เครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน

Water Level Meter With Pressure Sensor

นายอนุวัฒน์ แก่นจรรยา รหัส 47361738
นายเสาวภาคย์ พุฒิหน่อย รหัส 47364245

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2550

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	เครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน
ผู้ดำเนินโครงงาน	นายอนุวัฒน์ แก่นจรรยา รหัส 47361738
	นายเสาวภาคย์ พุฒิหน่อย รหัส 47364245
อาจารย์ที่ปรึกษา	ผศ.ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง
อาจารย์ที่ปรึกษาร่วม	ดร.อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติโครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ผศ.ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง)

.....กรรมการ
(ดร.อัครพันธ์ วงศ์กังแห)

.....กรรมการ
(ดร.แคทรียา สุวรรณศรี)

หัวข้อโครงงาน	เครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน
ผู้ดำเนินโครงงาน	นายอนุวัฒน์ แก่นจรรยา รหัส 47361738
	นายเสาวภาคย์ พุฒิหน่อย รหัส 47364245
อาจารย์ที่ปรึกษา	ผศ.ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง
อาจารย์ที่ปรึกษาร่วม	ดร.อัครพันธ์ วงศ์กังแห
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2550

บทคัดย่อ

โครงงานนี้เป็นการศึกษาหลักการทำงานของไมโครคอนโทรเลอร์ AVR ซึ่งเป็นไมโครคอนโทรเลอร์ที่พัฒนามาจาก MCS-51 และการพัฒนาโปรแกรมด้วยภาษาซีในการวัดระดับน้ำ เพื่อใช้ในการบอกความสูงของน้ำในขณะนั้น รวมทั้งยังประยุกต์ใช้ในการเตือนภัยน้ำท่วมอีกด้วย

ผลที่ได้จากการทำโครงการนี้ คือ รู้หลักการทำงานของไมโครคอนโทรเลอร์ AVR ซึ่งเป็นไมโครคอนโทรเลอร์ที่มีพอร์ตรับค่าสัญญาณอนาล็อกถึง 8 ขา อยู่ในตัวที่ถูกพัฒนาโดยบริษัท ATMEL รวมทั้งยังสามารถประยุกต์ใช้งานในการรับค่าสัญญาณอนาล็อกจากตัวรับสัญญาณอนาล็อกอื่นๆได้ เช่น สัญญาณอนาล็อกจากเซนเซอร์วัดแรงดัน เป็นต้น

Project Title	Water Level Meter With Pressure Sensor
Name	Mr.Anuwat Kaenjanya ID.47361738
	Mr.Saowapark Putnoi ID.47364245
Project Advisor	Asst.Dr.yongyut Chonbodeechalermroong
Co-Project Advisor	Dr.Akaraphunt Vongkunghae
Major	Electrical Engineering
Department	Electrical and computer Engineering
Academic Year	2007

.....

ABSTRACT

This project is to study the working principles of the AVR microcontroller which is developed from MCS-51 and also to develop programs with C language in order to measure water depth for an instant. Furthermore, the project can be applied to flood warning as well.

The results of the project are knowledge of working with the AVR microcontroller developed by ATMEL, which has ports for receiving analog signal up to 8 ports, and moreover, the abilities to apply it to obtaining any analog signals such as those from pressure sensors.

กิตติกรรมประกาศ

โครงงานวิศวกรรมศาสตร์เรื่องเครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน สำเร็จได้ด้วยดี เนื่องจากได้รับคำแนะนำและความช่วยเหลือรวมทั้งข้อคิดเห็นต่างๆ อันเป็นประโยชน์ในการทำโครงงานนี้จาก ผศ.ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง อาจารย์ที่ปรึกษาโครงงานและ นายปราโมทย์ บุญกอแก้ว เจ้าหน้าที่ควบคุมห้อง control ณ เขื่อนนเรศวร อำเภอพรหมพิราม จังหวัดพิษณุโลก ที่เอื้อเฟื้อสถานที่ในการทดลองโครงงาน

ขอขอบคุณอาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ทุกๆท่านที่ให้ความรู้ตลอดการเรียนที่ผ่านมาและเพื่อนๆ ที่คอยให้ความช่วยเหลือในทุกๆด้าน รวมทั้งคณะวิศวกรรมศาสตร์ที่ให้ความเอื้อเฟื้อเครื่องมือและอุปกรณ์ในการทำโครงงานนี้

คณะผู้จัดทำโครงงาน
นายอนุวัฒน์ แก่นจรรยา
นายเสาวภาคย์ พุฒิหน่อย

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ในอดีตเมื่อเกิดอุทกภัยขึ้นทำให้เกิดความเสียหายแก่ประชาชน และประเทศชาติอย่างมาก ซึ่งเมื่อเกิดเหตุการณ์น้ำท่วมหรืออุทกภัยในสถานที่ต่างๆประชาชนในสถานที่นั้นจะไม่รู้ล่วงหน้าว่าจะเกิดน้ำล้นตลิ่งเมื่อใด จึงทำให้ไม่สามารถเตรียมตัวรับมือได้ทัน ทำให้เกิดการเสียหายต่อชีวิตและทรัพย์สินของประชาชนได้เป็นจำนวนมาก

ในโครงงานนี้จึงมุ่งเน้นไปที่การประดิษฐ์เครื่องมือที่สามารถเตือนล่วงหน้าได้ว่าขณะนี้น้ำในแม่น้ำมีระดับน้ำสูงเพียงใดแล้ว เมื่อสามารถทราบล่วงหน้าก็สามารถที่จะหาวิธีป้องกันเพื่อไม่ให้เกิดความเสียหายได้ หรือลดความรุนแรงอันเนื่องมาจากการเกิดอุทกภัยได้ เช่น อพยพและโยกย้ายสิ่งของเอาไปใช้ในที่สูง เป็นต้น

จะเห็นได้ว่าในปัจจุบันเครื่องเตือนภัยน้ำท่วมมีราคาสูงมากเมื่อเทียบกับงบประมาณของชุมชนนั้นๆ ที่จะหามาติดตั้งเพื่อป้องกันและเตือนภัย ดังนั้นเมื่อประดิษฐ์อุปกรณ์นี้ขึ้นมา ซึ่งมีราคาถูก

ในโครงงานนี้เป็นการประดิษฐ์เครื่องวัดระดับความสูงของน้ำโดยใช้เซนเซอร์วัดแรงดันและใช้ความสัมพันธ์ระหว่างแรงดันกับความสูง เพื่อบอกว่าขณะนี้ระดับน้ำมีความสูงที่เท่าไร ในโครงงานนี้จะประดิษฐ์อุปกรณ์วัดความสูงของน้ำที่มีสายโยง (สายไฟ) จากตัวรับค่าแรงดันของดัน แล้วใช้ความสัมพันธ์ระหว่างแรงดันและความสูงเพื่อที่จะแปลงแรงดันน้ำให้เป็นความสูงของระดับน้ำ จากนั้นรายงานผลมายังตัวรายงานผล ซึ่งการใช้สายไฟเป็นตัวเชื่อมระหว่างตัวรับค่า และตัวแสดงผลนั้นทำให้ได้ค่าที่มั่นคง และไม่ยุ่งยากจนเกินไปในการทำเครื่องวัดระดับน้ำ รวมทั้งชิ้นงานที่ได้มีขนาดเล็ก สะดวกในการนำไปติดตั้งที่ใต้น้ำ แต่ก็จะทำให้มีข้อจำกัดในการใช้งานในพื้นที่ใกล้ๆแหล่งน้ำ ซึ่งในส่วนนี้จะคำนึงการรับค่าจากตัวรับเพื่อมาประมาณผลที่ตัวแสดงผลมากกว่าระยะทาง ซึ่งในส่วนระยะทางจากตัวรับค่าและตัวแสดงผลนั้นจะเป็นการประยุกต์ใช้งานจากโครงงานนี้ ซึ่งถ้าจะทำให้สามารถแสดงผลได้ไกลขึ้นอาจจำเป็นที่จะต้องใช้ความถี่วิทยุเข้ามาช่วย ดังนั้นจึงจำเป็นที่จะต้องมีตัวรับและตัวส่งคลื่นวิทยุ ซึ่งจะเป็นการยุ่งยากมากขึ้นในการจัดทำโครงงานนี้

1.2 วัตถุประสงค์

1. เพื่อเป็นแนวทางในการแก้ไขปัญหาน้ำท่วม
2. เพื่อเป็นแนวทางให้มีการพัฒนาอุปกรณ์นี้ให้มีระยะทางระหว่างตัวรับค่ากับตัวแสดงผลได้ไกลขึ้น (ไร้สาย)
3. เพื่อนำความรู้ที่เรียนมาประยุกต์ใช้กับโครงงานนี้

1.3 ขอบข่ายของงาน

1. ศึกษาการใช้งานของตัวเซนเซอร์วัดแรงดัน (MPX 5100DP)
2. ศึกษาการความสัมพันธ์ระหว่างแรงดันน้ำและความสูงของระดับน้ำ
3. ศึกษาตัวรับค่าที่ได้จากเซนเซอร์
4. ออกแบบและสร้างอุปกรณ์วัดระดับน้ำ

1.4 ตารางกิจกรรมการดำเนินงาน

กิจกรรมการดำเนินงาน	2550			2551				
	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1.ศึกษาความรู้พื้นฐานเกี่ยวกับเครื่องมือวัด	←	—	→					
2.รวบรวมข้อมูลและเอกสาร			←	—	→			
3.วิเคราะห์และออกแบบชิ้นงาน					←	—	→	
4.ทำเครื่องวัดระดับน้ำเสร็จสมบูรณ์						←	—	→
5.จัดทำรูปเล่มรายงาน							←	— →
6.เสนอโครงงาน								← →

1.5 ผลที่คาดว่าจะได้รับ

1. นำอุปกรณ์วัดระดับน้ำที่สร้างขึ้นไปใช้งานได้จริง
2. ทำให้ทราบแนวคิดและหลักการทำงานเบื้องต้นของเครื่องวัดระดับน้ำมากขึ้น

1.6 งบประมาณ

1.	เซนเซอร์วัดแรงดัน	900 บาท
2.	ค่าถ่ายเอกสาร	100 บาท
3.	ค่าสายไฟ	200 บาท
4.	ค่าแผ่นซีดี	100 บาท
5.	เอกสารประกอบโครงงาน	200 บาท
6.	อุปกรณ์อิเล็กทรอนิกส์	500 บาท
	รวม	2000 บาท

บทที่ 2
หลักการและทฤษฎี

2.1 ความสัมพันธ์ระหว่างความดันกับความสูงของของเหลว

ความดัน หมายถึง แรงต่อหน่วยพื้นที่

$$P = \frac{F}{A} \tag{2.1}$$

เมื่อ P คือ ความดัน (หน่วยเป็นนิวตัน/ตารางเมตร)

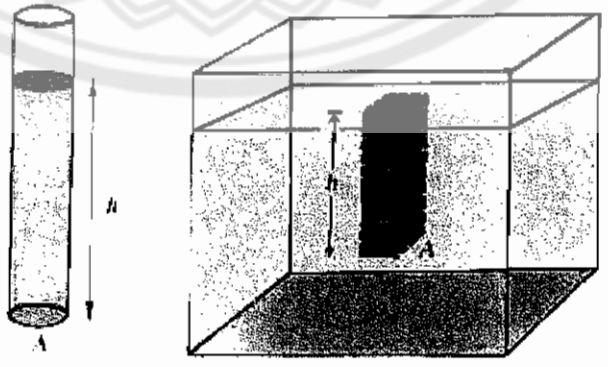
F คือ แรงกระทำที่ตั้งฉากกับพื้นที่ (หน่วยเป็นนิวตัน)

A คือ พื้นที่ที่รับแรง (หน่วยเป็นตารางเมตร)

ในหน่วย SI ความดันมีหน่วยเป็น ปาสคาล (Pa) หรือ นิวตันต่อตารางเมตร(N/m^2) หรือ กิโลกรัมต่อเมตรต่อวินาทีกำลังสอง ($\frac{kg.s^{-2}}{m}$) ส่วนความดันในหน่วย มิลลิเมตรปรอท (mmHg) ซึ่ง 760 mmHg = 101325 Pascal หรือ 1 atm = 101325 Pa = 101.325 kPa แต่อย่างไรก็ตามความดันในหน่วย mmHg ไม่ใช่ หน่วย SI แต่ก็อนุโลมให้ใช้ค่าความดันในหน่วย mmHg หรือ ความดันบรรยากาศเป็นหน่วยความดันมาตรฐาน

2.1.1 ความดันของของเหลว

ของเหลวมีมวลและน้ำหนักเช่นเดียวกับของแข็ง ดังนั้นบริเวณใดก็ตามที่ถูกของเหลวทับอยู่ จะถูกกดด้วยแรงเท่ากับน้ำหนักของของเหลวนั้น และมีค่าของความดันเท่ากันทุกทิศทาง ณ ระดับลึกเดียวกัน ลองพิจารณารูปภาพและขั้นตอนคิดดังต่อไปนี้



รูปที่ 2.1 ความสัมพันธ์ระหว่างความดันและความสูงของของเหลว

พิจารณา ลำของเหลว มีพื้นที่หน้าตัด A ตารางเมตร สูง h เมตร

ปริมาตร V=Ah ลบ.ม.

มวล $m = \rho V = \rho Ah$ กิโลกรัม

เพราะฉะนั้น น้ำหนัก $mg = \rho gAh$ นิวตัน

พื้นที่ที่ถูกน้ำหนักของของเหลวกดทับ = A ตารางเมตร

เพราะฉะนั้น ความดันที่ตกลงบนพื้นที่ A ตารางเมตร $P = \frac{\rho gAh}{A} = \rho gh$ นิวตันต่อตารางเมตร

P = ความดัน (นิวตัน/ตารางเมตร)หรือ (ปาสคาล Pa)

ρ = ความหนาแน่นของเหลว (กิโลกรัม/ลูกบาศก์เมตร)

g = ความเร่งจากแรงดึงดูดโลก (เมตร/วินาที^{2})

h = ความลึกของของเหลว (เมตร)

จากสมการข้างต้น ทำให้ทราบว่าความดันของของเหลว ขึ้นอยู่กับปัจจัย 3 ประการ คือ ความลึกหรือความสูง ความหนาแน่นของของเหลว และแรงโน้มถ่วงของโลก

ตารางที่ 2.1 ความหนาแน่นของของแข็ง ของเหลว และของก๊าซ

substant	ความหนาแน่น (kg/m^3)	Substant	ความหนาแน่น (kg/m^3)
Ice	917	Water	1000
Aluminum	2700	Glycerin	1260
Iron	7860	Ethyl alcohol	806
Copper	8920	Benzene	879
Silver	10500	Mercury	13600
Lead	11300	Air	1.29
Gold	19300	Oxygen	1.43
platinum	21400	Hydrogen	0.0899

2.1.2 ความสำคัญของความดันของของเหลว

1. ณ ตำแหน่งใดๆในของของเหลว แรงดันของของเหลวมีทุกทิศทางรอบตำแหน่งนั้นๆ

2. ของเหลวที่อยู่ติดกับภาชนะจะส่งแรงดันออกไปทุกทิศทางตั้งฉากกับผิวภาชนะที่ของเหลวนั้นสัมผัสอยู่

3. ภายใต้สภาพแรงดึงดูดของโลก ความดันของของเหลว ณ ตำแหน่งใดๆขึ้นกับความลึกของตำแหน่งนั้นวัดจากผิวของเหลว และความหนาแน่นของของเหลว ตามสมการ $P = \rho gh$

4. ความดันของของเหลวภายใต้แรงดึงดูดของโลก จะขึ้นอยู่ระดับความลึกวัดจากผิวของเหลวโดยไม่ขึ้นกับรูปร่างของภาชนะ

2.2 AVR Microcontroller

AVR Microcontroller เป็นไมโครคอนโทรลเลอร์ที่ผลิตโดยบริษัท Atmel ซึ่งได้พัฒนามาจาก MCS-51 โดยที่ AVR ไมโครคอนโทรลเลอร์จะมีระบบ RISC (Reduce Instruction Set Computer) core running หรือมีสถาปัตยกรรมแบบ RISC ที่ใช้คำสั่ง 1 คำสั่งใช้สัญญาณนาฬิกาเพียง 1 ลูก ทำให้เร็วกว่าการใช้ MCS-51 และทำให้มีประสิทธิภาพมากกว่าอีกด้วย

AVRเป็นไมโครคอนโทรลเลอร์ (MCU) ที่ได้รวบรวมอุปกรณ์สนับสนุนการทำงานของ CPU ไว้มากมาย อาทิเช่น Analog to Digital , SPI (Serial Peripheral Interface) , UART (Universal Asynchronous Receiver Transmitters) , Timer , Counter , PWM (Pulse Width Modulator) ซึ่งอุปกรณ์สนับสนุนการทำงานเหล่านี้ทำให้ MCUสามารถทำงานได้กว้างและใช้อุปกรณ์ต่อร่วมจากภายนอกน้อยมาก และสามารถประมวลคำสั่งได้ภายใน 1 clock ในบทนี้จะนำเสนอข้อมูลบางส่วนที่เป็นการทำงานภายในของ AVR – MCU (AVR Microcontroller Unit) แนะนำคุณสมบัติและขาต่อใช้งานของไมโครคอนโทรลเลอร์ สถาปัตยกรรมภายในและรีจีสเตอร์ใช้งานทั่วไป ตำแหน่ง I/O รีจีสเตอร์สถานะและการใช้งาน EEPROM การรีเซ็ตและการอินเตอร์รัพท์ การสื่อสารอนุกรม การเปรียบเทียบสัญญาณอนาลอกและการแปลงสัญญาณอนาล็อกเป็นดิจิตอล การทำงานของพอร์ตอินพุต/เอาท์พุตการทำงานของ Timer / Counter & Watch dog และการใช้กลุ่มคำสั่งต่าง ๆ

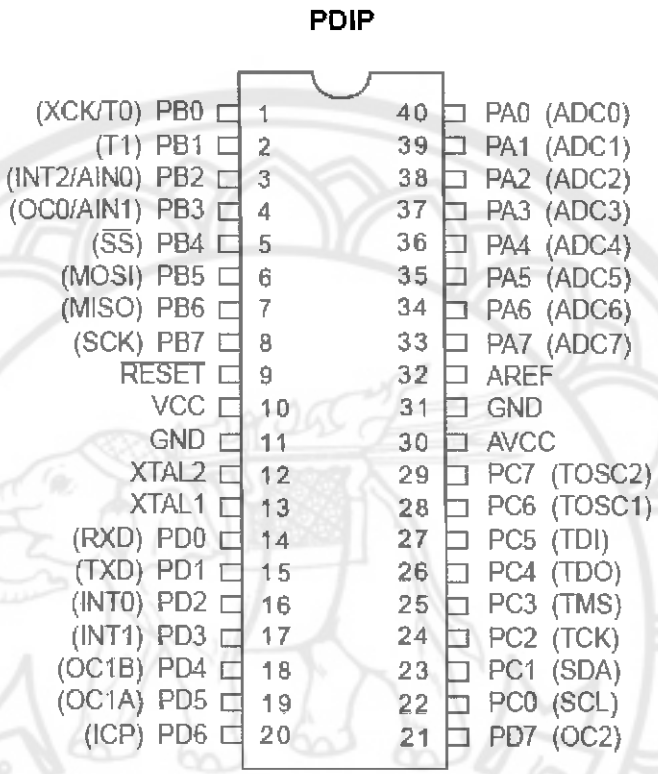
2.2.1 คุณสมบัติและขาต่อใช้งานของไมโครคอนโทรลเลอร์

