สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมไฟอัตโนมัติ

นาย พีรเชษฐ์ สันติเวส
นาย สืบพงศ์ เจิดจำนงวิทยากุล

วพ.
พ7913
2550

เลขหมู่.....
เลขทะเบียน.....81618
วัน,เดือน,ปี.....19 มิ.ย. 2551

b.11933677
i.....

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Automatic Light Controller

Mr. Peerachet Suntiwes

Mr. Suebpong Juebjumnongwittayakun

A Special Project Submitted in Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Monkut's Institute of Technology Ladkrabang

Academic Year 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง	ระบบควบคุมไฟอัตโนมัติ
นักศึกษา	นาย พีรเชษฐ์ สันติเวส
	นาย สืบพงศ์ เจิดจำนงวิทยากุล
ภาควิชาฟิสิกส์	ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**
สาขาวิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	รศ. วิชิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

คณะกรรมการตรวจสอบ		ลายมือชื่อ
ประธานกรรมการ	ผศ.สาหร่าย เล็กชะอุ่ม	
กรรมการ	อ.กาจปัญญา สุวรรณสุโข	
กรรมการ	ดร.ประธาน บุรณศิริ	
กรรมการที่ปรึกษา	รศ.วิชิต ศิริโชติ	

.....

(รองศาสตราจารย์วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษ ระบบควบคุมไฟอัตโนมัติ
นักศึกษา นาย พีรเชษฐ์ สันติเวส
นาย สืบพงศ์ เจิดจำนงวิทยากุล
ภาควิชา ฟิสิกส์ประยุกต์ **คณะวิทยาศาสตร์**
สาขาวิชา ฟิสิกส์ประยุกต์
ปีการศึกษา 2550
อาจารย์ที่ปรึกษา รศ.วิชิต ศิริโชติ

บทคัดย่อ

ได้สร้างเครื่องมือเพื่อช่วยประหยัดพลังงานไฟฟ้าส่องสว่างสำหรับบ้านพักอาศัย วงจรนี้สร้างขึ้นด้วยไมโครคอนโทรลเลอร์ชนิด AVR MEGA128 และ เซนเซอร์อินฟราเรด ซอฟแวร์ที่นำมาใช้พัฒนาคือ Codevision c compiler ไมโครคอนโทรลเลอร์จะทำการตรวจจับการ เข้า-ออกของบุคคล โดยโหมดการใช้งานสามารถเลือกได้ 2 แบบ คือโหมดการตรวจจับด้วยเซนเซอร์อินฟราเรด และโหมดตั้งเวลาเปิด-ปิด เอาท์พุตมี 4 ช่องสร้างด้วย รีเลย์ขนาด 10 แอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Automatic Light Controller

Name Mr. Peerachet Suntiwes

Mr. Suebpong Juebjumnongwittayakun

Department Applied Physics **Faculty of Science**

Program Applied Physics

Academic Year 2007

Special Project Advisor Wichit Sirichote

ABSTRACT

A device used for energy saving of the home lighting has been developed. The circuit is built with the AVR MEGA604 microcontroller and the Infrared sensor. The software is developed using Codevision c compiler. The microcontroller detects the number of people coming in or going out from the room. The selectable control modes include infrared sensing or preset time ON/OFF scheduler. The output provides 4-channel 10A electromechanical relay.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.วิชิต ศิริโชติ ที่แนะนำ ชี้แนะ และช่วยเหลือ ซึ่งเป็นส่วนสำคัญของโครงงานพิเศษนี้ ที่ทำให้สามารถสำเร็จลุล่วงไปได้ด้วยดี ขอขอบพระคุณ ท่าน Google และท่าน Wikipedia ที่เป็นอาจารย์ อีกท่านหนึ่งในยามที่สับสน ขอบคุณเพื่อนๆในห้องโปรเจค ที่ช่วยเหลือในการหาข้อมูล การแก้ไขส่วนต่างๆร่วมกัน และขอพระคุณผู้สนับสนุนทุกท่าน ทั้งที่เป็นทางการ และ ไม่เป็นทางการทั้งหมด

