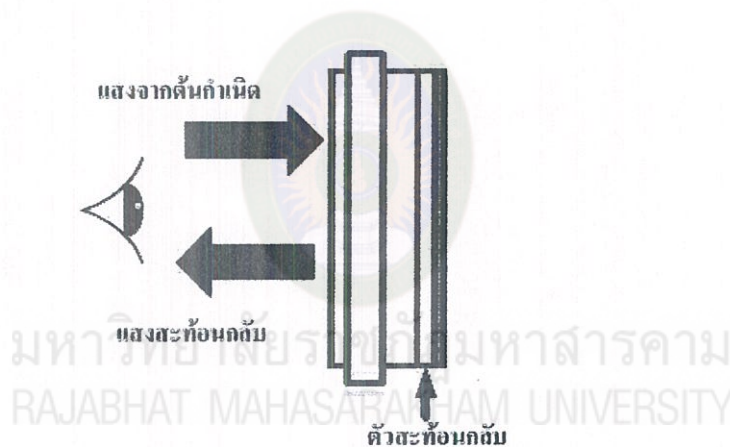
รูปที่ 2.13 การแสดงผลของแสดงผลแบบผลึกเหลว



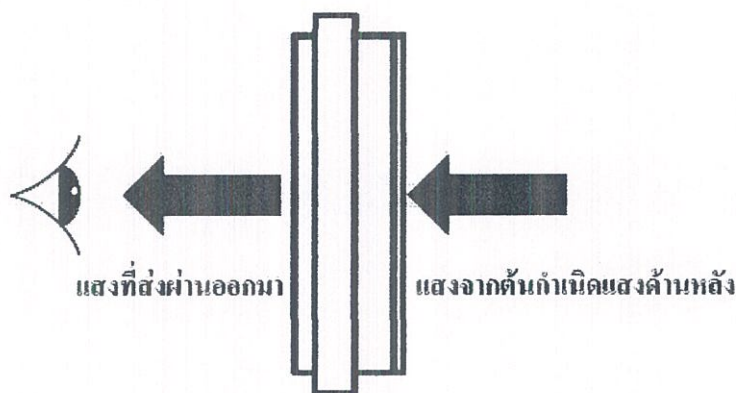
รูปที่ 2.14 ส่วนประกอบมอดูลตัวแสดงผลแบบผลึกเหลว

ชนิดของตัวแสดงผลแบบผลึกเหลวแบ่งตามลักษณะการแสดงผลสามารถแบ่งออกได้เป็น 2 ชนิดคือ แสดงเป็นตัวอักษร (Character) และ แสดงเป็นกราฟิก (Graphics) ในกรณีแบ่งตามการกำเนิดรูปสามารถแบ่งออกได้เป็น 3 ชนิด ดังนี้คือ

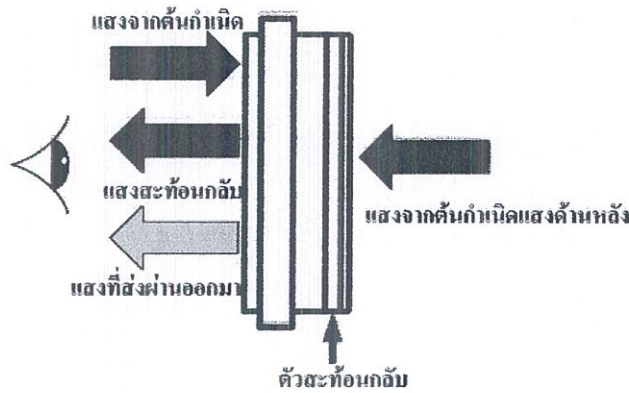
1. ชนิดสะท้อนกลับ (Reflective) เป็นชนิดที่ต้องอาศัยแสงจากภายนอกส่องผ่านเข้าไปและสะท้อนกลับออกมา ซึ่งการจะสะท้อนกลับออกมาได้หรือไม่ขึ้นอยู่กับการบิดตัวของผลึกเหลว
2. ชนิดส่องผ่าน (Transmissive) เป็นชนิดที่ต้องมีตัวกำเนิดแสงด้านหลัง (Backlight) ส่องแสงผ่านออกมา
3. ชนิดสะท้อนกลับและส่องผ่าน (Transflective) เป็นชนิดที่ทั้งตัวสะท้อนแสงอยู่ด้านหลังและมีตัวกำเนิดแสงด้านหลังส่องแสงผ่านออกมา สามารถเลือกให้ทำงานในแบบใดแบบหนึ่งหรือทั้งสองแบบได้ในเวลาเดียวกัน



รูปที่ 2.15 ตัวแสดงผลแบบผลึกเหลวชนิดสะท้อนกลับ

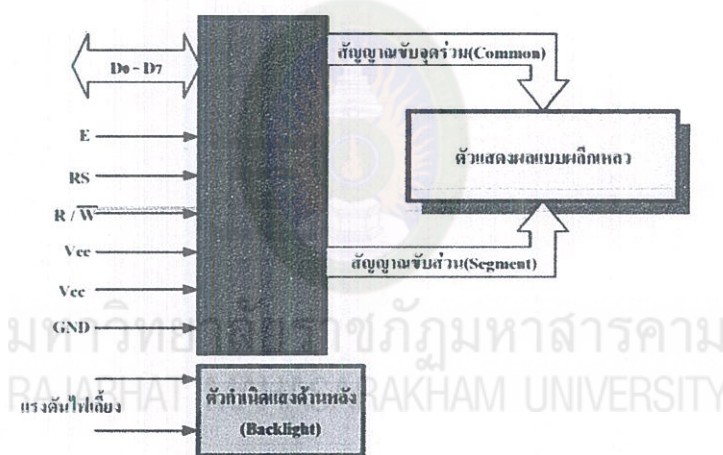


รูปที่ 2.16 ตัวแสดงผลแบบผลึกเหลวชนิดส่องผ่าน



รูปที่ 2.17 ตัวแสดงผลแบบผลึกเหลวชนิดสะท้อนกลับและส่งผ่าน

2.6.2 โครงสร้างภายในมอดูลตัวแสดงผลแบบผลึกเหลว



รูปที่ 2.18 โครงสร้างภายในมอดูลตัวแสดงผลแบบผลึกเหลว

2.6.2.1 ขาสัญญาณสำหรับเชื่อมประสานเพื่อการควบคุมการทำงาน ประกอบด้วยขาสัญญาณดังต่อไปนี้

1. D0- D7 เป็นขาอินพุต/เอาต์พุต ทำหน้าที่เป็นทางผ่านของข้อมูล
2. E (Enable) เป็นขาสัญญาณอินพุตแอกทีฟที่ตรรกะ 1 ใช้สำหรับสั่งให้

ชิปควบคุมทำงาน

3. (Register select) เป็นขาสัญญาณอินพุต ใช้สำหรับเลือกการติดต่อ

รีจิสเตอร์ภายในชิปควบคุม ถ้ามีตรรกะเป็น 1 หมายถึง เลือกติดต่อกับรีจิสเตอร์ข้อมูล ถ้ามีตรรกะเป็น 0 หมายถึงเลือกติดต่อกับรีจิสเตอร์คำสั่ง RS

4. WR / (Read/write) เป็นขาสัญญาณอินพุต ใช้สำหรับควบคุมการอ่านหรือเขียนกับชิปควบคุม ถ้ามีตรรกะเป็น 1 หมายถึง อ่านข้อมูลจากชิปควบคุม ถ้ามีตรรกะเป็น 0 หมายถึงเขียนข้อมูลไปยังชิปควบคุม

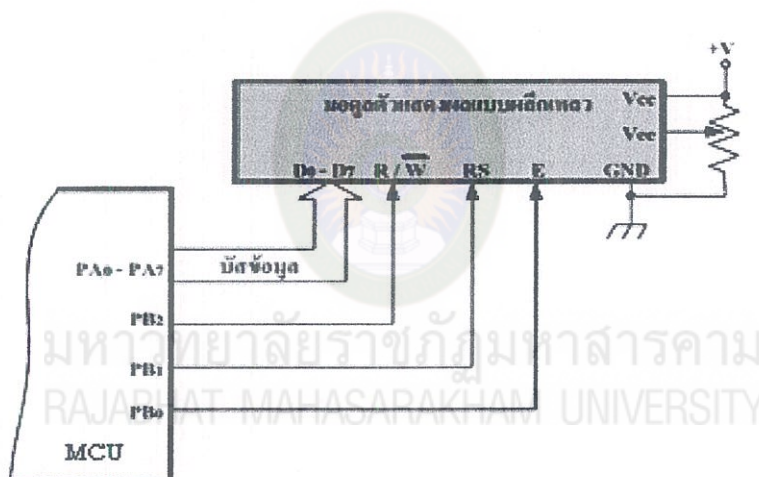
5. Vee เป็นขาสำหรับป้อนแรงดันไฟเพื่อปรับความสว่างของการแสดงผล

6. Vcc เป็นขาสำหรับต่อกับแรงดันไฟเลี้ยงวงจรบวก 5 โวลต์

7. GND เป็นขาสำหรับต่อกับแรงดันไฟเลี้ยงวงจรขั้วกราวด์

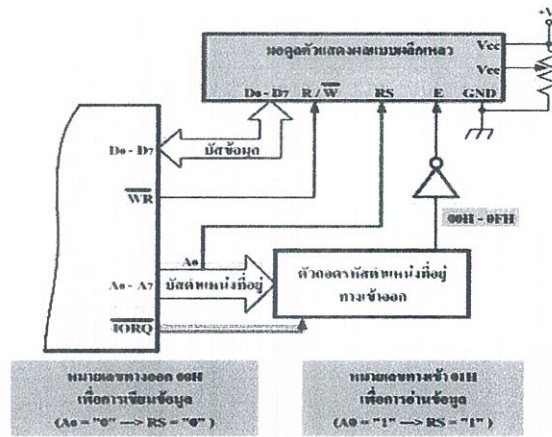
### 2.6.2.2 การเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับมอดูลตัวแสดงผลสามารถทำได้ 2 วิธีคือ

1. เชื่อมต่อผ่านทางออก ดังแสดงในรูปที่ 2.19 วิธีนี้ไมโครคอนโทรลเลอร์จะไม่สามารถอ่านข้อมูลจากตัวแสดงผลได้เนื่องจากใช้ทางออกเป็นตัวเชื่อมต่อกับขาสัญญาณข้อมูลของตัวแสดงผล



รูปที่ 2.19 การเชื่อมต่อมอดูลตัวแสดงผลกับไมโครคอนโทรลเลอร์โดยตรง

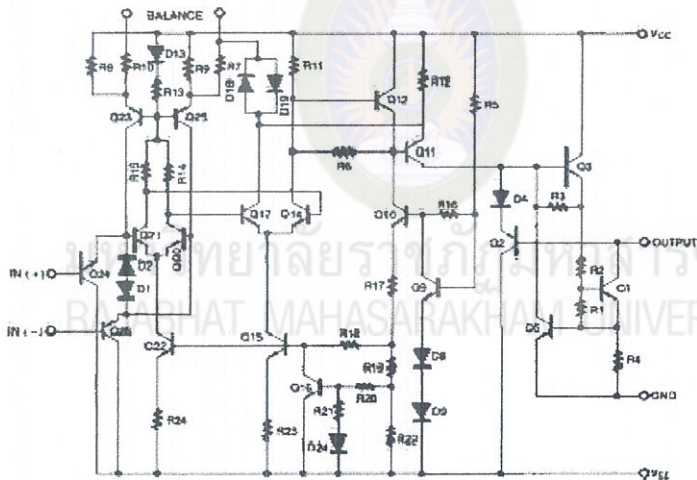
2. เชื่อมต่อโดยตรงกับไมโครคอนโทรลเลอร์อาศัยชิปควบคุมภายในทำหน้าที่ทางเข้าออก ดังแสดงในรูปที่ 2.20 ซึ่งวิธีนี้ไมโครคอนโทรลเลอร์จะสามารถอ่านและเขียนข้อมูลกับตัวแสดงผลได้



รูปที่ 2.20 การเชื่อมต่อโมดูลตัวแสดงผลกับไมโครคอนโทรลเลอร์โดยอาศัยชิปควบคุม

## 2.6 ไอซีเปรียบเทียบแรงดัน

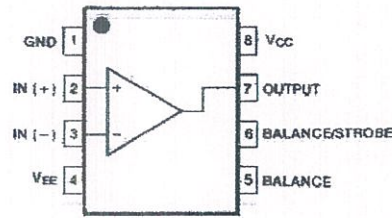
ไอซีเปรียบเทียบแรงดัน LM311 จะมีคุณสมบัติที่ดีกว่าเมื่อเปรียบเทียบกับการใช้อปแอมป์ วงจรแสดงดังรูปที่ 2-11



รูปที่ 2.21 วงจรภายในของ ไอซีเบอร์ LM311

ไอซี Comparator แบบตัวเดียวโดยขาไฟลบ ( $V_{EE}$ ) ถูกแยกจากขากราวด์ของเอาต์พุต (ใน LM339 จะรวมกันอยู่) และเอาต์พุตยังคงเป็นแบบ open collector เหมือนเดิม การวางขาไอซีแสดงดังรูปที่ 2-12