คุณสมบัติ

1. สถาปัตยกรรมภายในถูกออกแบบให้ใช้สถาปัตยกรรมแบบ RISE (Reduce Instruction Set Computer) ทำให้การประมวลผลมีความเร็ว 1 คำสั่ง / 1 Clock
2. มีคำสั่งในการควบคุมการทำงานของไมโครคอนโทรลเลอร์จำนวน 118 คำสั่ง
3. หน่วยความจำบันทึก PROGRAM MEMORY ขนาด 32 Kbyte
4. หน่วยความจำแบบ EEPROMสำหรับบันทึก DATA MEMORY ขนาด 1024 Byte
5. หน่วยความจำแบบ RAM ขนาด 2K Byte
6. ระบบการเปลี่ยนสัญญาณ ANALOG TO DIGITAL ขนาด 10 บิต จำนวน 8 CHANNEL
7. กลุ่มรีจีสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว
8. พอร์ตอินพุตและเอาท์พุตขนาด 8 บิต จำนวน 4 พอร์ต
9. ระบบการสื่อสารข้อมูลดิจิตอลแบบอะซิงโครนัส(UART) 1 CHANEL
10. ระบบการสื่อสารข้อมูลดิจิตอลแบบซิงโครนัส(SPI) 1 CHANEL
11. ความถี่สัญญาณนาฬิกา 0 - 16 MHz (ATMEGA 32)
12. ระบบการรีเซ็ตแบบอัตโนมัติเมื่อเริ่มจ่ายกระแสไฟฟ้าเข้าไมโครคอนโทรลเลอร์ (Power on reset)
13. ระบบการกำเนิดความถี่สัญญาณแบบ PWM จำนวน 4 CHANNEL (ATMEGA 32)
14. ระบบการตรวจจับระดับสัญญาณอนาล็อก(Analog Comparator)
15. 6 SLEEP MOD:IDEL ,POWER SAVE , POWER DOWN ,ADC Noise , Reduction , Standby, and Extended standby
16. ระบบการป้องกันการ COPY ข้อมูลภายในหน่วยความจำ (LOCK FOR SOLFWARE SECURITY)
17. ระบบตรวจจับการทำงานผิดพลาดของ CPU (WATCHDOG TIMER WITH ON-CHIP OSCILATOR)
18. ระบบการอินเตอร์รัพท์จากภายนอก (EXTERNAL INTERRUPT)
19. TIMER/COUNTER ขนาด 16 บิต 1 CHANNEL
20. TIMER/COUNTER ขนาด 8 บิต 2 CHANNEL
21. Vcc: 4.5 - 5.5 for ATMEGA 32

2.2.2 โครงสร้างภายนอกและตำแหน่งขา

ภายในประกอบด้วยรีจีสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัวซึ่งแต่ละตัวจะต่อเข้ากับALU โดยตรง ทำให้การประมวลผล ต่อ 1 คำสั่งมีความเร็วกว่า CPU ที่มีสถาปัตยกรรมแบบ RISC



รูปที่ 2.2 โครงสร้างภายนอกและตำแหน่งขา AVR Microcontroller

2.2.3 รายละเอียดของขาสัญญาณและการใช้งาน

Vcc คือ ขาจ่ายไฟให้กับ CPU และ GND คือ กราวด์

Port A (PA7..PA0)

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตสามารถ PULL UP ภายในแยกจากกันซึ่งสามารถรับกระแส SINK 20mA โดยพอร์ต Aยังใช้เป็นขาอินพุตเพื่อรับสัญญาณอนาล็อกในส่วนของการแปลงสัญญาณ ANALOG TO DIGITAL

Port B (PB7..PB0)

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตสามารถ PULL UP ภายในอิสระแยกจากกันซึ่งแต่ละขาสามารถรับกระแส SINK 20 mA และยังถูกนำไปใช้งานอื่นๆอีก

Port C (PC7..PC0)

เป็นพอร์ต 2 ทิศทางขนาด 8 บิต โดยสามารถกำหนดให้แต่ละขาของพอร์ตสามารถ PULL UP ภายในอิสระแยกจากกันซึ่งแต่ละขาสามารถรับกระแส SINK 20 mAและยังถูกนำไปใช้งานอื่นๆอีก

Port D (PD7..PD0)

เป็นพอร์ต 2 ทิศทางขนาด 8 บิตโดยสามารถกำหนดให้แต่ละขาของพอร์ตสามารถ PULL UP ภายในอิสระแยกจากกันซึ่งแต่ละขาสามารถรับกระแส SINK 20 mAและยังถูกนำไปใช้งานอื่นๆอีก

Reset คือ ขารีเซ็ต

XTAL 1 เป็นขาอินพุตของคริสตอลออสซิเลเตอร์

XTAL 2 เป็นขาเอาท์พุตของคริสตอลออสซิเลเตอร์

AVcc ใช้จ่ายไฟให้กับวงจร Analog to Digital

AREF เป็นขาแรงดันอ้างอิงที่ใช้งานในส่วนของวงจร Analog to Digital

AGND เป็นขากราวต์ของวงจร Analog to Digital

2.2.4 ฟังก์ชัน ADC

การแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลนั้นมีความจำเป็นมากเพราะว่าในตัวไมโครคอนโทรลเลอร์ไม่สามารถประมวลผลแบบอนาล็อกได้มันจะประมวลผลแบบดิจิตอลเท่านั้นดังนั้นจึงจำเป็นต้องมีการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล

ปกติใน CPU ของ AVR - ATMEGA32 นั้นจะมีฟังก์ชัน ADC อยู่ภายในตัวไอซีดังนั้นไม่จำเป็นต้องใช้ ไอซี ADC ต่อภายนอก สำหรับฟังก์ชัน ADC นี้สามารถรับสัญญาณอนาล็อกได้สูงสุด 8 Channel โดยรับสัญญาณเข้ามาทาง พอร์ต A เราสามารถเลือกใช้ฟังก์ชันนี้ทำการแปลงสัญญาณอนาล็อกที่ละ Channel อย่างต่อเนื่อง หรือจะให้ทำการแปลงสัญญาณเฉพาะchannelที่เราต้องการ ก็ได้เช่นกัน โดยสัญญาณดิจิตอลที่แปลงได้จะมีความละเอียด 10 บิต โดยการรับสัญญาณแต่ละขาของพอร์ท A โดยจะมีวงจร SAMPLE AND HOLD เพื่อ ช่วยให้สัญญาณอนาลอกที่รับเข้ามาเพื่อแปลงเป็นสัญญาณดิจิตอลที่มีระดับสัญญาณคงที่โดยปกติการใช้งานฟังก์ชันนี้เราจำเป็นต้องจัดแรงดัน AVCC AREF และ AGND ให้กับฟังก์ชันด้วย

คุณสมบัติ

1. 10 bit resolution
2. 0.5 LSB integral non-linearity
3. ±2 LSB Absolute Accuracy
4. 8 Multiplexed Single Ended Input Channels

5. 7 Differential Input Channels

6. 2 Differential Input Channels With Gain of 10× and 200×

7. Optional left adjustment for ADC result readout

8. 0-Vcc ADC Input Voltage Range

9. selectable 2.56V ADC Reference Voltage

10. Free Running or Single conversion Complete

11. Sleep mode noise Canceler

การทำงาน

ในส่วนของการแปลงสัญญาณ อนาล็อก เป็นดิจิตอล สามารถทำได้ 2 mode คือ

1. Single Conversion Mode

2. Free Running Mode

การทำงาน Single Conversion Mode ผู้ใช้ต้องเป็นผู้กำหนดการใช้งานขึ้นเอง แต่ในส่วนของ Free Running Mode วงจร Analog to digital จะเป็นตัวจัดการอ่านข้อมูลและเก็บใน ADC Data Register ซึ่งบิต ADFR ใน Register ADCSR จะเป็น บิตที่ใช้เลือก Mode การใช้งานของวงจร Analog to digital สำหรับการกำหนดให้วงจร Analog to digital ทำงานนั้น สามารถทำได้โดยการเซ็ตบิต ADEN ในรีจีสเตอร์ ADCHRA ให้เป็น 1 โดยบิตนี้จะเป็น 1 ไปตลอดจนกระทั่ง Conversion ของสัญญาณจะเรียบร้อยแล้วจึงทำให้บิตนี้เป็น 0 โดยอัตโนมัติ แต่ถ้าเป็นการเปลี่ยน Channel ของการแปลงสัญญาณขณะที่ Channel เดิมยัง Conversion อยู่ วงจร Analog to digital จะ Conversion สัญญาณ Channel เดิมให้เสร็จก่อนแล้วจึง Conversion สัญญาณ Channel ถัดไป โดยข้อมูลที่ได้จากการแปลงสัญญาณอนาล็อกเป็นดิจิตอลจะเก็บไว้ในรีจีสเตอร์ ADCH และ ADCL

การใช้งานโมดูลเปรียบเทียบแรงดัน [1]

ไมโครคอนโทรเลอร์ AVR เบอร์ ATmega16 มีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล หรือ ADC (Analog to digital Converter) ความละเอียดขนาด 10 บิต (10-bit Resolution) ที่แรงดัน+5 V หมายถึงการเปลี่ยนแปลงสัญญาณดิจิตอลแล้วจะได้ค่าตัวเลขอยู่ระหว่าง 0-1024 โดยมีรูปแบบการแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบซัคเซสซีฟ แอพพร็อกซิเมชั่น (Successive Approximation ADC) คือการแปลงแบบประมาณค่า โดยการสุ่มค่าดิจิตอลแล้วแปลงเป็นแรงดันอนาล็อกภายในโมดูล เพื่อใช้เปรียบเทียบกับแรงดันอนาล็อกทางด้านอินพุต เมื่อเปรียบเทียบได้ค่าแรงดันเท่ากัน โมดูล ADC จะให้ผลลัพธ์ออกมาเป็นค่าดิจิตอล ซึ่งการใช้วิธีนี้เป็นที่นิยมเพราะมีความเที่ยงตรงสูงและทำงานได้อย่างรวดเร็ว

ในโมดูล ADC จะมีจำนวน 8 ช่องอินพุตสัญญาณใช้หลักการมัลติเพล็กซ์ (Multiplexer) เพื่อเลือกการทำงานในแต่ละช่อง กำหนดไว้ที่ขาพอร์ต A โดยมีแรงดันอินพุต

ระหว่าง 0 V(GND) ถึง VCC (แรงดันอินพุตที่ขา VCC ของไมโครคอนโทรเลอร์ AVR) ผ่านวงจรสุ่มและเก็บค่า (Sample and Hold)

รีจิสเตอร์ที่เกี่ยวข้องกับโมดูลเปรียบเทียบแรงดันอนาล็อก

1. รีจิสเตอร์ ADMUX (ACD Multiplexer Selection Register)

รีจิสเตอร์กำหนดแรงดันอ้างอิง (Voltage Reference) รูปแบบการเก็บข้อมูลและการกำหนด อินพุตอนาล็อกอ้างอิงด้านบวกและลบ

2. รีจิสเตอร์ ADCSRA (ADC Control and Status Register)

รีจิสเตอร์กำหนดสถานการณ์ทำงานของโมดูล ADC

3. รีจิสเตอร์ ADCL และ ADCH (The ADC Data Register)

รีจิสเตอร์เก็บข้อมูลที่ได้จากการแปลงสัญญาณอนาล็อกเป็นดิจิตอล

4. รีจีสเตอร์ SFIOR (Special Function IO Register)

รีจิสเตอร์กำหนดการกระตุ้นจากแหล่งสัญญาณภายนอกให้กับโมดูล ADC

รายละเอียดของรีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานโมดูลมีดังนี้

1. รีจิสเตอร์ ADMUX (ACD Multiplexer Selection Register)

ตารางที่ 2.2 ข้อมูลของรีจิสเตอร์ ADMUX

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7:6 บิต REFS1:0 (Reference Selection Bite)

  บิตกำหนดแรงดันอ้างอิงโมดูล ADC กำหนดรายระเอียดตามตารางที่ 2.2

ตารางที่ 2.3 แรงดันอ้างอิงสำหรับโมดูล ADC

REFS1	REFS0	แหล่งแรงดันอ้างอิง
0	0	AREF , ปิดการใช้งานแรงดันอ้างอิงภายใน (Vref)
0	1	ใช้แรงดัน AVCC กับตัวที่เก็บประจุภายนอกที่ขา AREF
1	0	สงวนไว้
1	1	ใช้แรงดันอ้างอิงภายในที่ 2.56 V กับตัวเก็บประจุภายนอกที่ขา AREF

- บิตที่ 5: บิต ADLAR (ADC Left Adjust Result)

บิตกำหนดรูปแบบการเก็บข้อมูลในรีจิสเตอร์ ADC Data ขนาด 16 บิต มี 2 รูปแบบคือเก็บชิดบิตสูงสุด (MST bit) ดูเพิ่มในรีจิสเตอร์ ADC Data Register (ADCH,ADCL)

ตารางที่ 2.4 ตัวคูณและค่าบิต

MUX4..0	อินพุตเดี่ยว	ความแตกต่างของอินพุตด้านบวก	ความแตกต่างของอินพุตด้านลบ	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x

ตารางที่ 2.4 (ต่อ) ตัวคูณและค่าบิต

MUX4..0	อินพุตเดี่ยว	ความแตกต่างของอินพุตด้านบวก	ความแตกต่างของอินพุตด้านลบ	Gain
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x
11101		ADC5	ADC2	1x
11110	1.22 V (V_{BG})	N/A		
11111	0 V (GND)			

โดยที่ผลการเปลี่ยนแปลงสัญญาณอนาล็อกเป็นดิจิตอล คำนวณผลลัพธ์ที่ได้จากสูตรต่อไปนี้

- เมื่อทำงานในโหมดสัญญาณเดี่ยว
-

$$ADC = \frac{V_{IN} \times 1024}{V_{REF}} \qquad (2.2)$$

โดยที่ V_{IN} : แรงดันด้านขาอินพุต

V_{REF} : แรงดันอ้างอิง

รายละเอียดการกำหนดแรงดันอินพุตและแรงดันอ้างอิง ตามตารางที่ 2.3 ถ้าทำงานในช่องที่มีความต่างของสัญญาณทางด้านบวกและลบ

$$ADC = \frac{(V_{POS} - V_{NEG}) \times GAIN \times 512}{V_{REF}} \quad (2.3)$$

โดย V_{POS} : แรงดันอินพุตทางด้านบวก

V_{NEG} : แรงดันอินพุตทางด้านลบ

V_{REF} : แรงดันอ้างอิง

$GAIN$: ตัวคูณอัตราขยาย

2. รีจิสเตอร์ ADCSRA (ADC Control and Status Register)

ตารางที่ 2.5 ข้อมูลของริสเตอร์ ADCSRA

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

-

- บิตที่ 7 : บิต ADEN (ADC Enable)

เซตบิต ADEN เป็น "1"เพื่อเปิดการใช้งานโมดูล ADC

- บิตที่ 6 : บิต ADSC (ADC Start Conversion)

เซต ADSC เป็น "1" เพื่อกำหนดให้โมดูลเริ่มต้นการแปลสัญญาณอนาล็อกเป็นดิจิตอล เมื่อแปลงเสร็จสมบูรณ์ บิต ADSC จะถูกเซตเป็น "0" จะไม่มีผลใดๆกับโมดูล

- บิตที่ 5 : บิต ADATE (ADC Auto Trigger Enable)

เซตบิต ADATE เป็น "1" เพื่อเปิดการกระตุ้น (Trigger) สัญญาณอัตโนมัติ โดยแหล่งสัญญาณในการกระตุ้นกำหนดให้ในบิต ADTS ที่รีจิสเตอร์ SFIOR

- บิตที่ 4 : ADIF (ADC Interrupt Flag)

บิต ADIF จะถูกเซตเป็น "1" เมื่อโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอลสมบูรณ์ และข้อมูลได้ถูกเขียนไปที่รีจิสเตอร์ ADCD (ADCH,ADCL) แล้ว หากมีการเปิดใช้งานอินเตอร์รัปต์เนื่องจากโมดูล ADC และเปิดใช้งานอินเตอร์รัปต์โดยจะส่งผลให้เกิดอินเตอร์รัปต์ขึ้น

- บิตที่ 3 : บิต ADIE (ADC Interrupt Enable)

เซตบิต ADIE เป็น "1" เพื่อเปิดการใช้งานอินเตอร์รัปต์เนื่องจากโมดูล ADC (ต้องเปิดอินเตอร์รัปต์โดยรวมด้วย)

- บิตที่ 2 : บิต ADPS2:0 (ADC Prescaler Select Bits)

บิตกำหนดปรีสเกลเลอร์สำหรับใช้ในการหารสัญญาณนาฬิกาสำหรับโมดูล ADC รายละเอียดแสดงดังตาราง 2.6

ตารางที่ 2.6 ปรีสเกลเลอร์สำหรับโมดูล ADC

ADSP2	ADSP1	ADSP0	ปรีสเกลเลอร์หารสัญญาณนาฬิกาความถี่ (XTAL) และสัญญาณนาฬิกาโมดูล ADC
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

3. รีจิสเตอร์ ADCL และ ADCH (The ADC Data Register)

ตารางที่ 2.7 ข้อมูลของรีจิสเตอร์ ADCL และ ADCH เมื่อ ADLAR=0

บิตที่	15	14	13	12	11	10	9	8
ชื่อบิต	-	-	-	-	-	-	ADC9	ADC8
ชื่อบิต	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
บิตที่	7	6	5	4	3	2	1	0
Read/Write	R R	R R	R R	R R	R R	R R	R R	R R
ค่าเริ่มต้น	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

- บิตที่9:0 บิต ADC9:0 (ADC Data Register)

ผลลัพธ์ของข้อมูลที่ได้จากโมดูล ADC จะเก็บทางขวาสุดของข้อมูล โดยการเซตบิต ADLAR (ADC Lift Adjust Result) เป็น “0” ในรีจิสเตอร์ ADMUX

ตารางที่ 2.8 ข้อมูลรีจิสเตอร์ ADCL และ ADCH เมื่อ ADLAR = 1

บิตที่	15	14	13	12	11	10	9	8
ชื่อบิต	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2
ชื่อบิต	ADC1	ADC0	-	-	-	-	-	-
บิตที่	7	6	5	4	3	2	1	0
Read/Write	R R	R R	R R	R R	R R	R R	R R	R R
ค่าเริ่มต้น	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0

- บิตที่ 15:6 บิต ADC9:0(ADC Data Register)

ผลลัพธ์ของข้อมูลที่ได้จากโมดูล ADC จะเก็บทางซ้ายสุดของข้อมูล โดยการเซตบิต ADLAR (ADC Lift Adjust Result) เป็น “1” ในรีจิสเตอร์ ADMUX

4 รีจีสเตอร์ SFIOR (Special Function IO Register)

ตารางที่ 2.9 ข้อมูลรีจิสเตอร์ SFIOR

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

- บิตที่ 7:5 :บิต ADTS2:0(ADC Auto Trigger Source)

บิตกำหนดการกระตุ้นโมดูล ADC จากแหล่งสัญญาณภายนอกโดยขึ้นอยู่กับบิต ADATE ในรีจิสเตอร์ ADCSRA หาก ADATE ถูกเซตเป็น “1” การกำหนดบิตกระตุ้นการทำงานของโมดูล ADC จะขึ้นอยู่กับสัญญาณจากภายนอกตามบิต ADTS2:0 หากบิต ACSRA ถูกเซตเป็น “0” บิต ADTS2:0 จะไม่มีผลใดๆกับโมดูล ADC หากกำหนดแหล่งกระตุ้นสัญญาณจากภายนอกกำหนดได้ดังตารางที่ 2.10

ตารางที่ 2.10 แหล่งกระตุ้นสัญญาณอัตโนมัติ

ADTS2	ADTS1	ADTS0	แหล่งกระตุ้นสัญญาณอัตโนมัติ
0	0	0	โหมดทำงานอิสระ
0	0	1	เปรียบเทียบแรงดันอนาล็อก
0	1	0	อินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ช่องที่ 0
0	1	1	โมดูลเปรียบเทียบสัญญาณของไทเมอร์/เคาน์เตอร์ 0
1	0	0	ไทเมอร์/เคาน์เตอร์0 โอเวอร์โฟลว์
1	0	1	โมดูลเปรียบเทียบสัญญาณของไทเมอร์/เคาน์เตอร์1
1	1	0	ไทเมอร์/เคาน์เตอร์1 โอเวอร์โฟลว์
1	1	1	โมดูลตรวจจับสัญญาณอินพุตของไทเมอร์/เคาน์เตอร์1

- บิตที่ 4 : บิต Res(Reserved Bit)

บิตนี้สงวนไว้ไม่ได้ใช้งาน อ่านค่าได้เป็น “0”

2.2.5 ฟังก์ชัน PWM

การมอดูเลตความกว้างของพัลส์(PWM) เป็นเทคนิคสำคัญที่ใช้ในการปรับปรุงสมรรถนะของอินเวอร์เตอร์ ดังนั้นการศึกษาเกี่ยวกับ PWM จึงมีความจำเป็นอย่างยิ่งสำหรับอินเวอร์เตอร์ เพื่อที่อินเวอร์เตอร์จะได้มีสมรรถนะและประสิทธิภาพในการทำงานที่ดีขึ้น เนื่องจากว่า PWM เป็นฟังก์ชั่นการทำงานหนึ่งในโหมด PWM ของ Timer/Counter ที่อยู่ภายใน AVR – ATMEGA32 ดังนั้นในหัวข้อต่อไปจะแนะนำเกี่ยวกับการทำงานของ Timer/Counter ของ AVR – ATMEGA32