นาย พีรเชษฐ์ สันติเวส

นาย สืบพงศ์ เจิดจำนงวิทยากุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	จ
สารบัญรูป	ฉ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและเหตุจูงใจของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 ประโยชน์และผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 มาทำความรู้จักกับไมโครคอนโทรลเลอร์ AVR	3
2.2 หลักการทำงานของ IR Sensor	16
2.3 การคิดค่าไฟฟ้า	17
บทที่ 3 วิธีการดำเนินการวิจัย	19
3.1 หลักการอย่างง่าย	19
บทที่ 4 การทดลอง และผลการทดลอง	22
4.1 การทดลอง ทดสอบโครงงานพิเศษควบคุมไฟอัตโนมัติ	22
บทที่ 5 สรุปผล และข้อเสนอแนะ	30
5.1 สรุป	31
5.2 ข้อเสนอแนะ และแนวทางในการพัฒนา	31
เอกสารอ้างอิง	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 รีเซตและอินเตอร์รัปต์เวกเตอร์	7
ตารางที่ 2.2 รีจิสเตอร์ MCUCSR	9
ตารางที่ 2.3 รีจิสเตอร์ GICR	9
ตารางที่ 2.4 รีจิสเตอร์ GIFR	9
ตารางที่ 2.5 รีจิสเตอร์ GIFR	10
ตารางที่ 2.6 รีจิสเตอร์ TCCR0	12
ตารางที่ 2.7 บิตที่กำหนดรูปแบบสัญญาณของไทเมอร์/เคาน์เตอร์	12
ตารางที่ 2.8 ค่า BOTTOM, MAX และ TOP	13
ตารางที่ 2.9 บิตกำหนดแหล่งสัญญาณนาฬิกาของไทเมอร์ 0	13
ตารางที่ 2.10 รีจิสเตอร์ TIFR	13
ตารางที่ 2.11 รีจิสเตอร์ SFIOR	15
ตารางที่ 2.12 อัตรารายเดือน : ค่าพลังงานไฟฟ้า	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 บล็อกไดอะแกรม AVR (ATmega 128)	3
รูปที่ 2.2 ขาพอร์ต AVR (ATmega 128)	4
รูปที่ 2.3 การทำงานของอินเตอร์รัปต์	6
รูปที่ 2.4 บล็อกไดอะแกรม 8 บิต ไทเมอร์/เคาน์เตอร์0	10
รูปที่ 2.5 ปรีสเกลเลอร์ไทเมอร์/เคาน์เตอร์ 0 และ 1	14
รูปที่ 2.6 ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ ไม่มีกำหนดปรีสเกลเลอร์	15
รูปที่ 2.7 ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ กำหนดปรีสเกลเลอร์เท่ากับ	15
รูปที่ 2.8 การสะท้อนของ IR Sensor	16
รูปที่ 2.9 วงจรภาคส่ง - วงจรภาครับ ของ IR Sensor	16
รูปที่ 3.1 วงจรแสงสว่างแบบหลอดไส้	19
รูปที่ 3.2 วงจรแสงสว่างแบบหลอดฟูออเรสเซ็น	19
รูปที่ 3.3 วงจร 7Segment LED Display	20
รูปที่ 3.4 วงจร Relay	20
รูปที่ 3.5 วงจร รับ/ส่ง Infrared	21
รูปที่ 3.6 วงจร Switch	21
รูปที่ 3.7 วงจร LCD Connector	21
รูปที่ 4.1 การต่อวงจรทดลอง	23
รูปที่ 4.2 การใช้ Scope จับวัดที่ IR Sensor	23
รูปที่ 4.3 การใช้ Multimeter วัดส่วน Input ของ Relay	24
รูปที่ 4.4 การต่อไฟบวกเข้าที่ Input ของ Relay	24
รูปที่ 4.5 การต่อไฟบวกเข้าที่ Input ของ Relay	24
รูปที่ 4.6 การต่อชุดไฟทดสอบ	25
รูปที่ 4.7 ผลจาก Scope จับที่ Sensor ตัวที่ 1 และ 2 ขณะยังไม่มีวัตถุผ่าน	25
รูปที่ 4.8 นำวัตถุมาบังที่ Sensor แทนการเดินผ่านของคน	25
รูปที่ 4.9 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1	26
รูปที่ 4.10 ผลจาก Scope จับที่ Sensor ตัวที่ 1 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1	26
รูปที่ 4.11 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1 และ 2	27
รูปที่ 4.12 ผลจาก Scope จับที่ Sensor ตัวที่ 1 และ 2 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1 และ 2	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