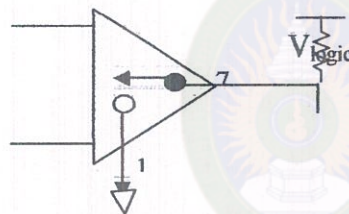




รูปที่ 2.22 โครงสร้างภายใน IC LM339

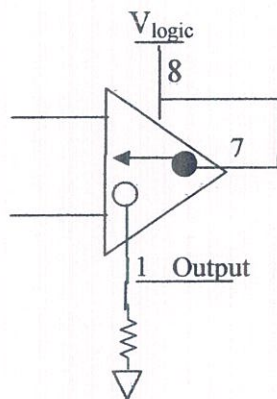
การใช้งานต้องต่อไฟแหล่งจ่ายบวก V (ไม่เกิน 36 V) ที่ขา 8 และไฟ -V (ไม่ต่ำกว่า -30V) (หรืออาจต่อกราวด์) ที่ขา 4 ส่วนขาเอาต์พุตเป็นอิสระจากแหล่งจ่ายไฟไม่จำเป็นต้องใช้แหล่งจ่ายเดียวกันก็ได้หรืออาจใช้แหล่งจ่ายเดียวกันก็ได้โดยมากถูกนำไปใช้ในวงจรดิจิตอลนั้นคือใช้ 5 V

การทำงานคล้ายกับ LM339 กล่าวคือที่เอาต์พุตจะต่อ ตัวต้านทานแบบ pull up และต่อกับเอาต์พุตของไอซี โดยภายในไอซีที่ของเอาต์พุตเสมือนต่ออยู่กับสวิทช์ดังรูปที่ 14 ถ้า inverting input มีค่ามากกว่า noninverting input เปรียบเหมือนสวิทช์ปิดขาเอาต์พุตต่อกับกราวด์ได้ลอจิกศูนย์



รูปที่ 2.23 ต่อวงจรต้องการลอจิก 0

ทำนองกลับกันถ้า Inverting input มีค่าน้อยกว่า noninverting input เปรียบเสมือนสวิทช์เปิดเอาต์พุตต่อกับ  $V_{logic}$  ได้ลอจิกหนึ่ง และเพราะขากราวด์และ  $V_{EE}$  แยกอิสระจากกันเมื่อปรับวงจรดังรูปที่ 2-14 จะได้เอาต์พุตลอจิกที่ตรงกันข้ามกับรูปที่ 2-13

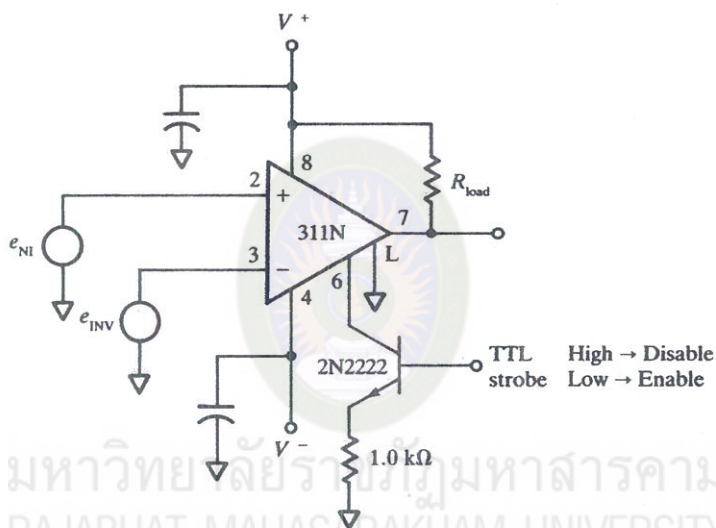


รูปที่ 2.24 ต่อวงจรต้องการลอจิก 1

กล่าวคือถ้า Noninverting input มีค่ามากกว่า Inverting input เปรียบเสมือนสวิตช์เปิด เอาท์พุตต่อกับกราวด์ ได้ลอจิกศูนย์ และในทางตรงกันข้าม Noninverting input มีค่าน้อยกว่า Inverting input เปรียบเสมือนสวิตช์ปิดเอาท์พุตต่อกับ  $V_{logic}$  ได้ลอจิกศูนย์

ในการใช้งานมีอีก 2 สิ่งที่สำคัญคือการเซต Balance offset ที่ขา 5 และ 6 สำหรับวิธีการเซตให้ดูที่คู่มืออีกที แต่ถ้าไม่ใช่ให้ทำการ short ขา 5 กับขา 6 เข้าด้วยกันเพื่อลดผลของสัญญาณรบกวนที่จะรบกวนอินพุต

และการเซตสโตรบ (ขา 6) เพื่อทำให้เอาท์พุตสวิตช์เปิดโดยไม่คำนึงถึงสัญญาณอินพุต โดยเมื่อกระแส 3mA ถึง 5mA ไหลออกจากขาสโตรบเอาท์พุตทรานซิสเตอร์จะอยู่ในสภาวะสวิตช์ปิด ตัวอย่างการใช้งานขาสโตรบแสดงดังรูปที่ 2-15

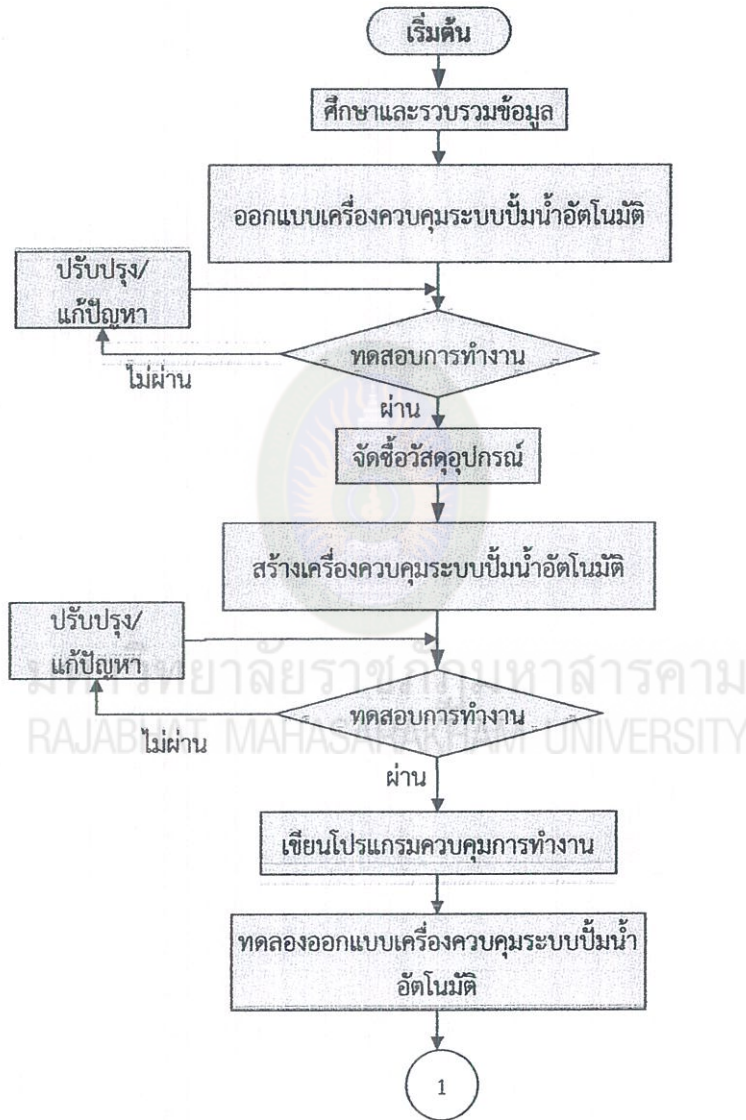


รูปที่ 2.25 การเซตสโตรบ (ขา 6)

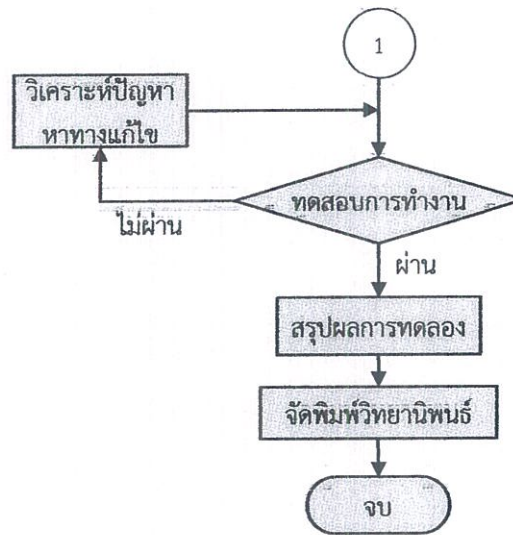
### บทที่ 3

#### วิธีการดำเนินการวิจัย

เมื่อผู้จัดทำโครงการได้ทำการศึกษาทฤษฎีที่เกี่ยวข้องแล้ว รวบรวมทฤษฎีที่เกี่ยวข้องและเนื้อหาที่ใช้ในการจัดทำโครงการไว้ในบทที่ 2 สามารถดำเนินการตามขั้นตอนดังรูปที่ 3-1



รูปที่ 3.1 แผนผังขั้นตอนการดำเนินโครงการ



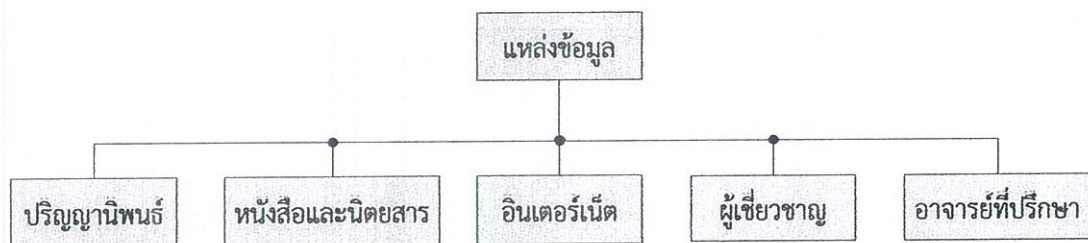
รูปที่ 3.1 (ต่อ) แผนผังขั้นตอนการดำเนินโครงการ

### 3.1 การศึกษาและรวบรวมข้อมูล

การศึกษาและรวบรวมข้อมูล เพื่อนำข้อมูลที่ได้นั้นมาทำการประยุกต์ใช้ในการสร้างโครงงาน เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ ได้ข้อมูลจากแหล่งข้อมูลต่างๆดังต่อไปนี้

- วิทยานิพนธ์
- อินเทอร์เน็ต
- หนังสือและนิตยสาร
- ผู้เชี่ยวชาญ
- อาจารย์ที่ปรึกษา

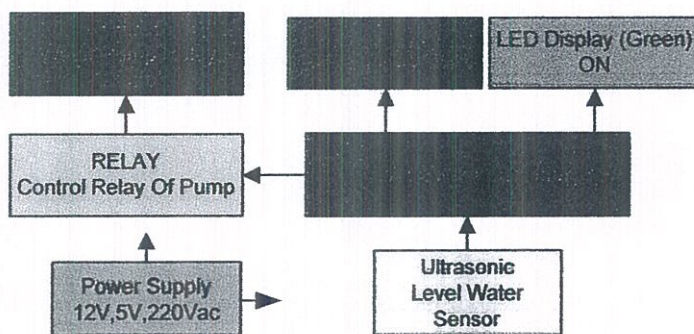
การศึกษาและรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆนั้นสามารถเขียนเป็นแผนผังได้ ดังนี้แสดงในรูปที่ 3-2



รูปที่ 3.2 แผนผังการรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ

### 3.2 การออกแบบและสร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ

การออกแบบและสร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ สามารถออกแบบบล็อกไดอะแกรมการทำงานได้ดังรูปที่ 3-3



รูปที่ 3.3 บล็อกไดอะแกรมการทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ

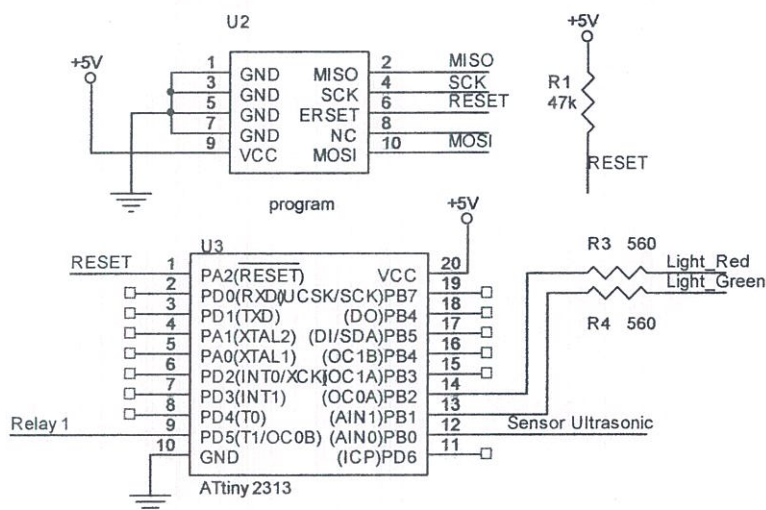
การทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ จะใช้ไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 เป็นตัวประมวลผล โดยหลักการทำงานมีดังนี้ เริ่มต้นจากภาคจ่ายไฟเลี้ยง 12 โวลต์ จะได้ไฟเลี้ยง 5 โวลต์ใช้เลี้ยง ไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 วงจรอุลตราโซนิกตรวจวัดระดับน้ำ (Ultrasonic Level Water Sensor) และใช้ไฟเลี้ยง 12 โวลต์ เลี้ยงรีเลย์ตัวที่ 1 จะควบคุมการทำงานโดยใช้คำสั่งจากไมโครคอนโทรลเลอร์ เมื่อวัดระดับน้ำอยู่ระดับใกล้หมด แล้วจะสั่งให้รีเลย์ตัวที่ 1 ทำงาน แล้วตัด-ต่อไฟ 220 โวลต์ เอซี ให้กับรีเลย์ตัวที่ 2,3 ทำหน้าที่คล้ายแมกเนติก แต่มีขนาดเล็กกว่า ทำงานเพื่อต่อไฟให้กับปั้มน้ำ พร้อมมีไฟแสดงสถานะการทำงานของปั้มน้ำ ถ้าปั้มน้ำทำงานไฟสีแดงจะติด และถ้าปั้มน้ำไม่ทำงานไฟสีแดงจะติด