Timer /Counter

ภายใน AVR - ATMEGA32 จัดให้มี Timer/Counter 3 ชุด โดยจัดเป็น Timer/Counter ขนาด 8 บิต92 ชุด และ Timer/Counter ขนาด 16 บิต 1 ชุด ดังนี้คือ Timer/Counter2 และ Timer/Counter0 และ Timer/Counter1 ซึ่ง Timer/Counter2 สามารถรับสัญญาณ Clock จากภายนอก ซึ่งเป็น Option ที่จะนำ Timer/Counter2 มาทำเป็น RTC โดยใช้ XTAL ที่มีค่าความถี่เท่ากับ 32.768KHz มาเป็นฐานเวลา และTimer/Counter0 และ Timer/Counter1 ใช้วงจร Prescaling ขนาด 10 บิตร่วมกัน ส่วน Timer/Counter2 ใช้วงจร Prescaling แยกออกต่างหาก

แนะนำการใช้งาน Timer/Counter แต่ละประเภท

1. Timer/Counter0

โครงสร้างของ Timer/Counter0 ขนาด 8 บิต แสดงในรูปที่ 50 ซึ่งสามารถเลือกสัญญาณ Clock ได้จาก CK (Clock ของระบบ) หรือสัญญาณ Clock ของระบบที่ถูกหาร (Prescaling) หรือ สัญญาณจากภายนอก โดยการใช้งานจะอธิบายในรีจีสเตอร์ TCCRO และ TIFR ส่วนสัญญาณควบคุมสามารถทราบรายละเอียดได้จาก รีจีสเตอร์ TCCRO ซึ่งการควบคุมการอินเตอร์รับจะควบคุมไดจาก รีจีสเตอร์ TIMSK เมื่อ Timer/Counter0 ได้รับสัญญาณจากภายนอก ซึ่งสัญญาณดังกล่าวจะซิงโครไน (Synchronized) กับสัญญาณนาฬิกาภายในCPU โดย TIMER/COUNTER 0 จะเป็นวงจรนับขึ้นที่สามารถเขียนและอ่านข้อมูลได้ตลอดเวลาโดยเมื่อทำการเขียนข้อมูลลงใน TIMER/COUNTER 0 ในขณะที่มีสัญญาณ Clock จะทำให้ TIMER/COUNTER 0 นับค่าต่อเนื่องจากค่าที่ถูกเขียนลงไป

2. Timer/Counter 1

จะมีขนาด 16 บิต โดยสามารถเลือกสัญญาณนาฬิกาได้จาก CK หรือสัญญาณที่ได้รับการหารจาก CK (Prescelling) ซึ่งการหยุด Timer/Counter 1 จะอธิบายไว้ในรีจีสเตอร์ TCCR1A (Timer/Counter 1 Control Register) และ TCCR1B โดยแฟร์กที่แสดงสถานะต่างๆ (Overflow,Compare math,Capture even) ส่วนสัญญาณควบคุมจะอธิบายไว้ในรีจีสเตอร์ TCCR1A และTCCR1B การควบคุมสัญญาณอินเตอร์รัพต์จะควบคุมได้จากรีจีสเตอร์TIMSK (TIMER/COUNTER INTERRUPT MASK REGISTER)

เมื่อ TIMER1/COUNTER1 จะประกอบด้วยส่วนของการเปรียบเทียบเอาท์พุต (Output Compear Function) 2 ฟังก์ชั่น โดยจะใช้ รีจีสเตอร์ OCR1A (Output Compare Register 1 A) และ OCR1B (Output Compare Register 1B) เป็นส่วนของการเก็บค่าข้อมูลของการเปรียบเทียบ TIMER1/COUNTER1 จะสามารถเลือกใช้ฟังชั่น PWM ได้ทั้ง 8,9 และ 10 บิต

The Timer/Counter Control Register

Bits 7, 6-COM1A1, COM1A0: Compare Output Mode 1 A, bit 1 and 0 บิต COM1A1 และ COM1A0 เป็นบิตที่ใช้ในการกำหนดลักษณะของสัญญาณที่เกิดขึ้นที่ขา OC1A เมื่อ Timer/Counter1 เกิด Compare Match ซึ่งเมื่อใช้ฟังชั่นOutput Compare Match ของTimer/Counter1 จะต้องควบคุมให้ขา OCIA มีสถานะเป็นเอาท์พุต โดยการเลือกลักษณะของสัญญาณแสดงในตาราง

Bit 5, 4-COM1B1, COM1B0: Compare Output Mode 1 B, bit 1 and 0 บิต Com1A1 และ COM1A0 เป็นบิตที่ใช้ในการกำหนดลักษณะของสัญญาณที่เกิดขึ้นที่ขาOC1A เมื่อ

Timer/Counter1 เกิดCompare Match ซึ่งเมื่อใช้ฟังชั่นOutput Compare Match ของ Timer/Counter1 จะต้องควบคุมให้ขา OC1A มีสถานะเป็นเอาท์พุต

Bit 3…2-Res: Reserved bits ในส่วนของ AT mega32 จะสงวนบิตในกลุ่มนี้ไว้

Bit 1…0 - PWM11, PWM10: Pulse Width Modulator Select Bit เป็นบิตที่ใช้ในการกำหนดการทำงานของ PWM

The Timer/Counter1 Control Register B-TCCRI1B

Bit 7-ICN1: Input Capture 1 Noise Canceler (4 CKs) บิตนี้เป็นบิตที่กำหนดให้ Input Capture 1 Noise Canceler ทำงานหรือไม่ทำงาน โดยเมื่อบิตนี้เป็น 1 จะเป็นการกำหนดให้ Input Capture 1 Noise Canceler ทำงาน แต่เมื่อบิตนี้เป็น 0 จะเป็นการกำหนดไม่ให้ Input Capture 1 Noise Canceler ทำงาน

ชุด Noise Canceler จะถูกกำหนดให้ทำงานโดยการ Sampling สัญญาณที่เข้ามาที่ชุด Input Capture 1 โดยสัญญาณ Sampling แรกจะเริ่มที่ขอบแรกของสัญญาณขาขึ้นหรือขาลงขึ้นอยู่กับการกำหนดในบิต ICES1 โดยชุด Noise Canceler จะ Sampling ด้วยความถี่เท่ากับความถี่ของ XTAL ซึ่งจะ Sampling ทั้งหมด 4 ครั้ง โดยลอจิกที่ได้จากการ Sampling จะต้องมีลอจิกเดียวกันกับลอจิกที่กำหนดในบิต ICES1

Bit 6-ICES1: Input Capture 1 Edge Select เป็นบิตที่ใช้กำหนดให้ชุด Input Capture 1 จะต้อง Detect ถ้าบิต ICES1 เซ็ต เป็น 1 จะเป็นการกำหนดให้ชุด Input Capture1 ทำหน้าที่ Detect สัญญาณที่ขอบขาขึ้น แต่ถ้าบิต ICES 1 ถูกเคลียเป็น 0 จะเป็นการกำหนดให้ชุด Input Capture 1 ทำหน้าที่ Detect สัญญาณที่ขอบขาลง

Bit 5, 4-RES: Reserved bits บิตนี้ถูกสงวนไว้

Bit 3: CTC1: Clear Timer1/Counter1 on Compare Match บิตนี้เป็นที่ใช้ในการกำหนดว่าเมื่อเกิด Output Compare แล้วจะให้เกิดการนับต่อไปหรือจะให้มีการรีเซ็ตค่าให้เป็น 00000 แล้วจึงทำการนับต่อไป โดยถ้าเป็นบิตนี้เป็น 1 จะเป็นการกำหนดให้มีการรีเซ็ตค่าให้เป็น 0 เมื่อเกิดการ Output Compare แต่ถ้าบิตนี้เคลียร์เป็น 0 จะเป็นการกำหนดให้มีการนับค่าต่อเมื่อเกิด Output Compare Bit 2, 1, 0-CS12, CS11, CS10: Clock Select1, bit 2, 1 and 0 เป็นบิตที่ใช้ในการเลือกสัญญาณ Clock

The Timer/Counter In Capture Register – 1CR1H AND ICR1L

เป็นรีจีสเตอร์ขนาด 16 ที่ใช้เก็บค่า Timer/Couner1 ที่อยู่ในรีจีสเตอร์ TCNT1 เมื่อ Input Capture สามารถ Detect ได้ เมื่อ Input Capture สามารถ Detect สัญญาณได้ตามที่กำหนดในบิตICES1 จะทำให้ CPU โหลดค่าในรีจีสเตอร์ TCNT1 ลงในรีจีสเตอร์ และในเวลาเดียวกับบิต ICF1จะเซ็ตเป็น 1 โดยการอ่านค่าจากรีจิสเตอร์ ICR1 ของ CPU จะใช้รีจีสเตอร์ TEMP เป็นรีจีสเตอร์พักข้อมูล ซึ่งการใช้รีจีสเตอร์ TEMP ช่วยในการอ่านข้อมูลเพื่อให้ค่าที่อยู่ในรีจีสเตอร์

ICR1H และ ICR1L เสมือนถูกอ่านออกมาพร้อมกัน การอ่านค่าจาก รีจีสเตอร์ ICR1 จะต้องอ่านค่าจากรีจีสเตอร์ ICR1L ก่อน โดยเมื่อ CPU อ่านค่าจาก ICR1L จะทำให้ค่าในรีจีสเตอร์ ICR1H ถูกโหลดลงในรีจีสเตอร์ TEMP เมื่อ CPU อ่านค่าจาก ICR1H จะทำให้ค่าในรีจีสเตอร์ TEMP ถูกส่งให้ CPU

การใช้งาน Timer/Counter1 ในโหมด PWM

การทำงานในโหมด PWM ของ Timer/Counter1 จะสามารถเลือกใช้งานได้ 8, 9 หรือ 10 บิต โดยเอาท์พุตที่ได้จะออกที่ขา PD5(OC1A) และขา PD(OC1B) ในการทำงาน Timer/Counter1 จะนับขึ้นและนับลง ซึ่งจะนับขึ้นจาก 0000 ถึงค่าสูงสุด(ตามที่กำหนดในตารางที่ 13) และจะนับจากค่าสูงสุดลงมาที่ 0000 แล้วจึงนับขึ้นอีกครั้ง

เมื่อค่าใน Timer/Counter1 เท่ากับค่าในรีจีสเตอร์ OCR1A หรือ OCR1B จะทำให้ขา PD5 (OC1A) /PD1 (OC1B) เปลี่ยนแปลงตามที่กำหนดในบิต COM1A/CoM1A0 หรือ CoM1B/COM1B0 เมื่อ OCR1 มีค่าเท่ากับ 0000 หรือค่าสูงสุดจะทำให้เอาท์พุตขา OC1A/OCA1B มีลอจิกเป็น LOW หรือ HIGH ตามที่กำหนดในบิต COM1A1/COM1A0 หรือ COM1B1/COM1B0 และเมื่อ Timer/Counter1 เกิด Overflow และค่าการนับเป็น 0000 จะทำให้บิต TOV1 เซ็ตเป็น 1

3. Timer2&Counter

เป็น Timer / Counter ขนาด 8 บิต ต่อไปจะกล่าวถึงรายละเอียดของรีจีสเตอร์ใช้งานใน Timer/Counter 2

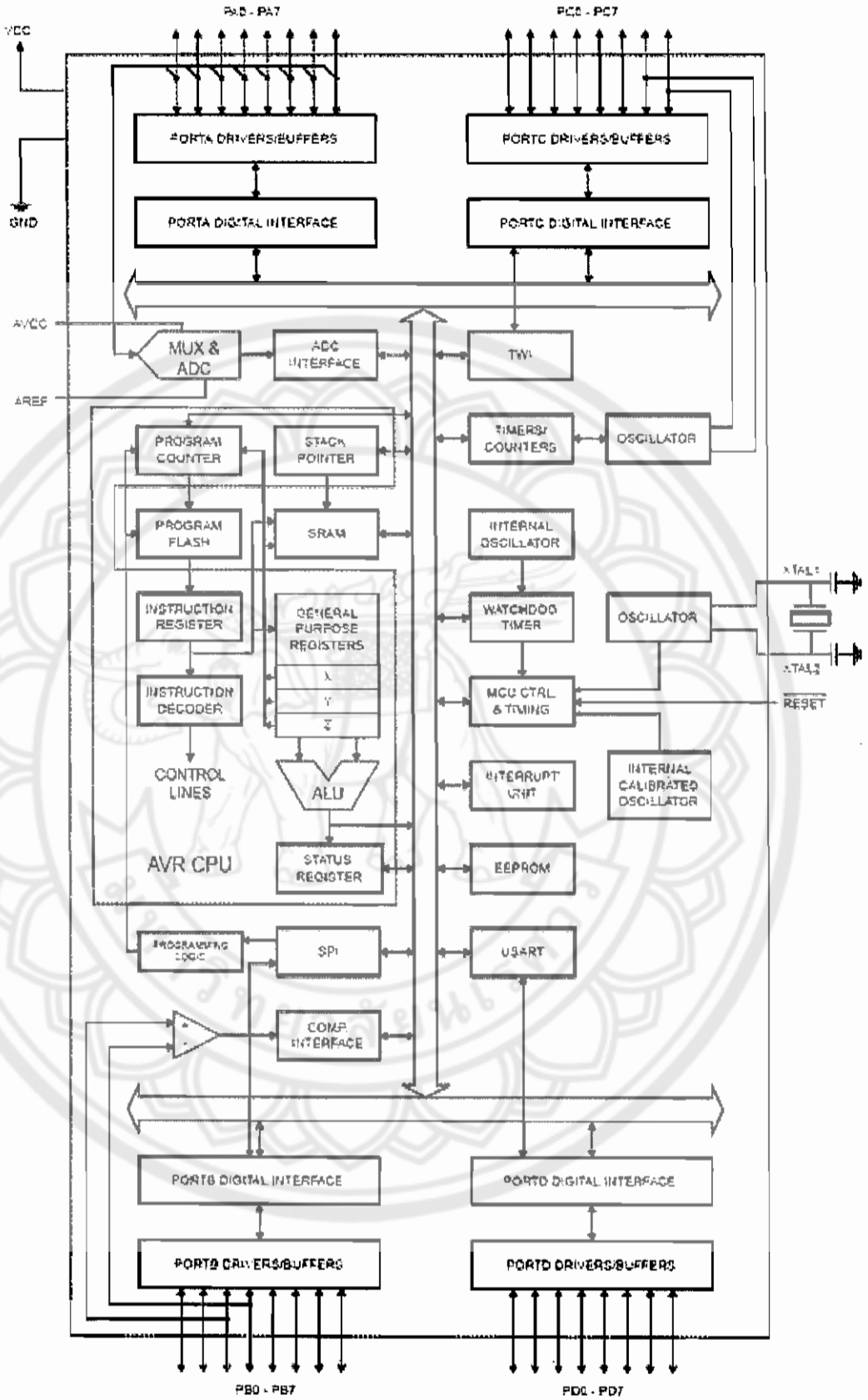
The Timer/Counter 2 Control Register – TCCR2

Bit 7 - Res:Reserved Bit ใน AT90S4434/8535 บิตนี้สงวนไว้

Bit 6 - PWM2: Pulse Width Modulator Enable เป็นบิตที่ใช้ Enable ให้โหมด PWM ใน Timer/Counter2 ให้ทำงาน โดยถ้าบิตนี้เซ็ตเป็น 1 จะเป็นการกำหนดให้โหมด PWM ถูก Enable ให้ทำงานแต่ถ้าบิตนี้ถูกเคลียเป็น 0 จะเป็นการ Disable ไม่ให้โหมด PWM ใน Timer/Counter2 ทำงาน Bit 5, 4 - Com21, Com20: Compare Output Mode, bit 1 and 0 เป็นบิตที่ใช้กำหนดลักษณะสัญญาณที่ขา PD7 (OC2) เมื่อ Timer/Counter2 ทำงานในโหมด Compare โดยเมื่อ Compare Output Match จะทำให้ขา PD7 (OC2) เป็นไปตามที่กำหนดในบิตCom21 และ Com20

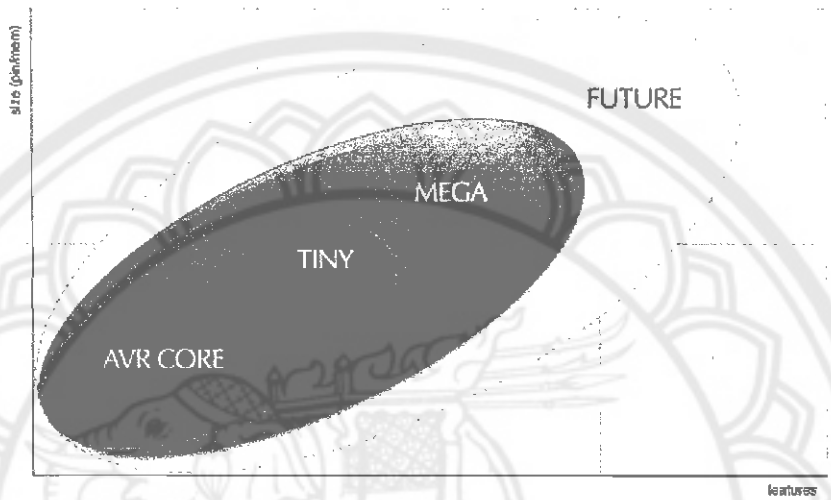
Bit 3-CTC2: Clear Timer/Counter on Compare Match เป็นบิตที่ใช้กำหนดให้ Timer2/Counter2 ทำการ RESET ค่าเป็น 00 หลังจากที่ค่าในรีจีสเตอร์ TCNT2 มีค่าเท่ากับค่าที่ตั้งไว้ในรีจีสเตอร์ OCR หรือ Compare Output Match ถ้าบิตนี้เซ็ตเป็น 1 จะทำให้ Timer/Counter2 รีเซ็ต

Bits 2, 1, 0-CS22, CS21, CS20: Clock Select bit 2, 1 and 0 เป็นบิตใช้ในการกำหนด ค่า Prescaling



รูปที่2.3 โครงสร้างภายในของ AVR

AVR Microcontroller มีให้เลือกใช้หลายเบอร์หลายแบบ โดยจะยกตัวอย่าง ATmega32 ซึ่งเป็น AVR ไมโครคอนโทรลเลอร์ที่มีหน่วยความจำ ROM แบบ Flash ที่สามารถเขียนและลบได้มากกว่า 10000 ครั้ง ถึง 32 Kbyte ทั้งนี้ AVR จะมีหน่วยความจำแตกต่างกันตามเบอร์ของ AVR โดย AVR นั้นจะแบ่งชื่อเรียกของไอซีตามขนาดของความจุ ดังนี้

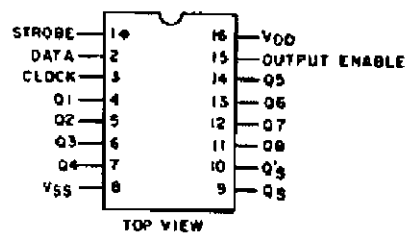


รูปที่ 2.4 ขนาดความจุของ AVR Microcontroller

ส่วนระบบการโปรแกรมนั้นสามารถทำได้โดยตรงได้ที่ตัวไมโครคอนโทรลเลอร์ได้เลย โดยไม่ต้องถอดเสียบ มีโมดูลสร้างสัญญาณ Pulse Width Modulator มีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิตอล มีขาให้ใช้งานตั้งแต่ 8 ขา ไปจนถึง 100 ขา

2.3 ไอซี CD4094B

ไอซีชนิดนี้เป็นไอซีตระกูล CMOS ทำหน้าที่เป็นชิปรีจิสเตอร์ โดยจะรับข้อมูลมาแบบอนุกรม(serial) แล้วแปลงให้เป็นข้อมูลแบบขนาน (parallel) ขนาด 8 บิต นอกจากนี้ไอซีเบอร์นี้ยังมีเอาท์พุตที่เป็นแบบอนุกรมด้วย (serial output) [3]



รูปที่2.5 โครงสร้างภายนอกและตำแหน่งขาไอซีเบอร์ CD4094B

บทที่ 3
การออกแบบและการสร้าง

3.1 การเลือกเซนเซอร์แรงดัน (Pressure sensor)