รูปที่ 4.13 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 2 28
รูปที่ 4.14 ผลจาก Scope จับที่ Sensor ตัวที่ 2 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 2 28
รูปที่ 4.15 เมื่อวัตถุผ่าน Sensor ตัวที่ 1 และ 2 ตามลำดับ 29
รูปที่ 4.16 เมื่อวัตถุผ่าน Sensor ตัวที่ 1 และ 2 29
รูปที่ 4.17 ชุดไฟทดสอบจะสว่างขึ้น 31
รูปที่ 4.18 ชุดไฟทดสอบที่ได้รับคำสั่งเปิด จากโครงงานพิเศษควบคุมไฟอัตโนมัติ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1. ความเป็นมาและเหตุจูงใจของโครงงาน

ในชีวิตประจำวันของคนเรานั้น ใช้แสงสว่างในการมองเห็นและดำเนินชีวิต แสงสว่างที่เราใช้นั้นได้มาจากแหล่งกำเนิดแสงจากธรรมชาติ และแหล่งกำเนิดแสงจากสิ่งที่มนุษย์ประดิษฐ์คิดค้นขึ้น โดยแสงที่ได้จากสิ่งประดิษฐ์ของมนุษย์นั้น ต้องอาศัยพลังงานจากไฟฟ้าเปลี่ยนเป็นแสงสว่าง

การกำเนิดไฟฟ้าเพื่อนำไปใช้ในการกำเนิดแสงนั้นต้องอาศัยการกำเนิดพลังงานไฟฟ้าจากสิ่งอื่น เช่น การสร้างเขื่อนเพื่อนำเอาพลังงานศักย์ของน้ำเปลี่ยนเป็นพลังงานกลไปหมุนเครื่องกำเนิดไฟฟ้า การเผาไหม้น้ำมันในการจุดระเบิดเกิดพลังงานความร้อนเป็นพลังงานกลในการหมุนเครื่องปั่นไฟ การใช้ถ่านหินเผาได้ความร้อนไปต้มน้ำเพื่อเกิดแรงดันไปหมุนเครื่องกำเนิดไฟฟ้า จนถึงการใช้พลังงานนิวเคลียร์เป็นต้น ซึ่งพลังงานไฟฟ้าที่สร้างจากสิ่งเหล่านี้ล้วนเป็นการสิ้นเปลืองทรัพยากรธรรมชาติเป็นอย่างมาก และที่สำคัญทำให้เกิดมลพิษต่อสิ่งแวดล้อมของโลก

ด้วยเหตุนี้จึงมีความคิดที่จะคิดค้นอุปกรณ์เพื่อช่วยในการทำให้เราใช้พลังงานจากไฟฟ้าน้อยลง และประหยัดพลังงานไฟฟ้าได้ โดยควบคุมระบบแสงสว่างในบ้านหรืออาคาร
โดยจะ เปิด - ปิด ตามเวลาที่เรากำหนดอัตโนมัติ หรือ เปิด – ปิด เมื่อเรา เข้า – ออก จากห้องที่ติดตั้งซึ่งจะเป็นผลทำให้การบริโภคไฟฟ้าน้อยลง และประหยัดค่าใช้จ่ายได้ระยะยาวในระดับบ้านเรือน ระดับองค์กร ระดับประเทศ และระดับโลก ในที่สุด

1.2. วัตถุประสงค์ของโครงงาน

- เพื่อประหยัดพลังงานไฟฟ้า และส่งผลไปถึงการประหยัดพลังงานที่ใช้กำเนิดไฟฟ้า
- เพื่อความสะดวกในการจัดการระบบแสงสว่างทั้งในและนอกอาคาร
- เป็นระบบควบคุมระบบแสงสว่างราคาถูก
- สามารถนำไปติดตั้งได้ง่ายในทุกครัวเรือน หรือ องค์กร
- สามารถประหยัดค่าใช้จ่ายในระยะยาว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3. ขอบเขตของโครงงาน

ขอบเขตของโครงงานนี้ คือ สร้างชุดควบคุม/ระบบควบคุม ไฟส่องสว่างตั้งแต่ 1 จุดหรือมากกว่า โดยสามารถสั่ง เปิด-ปิด ไฟส่องสว่างอัตโนมัติ หรือกำหนดเวลาโดยผู้ใช้ และสามารถนำไปประยุกต์ใช้เพิ่มกับระบบอื่นๆได้ในอนาคต