จากบล็อกไดอะแกรมการทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ สามารถแบ่งวงจรย่อยที่ตามหน้าที่ได้ดังนี้

#### 3.2.1 วงจรควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313

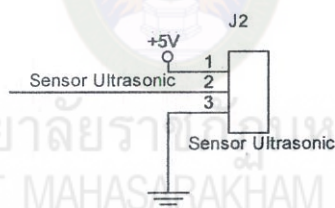
ในรูปที่ 3-4 แสดงวงจรในภาคส่งโดยใช้ไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 เนื่องจาก ไมโครคอนโทรลเลอร์ตระกูล เบอร์นี้ มีขนาดเล็กเหมาะกับงานขนาดเล็ก ด้วยเหตุนี้จึงใช้ไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 ซึ่งใช้ไฟเลี้ยงวงจร 5V ขารี่เซตจะทำงานที่ลอจิก '0' ซึ่งจะใช้ตัวต้านทาน (R1 47k) คริสตอลขนาด 8 MHz ภายในและ มีคอนเน็คเตอร์ ( U1) ในการต่อเพื่อทำการโหลดโปรแกรมที่เขียนควบคุมเข้าไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313 ขา

PD5 ใช้ควบคุมรีเลย์ ขา PB0 ใช้สื่อสารกับโมดูลตรวจวัดระยะทาง (Ultrasonic Sensor) และขา PB1, PB2 ควบคุมไฟแสดงสถานะของปั้ม สีแดงคือปั้มไม่ทำงาน สีเขียวคือปั้มทำงาน ตามลำดับ



รูปที่ 3.4 วงจรควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR ATTINY2313

### 3.2.2 วงจรตรวจวัดระดับน้ำ



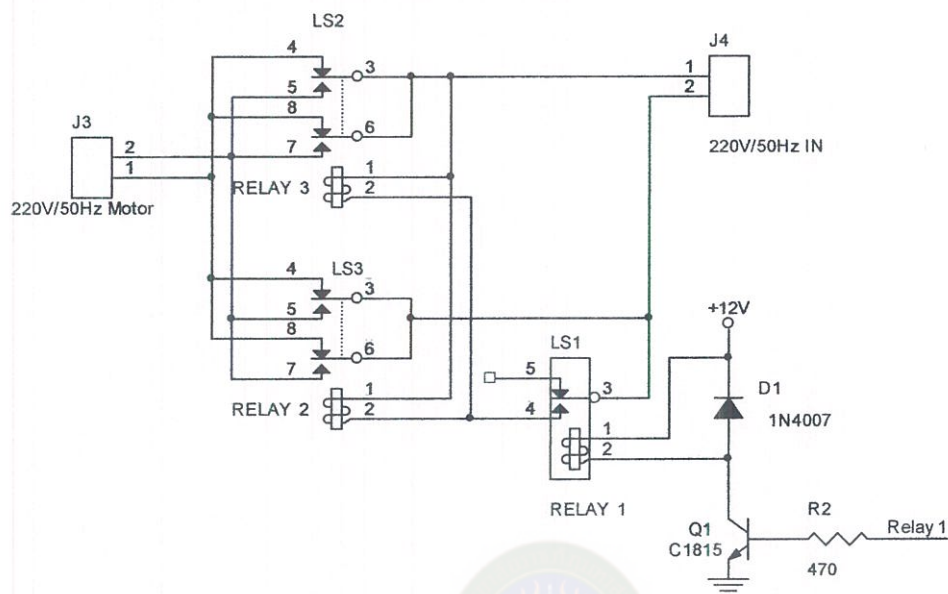
### รูปที่ 3.5 วงจรตรวจวัดระดับน้ำ

การใช้วงจรตรวจวัดระดับน้ำ จะใช้โมดูลสำเร็จรูป โดยจะมีตำแหน่งขาการต่อใช้งานตามวงจรด้านบน การอ่านค่าหรือการสื่อสารกับโมดูลตรวจวัดระดับน้ำ โดยการสื่อสารเพียงแค่ว่าเดียว เวลาจะสื่อสารต้องส่งพัลส์ไปก่อน ความกว้าง 20 ms จากนั้นรอสัญญาณที่สะท้อนกลับคืนนับคาบเวลาแล้วอ่านค่าเวลาแล้วแปลงเป็นระยะห่าง

### 3.2.3 วงจรควบคุมรีเลย์ ตัดต่อไฟสู่อุปกรณ์

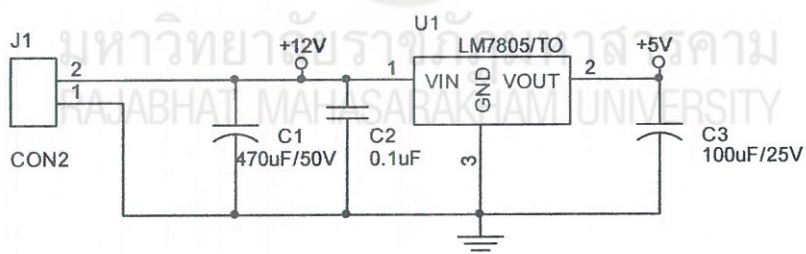
จะใช้รีเลย์ 3 ตัว ตัวที่ 1 ใช้ไฟเลี้ยง 12 โวลต์ ดีซี รับคำสั่งจากไมโครคอนโทรลเลอร์ผ่าน ทรานซิสเตอร์ Q1 C1815 ควบคุมให้รีเลย์ตัวที่ 1 ทำงาน เพื่อควบคุมการจ่ายไฟ 220 โวลต์ เอซี ไปจ่ายให้รีเลย์ตัวที่ 2 และรีเลย์ตัวที่ 3 โดยในวงจรนี้จะให้หน้าสัมผัสของรีเลย์ ตัวที่ 2 ขนานกันเพื่อ

เพิ่มความทนต่อกระแสไฟ รีเลย์ตัวที่ 3 เช่นเดียวกัน จะใช้ J3 เป็นจุดต่อไฟ 220 โวลต์ เอซี เป็นอินพุต และใช้ J4 เป็น เอาต์พุตไปจ่ายไฟให้ปั้มน้ำ ดังรูปที่ 3-5



รูปที่ 3.6 วงจรควบคุมปั้มน้ำ

### 3.2.4 วงจรภาคจ่ายไฟ



รูปที่ 3.7 วงจรภาคจ่ายไฟ

ในรูปที่ 3-6 แสดงวงจรภาคจ่ายไฟจะนำแรงดันจากอะแดปเตอร์ 12V ไปเลี้ยงรีเลย์และนำมาทำการแปลงแรงดันให้ลดลงเหลือ 5V นำแรงดันนี้ไปจ่ายให้กับไมโครคอนโทรลเลอร์ตระกูล AVR เบอร์ ATTINY2313 และวงจรต่างๆ

## บทที่ 4

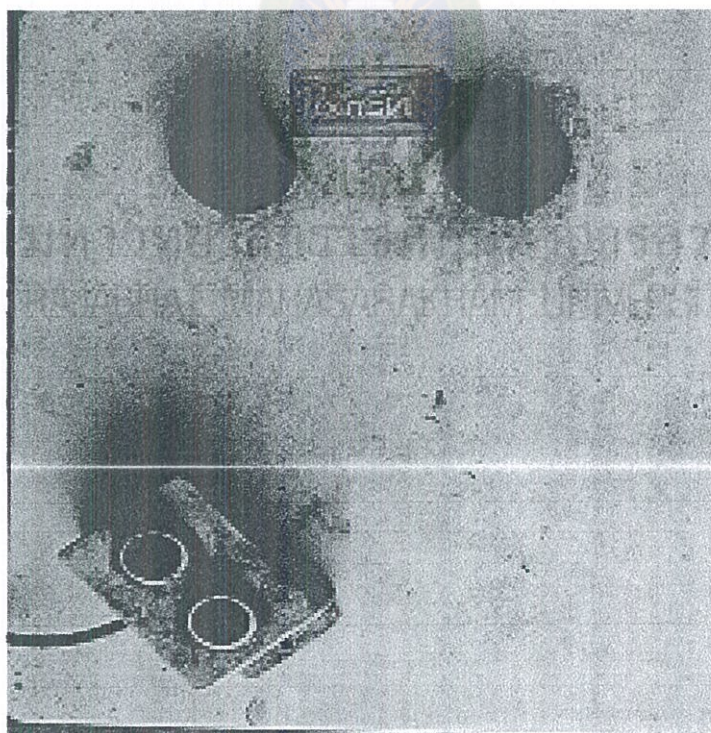
### ผลการวิจัย

ในบทนี้จะกล่าวถึงผลของการดำเนินงานของโครงการ เพื่อให้เป็นไปตามขอบเขตที่กำหนดไว้ ซึ่งหลังจากได้จัดทำเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ สมบูรณ์แล้วจึงได้ดำเนินการออกแบบโปรแกรมให้สามารถทำงานได้ตามขอบเขตที่ได้กำหนดไว้แล้ว การดำเนินงานของโครงการนี้ จะสามารถแบ่งผลที่ได้ออกเป็น 2 ส่วนคือ

1. ผลของโครงการ
2. ผลการทดสอบโครงการ

#### 4.1 ผลของโครงการ

เป็นผลจากการปฏิบัติงานเพื่อให้ได้ผลตามขอบเขตที่กำหนด ซึ่งมีผลของโครงการดังนี้



รูปที่ 4-1 ผลของโครงการ



## 4.2 ผลการทดสอบโครงการ

สามารถทดสอบออกตามขอบเขตที่ได้กำหนดไว้ดังต่อไปนี้

การทดสอบการทำงานของเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ โดยกำหนดดังนี้ ระดับน้ำจะทำงานได้ดีที่ระยะความลึกของน้ำประมาณ 2.5 - 3 เมตร ถึงจะสั่งให้ปั้มทำงาน และปั้มน้ำจะหยุดทำงานเมื่อระดับน้ำอยู่ห่างจากน้ำประมาณ 10 เซนติเมตร สามารถดำเนินการทดสอบได้ดังต่อไปนี้

4.2.1.1 ติดตั้งเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ พร้อมใช้งาน

4.2.1.2 ติดตั้งเซนเซอร์วัดระดับน้ำให้สูงห่างจากพื้น 3 เมตร แล้วทดสอบ

4.2.1.3 ทดสอบโดยการใช้สมุดจำลองเป็นระดับน้ำค่อยๆ เลื่อนขึ้นจากด้านล่างถึงน้ำ (ระดับน้ำต่ำสุดแล้วค่อยๆเพิ่มขึ้น) ในตารางที่ 4.1 และ(ระดับน้ำที่สูงที่สุดแล้วค่อยๆลดลง) ในตารางที่ 4.2

ตารางที่ 4.1 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำต่ำสุดแล้วค่อยๆเพิ่มขึ้น

ผิวหน้า ห่างจาก เซนเซอร์ (ซ.ม.)	การทำงานของปั้มน้ำ						รวม
	ออกแบบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
		ทดสอบ	ทดสอบ	ทดสอบ	ทดสอบ	ทดสอบ	
300	✓	✓	✓	✓	✓	✓	5
250	✓	✓	✓	✓	✓	✓	5
200	✓	✓	✓	✓	✓	✓	5
150	✓	✓	✓	✓	✓	✓	5
100	✓	✓	✓	✓	✓	✓	5
50	✓	✓	✓	✓	✓	✓	5
10	x	x	x	x	x	x	5
รวม							35/35
การทำงานของปั้มน้ำคิดเป็นร้อยละ							100

จากตารางที่ 4.1 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำต่ำสุดแล้วค่อยๆเพิ่มขึ้น ผลปรากฏว่าการทดสอบการทำงานของปั้มน้ำ จะเห็นได้ว่าปั้มน้ำจะทำงานเมื่อระดับน้ำห่างจากเซนเซอร์ 300 เซนติเมตร จนถึงระยะห่างประมาณ 10 เซนติเมตร หรือน้ำเต็มถึงเครื่องจะหยุดทำงานเมื่อทดสอบแต่ละครั้ง ที่ระดับความห่างแตกต่างกัน ได้ผลการทำงานของปั้มน้ำคิดเป็นร้อยละของการทำงาน 100

ตารางที่ 4.2 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำสูงสุดแล้วค่อยๆ ลดระดับลง