เซนเซอร์แรงดันมีหลายชนิด โดยแบ่งตามความสูงของน้ำที่วัดได้ และชนิดของน้ำที่จะวัดว่าเป็นน้ำนิ่งหรือน้ำไหล ความต้องการของโครงงานนี้ต้องการเซนเซอร์ที่วัดความสูงของน้ำได้ประมาณ 10 เมตร หรือสามารถรับแรงดันน้ำได้ 100 kPa จากการศึกษาและค้นคว้าเรื่องของตัวเซนเซอร์ก็ได้เลือกเซนเซอร์ MPX5100 DP มาใช้ในโครงงานนี้

เซนเซอร์ MPX5100 DP [4] เป็นทรานสดิวเซอร์ที่ให้เอาท์พุทอยู่ในช่วง 0-5 โวลต์ โดยวัดความสูงของระดับน้ำได้ 10 เมตร สัญญาณที่ได้จากตัวเซนเซอร์เป็นสัญญาณอนาล็อก ซึ่งเซนเซอร์แรงดันจะมีสมการที่ใช้คำนวณเอาท์พุตเป็นโวลต์ ดังนี้

$$V_{out} = V_s \times ((0.009 \times P) + 0.04) \qquad (3.1)$$

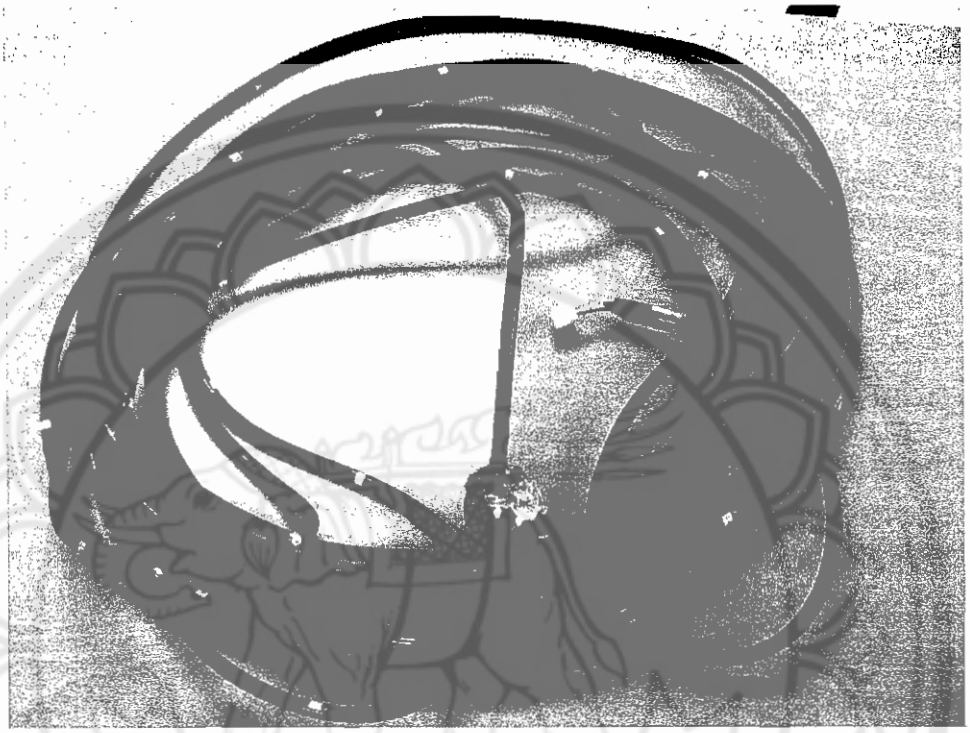
โดยที่ V_{out} คือ เอาท์พุตที่ได้จากเซนเซอร์แรงดัน

V_s คือ ไฟที่จ่ายให้แก่ตัวเซนเซอร์มีค่า 5 โวลต์

P คือ ความดันของน้ำ หน่วยเป็นกิโลปาสคาล (kPa)

รูปที่ 3.1 เซนเซอร์ MPX5100DP

ขาของเซนเซอร์ MPX5100DP มีอยู่ทั้งหมด 6 ขา แต่นำมาใช้จริงแค่ 3 ขา คือ ขา 1,2 และ 3 โดยที่ขา1 เป็นขาเอาท์พุท ขา2 เป็นกราวด์ และขา3 เป็นไฟที่ป้อนให้แก่ตัวเซนเซอร์ซึ่งใช้ 5 โวลต์ หลังจากที่เลือกเซนเซอร์แล้วก็มาถึงขั้นตอนการออกแบบเซนเซอร์เพื่อให้เหมาะกับการหย่อนลงไปในน้ำได้ โดยจะบัดกรีสายไฟกับขาแต่ละขาที่ใช้งานของเซนเซอร์ MPX5100DP



รูปที่ 3.2 เซนเซอร์ MPX5100DP หลังการออกแบบและเชื่อมต่อกับสายไฟ

3.2 ATmege32 ไมโครคอนโทรลเลอร์

ATmega32ไมโครคอนโทรลเลอร์เป็นส่วนที่จะต้องโปรแกรมเพื่อใช้ควบคุมการทำงานทั้งหมด โดยที่จะต้องกำหนดขาของ ATmega32 ก่อนว่าจะใช้ขาไหนทำอะไร

3.2.1 การกำหนดขาของ ATmega32

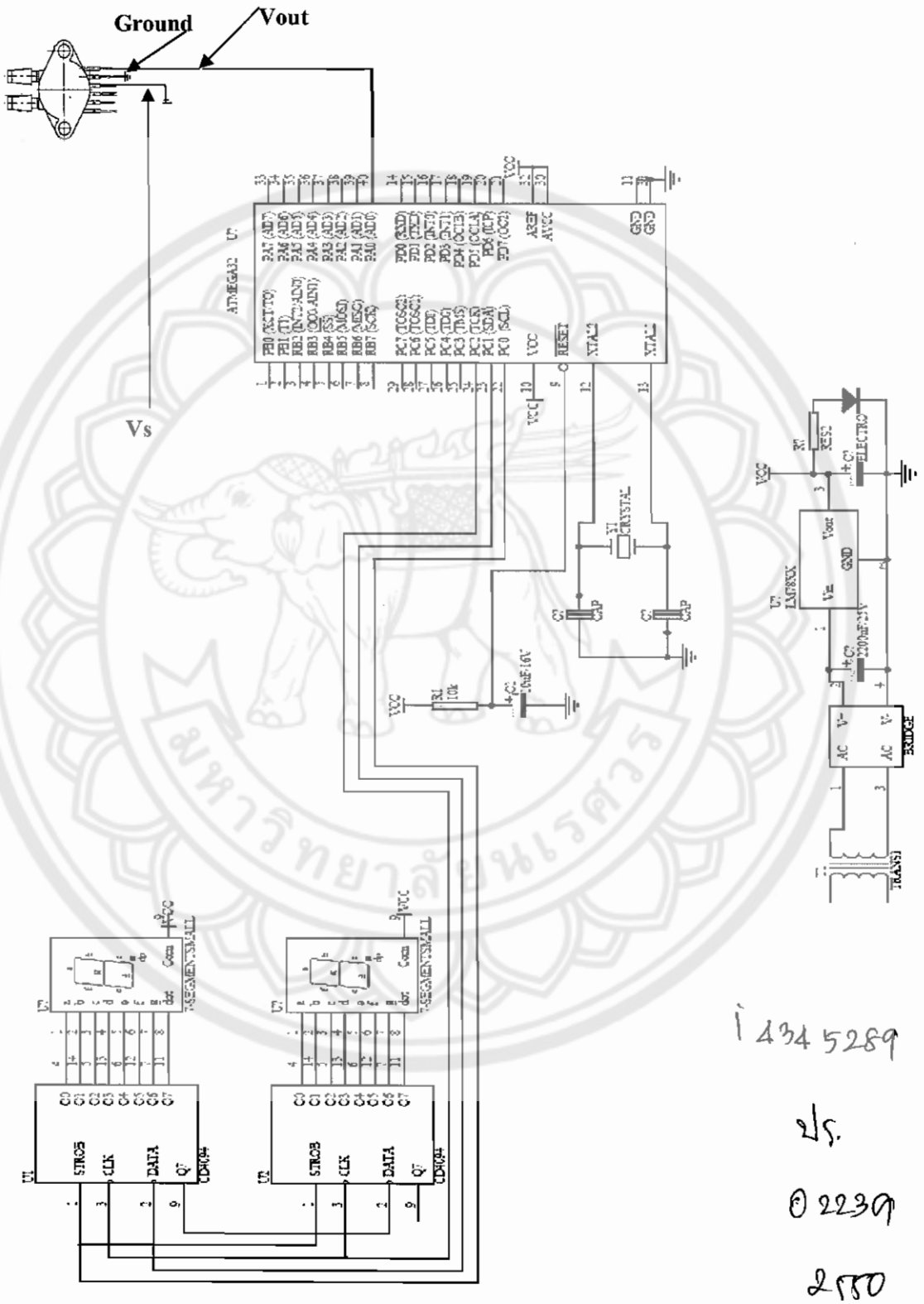
ขา 40 เป็นพอร์ด A ซึ่งพอร์ตนี้จะใช้เป็น AD Converter ได้ แต่กำหนดให้ใช้พอร์ต A0 เพียงพอร์ตเดียวเพื่อรับค่าจากเซนเซอร์ MPX5100DP โดยจะรับสัญญาณมาเป็นแบบสัญญาณอนาล็อก

ขา 22 เป็นพอร์ต C0 โดยจะกำหนดให้เป็นขาของ strobe

ขา 23 เป็นพอร์ต C1 กำหนดให้เป็นขาของ data

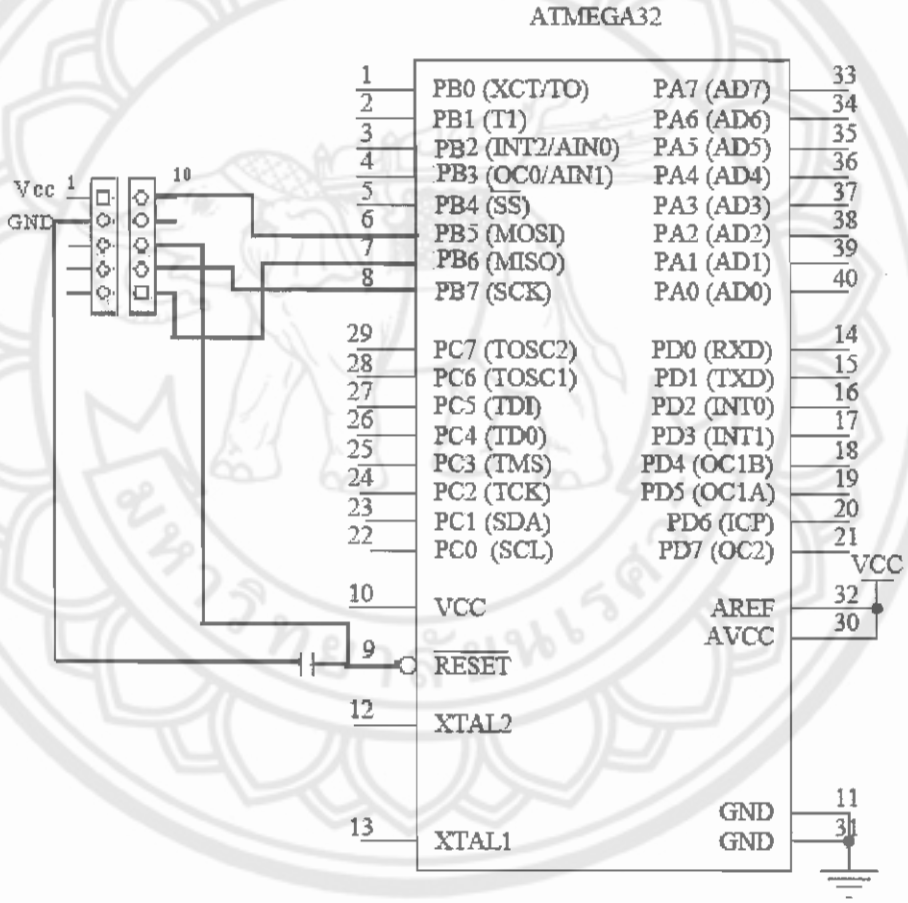
ขา 24 เป็นพอร์ต C2 กำหนดให้เป็นขาของ clock

3.3 การออกแบบวงจร



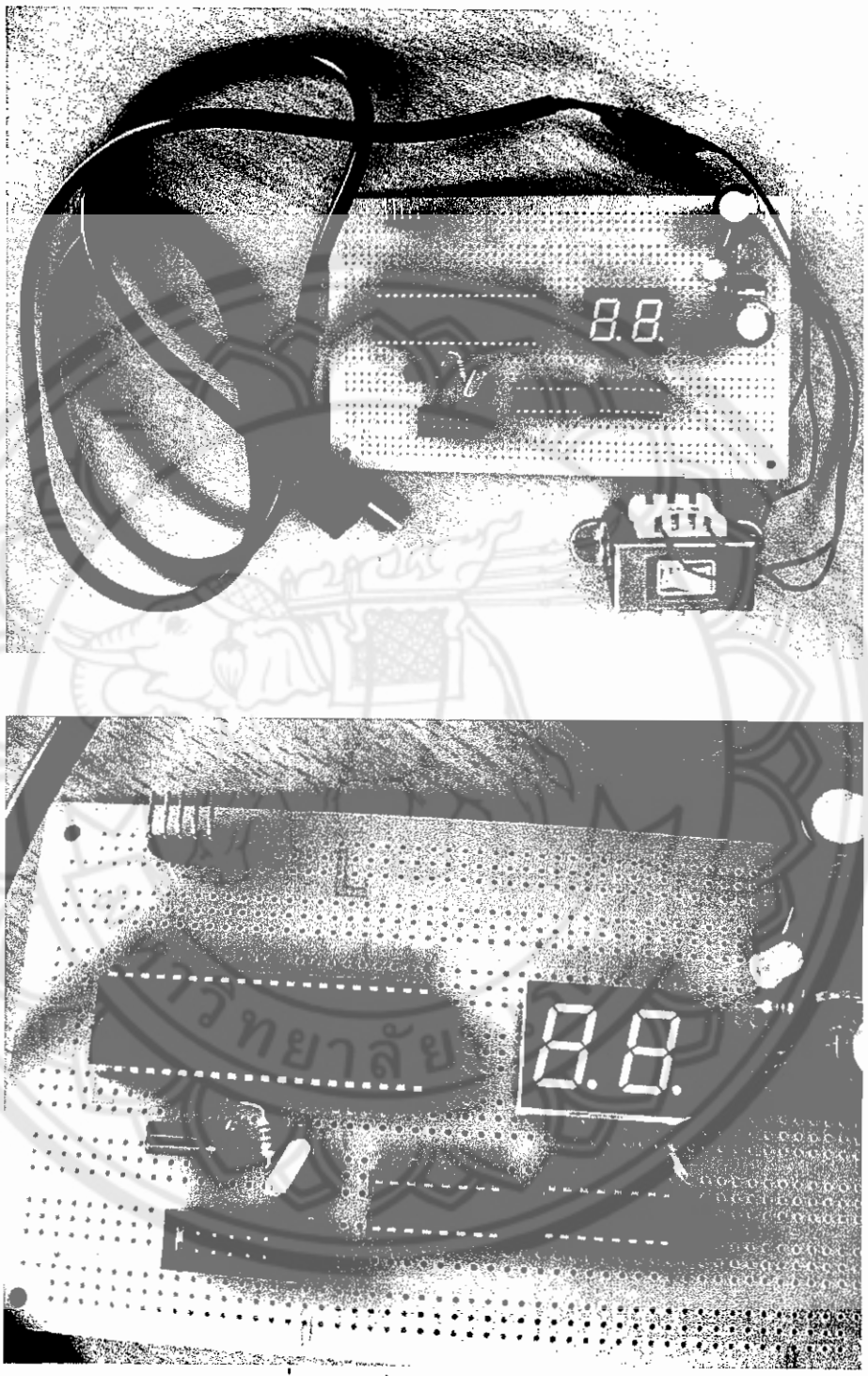
รูปที่ 3.3 การออกแบบวงจรเครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน

จากรูปวงจรอธิบายได้ดังนี้ ต่อเอาท์พุตของเซนเซอร์ MPX5100DP (ขา 1) เข้ากับพอร์ต A0 (ขา 40) ของ ATmega32 ไมโครคอนโทรลเลอร์ ATmega32 ก็จะทำการประมวลผลตามคำสั่ง แล้วส่งข้อมูลออกมาทางพอร์ต C1 โดยที่ข้อมูลจะถูกส่งไปยังไอซีเบอร์ CD4094 เพื่อแปลงข้อมูลที่รับมาแบบอนุกรมให้เป็นข้อมูลแบบขนาน ซึ่งขณะส่งข้อมูลไปนั้นจะต้องส่งสัญญาณ clock ซึ่งกำหนดให้เป็นพอร์ต C2 ไปด้วยเพื่อทำการส่งข้อมูลที่ได้ให้ไปถึงไอซีเบอร์ CD4094 ตัวที่หนึ่ง ไอซีเบอร์ CD4094 ตัวที่หนึ่งจะทำการส่งข้อมูลที่ได้ไปยังตัวที่สองเมื่อข้อมูลครบ 8 บิตแล้ว ก็หยุดการส่งข้อมูลไปยังตัวที่สอง ส่วนตัวที่หนึ่งเมื่อข้อมูลครบ 8 บิตแล้ว ก็พร้อมที่จะส่งข้อมูลออกมาแสดงผลทาง 7-segment สองหลัก โดยที่จะต้องส่งสัญญาณ strobe ให้กับไอซีเบอร์ CD4094 เพื่อทำการส่งข้อมูล ซึ่งข้อมูลที่ออกมาจะเป็นข้อมูลแบบขนานให้กับ 7-segment สองหลัก



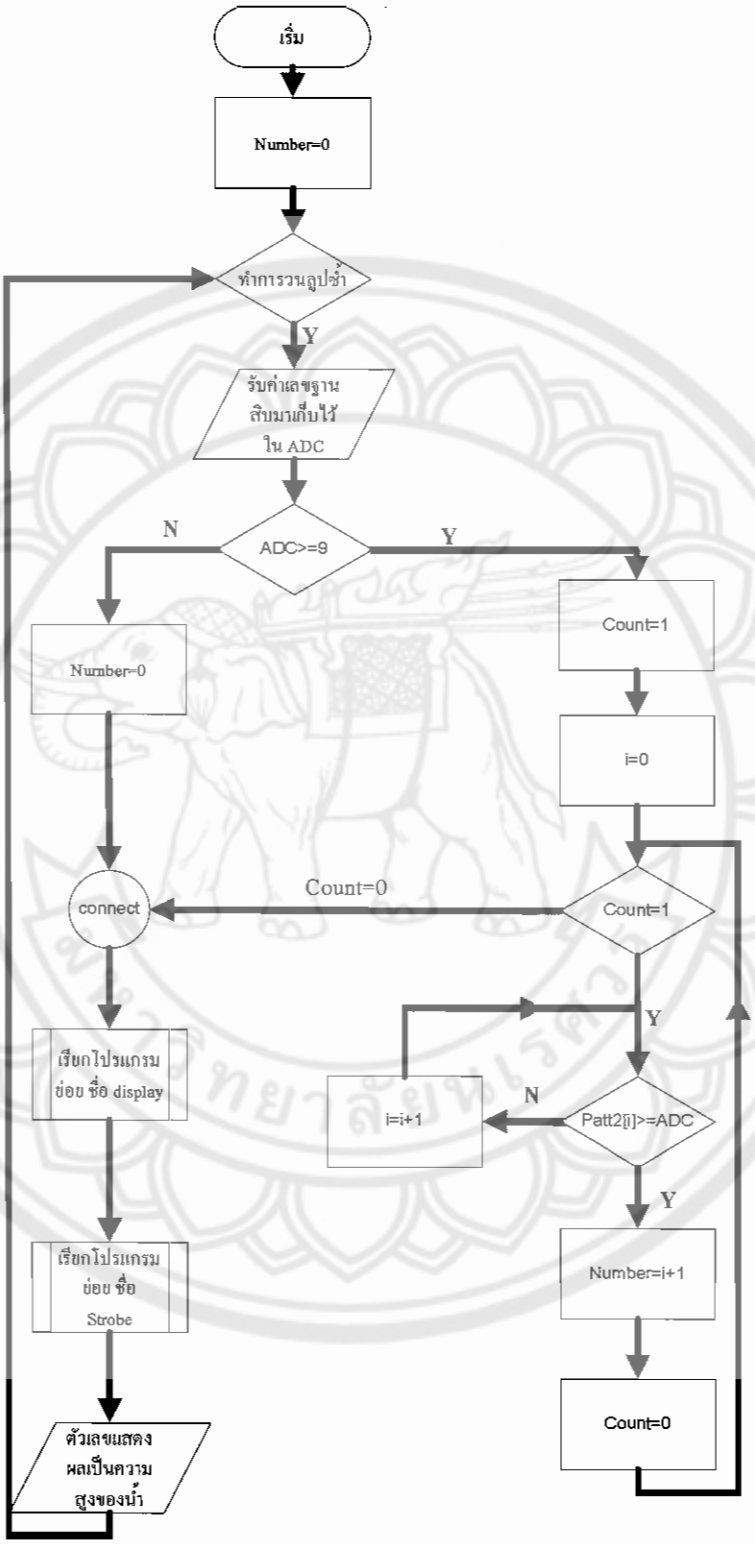
รูปที่ 3.4 วงจรเขียนโปรแกรมลงไมโครคอนโทรเลอร์

3.4 บอร์ดควบคุมและแสดงผล



รูปที่ 3.5 บอร์ดควบคุมและแสดงผล

3.5 การออกแบบโปรแกรม



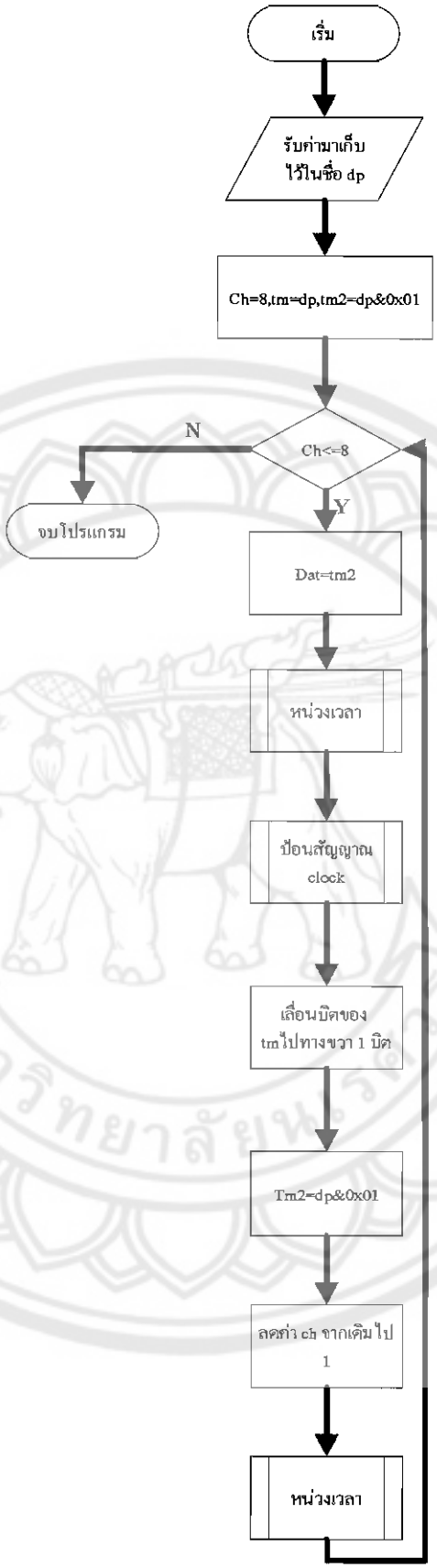
รูปที่ 3.6 แผนผังโปรแกรมทำเครื่องมือวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน

อธิบายแผนผังโปรแกรมทำเครื่องมือวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน

การเขียนโปรแกรมเริ่มต้นตั้งแต่ก่อนหย่อนเซนเซอร์ลงน้ำ ค่าที่อ่านได้จะเป็น 0.0 เมตร เนื่องจากเริ่มต้นโปรแกรมให้ number=0 ค่า number นี้จะเป็นค่าตัวเลขที่นำไปแสดงผล โดยหลักการคือ จะนำค่า number ไปหารแบบเอาเศษด้วย 10 (สัญลักษณ์ที่ใช้ %) แล้วจะได้ตำแหน่งของอาร์เรย์ของอาร์เรย์ชื่อ patt เพื่อแสดงตัวเลขหลักที่ 2 และนำค่า number หารแบบเอาส่วนด้วย 10 (สัญลักษณ์ที่ใช้ /) แล้วจะได้ตำแหน่งของอาร์เรย์ของอาร์เรย์ชื่อ patt เพื่อแสดงตัวเลขหลักที่ 1

เมื่อเริ่มหย่อนเซนเซอร์ลงไปในน้ำ ADC จะรับค่ามาเก็บไว้โดยจะเป็นเลขฐานสิบ 256 ค่า ที่เป็นเช่นนี้เพราะตอนเขียนโปรแกรมได้เลือกการใช้งานแบบ 8 บิต (แต่สามารถใช้งานได้ถึง 10 บิต แปลงเป็นเลขฐานสิบได้ 1024 ค่า)

เมื่อ ADC รับค่าเป็นเลขฐานสิบมาแล้วจะพิจารณาเงื่อนไขว่า ค่าที่รับมานั้นมากกว่าหรือเท่ากับ 9 หรือไม่ (9 คือ ตำแหน่งที่ 0 ในอาร์เรย์ patt2) ถ้าไม่จริงก็ให้ number=0 ก็คือแสดงตัวเลข 0.0 แต่ถ้าจริงก็ให้ count =1 และ i=0 (iคือ ตำแหน่ง) เพื่อเริ่มทำการวนลูปซ้ำ จากนั้นก็นำค่า ADC ที่ได้ไปเปรียบเทียบกับค่าในอาร์เรย์ชื่อ patt2 แล้วพิจารณาค่าในอาร์เรย์ว่ามากกว่าหรือเท่ากับค่า ADC ที่รับมาหรือไม่ ถ้าไม่จริงก็ให้เพิ่มค่า i ไปอีกหนึ่งเรื่อยๆ ถ้าจริงก็ให้ number=i+1 แล้วกำหนดให้ค่า count = 0 เพื่อออกจากลูป เพื่อไปยังโปรแกรมย่อย display และ strobe ตามลำดับ เพื่อแสดงค่าตัวเลขที่แปลงได้จากความดันน้ำเป็นความสูงของน้ำ จากนั้นก็ทำการวนลูปไปเรื่อยๆ



รูปที่ 3.7 แผนผังโปรแกรมย่อย display

อธิบายแผนผังโปรแกรมโปรแกรมย่อย display

เริ่มจากการรับค่าในอาร์เรย์ patt มาเก็บไว้ใน dp ซึ่งเป็นเลขฐานสิบหก แล้วกำหนดให้

ch=8

tm=dp

tm2=dp&0x01 (เพื่อให้ข้อมูลส่งไปยังไอซีเบอร์ CD4094 ทีละบิต)

จากนั้นพิจารณาเงื่อนไข ถ้า ch≤8 จริง ก็ให้ dat=tm2 (dat คือ ข้อมูล 1 บิต ที่ได้จาก dp&0x01ที่จะส่งไปยังไอซีเบอร์ CD4094) ต้องป้อนสัญญาณ clock ไปด้วยเพื่อทำการส่งข้อมูลให้กับไอซีเบอร์ CD4094 เพื่อรอทำการแปลงเป็นข้อมูลแบบขนาน จากนั้นเลื่อนบิต tm ไปทางขวา 1 บิตแล้วทำการ AND(&) กับ 0x01 แล้วทำการลดค่า ch ลงไป 1 แล้วเข้าไปวนลูปพิจารณาค่า ch ว่าตรงตามเงื่อนไขหรือไม่ จนกว่าข้อมูลจะครบ 8 บิต จึงจะออกจากโปรแกรม

บทที่ 4
ผลการทดลองและผลการวิเคราะห์

4.1 การทดลองวัดระดับน้ำจากเซนเซอร์แรงดัน

การทดสอบโปรแกรมเป็นการตรวจสอบความถูกต้องเพื่อให้สอดคล้องกับเครื่องวัดระดับน้ำที่แสดงผลเป็นแบบดิจิตอล 2 หลัก

4.1.1 จุดประสงค์

4.1.1.1 เพื่อศึกษาการทำงานสัมพันธ์กันระหว่างไมโครคอนโทรลเลอร์กับเซนเซอร์แรงดัน

4.1.1.2 เพื่อศึกษาวิเคราะห์ข้อมูล อินพุต/เอาท์พุต ของอุปกรณ์ที่ใช้ในวงจร

4.1.2 ขั้นตอนการเชื่อมต่อบอร์ดควบคุม

4.1.2.1 เชื่อมต่อเซนเซอร์แรงดันเข้ากับบอร์ดควบคุม

4.1.2.2 ตรวจสอบการเชื่อมต่อของอุปกรณ์ทั้งหมด

4.1.2.3 จ่ายไฟ 5 V DC ให้กับบอร์ดควบคุม

4.1.3 ทดสอบการควบคุมและการแสดงผล

การทดสอบจะหย่อนเซนเซอร์แรงดันลงไปในน้ำ โดยจะวัดความสูงของน้ำจากสายไฟที่หย่อนลงไป การทดสอบจะทดสอบที่ความสูงของระดับน้ำ 8 เมตร ซึ่งจะใส่น้ำเข้าไปในท่อยาว 8 เมตรเพราะฉะนั้นน้ำที่ใช้ทดลองจะต้องเป็นน้ำนิ่งจะไม่มีการไหลของน้ำ โดยจะหย่อนสายไฟลงไปทีละ 10 เซนติเมตร แล้วทำการบันทึกค่าความต่างศักย์ และผลที่แสดงออกมาทาง 7-segment 2 หลัก

4.2 ผลการทดลอง

4.2.1 ผลการทดสอบการควบคุมและการแสดงผล

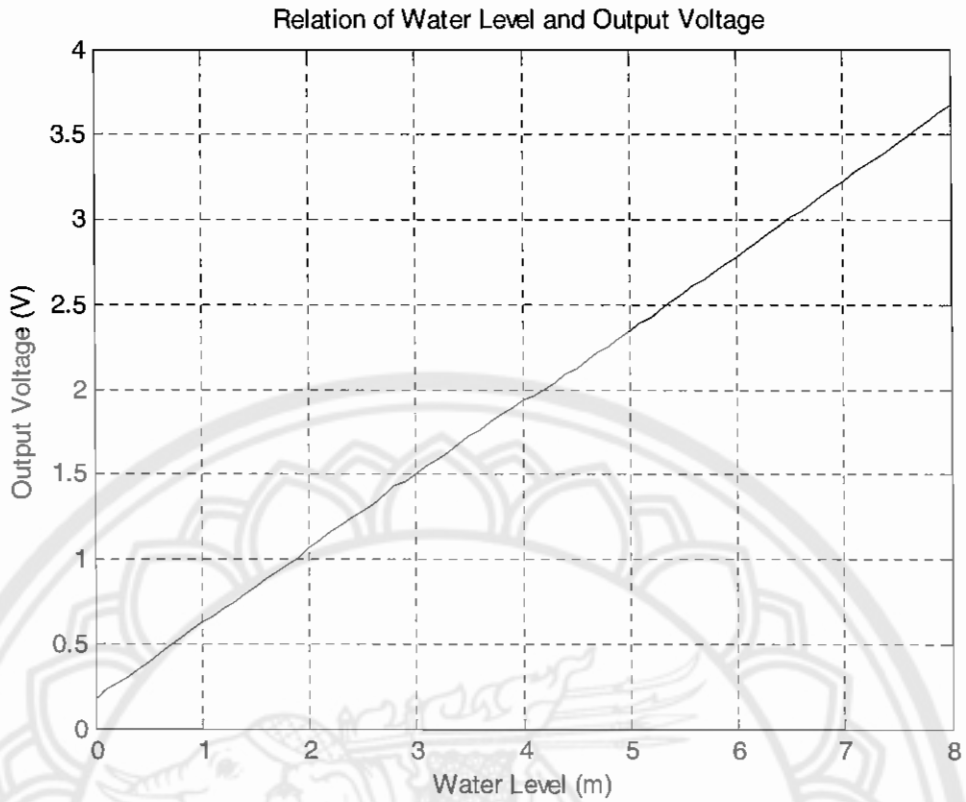
ตารางที่ 4.1 ผลจากการทดสอบเครื่องวัดระดับน้ำ

ความยาวสายไฟที่หย่อนลงน้ำ (เมตร)	ความต่างศักย์ที่วัดได้ (โวลต์)	ผลที่แสดง (เมตร)	ความยาวของสายไฟที่หย่อนลงน้ำ (เมตร)	ความต่างศักย์ที่วัดได้ (โวลต์)	ผลที่แสดง (เมตร)
0.0	0.17	0.0	2.1	1.107	2.1
0.1	0.231	0.1	2.2	1.164	2.2
0.2	0.259	0.2	2.3	1.199	2.3
0.3	0.303	0.3	2.4	1.241	2.4
0.4	0.349	0.4	2.5	1.285	2.5
0.5	0.392	0.5	2.6	1.325	2.6
0.6	0.435	0.6	2.7	1.373	2.7
0.7	0.485	0.7	2.8	1.436	2.8
0.8	0.529	0.8	2.9	1.459	2.9
0.9	0.574	0.9	3.0	1.500	3.0
1.0	0.620	1.0	3.1	1.546	3.1
1.1	0.659	1.1	3.2	1.589	3.2
1.2	0.702	1.2	3.3	1.631	3.3
1.3	0.748	1.3	3.4	1.675	3.4
1.4	0.794	1.4	3.5	1.720	3.5
1.5	0.835	1.5	3.6	1.765	3.6
1.6	0.882	1.6	3.7	1.806	3.7
1.7	0.924	1.7	3.8	1.850	3.8
1.8	0.969	1.8	3.9	1.894	3.9
1.9	1.013	1.9	4.0	1.930	4.0
2.0	1.065	2.0			

ตารางที่4.1(ต่อ) ผลจากการทดสอบเครื่องวัดระดับน้ำ

ความยาวสายไฟที่หย่อนลงน้ำ (เมตร)	ความต่างศักย์ที่วัดได้ (โวลต์)	ผลที่แสดง (เมตร)	ความยาวของสายไฟที่หย่อนลงน้ำ (เมตร)	ความต่างศักย์ที่วัดได้ (โวลต์)	ผลที่แสดง (เมตร)
4.1	1.956	4.1	6.1	2.830	6.1
4.2	1.998	4.2	6.2	2.875	6.2
4.3	2.038	4.3	6.3	2.918	6.3
4.4	2.083	4.4	6.4	2.964	6.4
4.5	2.123	4.5	6.5	3.007	6.5
4.6	2.174	4.6	6.6	3.052	6.6
4.7	2.213	4.7	6.7	3.094	6.7
4.8	2.256	4.8	6.8	3.143	6.8
4.9	2.301	4.9	6.9	3.182	6.9
5.0	2.345	5.0	7.0	3.228	7.0
5.1	2.388	5.1	7.1	3.275	7.1
5.2	2.430	5.2	7.2	3.318	7.2
5.3	2.479	5.3	7.3	3.364	7.3
5.4	2.521	5.4	7.4	3.402	7.4
5.5	2.568	5.5	7.5	3.449	7.5
5.6	2.611	5.6	7.6	3.493	7.6
5.7	2.654	5.7	7.7	3.537	7.7
5.8	2.698	5.8	7.8	3.580	7.8
5.9	2.742	5.9	7.9	3.626	7.9
6.0	2.783	6.0	8.0	3.674	8.0

<u>หมายเหตุ</u> ความสูงของน้ำที่วัดได้จะมีข้อผิดพลาดไม่เกิน ±0.2 เซนติเมตร



รูปที่ 4.1 กราฟความสัมพันธ์ระหว่างระดับน้ำกับความต่างศักย์ที่ได้จากเซนเซอร์แรงดัน

จากกราฟที่ได้อธิบายได้ว่าเมื่อระดับน้ำเพิ่มขึ้นส่งผลให้ความต่างศักย์ที่ได้จากเซนเซอร์แรงดันมีค่าเพิ่มขึ้นด้วย ซึ่งค่าความต่างศักย์สูงสุดที่ได้เซนเซอร์แรงดันจะมีค่า 5 โวลต์ และระดับน้ำที่วัดได้สูงสุดอยู่ที่ 10 เมตร

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลของโครงงาน

จากการทดลองนำเซนเซอร์ MPX5100DP ของ Motorola ซึ่งเป็นเซนเซอร์ที่แปลงความดันของเหลวให้เป็นแรงดันไฟฟ้า 0-5 โวลต์ เมื่อเราต่อสายสัญญาณและจุ่มเซนเซอร์ลงในน้ำโดยทำการวัดระยะที่สายไฟทุกๆ 10 เซนติเมตร ปรากฏว่าผลที่แสดงทาง 7-segment ตรงกับตำแหน่งของสายไฟที่จุ่มลงไปในน้ำ

จะเห็นว่าจากการทดลองวัดระดับน้ำด้วยเครื่องวัดระดับน้ำโดยใช้เซนเซอร์แรงดัน ต่อเข้ากับส่วนควบคุมเพื่อแสดงผลแบบดิจิตอล 2 หลัก เครื่องวัดระดับน้ำสามารถอ่านค่าได้อย่างถูกต้อง

5.2 ปัญหาที่เกิดขึ้นในระหว่างทำโครงงาน

5.2.1 เนื่องไมโครคอนโทรเลอร์มีความละเอียดสูงในการทดลองบางครั้งจึงทำให้ค่าที่ได้คลาดเคลื่อนจากความเป็นจริงไปบ้าง

5.2.2 เนื่องจากเซนเซอร์มีค่าผิดพลาด+0.2,-0.2โวลต์ซึ่งเป็นค่าที่มากพอที่จะทำให้ไมโครคอนโทรเลอร์ตัดสินใจผิดพลาด

5.3 แนวทางในการแก้ไขปัญหา

5.3.1 เราจะทำการทดลองหลายๆครั้งเพื่อให้ได้ค่าที่แน่นอนในแต่ละระดับของน้ำ

5.3.2 เราจำเป็นที่จะต้องวัดระดับของน้ำในขณะที่น้ำนิ่งจริงๆเท่านั้น

5.3.3 จ่ายไฟให้กับไมโครคอนโทรเลอร์และเซนเซอร์ด้วยระดับที่คงที่

5.4 ข้อเสนอแนะ

จากผลการทดลองจะเห็นว่าค่าที่แสดงมีความละเอียดแค่ทศนิยม 1 ตำแหน่งนั่นคือความละเอียด 10 เซนติเมตร ในความเป็นจริงไมโครคอนโทรเลอร์ AVR สามารถที่จะใช้งานที่ความละเอียด 10 บิต นั่นคือเราสามารถประยุกต์ใช้งานโครงงานนี้ให้แสดงผลได้ถึงทศนิยมตำแหน่งที่ 2 คือความละเอียด 1 เซนติเมตร แต่เราจำเป็นที่จะต้องหาเซนเซอร์ที่มีค่าความผิดพลาดให้น้อยกว่าเซนเซอร์ตัวนี้ ในโครงงานนี้เราใช้เซนเซอร์ MPX5100DP ซึ่งมีค่าความผิดพลาด +0.2,-0.2 โวลต์ นั่นคือจะทำให้มีค่าความผิดพลาดมากยิ่งขึ้นถ้าเราจะใช้การแสดงผลแบบทศนิยม 2 ตำแหน่ง เมื่อเราได้เซนเซอร์ที่มีความเที่ยงตรงสูง (ควรมีค่าความผิดพลาดไม่เกิน +0.01,-0.01 โวลต์) เราก็นำเซนเซอร์มาศึกษาถึงความสำพันธ์ระหว่างเอาท์พุตที่ได้กับแรงดันของน้ำว่าเป็นแบบเชิงเส้นหรือไม่ เพราะถ้าเป็นเชิงเส้นก็จะง่ายต่อการคำนวณ รวมทั้งง่ายต่อการเขียนโปรแกรมเพื่อรับค่าและแสดงผล ในส่วนของการเขียนโปรแกรมนั้นเราต้องกำหนดค่าในตาราง Array ใหม่ เพื่อให้มีความ

ละเอียดเพิ่มขึ้นและการใช้ชิปรีจิสเตอร์ก็ต้องเพิ่มหลักเข้าไปในส่วนของโปรแกรมที่ต้องเปลี่ยนแปลงคือ หลักแรกใช้ 100 หารเพื่อเก็บส่วน ทศนิยมตำแหน่งที่ 1 ใช้ 10 หารเพื่อเก็บส่วน ทศนิยมตำแหน่งที่ 2 ใช้ 10 หารเพื่อเก็บเศษ เช่น Array ที่ i=51

51 % 100 = 0

51 / 10 = 5

51% 10 = 1

% = หารเพื่อเก็บค่าสวน

/ =หารเพื่อเก็บค่าเศษ

จะแสดงผล 0.51 โดยหลักแรกเราจะแสดงผลให้มีจุดติดมาด้วยดังนั้นเราจำเป็นที่จะต้องนำไป OR กับ 0x01 เพื่อจะแสดงผลในตำแหน่ง h ของ 7-segment

เอกสารอ้างอิง

[1] ประจิน พลังสันติกุล. **การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับ WinAVR (C Compiler)**. กรุงเทพมหานคร: แอฟซอฟต์เทค, 2549.