ผิวน้ำห่างจาก เซนเซอร์ (ซ.ม.)	การทำงานของปั้มน้ำ						รวม
	ออกแบบ	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
		ทดสอบ	ทดสอบ	ทดสอบ	ทดสอบ	ทดสอบ	
10	x	x	x	x	x	x	5
50	x	x	x	x	x	x	5
100	x	x	x	x	x	x	5
150	x	x	x	x	x	x	5
200	x	x	x	x	x	x	5
250	√	√	√	√	√	√	5
300	√	√	√	√	√	√	5
	รวม						35/35
การทำงานของปั้มน้ำคิดเป็นร้อยละ							100

จากตารางที่ 4.1 ผลการทดสอบการทำงานของปั้มน้ำ จากระดับน้ำสูงสุดแล้วค่อยๆ ลดลง ผลปรากฏว่าการทดสอบการทำงานของปั้มน้ำ จะเห็นได้ว่าปั้มน้ำจะทำงานเมื่อระดับน้ำห่างจากเซนเซอร์ 250 เซนติเมตร จนถึงระยะห่างประมาณ 300 เซนติเมตร หรือน้ำจะหมดถึงเครื่องจะทำงาน เมื่อทดสอบแต่ละครั้ง ที่ระดับความห่างแตกต่างกัน ได้ผลการทำงานของปั้มน้ำคิดเป็นร้อยละของการทำงาน 100

## บทที่ 5

### สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิจัยชิ้นนี้เมื่อดำเนินการเสร็จสิ้นเพื่อเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ ให้สามารถใช้งานได้จริงในชีวิตประจำวัน เพื่อควบคุมระบบปั้มน้ำบ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น โดยสามารถที่จะควบคุมการทำงานโดยใช้คำสั่งจากไมโครคอนโทรลเลอร์ เมื่อวัดระดับน้ำอยู่ระดับใกล้หมด แล้วจะสั่งให้รีเลย์ตัวที่ 1 ทำงาน แล้วตัด-ต่อไฟ 220 โวลต์ เอซี ให้กับรีเลย์ตัวที่ 2,3 ทำหน้าที่คล้ายแมกเนติก แต่มีขนาดเล็กกว่า ทำงานเพื่อต่อไฟให้กับปั้มน้ำ พร้อมมีไฟแสดงสถานะการทำงานของปั้ม ถ้าปั้มทำงานไฟสีแดงจะติด และถ้าปั้มไม่ทำงานไฟสีแดงจะติด ได้ผลสรุปตามวัตถุประสงค์ และขอบเขตที่ตั้งกล่าวไว้ในบทที่ 1 โดยได้สร้างเครื่องควบคุมระบบปั้มน้ำอัตโนมัติทำงานตามที่ตั้งไว้ได้ และสั่งให้เปิด-ปิดแบบปกติได้

จากการทดสอบปั้มน้ำจะทำงานเมื่อระดับน้ำห่างจากเซนเซอร์ 300 เซนติเมตร จนถึงระยะห่างประมาณ 10 เซนติเมตร หรือน้ำเต็มถึงเครื่องจะหยุดทำงาน เมื่อทดสอบแต่ละครั้งและปั้มน้ำจะทำงานเมื่อระดับน้ำห่างจากเซนเซอร์ 250 เซนติเมตร จนถึงระยะห่างประมาณ 300 เซนติเมตร หรือน้ำจะหมดถึงเครื่องจะทำงาน เมื่อทดสอบแต่ละครั้ง ที่ระดับความห่างแตกต่างกัน

#### 5.2 ปัญหาและแนวทางการแก้ไข

ปัญหาที่เกิดขึ้นในระหว่างดำเนินการ เป็นปัญหาที่จำเป็นต้องได้รับการแก้ไข โดยจะมีลักษณะของปัญหาและวิธีการแก้ไขดังต่อไปนี้

5.2.1 เขียนโปรแกรมมากเป็นพิเศษ ซึ่งหน่วยความจำของไมโครคอนโทรลเลอร์ไม่เพียงพอที่จะเขียนฟังก์ชันย่อย

-แนวทางแก้ไข ใช้ไมโครคอนโทรลเลอร์ที่มีหน่วยความจำเพิ่มมากขึ้น

5.2.2 จอแอลซีดี (LCD Display) ไมโครคอนโทรลเลอร์ วงจรสวิตซ์ ไอซีรักษาระดับแรงดัน ซึ่งส่วนประกอบเหล่านี้ เมื่อนำมารวมต่อเข้าวงจร จะดึงกระแสสูงพอสมควร ถ้าใช้แบตเตอรี่ 9V แบบทั่วไป ทำให้ระยะเวลาการควบคุมได้ไม่นาน

-แนวทางแก้ไข ใช้แบตเตอรี่ที่สามารถชาร์ตประจุได้

### 5.3 ข้อเสนอแนะ

โครงการนี้ แม้ว่าจะใช้งานได้อย่างถูกต้องตามขอบเขตแล้วก็ตาม แต่ยังมีบางส่วนที่สามารถเพิ่มเติมเพื่อทำให้มีประสิทธิภาพมากขึ้น จึงได้จัดทำข้อเสนอแนะ ไว้เป็นแนวทางแก่ผู้สนใจที่จะนำไปพัฒนา หรือจะนำไปประยุกต์ใช้กับโครงการอื่น ๆ ดังต่อไปนี้

5.3.1 พัฒนาเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ บ้านหนองหัววัว ต.โคกสี อ.เมือง จ.ขอนแก่น ให้ควบคุมอุปกรณ์เครื่อง ณ เวลาใดๆก็ได้ หรือสามารถที่จะประยุกต์ใช้ในยานยนต์ต่างๆได้

5.3.2 ออกแบบให้ควบคุมระบบปั้มน้ำอัตโนมัติ ให้มีขนาดกะทัดรัดเพื่อความสะดวกในการพกพา

5.3.3 พัฒนาให้เครื่องควบคุมระบบปั้มน้ำอัตโนมัติ ตรวจสอบระดับน้ำอัตโนมัติ



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## บรรณานุกรม

- นิรุธ อำนวยศิลป์. คู่มือการเขียนโปรแกรม C. กรุงเทพมหานคร: ชัคเชศ มีเดีย จำกัด, 2544
- บรรเจิด ตันติกัลยาภรณ์. เครื่องรับส่ง. พิมพ์ครั้งที่ 1 กรุงเทพฯ : 2538
- บริษัท ซีเอ็ดดูเคชั่น จำกัด(มหาชน). HOBBY ELECTRONICS ฉบับที่ 143. กรุงเทพมหานคร: ส.เอเชียเพรส (1989), 2547
- บริษัท ซีเอ็ดดูเคชั่น จำกัด(มหาชน). HOBBY ELECTRONICS ฉบับที่ 155. กรุงเทพมหานคร: ส.เอเชียเพรส (1989), 2549
- ประจัน พลังสันติกุล. การเขียนโปรแกรมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C WinAVR (C Compiler). กรุงเทพมหานคร: แอพซอฟต์แวร์เทคโนโลยี จำกัด ,2549
- รังสรรค์, เชิด . เครื่องควบคุมอุปกรณ์ไฟฟ้าระยะไกลแบบไร้สาย. มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน (วิทยาเขตขอนแก่น), 2550



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

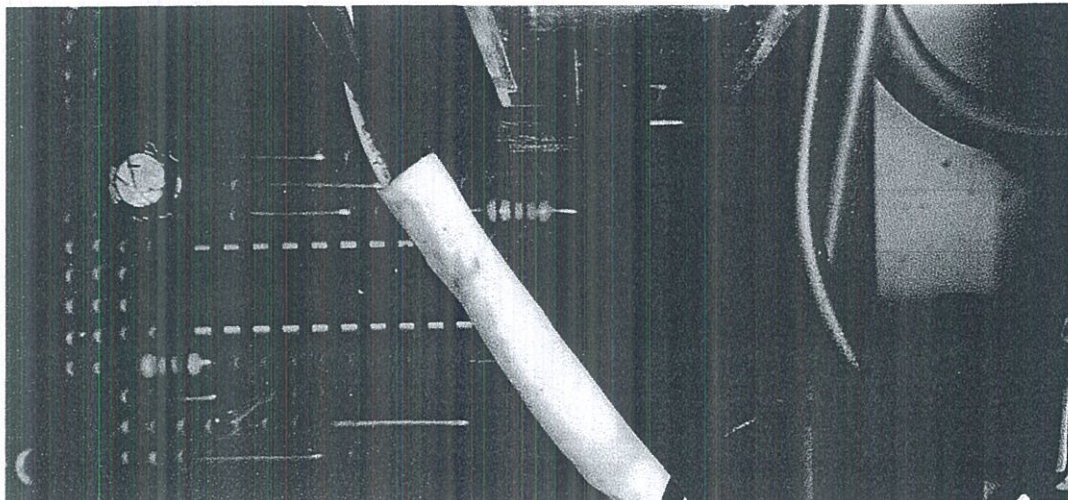


มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

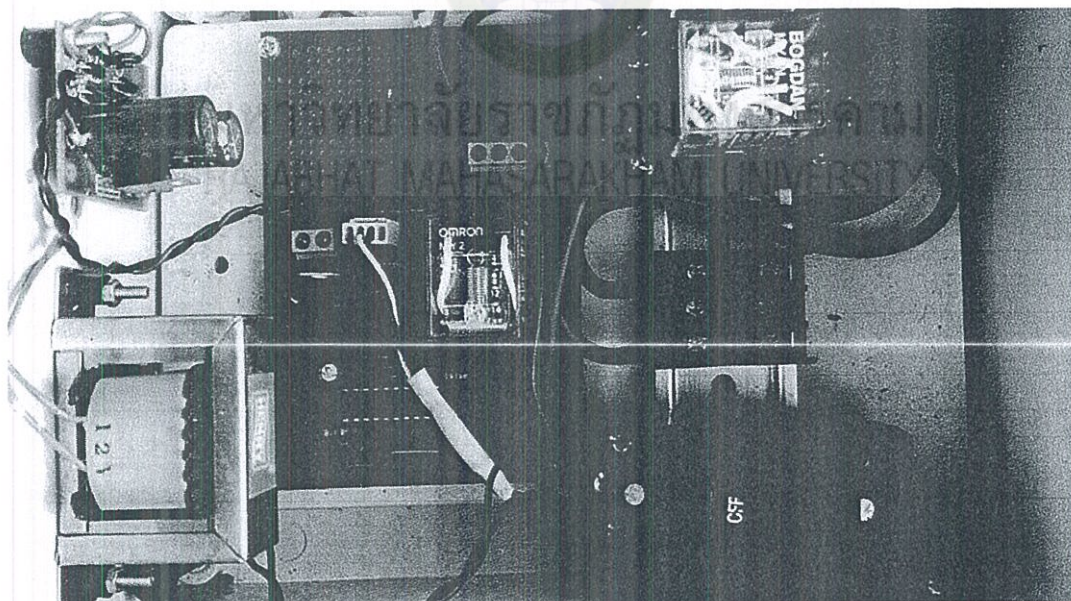


รูปวงจรเครื่องควบคุมระบบปั้มน้ำอัตโนมัติ

มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

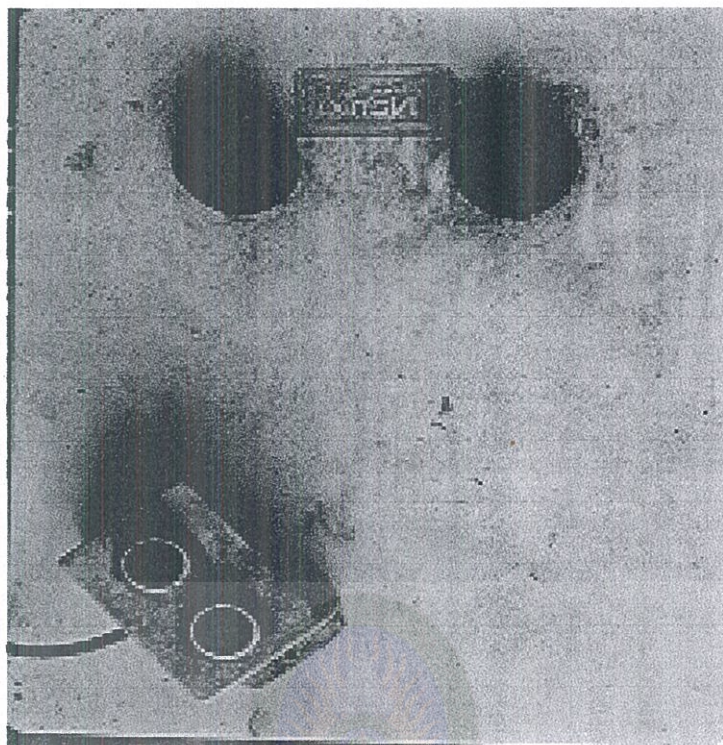


รูปที่ ก-1 แสดงไมโครคอนโทรลเลอร์ (AVR ATTINY2313)



รูปที่ ก-2 การเชื่อมต่อระหว่างอุปกรณ์





รูปที่ ก-3 ลักษณะภายนอกตู้เครื่อง

มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY



โปรแกรมควบคุมเครื่อง

มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

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

This program was produced by the

CodeWizardAVR V2.05.0 Professional

Automatic Program Generator

© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 12/4/2012

Author : NeVaDa

Company :

Comments:

Chip type : ATtiny2313V

AVR Core Clock frequency: 8.000000 MHz

Memory model : Tiny

External RAM size : 0

Data Stack size : 32

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

#include <tiny2313.h>

#include <delay.h>

// Standard Input/Output functions

#include <stdio.h>

// Declare your global variables here

unsigned int Cx,i,ADC;