[2] Atmel Company. " AVR Microcontroller." [Online]. Avialable: Http:// www.atmel.com/dyn/resources/prod_documents/doc2503.pdf. 2007.

[3] Texas Instruments."CD4094BE - CMOS 8-STAGE SHIFT-AND-STORE BUS REGISTER." [Online]. Avialable: Http:// www.alldatasheet.com/datasheetpdf/pdf/26897/TI/CD4094BE.html. 2008

[4] Freescale Semiconductor. " MPX5100DP." [Online]. Avialable: Http:// www.alldatasheet.com/datasheet-pdf/pdf/84246/MOTOROLA/MPX5100DP.html. 2008.

ภาคผนวก

ภาคผนวก ก.
ภาษาซีที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์

```
#include <mega32.h>     //ประกาศ Preprocessor directive เพื่อใช้ไมโครคอนโทรลเลอร์
                        ATmega32
#include <delay.h>      //ประกาศ Preprocessor directive เพื่อใช้ฟังก์ชัน delay
#include <stdio.h>      //ประกาศ Preprocessor directive เพื่อเรียกใช้ฟังก์ชันพื้นฐานของ
                        compiler

#define strobe  PORTC.0         //ประกาศให้พอร์ต C0 เท่ากับ strobe
#define clk     PORTC.2         //ประกาศให้พอร์ต C2 เท่ากับ clk
#define dat     PORTC.1         //ประกาศให้พอร์ต C1 เท่ากับ dat

    // Place your code here

    //          a b c d e f g h
    //  0       1 1 1 1 1 1 0 0     fc
    //  1       0 1 1 0 0 0 0 0     60
    //  2       1 1 0 1 1 0 1 0     da
    //  3       1 1 1 1 0 0 1 0     f2
    //  4       0 1 1 0 0 1 1 0      66
    //  5       1 0 1 1 0 1 1 0     b6
    //  6       1 0 1 1 1 1 1 0     be
    //  7       1 1 1 0 0 0 0 0     e0
    //  8       1 1 1 1 1 1 1 0     fe
    //  9       1 1 1 1 0 1 1 0     f6
int number=0,i=0 ;
unsigned char ADC,patt[]={0xfc,0x60,0xda,0xf2,0x66,
                0xb6,0xbe,0xe0,0xfe,0xf6,0x00,0x01};  //อาร์เรย์ตัวเลข 0-9ที่ใช้แสดงผล
char count;
unsigned int    patt2[]={9,11,13,16,18,20,23,25,27,29,
                 32,34,36,38,41,43,45,48,50,52,
                 55,57,60,64,66,69,68,71,73,76,
                 78,80,82,85,87,89,91,94,96,97,
                 99,101,104,106,108,110,112,115,
```

```
                117,119,121,124,126,129,131,133, //อาร์เรย์เพื่อนำค่าที่ ADC ได้มาเปรียบเทียบ
                136,138,140,143,145,147,150,152,
                154,156,159,161,163,166,168,170,172,
                175,177,179,181,184,186,189,191,193,
                195,198,201,203,205,208,210,212,214,
                217,219,222,224,226,228,231,233};

unsigned char adc_data;

#define ADC_VREF_TYPE 0x20          //ส่วนประกาศใช้งานรีจิสเตอร์ ADC

unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}



// Declare your global variables here
void strob(void)
{

                strobe=1; delay_ms(1);
                strobe=0; delay_ms(1);      //ฟังก์ชัน strobe เพื่อให้แสดงผล
}
```

```
void pulse(void)
{
        clk=1; delay_ms(1);          //ฟังก์ชัน pulse เพื่อส่งข้อมูลทีละบิต
        clk=0; delay_ms(1);
}

void display(char dp)
{
        unsigned char tm,tm2;
        char ch=8;
        tm =  dp;
        tm2 = (dp&0x01);
        while(ch)                    //ฟังก์ชัน display เพื่อแปลงข้อมูลเป็นแบบอนุกรม
        {
          dat=tm2; delay_ms(1);
          pulse();
          tm = tm>>1;
          tm2 = (tm & 0x01);
          ch--;  delay_ms(1);

        }
}

void main(void)
{

PORTA=0x00;                  //ประกาศให้พอร์ต A เป็นอินพุต
DDRA=0x00;

PORTB=0x00;                  //ประกาศให้พอร์ต B เป็นอินพุต
DDRB=0x00;
```

```
PORTC=0x00;              //ประกาศให้พอร์ต C เป็นเอาท์พุต
DDRC=0xff;

PORTD=0x00;              //ประกาศให้พอร์ต D เป็นอินพุต
DDRD=0x00;

TCCR0=0x00;
TCNT0=0x00;              //timer/counter 0 initialization
OCR0=0x00;

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;              //timer/count 1 initialization
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

ASSR=0x00;
TCCR2=0x00;              //timer/connt 2 initialization
TCNT2=0x00;
OCR2=0x00;

MCUCR=0x00;              //External Interrupt(s) initialization
MCUCSR=0x00;
```

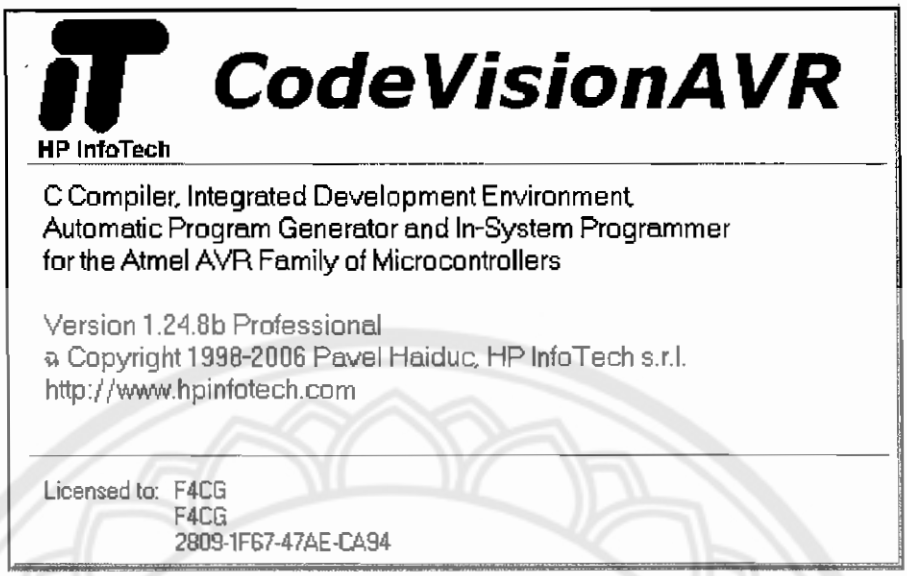
```
TIMSK=0x00;                    //timer(s)/count(s) interrupt(S) initialization

ACSR=0x80;                     //Analog Comparator initialization
SFIOR=0x00;

ADMUX=ADC_VREF_TYPE;           //ADC initialization
ADCSRA=0x86;


#asm("sei")                    // Global enable interrupts
 number=0;
while (1)
    {

    ADC=read_adc(0);           //รับค่ามาจากพอร์ต A0 มาเก็บไว้ใน ADC

     if ( ADC>=9)              //พิจารณาเงื่อนไข
    {
    count=1;
    i=0;
    while (count)
    {
    if (patt2[i]>=ADC)         //นำค่า ADC ไปเปรียบเทียบกับอาร์เรย์ patt2
      {
            number=i+1;
            count=0;
      }
      i++;
      if(i>=130){count=0;}     //ตรวจสอบค่า i ไม่ให้เกิน 130
      }
    }
```

```
else number=0;

display(patt[number%10]);    //แสดง 7-segment หลักที่ 2 โดยนำตัวเลขจากอาร์เรย์ patt
display(patt[number/10]|0x01); //แสดง 7-segment หลักที่ 1 โดยนำตัวเลขจากอาร์เรย์ patt
strob();
delay_ms(1000);


};
}
```

ภาคผนวก ข

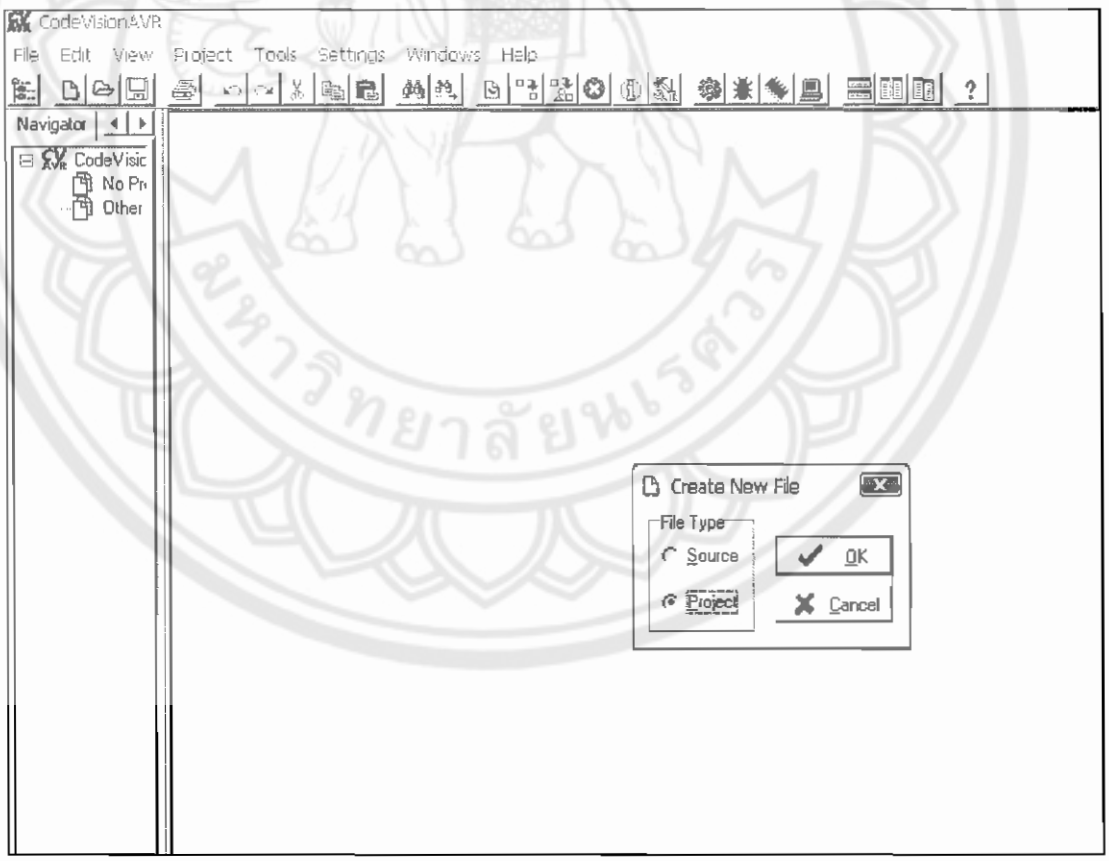
ขั้นตอนการเขียนโปรแกรมและเบิร์นโปรแกรมลงบอร์ด

1. เปิดโปรแกรม CodeVisionAVR



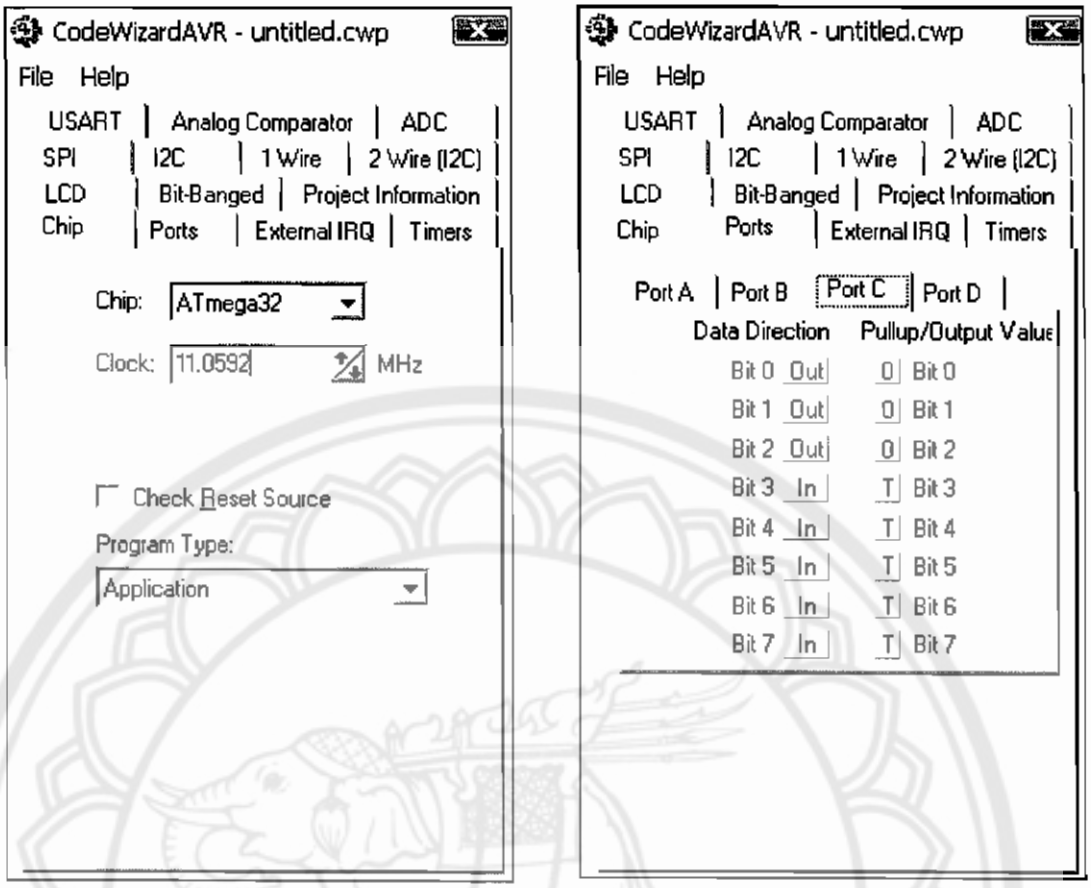
ผ-1 หน้าต่างเข้าโปรแกรม Code VisionAVR

2. เลือกที่ Project แล้วคลิกที่ OK



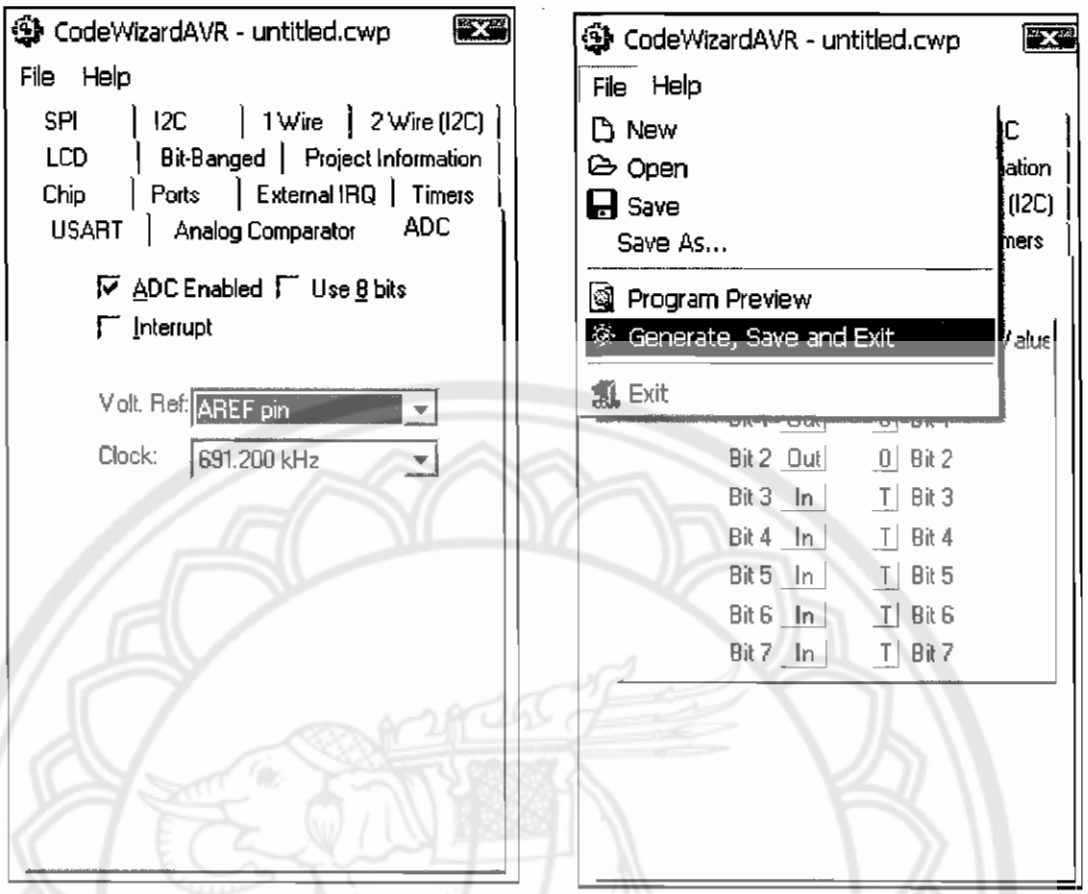
ผ-2 หน้าต่างเลือกชนิดของไฟล์

3. เลือก Chip ที่เราใช้งานและตั้งค่าความถี่ที่ใช้งาน และส่วนประกอบต่างๆที่เราจะใช้งาน



ผ-3 หน้าต่างตั้งค่าความถี่และค่าต่างๆให้กับ Chip

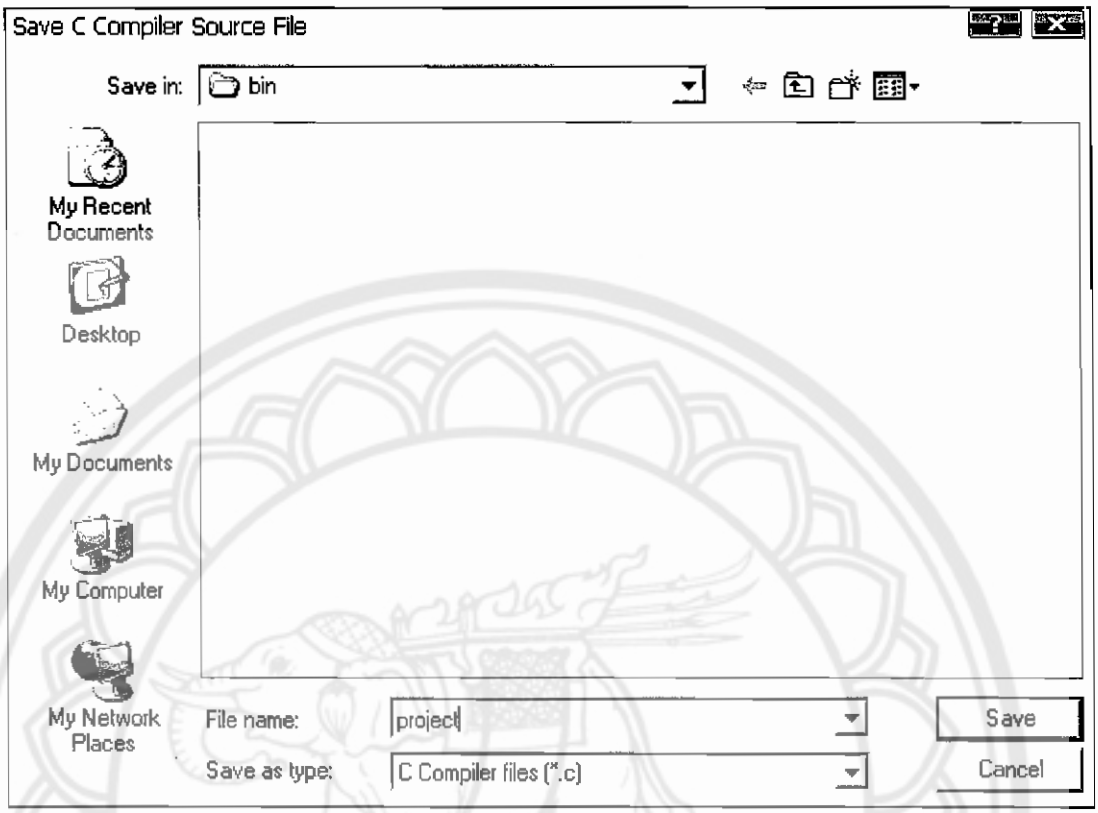
ในที่นี้เลือกการใช้งาน Port C เป็น Port ที่ส่งค่า Data Strop Clock จึงกำหนดเป็น out ตามรูป และเลือกการใช้งาน ขา ADC เทียบกับ AREF ตามรูปข้างต้น



ผ-4 หน้าต่างสร้างและบันทึกค่าที่ตั้งไว้

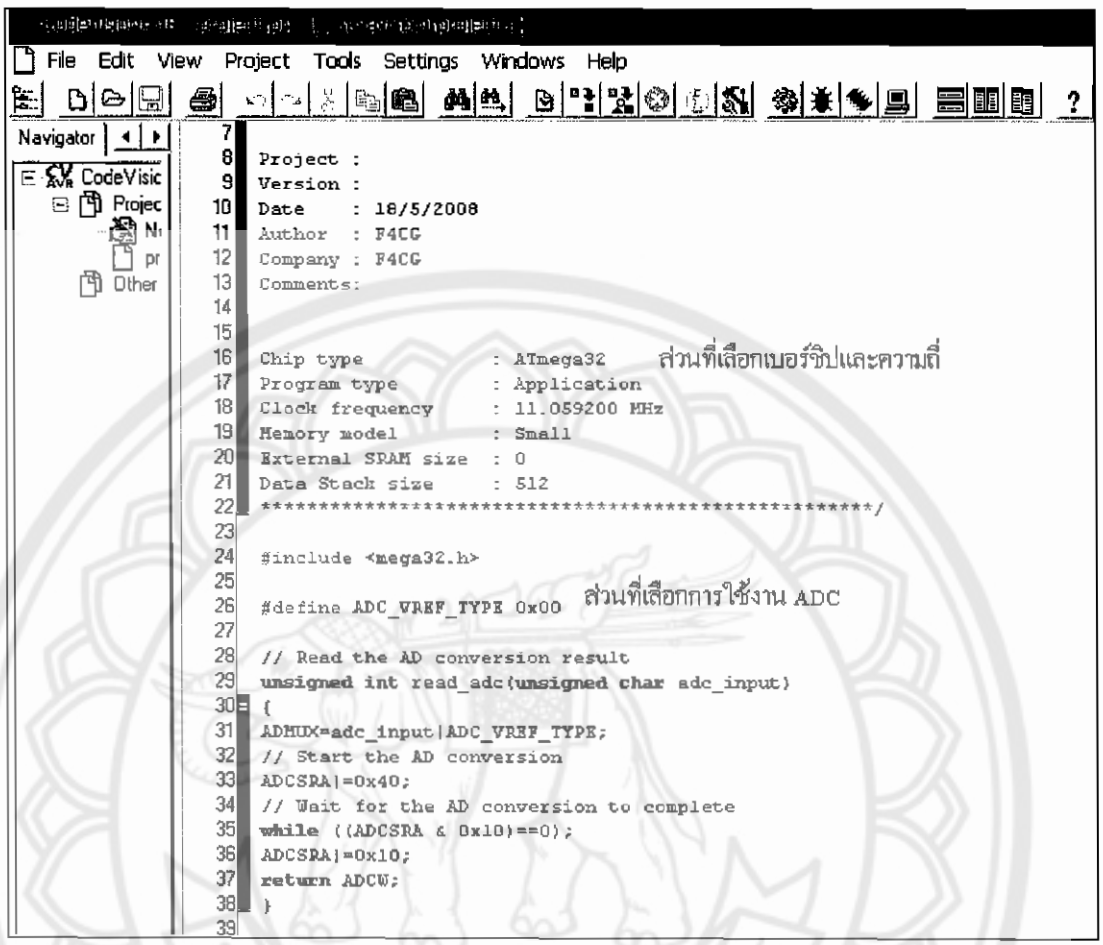
เมื่อเรากำหนดส่วนต่างๆที่จะใช้งานเสร็จแล้ว เลือกที่ File แล้วคลิกที่ Generate, Save and Exit

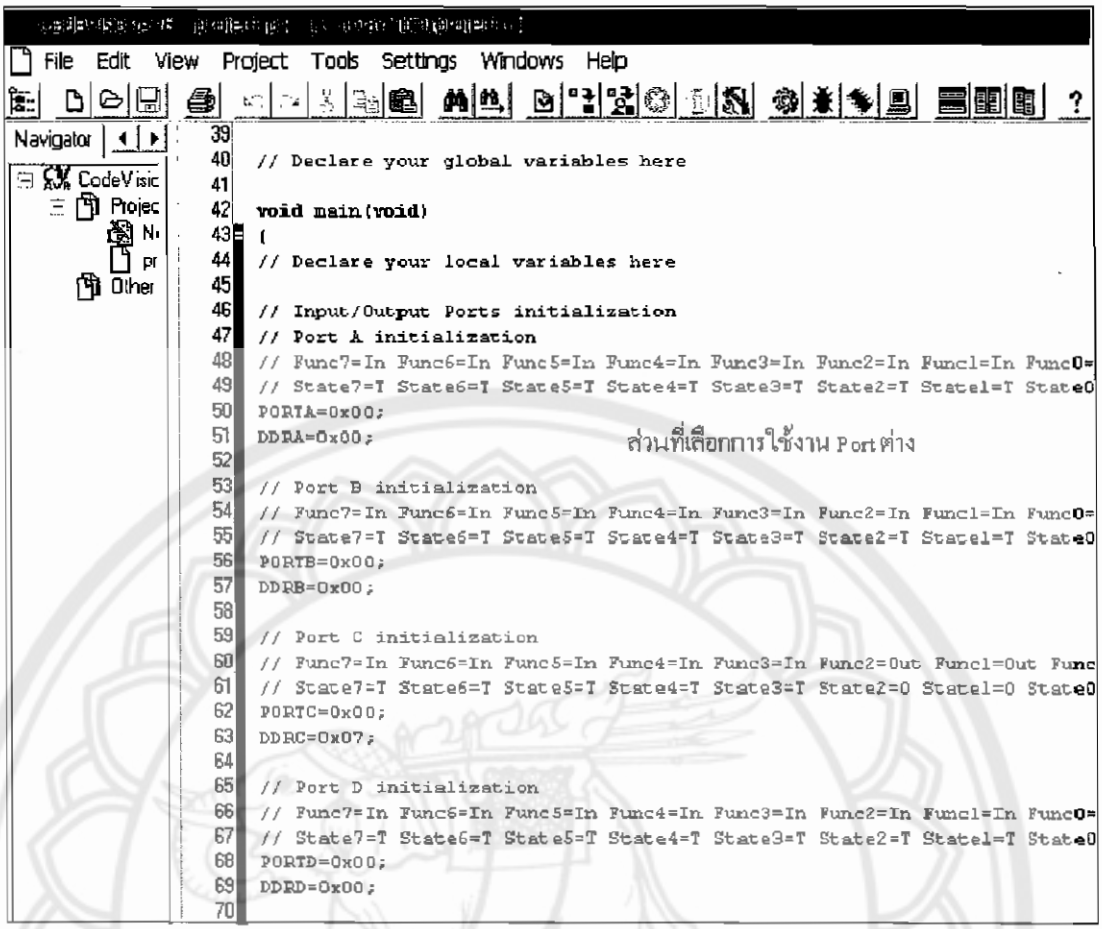
4. โปรแกรมก็จะให้ใส่ชื่อในการ Save ในที่นี้ตั้งเป็น Project



ผ-5 หน้าต่างบันทึกไฟล์ชื่อ project

5. โปรแกรมจะกำหนดส่วนต่างๆที่เราเลือกในช่วงแรกใส่ลงไปในโปรแกรม





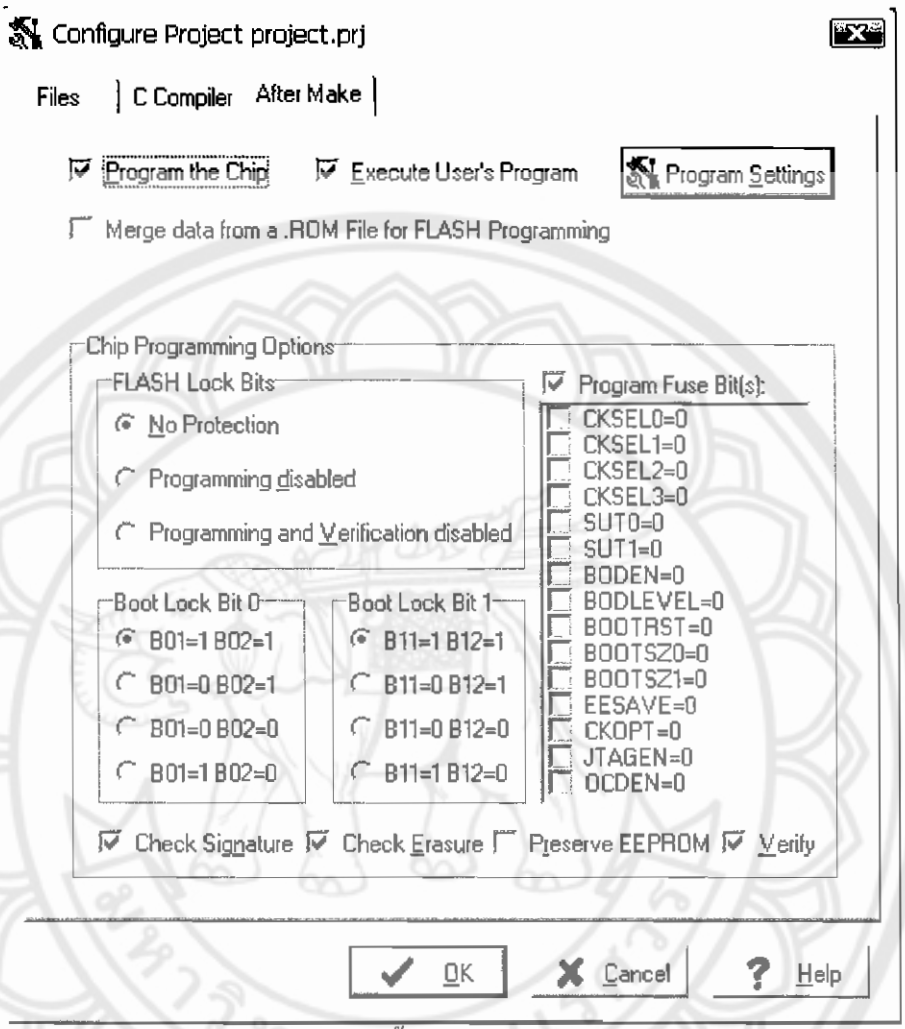
ผ-6 ค่าที่โปรแกรมกำหนดขึ้น

6. เราก็ทำการเขียนโปรแกรมในส่วนที่โปรแกรมกำหนดให้

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 691.200 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x84;