// Declare your global variables here

unsigned int value\_ADC[30];

unsigned int RxBit1(void)

{

มหาวิทยาลัยราชภัฏมหาสารคาม  
 RAJABHAT MAHASARAKHAM UNIVERSITY

```

Cx=0;
for(i=0;i<10;i++)
{
while(PINB.0==1){Cx++;//delay_us(1);
};
value_ADC[i]=Cx;

}
ADC =
value_ADC[0]+value_ADC[1]+value_ADC[2]+value_ADC[3]+value_ADC[4]+value_ADC[5
]+value_ADC[6]+value_ADC[7]+value_ADC[8]+value_ADC[9];
ADC = ADC/300;

putchar('\r');
putchar('\n');
putchar('A');
putchar(ADC%100000/10000+'0');
putchar(ADC%10000/1000+'0');
putchar(ADC%1000/100+'0');
putchar('.');
putchar(ADC%100/10+'0');

putchar(ADC%10+'0');
putchar('m');
}
void main(void)
{
#pragma optsize-
CLKPR=0x80;
CLKPR=0x00;
#ifdef _OPTIMIZE_SIZE_

```

```
#pragma optsize+  
#endif
```

```
PORTA=0x00;  
DDRA=0x00;  
PORTB=0x00;  
DDRB=0x00;  
PORTD=0x00;  
DDRD=0x7F;  
TCCR0A=0x00;  
TCCR0B=0x00;  
TCNT0=0x00;  
OCR0A=0x00;  
OCR0B=0x00;  
TCCR1A=0x00;  
TCCR1B=0x00;  
TCNT1H=0x00;  
TCNT1L=0x00;  
ICR1H=0x00;  
ICR1L=0x00;  
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
GIMSK=0x00;  
MCUCR=0x00;  
TIMSK=0x00;  
USICR=0x00;  
UCSRA=0x02;  
UCSRB=0x08;  
UCSRC=0x06;
```



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

```
UBRRH=0x00;
```

```
UBRRL=0x67;
```

```
ACSR=0x80;
```

```
DIDR=0x00;
```

```
while (1)
```

```
{
```

```
  DDRB=0xf1;
```

```
  PORTB.0=1;
```

```
  delay_us(10);
```

```
  PORTB.0=0;
```

```
  DDRB=0xf0;
```

```
  delay_us(800);
```

```
  RxBit1();
```

```
  delay_ms(50);
```

```
}
```

```
}
```



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller  
Advanced RISC Architecture
- 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 × 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
- 32Kbytes of In-System Self-programmable Flash program memory
  - 1024Bytes EEPROM
  - 2Kbytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
- In-System Programming by On-chip Boot Program  
True Read-While-Write Operation
- Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
- Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - 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
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
- 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
- 2.7V - 5.5V for ATmega32L
  - 4.5V - 5.5V for ATmega32
- Speed Grades
- 0 - 8MHz for ATmega32L
  - 0 - 16MHz for ATmega32
- Power Consumption at 1MHz, 3V, 25°C
- Active: 1.1mA
  - Idle Mode: 0.35mA
  - Power-down Mode: < 1µA



## 8-bit AVR<sup>®</sup> Microcontroller with 32KBytes In-System Programmable Flash

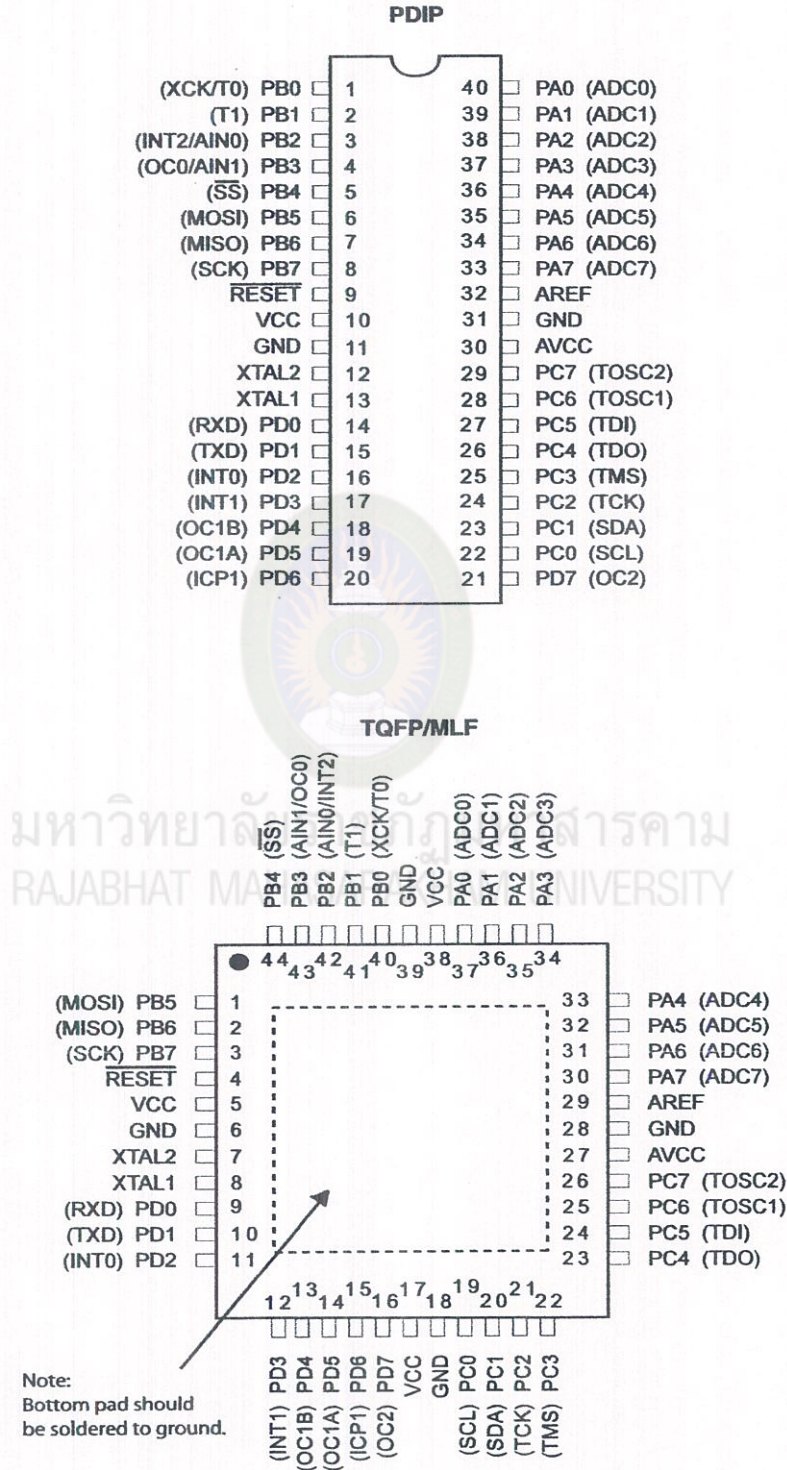
ATmega32  
ATmega32L





## Pin Configurations

Figure 1. Pinout ATmega32

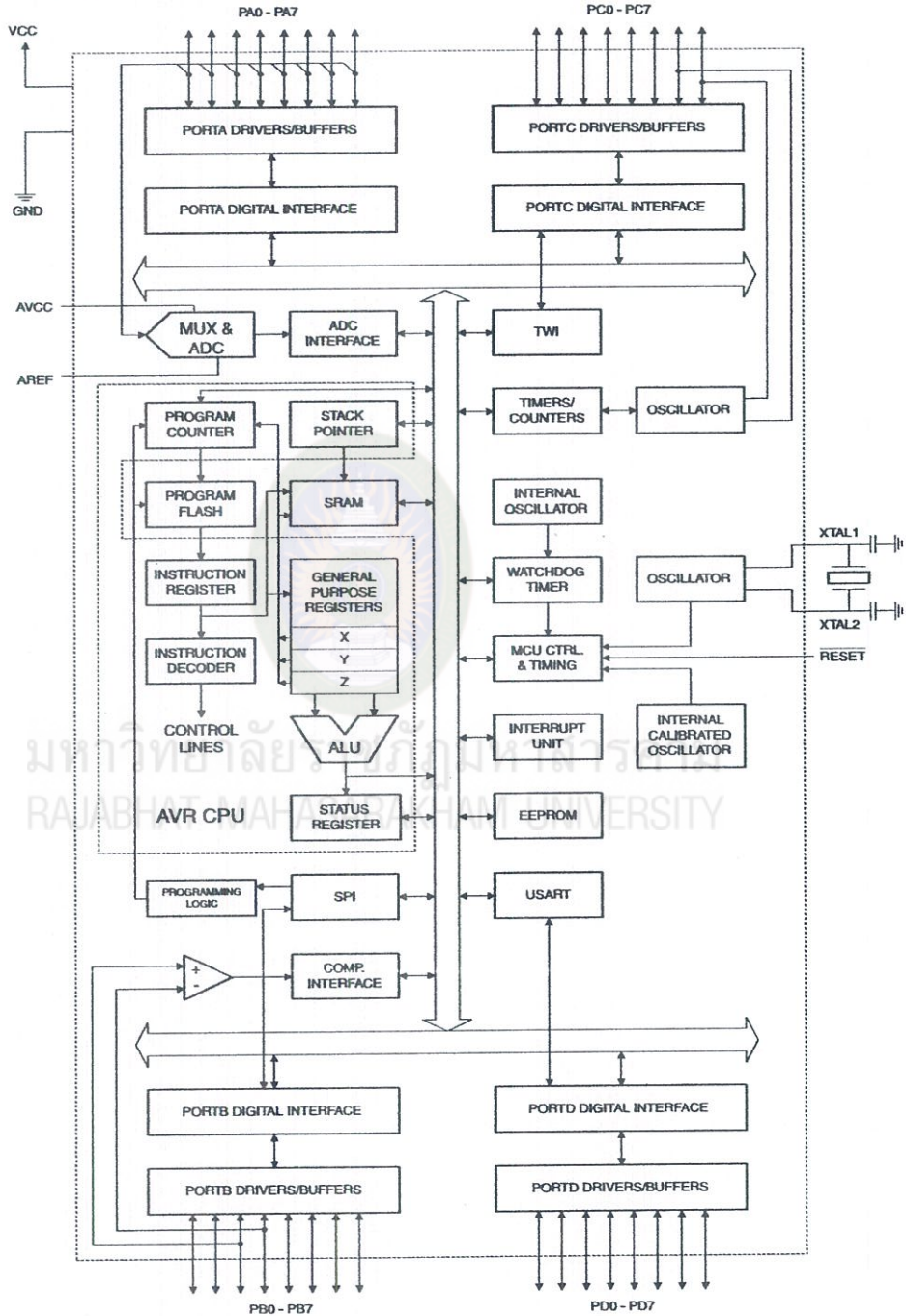


## Overview

The Atmel®AVR®ATmega32 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 ATmega32 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Block Diagram

Figure 2. Block Diagram



The Atmel®AVR®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 clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega32 provides the following features: 32Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 1024bytes EEPROM, 2Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, 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 External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. 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. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

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 ATmega32 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The Atmel AVR ATmega32 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.

## Pin Descriptions

**VCC** Digital supply voltage.

**GND** Ground.

**Port A (PA7..PA0)** Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide 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 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 ATmega32 as listed on page 57.

## 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. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

The TD0 pin is tri-stated unless TAP states that shift out data are entered.

Port C also serves the functions of the JTAG interface and other special features of the ATmega32 as listed on page 60.

## 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 ATmega32 as listed on page 62.

## RST

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 15 on page 37. Shorter pulses are not guaranteed to generate a reset.

## OSC1

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

## OSC2

Output from the inverting Oscillator amplifier.

## VCC

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

## AREF

AREF is the analog reference pin for the A/D Converter.



## Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

## Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

## About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

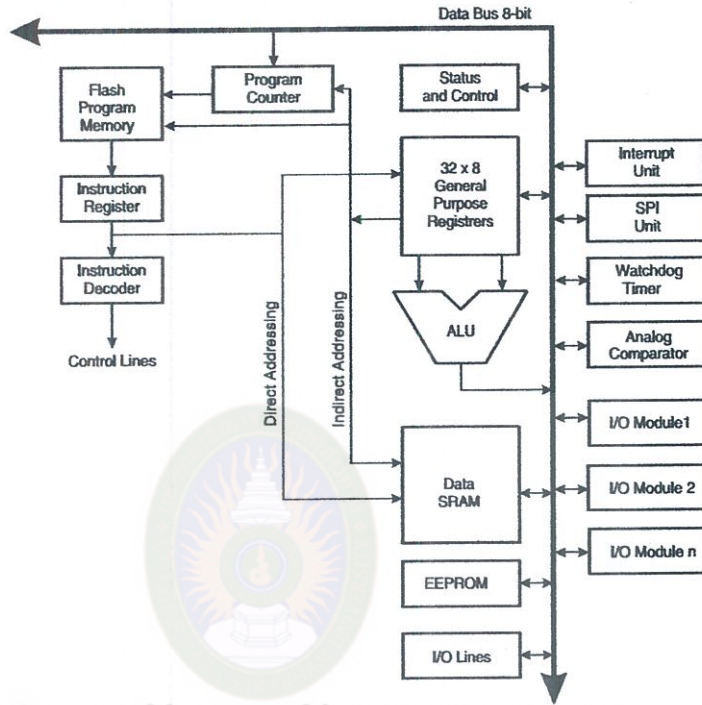
## AVR CPU Core

### Introduction

This section discusses the Atmel® AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

### Architectural Overview

**Figure 3. Block Diagram of the AVR MCU Architecture**



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains  $32 \times 8$ -bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, \$20 - \$5F.