while (1)
      {
      // Place your code here

      };
}
```

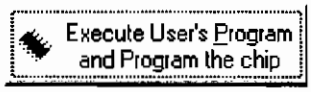
ผ7-โปรแกรมหลักที่จะต้องเขียน

7. ทำการตั้งค่า ก่อนทำการ Compile โดยการคลิกที่ ปุ่ม  แล้วเลือกใน ช่อง Program the Chip และ Execute User's Program



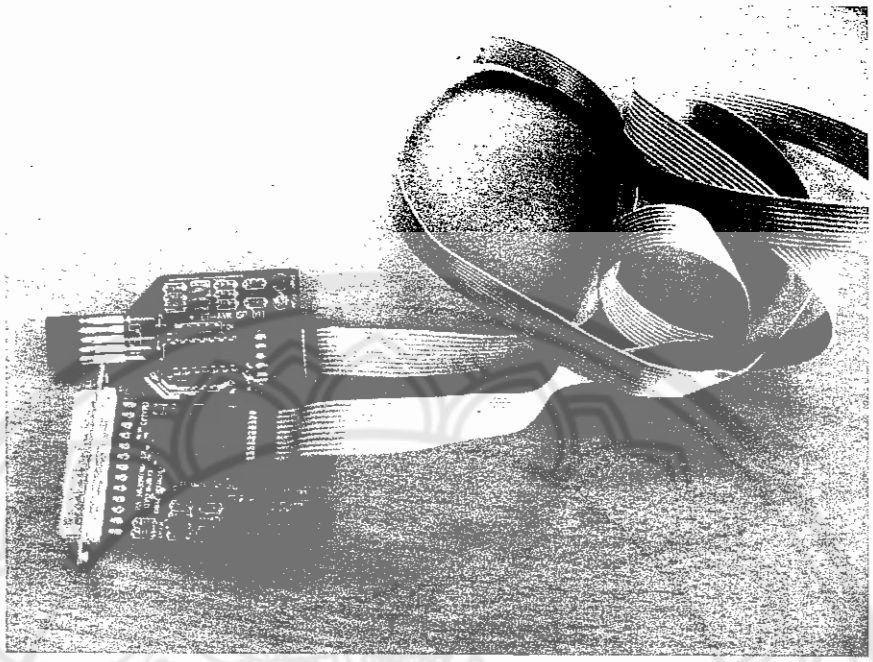
ผ-8 หน้าต่างตั้งค่าก่อนทำการคอมไพล์

8. คลิกที่ ปุ่ม เพื่อทำการ Compile และโหลดโปรแกรมลง ชิป โดยคลิกที่

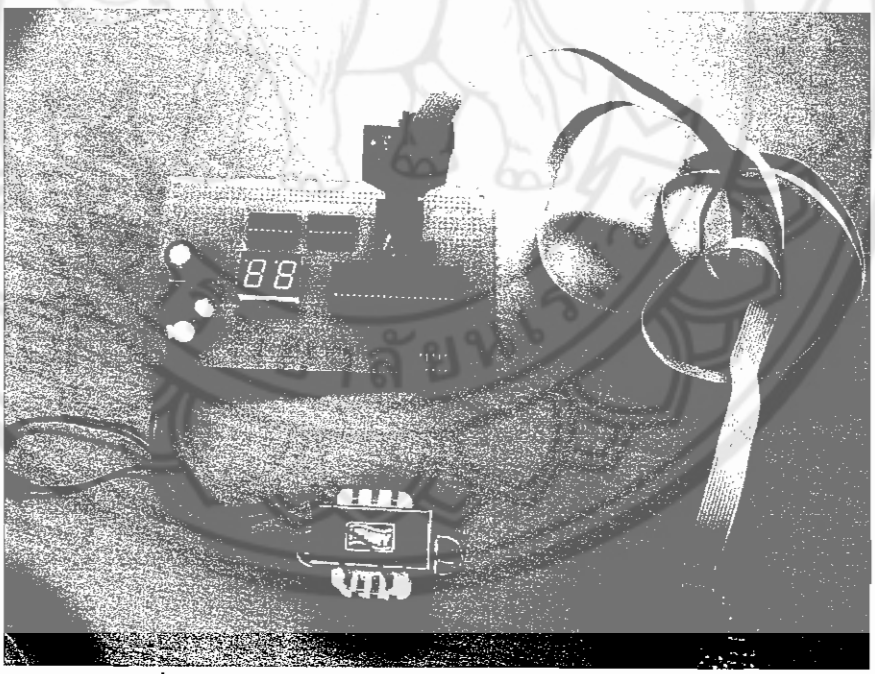


ผ-9 หน้าต่างการคอมไพล์และเตรียมโหลดโปรแกรมลง Chip

9. ทำการต่อสายเบิร์นโปรแกรมเข้ากับบอร์ดควบคุมและแสดงผล

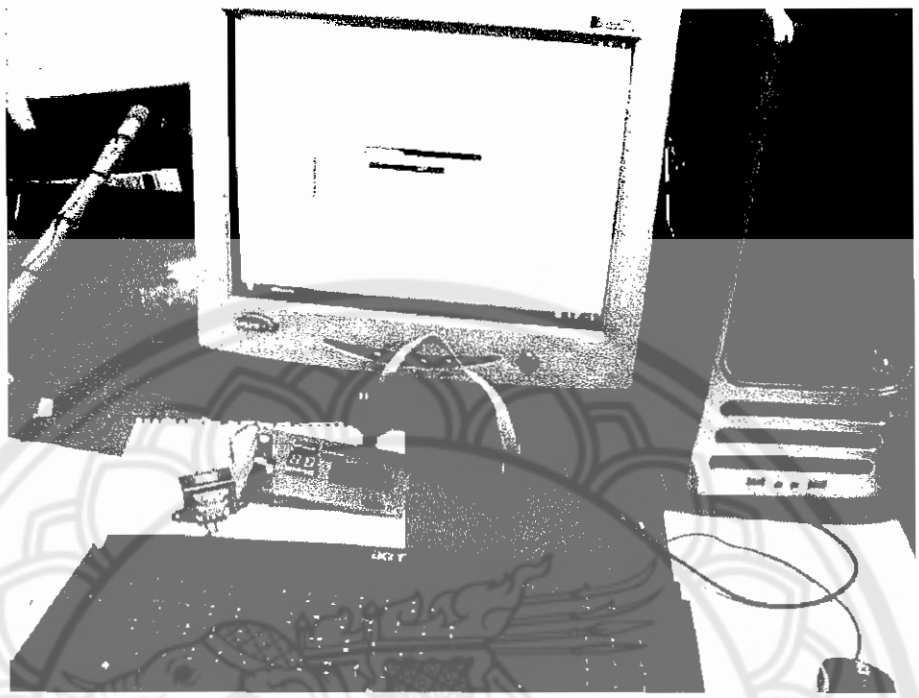


ผ-10 สายเบิร์นโปรแกรม



ผ-11 เชื่อมต่อสายเบิร์นโปรแกรมกับบอร์ดควบคุมและแสดงผล

10. ทำการโหลดข้อมูลลงบอร์ดควบคุมและแสดงผล



ผ-12 โหลดโปรแกรมลงบอร์ด

11. หลังจากโหลดโปรแกรมเสร็จแล้ว ก็สามารถนำบอร์ดไปใช้งานต่อได้เลย

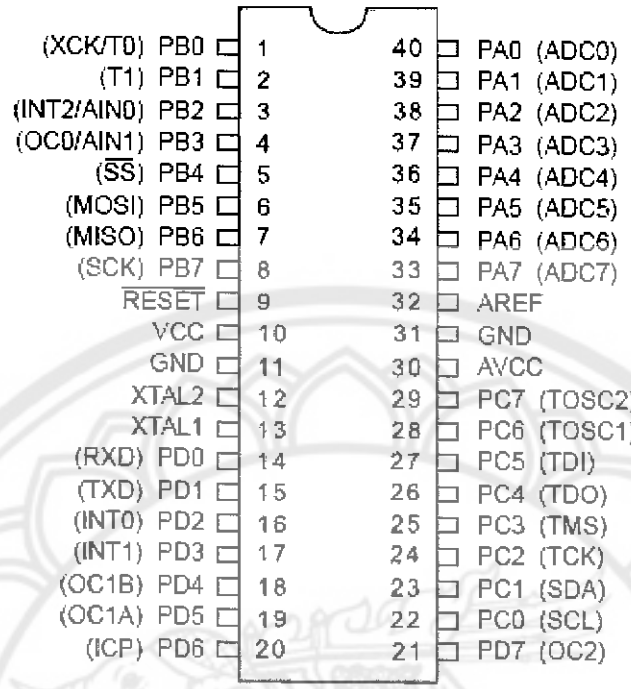
ภาคผนวก ค.

รายละเอียดของอุปกรณ์อิเล็กทรอนิกส์

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 32K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 1024 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 2K Byte Internal SRAM
  - 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 MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega32L
  - 4.5 - 5.5V for ATmega32
- Speed Grades
  - 0 - 8 MHz for ATmega32L
  - 0 - 16 MHz for ATmega32
- Power Consumption at 1 MHz, 3V, 25°C for ATmega32L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA

PDIP



TQFP/MLF

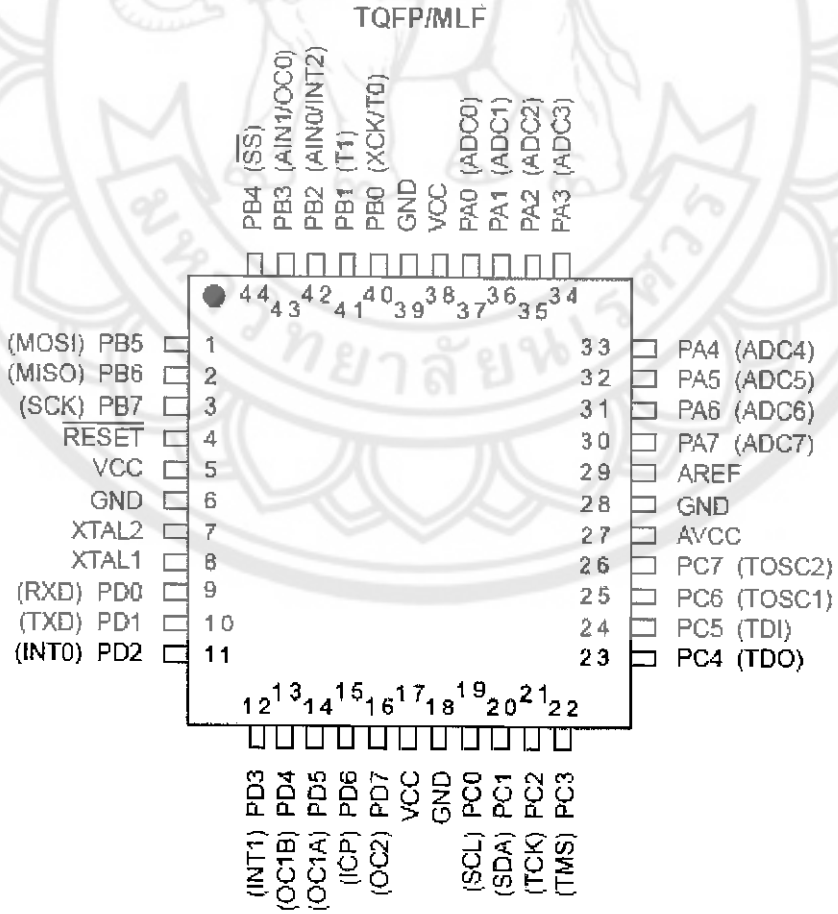
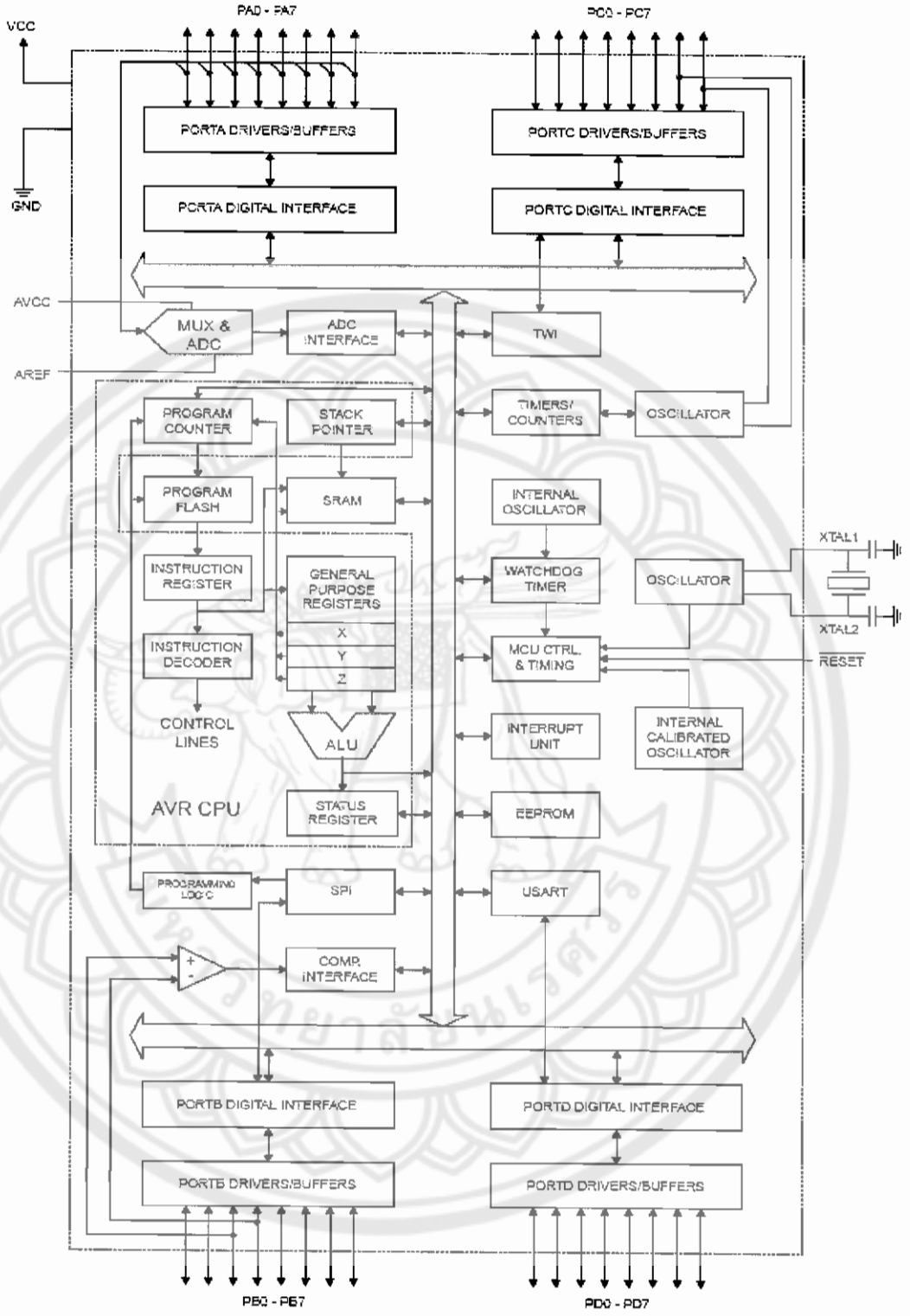


Figure 2. Block Diagram



Freescale Semiconductor
Technical Data

MPX5100
Rev 10, 05/2005

Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated, and Calibrated

MPX5100/MPXV5100 SERIES

INTEGRATED PRESSURE SENSOR
0 to 100 kpa (0 to 14.5 psi)
15 to 115 kPa
(2.2 to 16.7 psi)
0.2 to 4.7 V Output

The MPX5100 series piezoresistive transducer is a state-of-the-art monolithic silicon pressure sensor designed for a wide range of applications, but particularly those employing a microcontroller or microprocessor with A/D inputs. This patented, single element transducer combines advanced micromachining techniques, thin-film metallization, and bipolar processing to provide an accurate, high level analog output signal that is proportional to the applied pressure.

Features

- 2.5% Maximum Error over 0° to 85°C
- Ideally suited for Microprocessor or Microcontroller-Based Systems
- Patented Silicon Shear Stress Strain Gauge
- Available in Absolute, Differential and Gauge Configurations
- Durable Epoxy Unibody Element
- Easy-to-Use Chip Carrier Option

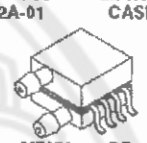
Typical Applications

- Patient Monitoring
- Process Control
- Pump/Motor Control
- Pressure Switching

SMALL OUTLINE PACKAGES

MPXV5100GC6U
CASE 482A-01

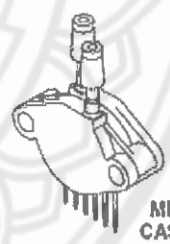
MPXV5100GC7U
CASE 482C-03



MPXV5100DP
CASE 1351-01

ORDERING INFORMATION

Device Type	Options	Case No.	MPX Series Order Number	Device Marking
UNIBODY PACKAGE (MPX5100 SERIES)				
Basic Elements	Absolute	867	MPX5100A	MPX5100A
	Differential	867	MPX5100D	MPX5100D
Ported Elements	Differential Dual Ports	867C	MPX5100DP	MPX5100DP
	Absolute, Single Port	867B	MPX5100AP	MPX5100AP
	Gauge, Single Port	867B	MPX5100GP	MPX5100GP
	Gauge, Axial PC Mount	867F	MPX5100GSX	MPX5100D
	Gauge, Axial Port, SMT	482A	MPXV5100GC6U	MPXV5100G
	Gauge, Axial Port, DIP	482C	MPX5V100GC7U	MPXV5100G
	Gauge, Dual Port, SMT	1351	MPXV5100DP	MPXV5100



MPX5100DP
CASE 867C-05

PIN NUMBER(1)			
1	N/C	5	N/C
2	V_S	6	N/C
3	GND	7	N/C
4	V_{OUT}	8	N/C

1. Pins 1, 5, 6, 7, and 8 are internal device connections. Do not connect to external circuitry or ground. Pin1 is noted by the notch in the lead.

PIN NUMBER(1)			
1	V_{OUT}	4	N/C
2	GND	5	N/C
3	V_S	6	N/C

1. Pins 4, 5, and 6 are internal device connections. Do not connect to external circuitry or ground. Pin 1 is noted by the notch in the lead.

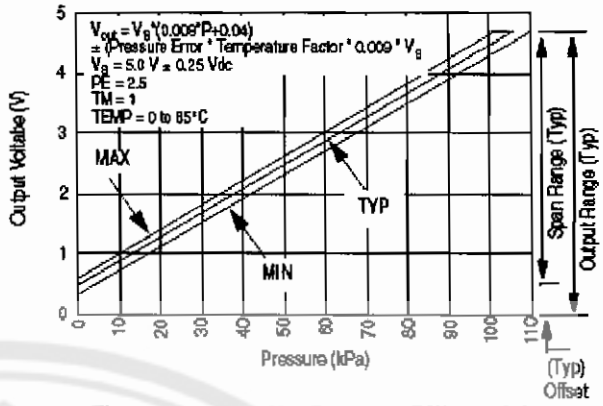


Figure 2. Output Vs. Pressure Differential

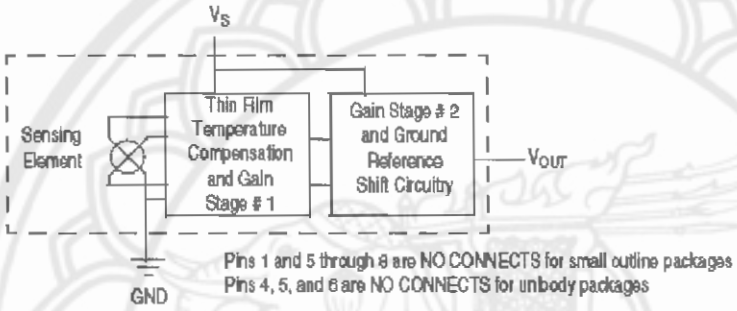


Figure 1. Fully Integrated Pressure Sensor Schematic

Figure 1. Fully Integrated Pressure Sensor Schematic

TABLE 1. Maximum Ratings(1)

Rating	Symbol	Value	Unit
Maximum Pressure (P1 > P2)	P_{MAX}	400	kPa
Storage Temperature	T_{STG}	-40° to +125°C	°C
Operating Temperature	T_A	-40° to +125°C	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

TABLE 2. Operating Characteristics (V_S = 5.0 V_{DC}, T_A = 25°C unless otherwise noted, P1 > P2. Decoupling circuit shown in Figure 4 required to meet electrical specifications.)

Characteristic		Symbol	Min	Typ	Max	Unit
Pressure Range[1] Gauge, Differential: MPX5100D/MPX5100G/MPXV5100G Absolute: MPX5100A		P_{OP}	 0 15	 — —	 100 115	kPa
Supply Voltage[2]		V_S	4.75	5.0	5.25	V_{DC}
Supply Current		I_O	—	7.0	10	mAdc
Minimum Pressure Offset[3] @ V_S = 5.0 V	(0 to 85°C)	V_{OFF}	0.088	0.20	0.313	V_{DC}
Full Scale Output[4] @ V_S = 5.0 V	Differential and Absolute (0 to 85°C)	V_{FSO}	4.587	4.700	4.813	V_{DC}
Full Scale Span[5] @ V_S = 5.0 V	Differential and Absolute (0 to 85°C)	V_{FSS}	—	4.500	—	V_{DC}
Accuracy[6]		—	—	—	±2.5	$\%V_{FSS}$
Sensitivity		V/P	—	45	—	mV/kPa
Response Time[7]		t_R	—	1.0	—	ms
Output Source Current at Full Scale Output		I_{O+}	—	0.1	—	mAdc
Warm-Up Time[8]		—	—	20	—	ms
Offset Stability[9]		—	—	±0.5	—	$\%V_{FSS}$

1. 0.1 kPa (kiloPascal) equals 0.145 psi.
2. Device is ratiometric within this specified excitation range.
3. Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.
4. Full Scale Output (V_{FSO}) is defined as the output voltage at the maximum or full rated pressure.
5. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
6. Accuracy (error budget) consists of the following:
   - Linearity: Output deviation from a straight line relationship with pressure over the specified pressure range.
   - Temperature Hysteresis: Output deviation at any temperature within the operating temperature range, after the temperature is cycled to and from the minimum or maximum operating temperature points, with zero differential pressure applied.
   - Pressure Hysteresis: Output deviation at any pressure within the specified range, when this pressure is cycled to and from minimum or maximum rated pressure at 25°C.
   - TcSpan: Output deviation over the temperature range of 0° to 85°C, relative to 25°C.
   - TcOffset: Output deviation with minimum pressure applied over the temperature range of 0° to 85°C, relative to 25°C.
   - Variation from Nominal: The variation from nominal values, for Offset or Full Scale Span, as a percent of V_{FSS} at 25°C.

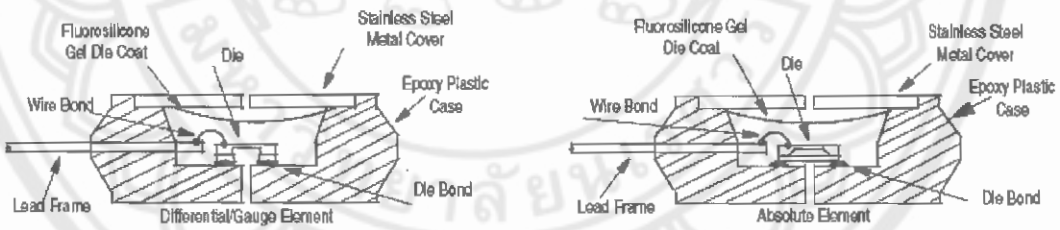


Figure 3. Cross Sectional Diagrams (Not to Scale)

Figure 4 shows the recommended decoupling circuit for interfacing the output of the integrated sensor to the A/D input of a microprocessor or microcontroller. Proper decoupling of the power supply is recommended.

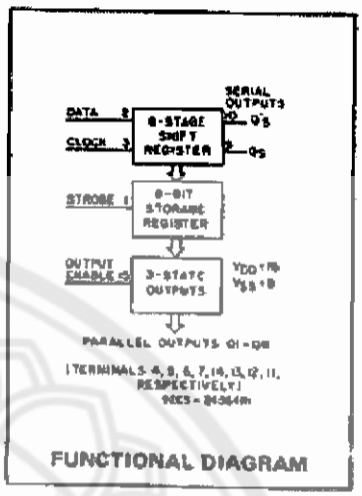
TEXAS INSTRUMENTS

CD4094B Types

Data sheet acquired from Harris Semiconductor
SCHS053

CMOS
8-Stage Shift-and-Store Bus Register

High-Voltage Types (20-Volt Rating)



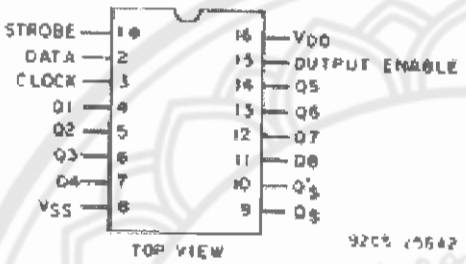


Fig. 1 – Terminal assignment.

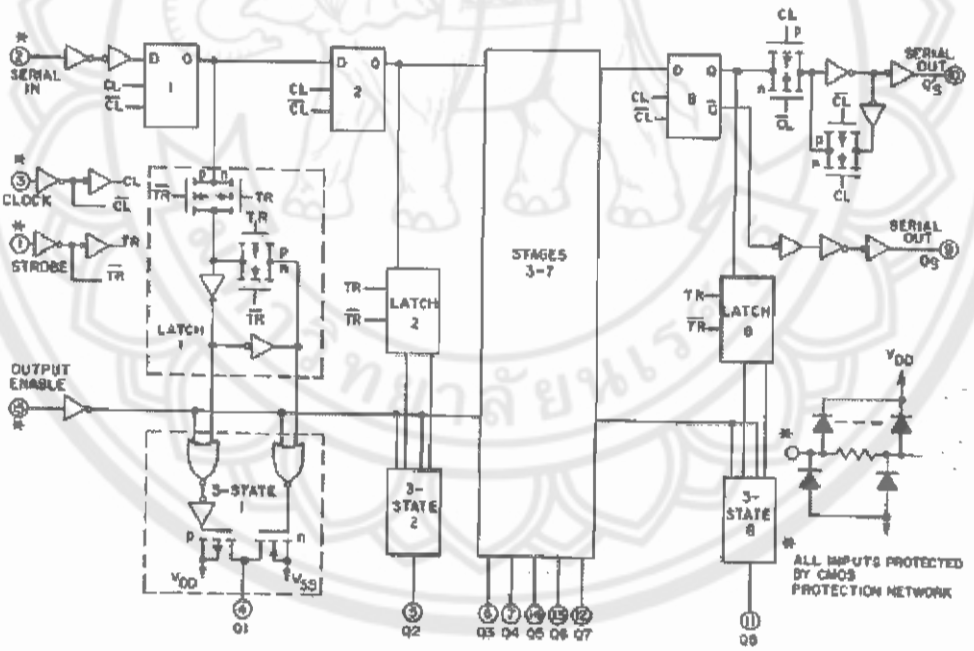


Fig. 2 – CD4094B Logic diagram.

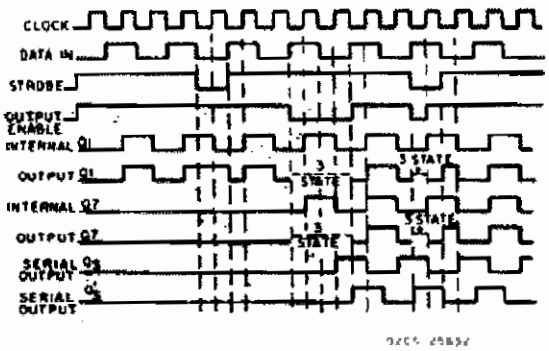


Fig. 3 – Timing diagram.

TRUTH TABLE

CL*	Output Enable	Strobe	Data	Parallel Outputs		Serial Outputs	
				Q1	QN	Q8*	Q'8
	0	X	X	OC	OC	Q7	NC
	0	X	X	OC	OC	NC	Q7
	1	0	X	NC	NC	Q7	NC
	1	1	0	0	Q_{N-1}	Q7	NC
	1	1	1	1	Q_{N-1}	Q7	NC
	1	1	1	NC	NC	NC	Q7

* = Level Change
X = Don't Care
NC = No Change
OC = Open Circuit

Logic 1 ≡ High
Logic 0 ≡ Low

CHARACTERISTIC	V_{DD} (V)	LIMITS		UNITS
		MIN.	MAX.	
Supply-Voltage Range (For T_A=Full Package-Temperature Range)		3	18	V
Data Setup Time, t_S	5 10 15	125 55 35	– – –	ns
Clock Pulse Width, t_W	5 10 15	200 100 83	– – –	ns
Clock Input Frequency, f_{CL}	5 10 15	dc	1.25 2.5 3	MHz
Clock Input Rise or Fall time, t_rCL, t_fCL:*	5 10 15	–	15 5 5	μs
Strobe Pulse Width, t_W	5 10 15	200 80 70	– – –	ns