## ALU – Arithmetic Logic Unit

The high-performance Atmel®AVR® ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY



## Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a half carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit,  $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 = N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.



- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

## General Purpose Register File

The Register File is optimized for the Atmel®AVR® Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

**Figure 4. AVR CPU General Purpose Working Registers**

	7	0	Addr.	
General Purpose Working Registers	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	X-register Low Byte
	R27		\$1B	X-register High Byte
	R28		\$1C	Y-register Low Byte
	R29		\$1D	Y-register High Byte
	R30		\$1E	Z-register Low Byte
	R31		\$1F	Z-register High Byte

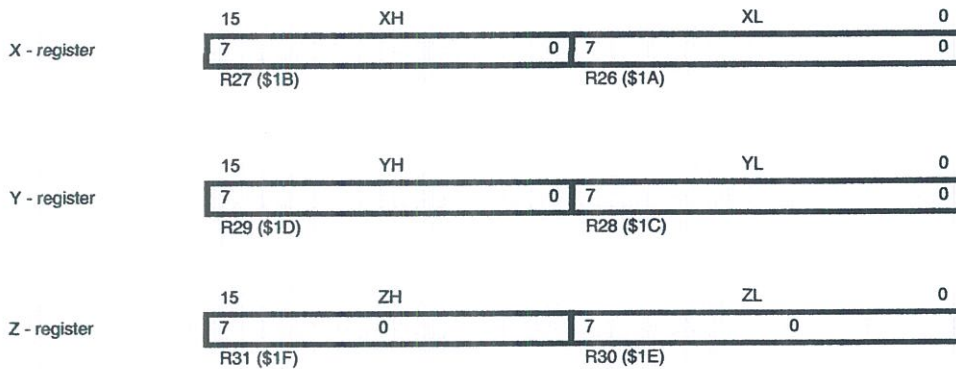
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

## The X-register, Y-register and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

**Figure 5.** The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the Instruction Set Reference for details).

## Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above \$60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

### FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

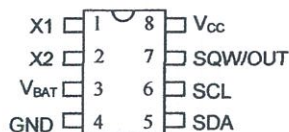
### ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

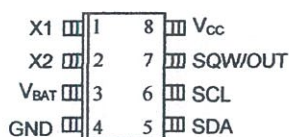
### DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

### PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)

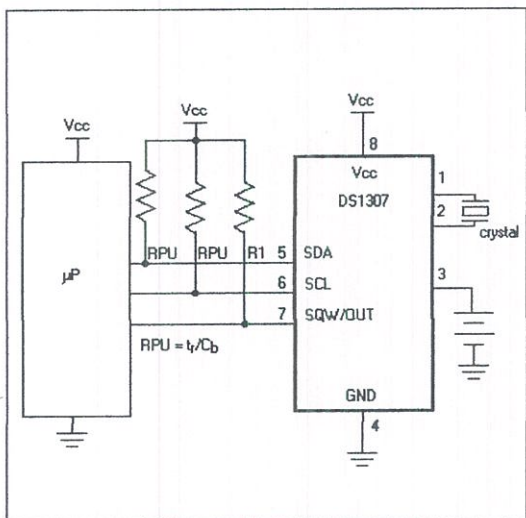


DS1307 8-Pin SOIC (150-mil)

### PIN DESCRIPTION

V <sub>CC</sub>	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V <sub>BAT</sub>	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

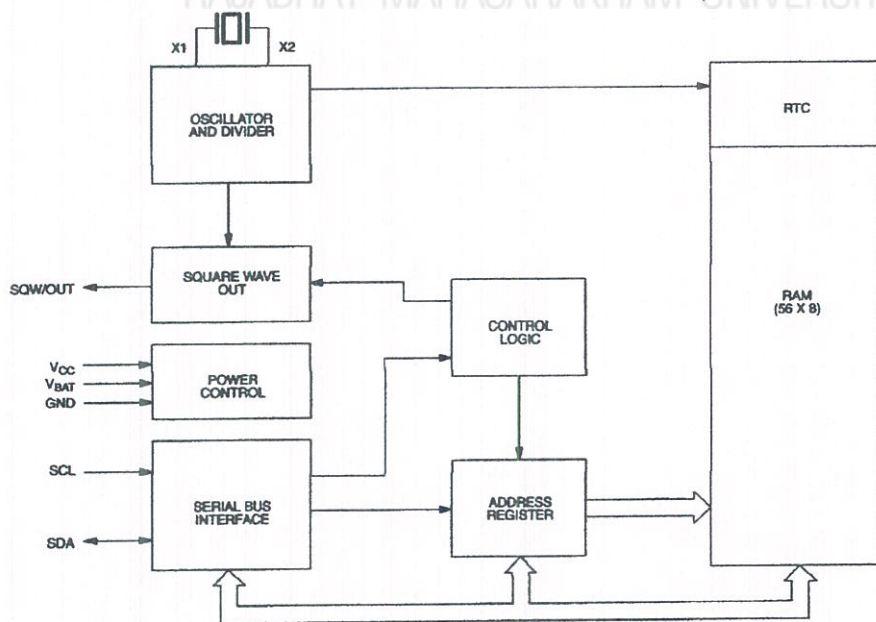
**TYPICAL OPERATING CIRCUIT**



**OPERATION**

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When  $V_{CC}$  falls below  $1.25 \times V_{BAT}$  the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When  $V_{CC}$  falls below  $V_{BAT}$  the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to  $V_{CC}$  when  $V_{CC}$  is greater than  $V_{BAT} + 0.2V$  and recognizes inputs when  $V_{CC}$  is greater than  $1.25 \times V_{BAT}$ . The block diagram in Figure 1 shows the main elements of the serial RTC.

**DS1307 BLOCK DIAGRAM Figure 1**



## SIGNAL DESCRIPTIONS

**V<sub>CC</sub>, GND** – DC power is provided to the device on these pins. V<sub>CC</sub> is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V battery is connected to the device and V<sub>CC</sub> is below 1.25 x V<sub>BAT</sub>, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V<sub>CC</sub> falls below V<sub>BAT</sub> the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V<sub>BAT</sub>.

**V<sub>BAT</sub>** – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V<sub>BAT</sub> nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See “Conditions of Acceptability” at <http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>.

**SCL (Serial Clock Input)** – SCL is used to synchronize data movement on the serial interface.

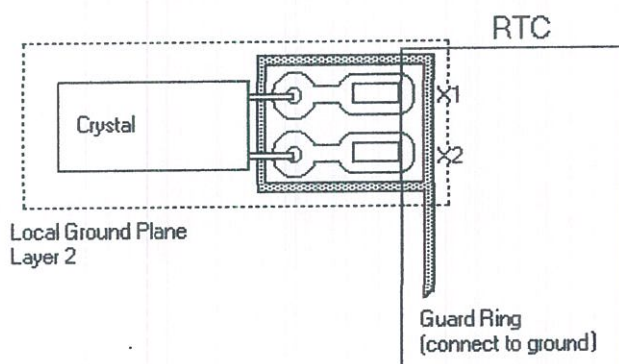
**SDA (Serial Data Input/Output)** – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

**SQW/OUT (Square Wave/Output Driver)** – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either V<sub>cc</sub> or V<sub>bat</sub> applied.

**X1, X2** – Connections for a standard 32.768kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks.” The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

## RECOMMENDED LAYOUT FOR CRYSTAL



## CLOCK ACCURACY

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. See Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks” for detailed information.

Please review Application Note 95, “Interfacing the DS1307 with a 8051-Compatible Microcontroller” for additional information.

## RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

### DS1307 ADDRESS MAP Figure 2

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

## CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The RTC registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. Bit 7 of register 0 is the clock halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

**Please note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.**

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

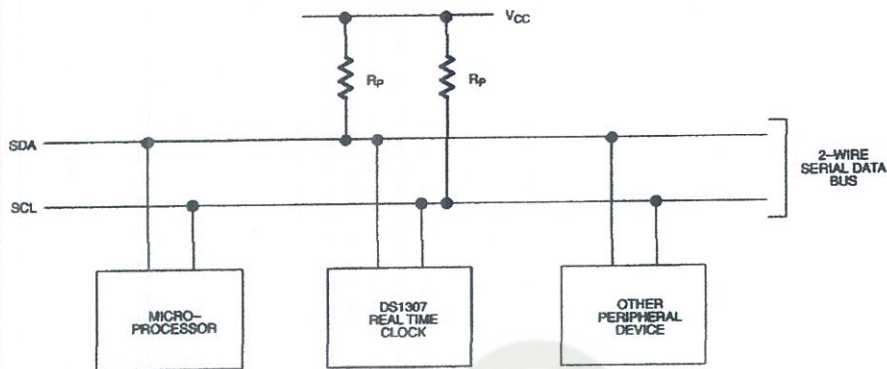




## 2-WIRE SERIAL DATA BUS

The DS1307 supports a bi-directional, 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2-wire bus. A typical bus configuration using this 2-wire protocol is shown in Figure 4.

### TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



Figures 5, 6, and 7 detail how data is transferred on the 2-wire bus.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

**Bus not busy:** Both data and clock lines remain HIGH.

**Start data transfer:** A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

**Stop data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

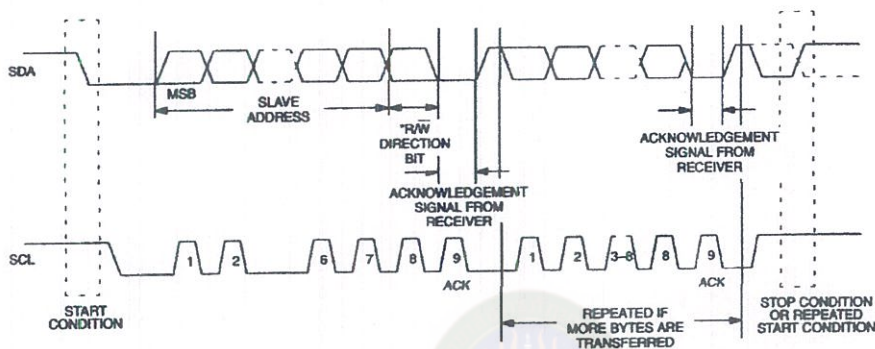
**Data valid:** The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the 2-wire bus specifications a regular mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the regular mode (100kHz) only.

**Acknowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

## DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



Depending upon the state of the  $\overline{R/\overline{W}}$  bit, two types of data transfer are possible:

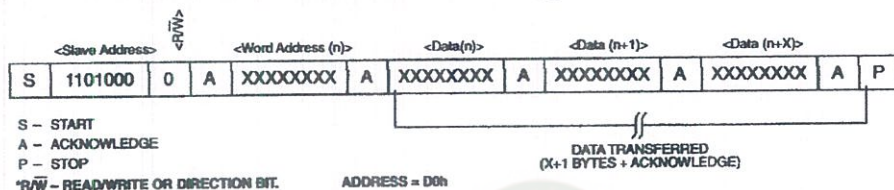
1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 may operate in the following two modes:

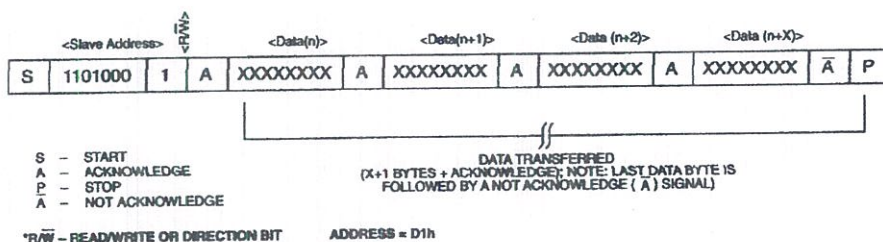
**Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and \*direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the \*direction bit ( $\overline{R/\overline{W}}$ ) which, for a write, is a 0. After receiving and decoding the address byte the device outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307 This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

**DATA WRITE – SLAVE RECEIVER MODE Figure 6**



**Slave transmitter mode (DS1307 read mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the \*direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit DS1307 address, which is 1101000, followed by the \*direction bit ( $\overline{R/\overline{W}}$ ) which, for a read, is a 1. After receiving and decoding the address byte the device inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a “not acknowledge” to end a read.

**DATA READ – SLAVE TRANSMITTER MODE Figure 7**



**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground  
 Storage Temperature  
 Soldering Temperature

-0.5V to +7.0V  
 -55°C to +125°C  
 260°C for 10 seconds DIP  
 See JPC/JEDEC Standard J-STD-020A for  
 Surface Mount Devices

\* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

Range	Temperature	V <sub>CC</sub>
Commercial	0°C to +70°C	4.5V to 5.5V V <sub>CC1</sub>
Industrial	-40°C to +85°C	4.5V to 5.5V V <sub>CC1</sub>

**RECOMMENDED DC OPERATING CONDITIONS**

(Over the operating range\*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V <sub>CC</sub>	4.5	5.0	5.5	V	
Logic 1	V <sub>IH</sub>	2.2		V <sub>CC</sub> + 0.3	V	
Logic 0	V <sub>IL</sub>	-0.5		+0.8	V	
V <sub>BAT</sub> Battery Voltage	V <sub>BAT</sub>	2.0		3.5	V	

\*Unless otherwise specified.

**DC ELECTRICAL CHARACTERISTICS**

(Over the operating range\*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage (SCL)	I <sub>LI</sub>			1	μA	
I/O Leakage (SDA & SQW/OUT)	I <sub>LO</sub>			1	μA	
Logic 0 Output (I <sub>OL</sub> = 5mA)	V <sub>OL</sub>			0.4	V	
Active Supply Current	I <sub>CCA</sub>			1.5	mA	7
Standby Current	I <sub>CCS</sub>			200	μA	1
Battery Current (OSC ON); SQW/OUT OFF	I <sub>BAT1</sub>		300	500	nA	2
Battery Current (OSC ON); SQW/OUT ON (32kHz)	I <sub>BAT2</sub>		480	800	nA	
Power-Fail Voltage	V <sub>PF</sub>	1.216 x V <sub>BAT</sub>	1.25 x V <sub>BAT</sub>	1.284 x V <sub>BAT</sub>	V	8

\*Unless otherwise specified.

## AC ELECTRICAL CHARACTERISTICS

(Over the operating range\*)

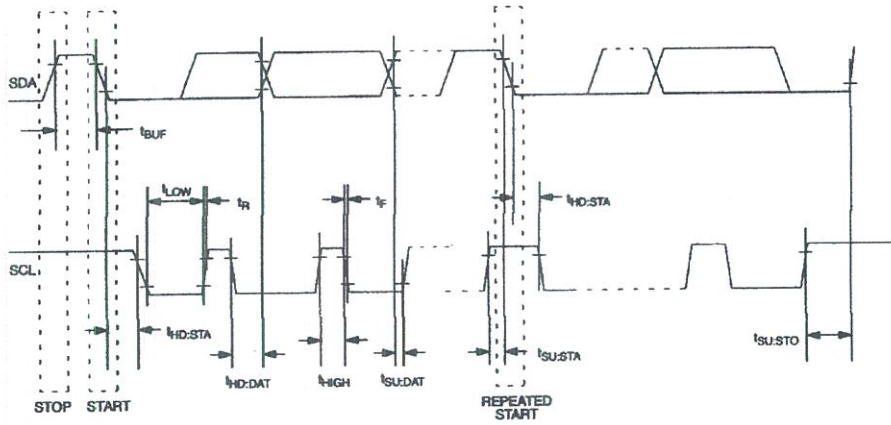
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	$f_{SCL}$	0		100	kHz	
Bus Free Time Between a STOP and START Condition	$t_{BUF}$	4.7			$\mu s$	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			$\mu s$	3
LOW Period of SCL Clock	$t_{LOW}$	4.7			$\mu s$	
HIGH Period of SCL Clock	$t_{HIGH}$	4.0			$\mu s$	
Set-up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			$\mu s$	
Data Hold Time	$t_{HD:DAT}$	0			$\mu s$	4,5
Data Set-up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	$t_R$			1000	ns	
Fall Time of Both SDA and SCL Signals	$t_F$			300	ns	
Set-up Time for STOP Condition	$t_{SU:STO}$	4.7			$\mu s$	
Capacitive Load for each Bus Line	$C_B$			400	pF	6
I/O Capacitance ( $T_A = 25^\circ C$ )	$C_{I/O}$		10		pF	
Crystal Specified Load Capacitance ( $T_A = 25^\circ C$ )			12.5		pF	

\*Unless otherwise specified.

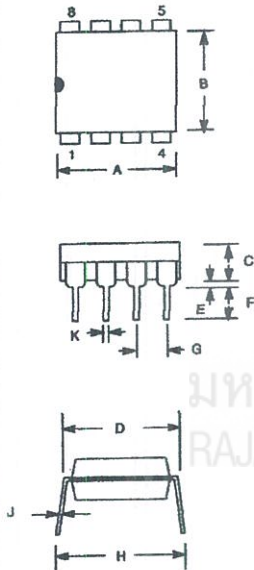
## NOTES:

- $I_{CCS}$  specified with  $V_{CC} = 5.0V$  and SDA, SCL = 5.0V.
- $V_{CC} = 0V$ ,  $V_{BAT} = 3V$ .
- After this period, the first clock pulse is generated.
- A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the  $V_{IHMIN}$  of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
- The maximum  $t_{HD:DAT}$  has only to be met if the device does not stretch the LOW period ( $t_{LOW}$ ) of the SCL signal.
- $C_B$  – Total capacitance of one bus line in pF.
- $I_{CCA}$  – SCL clocking at max frequency = 100kHz.
- $V_{PF}$  measured at  $V_{BAT} = 3.0V$ .

TIMING DIAGRAM Figure 8



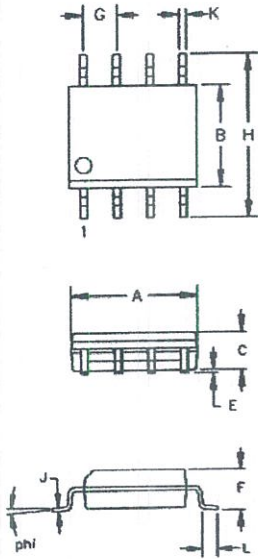
DS1307 64 X 8 SERIAL REAL-TIME CLOCK  
8-PIN DIP MECHANICAL DIMENSIONS



PKG DIM	8-PIN	
	MIN	MAX
A IN.	0.360	0.400
MM	9.14	10.16
B IN.	0.240	0.260
MM	6.10	6.60
C IN.	0.120	0.140
MM	3.05	3.56
D IN.	0.300	0.325
MM	7.62	8.26
E IN.	0.015	0.040
MM	0.38	1.02
F IN.	0.120	0.140
MM	3.04	3.56
G IN.	0.090	0.110
MM	2.29	2.79
H IN.	0.320	0.370
MM	8.13	9.40
J IN.	0.008	0.012
MM	0.20	0.30
K IN.	0.015	0.021
MM	0.38	0.53

# DS1307Z 64 X 8 SERIAL REAL-TIME CLOCK

## 8-PIN SOIC (150-MIL) MECHANICAL DIMENSIONS



PKG	8-PIN (150 MIL)	
	MIN	MAX
A IN.	0.188	0.196
MM	4.78	4.98
B IN.	0.150	0.158
MM	3.81	4.01
C IN.	0.048	0.062
MM	1.22	1.57
E IN.	0.004	0.010
MM	0.10	0.25
F IN.	0.053	0.069
MM	1.35	1.75
G IN.	0.050 BSC	
MM	1.27 BSC	
H IN.	0.230	0.244
MM	5.84	6.20
J IN.	0.007	0.011
MM	0.18	0.28
K IN.	0.012	0.020
MM	0.30	0.51
L IN.	0.016	0.050
MM	0.41	1.27
phi	0°	8°

56-G2008-001



มหาวิทยาลัยราชภัฏมหาสารคาม  
RAJABHAT MAHASARAKHAM UNIVERSITY

**DALLAS**  
SEMICONDUCTOR

## DS1820 1-Wire™ Digital Thermometer

### FEATURES

- Unique 1-Wire™ interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line
- Zero standby power required
- Measures temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  in  $0.5^{\circ}\text{C}$  increments. Fahrenheit equivalent is  $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$  in  $0.9^{\circ}\text{F}$  increments
- Temperature is read as a 9-bit digital value.
- Converts temperature to digital word in 200 ms (typ.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

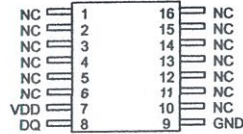
### PIN ASSIGNMENT



DS1820  
PR35 PACKAGE  
See Mech. Drawings  
Section



BOTTOM VIEW



DS1820S  
16-PIN SSOP  
See Mech. Drawings  
Section

### PIN DESCRIPTION

GND	–	Ground
DQ	–	Data In/Out
VDD	–	Optional V <sub>DD</sub>
NC	–	No Connect

### DESCRIPTION

The DS1820 Digital Thermometer provides 9-bit temperature readings which indicate the temperature of the device.

Information is sent to/from the DS1820 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS1820. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS1820 contains a unique silicon serial number, multiple DS1820s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and in process monitoring and control.



## DETAILED PIN DESCRIPTION

PIN 16-PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	Ground.
8	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V <sub>DD</sub>	Optional V <sub>DD</sub> pin. See "Parasite Power" section for details of connection.

DS1820S (16-pin SSOP): All pins not specified in this table are not to be connected.

## OVERVIEW

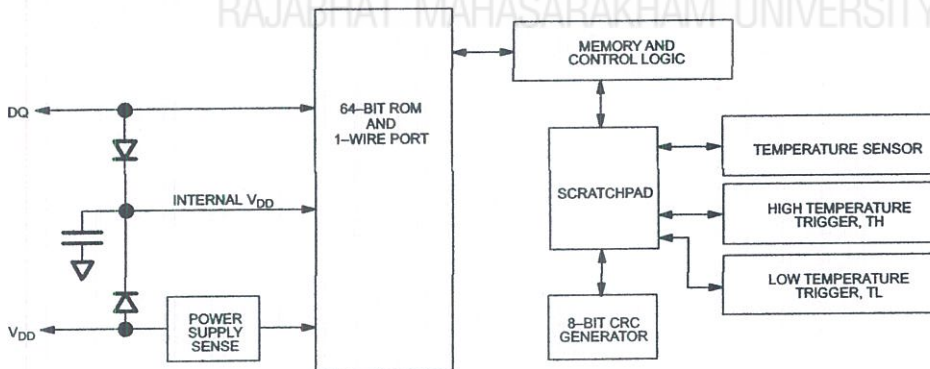
The block diagram of Figure 1 shows the major components of the DS1820. The DS1820 has three main data components: 1) 64-bit lasered ROM, 2) temperature sensor, and 3) nonvolatile temperature alarm triggers TH and TL. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS1820 may also be powered from an external 5 volts supply.

Communication to the DS1820 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out

a specific device if many are present on the 1-Wire line as well as indicate to the Bus Master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS1820 to perform a temperature measurement. The result of this measurement will be placed in the DS1820's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of one byte EEPROM each. If the alarm search command is not applied to the DS1820, these registers may be used as general purpose user memory. Writing TH and TL is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

DS1820 BLOCK DIAGRAM Figure 1



## PARASITE POWER

The block diagram (Figure 1) shows the parasite powered circuitry. This circuitry "steals" power whenever the I/O or  $V_{DD}$  pins are high. I/O will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled "1-Wire Bus System"). The advantages of parasite power are two-fold: 1) by parasiting off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS1820 to be able to perform accurate temperature conversions, sufficient power must be provided over the I/O line when a temperature conversion is taking place. Since the operating current of the DS1820 is up to 1 mA, the I/O line will not have sufficient drive due to the 5K pull-up resistor. This problem is particularly acute if several DS1820's are on the same I/O and attempting to convert simultaneously.

There are two ways to assure that the DS1820 has sufficient supply current during its active conversion cycle. The first is to provide a strong pull-up on the I/O line whenever temperature conversions or copies to the  $E^2$  memory are taking place. This may be accomplished by using a MOSFET to pull the I/O line directly to the power supply as shown in Figure 2. The I/O line must be switched over to the strong pull-up within 10  $\mu$ s maximum after issuing any protocol that involves copying to the  $E^2$  memory or initiates temperature conversions. When using the parasite power mode, the  $V_{DD}$  pin must be tied to ground.

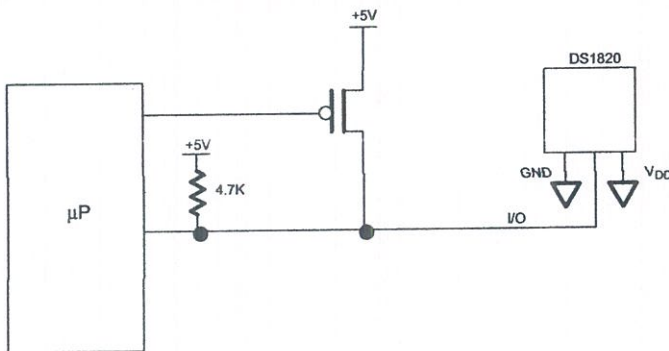
Another method of supplying current to the DS1820 is through the use of an external power supply tied to the

$V_{DD}$  pin, as shown in Figure 3. The advantage to this is that the strong pull-up is not required on the I/O line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1-Wire bus during the conversion time. In addition, any number of DS1820's may be placed on the 1-Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

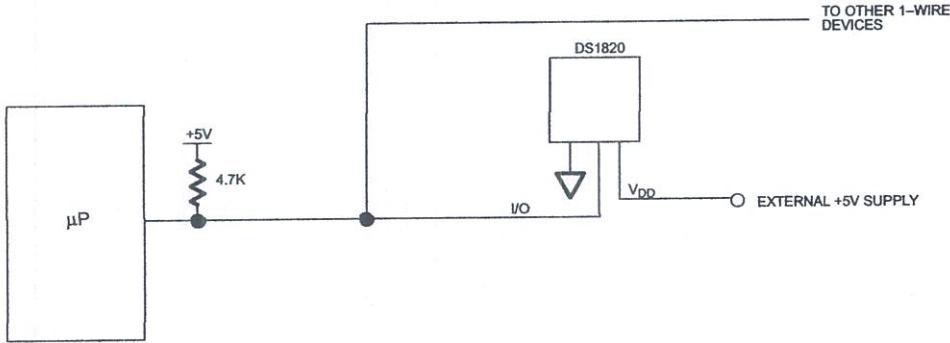
The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS1820 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that  $V_{DD}$  be applied to the DS1820.

For situations where the bus master does not know whether the DS1820's on the bus are parasite powered or supplied with external  $V_{DD}$ , a provision is made in the DS1820 to signal the power supply scheme used. The bus master can determine if any DS1820's are on the bus which require the strong pull-up by sending a Skip ROM protocol, then issuing the read power supply command. After this command is issued, the master then issues read time slots. The DS1820 will send back "0" on the 1-Wire bus if it is parasite powered; it will send back a "1" if it is powered from the  $V_{DD}$  pin. If the master receives a "0", it knows that it must supply the strong pull-up on the I/O line during temperature conversions. See "Memory Command Functions" section for more detail on this command protocol.

**STRONG PULL-UP FOR SUPPLYING DS1820 DURING TEMPERATURE CONVERSION** Figure 2



### USING $V_{DD}$ TO SUPPLY TEMPERATURE CONVERSION CURRENT Figure 3



#### OPERATION – MEASURING TEMPERATURE

The DS1820 measures temperature through the use of an on-board proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 4.

The DS1820 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to  $-55^{\circ}\text{C}$ . If the counter reaches zero before the gate period is over, the temperature register, which is also preset to the  $-55^{\circ}\text{C}$  value, is incremented, indicating that the temperature is higher than  $-55^{\circ}\text{C}$ .

At the same time, the counter is then preset with a value determined by the slope accumulator circuitry. This circuitry is needed to compensate for the parabolic behavior of the oscillators over temperature. The counter is then clocked again until it reaches zero. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the non-linear behavior of the oscillators over temperature, yielding a high resolution temperature measurement. This is done by changing the number of counts necessary for the counter to go through for each incremental degree in temperature. To obtain the desired resolution, therefore, both the value of the counter and the number of counts per degree C (the value of the slope accumulator) at a given temperature must be known.

Internally, this calculation is done inside the DS1820 to provide  $0.5^{\circ}\text{C}$  resolution. The temperature reading is

provided in a 16-bit, sign-extended two's complement reading. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially over the 1-Wire interface. The DS1820 can measure temperature over the range of  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  in  $0.5^{\circ}\text{C}$  increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

Note that temperature is represented in the DS1820 in terms of a  $1/2^{\circ}\text{C}$  LSB, yielding the following 9-bit format:

MSB								LSB
1	1	1	0	0	1	1	1	0

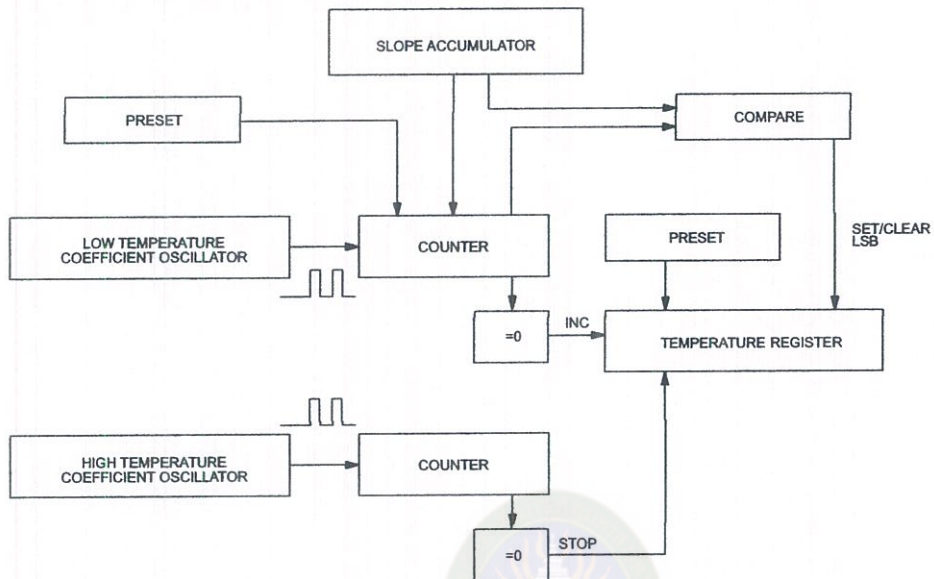
=  $-25^{\circ}\text{C}$

The most significant (sign) bit is duplicated into all of the bits in the upper MSB of the two-byte temperature register in memory. This "sign-extension" yields the 16-bit temperature readings as shown in Table 1.

Higher resolutions may be obtained by the following procedure. First, read the temperature, and truncate the  $0.5^{\circ}\text{C}$  bit (the LSB) from the read value. This value is TEMP\_READ. The value left in the counter may then be read. This value is the count remaining (COUNT\_REMAIN) after the gate period has ceased. The last value needed is the number of counts per degree C (COUNT\_PER\_C) at that temperature. The actual temperature may be then be calculated by the user using the following:

$$\text{TEMPERATURE} = \text{TEMP\_READ} - 0.25 + \frac{(\text{COUNT\_PER\_C} - \text{COUNT\_REMAIN})}{\text{COUNT\_PER\_C}}$$

## TEMPERATURE MEASURING CIRCUITRY Figure 4



TEMPERATURE/DATA RELATIONSHIPS Table 1

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

**OPERATION – ALARM SIGNALING**

After the DS1820 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, the 0.5°C bit is ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set.

This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS1820 will respond to the alarm search command. This allows many DS1820s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

### 64-BIT LASERED ROM

Each DS1820 contains a unique ROM code that is 64-bits long. The first eight bits are a 1-Wire family code (DS1820 code is 10h). The next 48 bits are a unique serial number. The last eight bits are a CRC of the first 56 bits. (See Figure 5.) The 64-bit ROM and ROM Function Control section allow the DS1820 to operate as a 1-Wire device and follow the 1-Wire protocol detailed in the section "1-Wire Bus System". The functions required to control sections of the DS1820 are not accessible until the ROM function protocol has been satisfied. This protocol is described in the ROM function protocol flowchart (Figure 6). The 1-Wire bus master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. After a ROM functions sequence has been successfully executed, the functions specific to the DS1820 are accessible and the bus master may then provide one of the six memory and control function commands.

### CRC GENERATION

The DS1820 has an 8-bit CRC stored in the most significant byte of the 64-bit ROM. The bus master can compute a CRC value from the first 56-bits of the 64-bit ROM and compare it to the value stored within the DS1820 to determine if the ROM data has been received error-free by the bus master. The equivalent polynomial function of this CRC is:

$$\text{CRC} = X^8 + X^5 + X^4 + 1$$

The DS1820 also generates an 8-bit CRC value using the same polynomial function shown above and pro-

vides this value to the bus master to validate the transfer of data bytes. In each case where a CRC is used for data transfer validation, the bus master must calculate a CRC value using the polynomial function given above and compare the calculated value to either the 8-bit CRC value stored in the 64-bit ROM portion of the DS1820 (for ROM reads) or the 8-bit CRC value computed within the DS1820 (which is read as a ninth byte when the scratchpad is read). The comparison of CRC values and decision to continue with an operation are determined entirely by the bus master. There is no circuitry inside the DS1820 that prevents a command sequence from proceeding if the CRC stored in or calculated by the DS1820 does not match the value generated by the bus master.

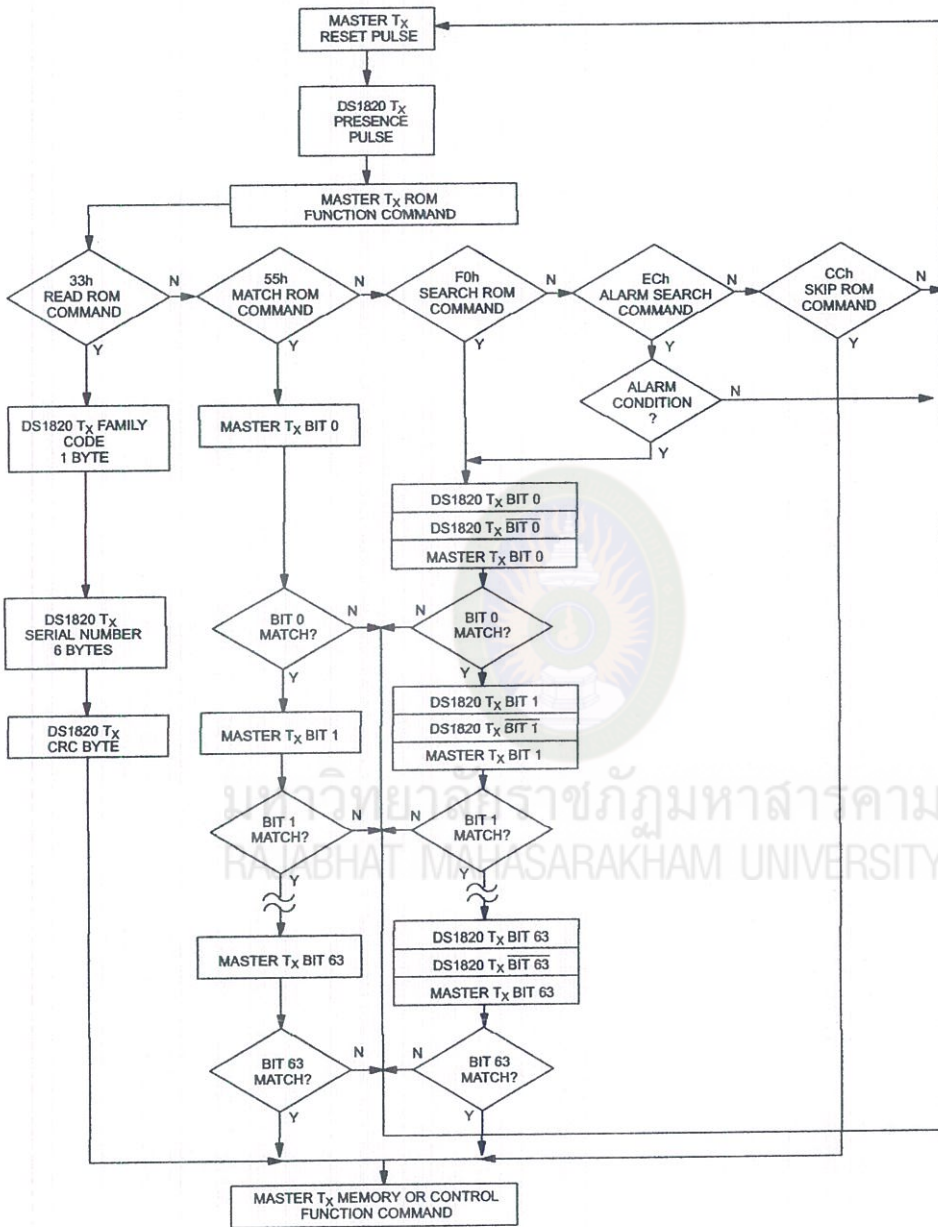
The 1-Wire CRC can be generated using a polynomial generator consisting of a shift register and XOR gates as shown in Figure 7. Additional information about the Dallas 1-Wire Cyclic Redundancy Check is available in Application Note 27 entitled "Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor Touch Memory Products".

The shift register bits are initialized to zero. Then starting with the least significant bit of the family code, one bit at a time is shifted in. After the 8th bit of the family code has been entered, then the serial number is entered. After the 48th bit of the serial number has been entered, the shift register contains the CRC value. Shifting in the eight bits of CRC should return the shift register to all zeros.

64-BIT LASERED ROM Figure 5

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

ROM FUNCTIONS FLOW CHART Figure 6



## ประวัติผู้วิจัย

ชื่อ นายกิตติพงษ์ ชินสุข

สถานที่ติดต่อ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏมหาสารคาม

80 ถนนนครสวรรค์ ตำบลตลาด อำเภอเมือง จังหวัดมหาสารคาม 44000

ห้องทำงาน: อาคาร 6 ชั้น 2, ห้อง 621(ห้องพักอาจารย์ 1)

โทรศัพท์: 0812657577, 0891255142

อีเมล: kittipongch.rmu@gmail.com

### ประวัติการศึกษา

[1] วท.บ.(สถิติ),มหาวิทยาลัยมหาสารคาม

[2] วท.ม.(วิทยาการคอมพิวเตอร์),สถาบันบัณฑิตพัฒนบริหารศาสตร์

### ประสบการณ์ด้านการฝึกอบรม

[1] Image Processing and Computer Vision Using OpenCV Programming โดย คณะสถิติ  
ประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์(NIDA) ระยะเวลา 1 วัน (1 กรกฎาคม 2557)

[2] ระบบสารสนเทศภูมิศาสตร์สำหรับผู้เริ่มต้นใช้งาน(GIS for Beginners) โดย สถาบันวิทยาการ  
อวกาศและภูมิสารสนเทศ(GISTDA) ระยะเวลา 5 วัน (21-25 กรกฎาคม 2557)

[3] การวิเคราะห์ข้อมูลเชิงพื้นที่และการสร้างแบบจำลองในระบบสารสนเทศภูมิศาสตร์ (Spatial  
Analysis and Modeling in GIS) โดย สถาบันวิทยาการอวกาศและภูมิสารสนเทศ(GISTDA)  
ระยะเวลา 5 วัน (28 กรกฎาคม - 1 สิงหาคม 2557)

### ประสบการณ์ด้านวิจัย

-