1.4. ประโยชน์และผลที่คาดว่าจะได้รับ

ประโยชน์กับตัวผู้ทำ

- สามารถพัฒนาทักษะการคิด ประดิษฐ์ และวางแผน
- สามารถออกแบบวงจร และสร้างวงจรที่เกี่ยวข้องได้
- เพิ่มทักษะในการเขียนโปรแกรม ในการควบคุม AVR Controller

ประโยชน์กับโครงงาน

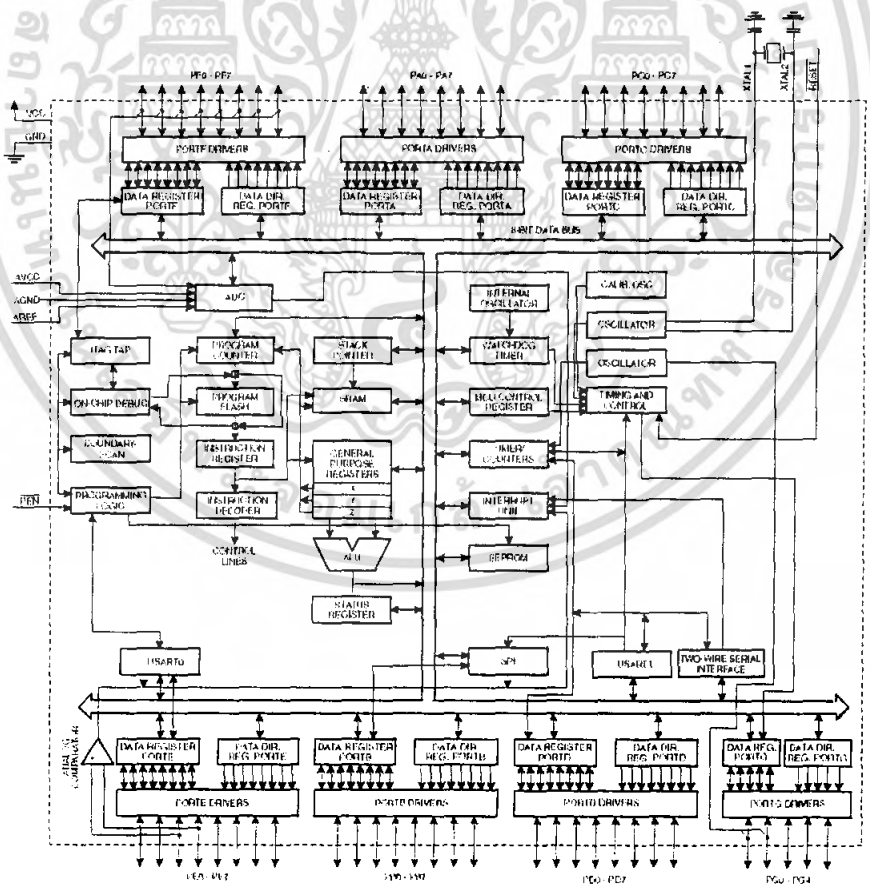
- ใช้งานได้จริง
- ช่วยประหยัดค่าไฟฟ้าตั้งแต่ระดับครัวเรือน องค์กร จนถึงประเทศ
- ลดปัญหาการสิ้นเปลืองไฟฟ้า
- ลดปัญหาด้านพลังงาน - โลกร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 มาทำความรู้จักกับไมโครคอนโทรลเลอร์ AVR

ไมโครคอนโทรลเลอร์ AVR นั้นผลิตโดยบริษัท ATMEL ซึ่งจัดเป็นไมโครคอนโทรลเลอร์ตระกูลใหม่จาก ATMEL มีสถาปัตยกรรมแบบ RISC คือ หนึ่งคำสั่งทำงานใช้สัญญาณนาฬิกาเพียง 1 ลูก เป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูง แบ่งออกเป็นหลายอนุกรม และแต่ละอนุกรมก็แบ่งออกเป็นหลายเบอร์ โดยการทดลองนี้เราจะใช้ไมโครคอนโทรลเลอร์ AVR เบอร์ ATmega128 รายละเอียดและคุณสมบัติภายในไมโครคอนโทรลเลอร์ ATmega128 แสดงดังไดอะแกรมรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรม AVR (ATmega 128)

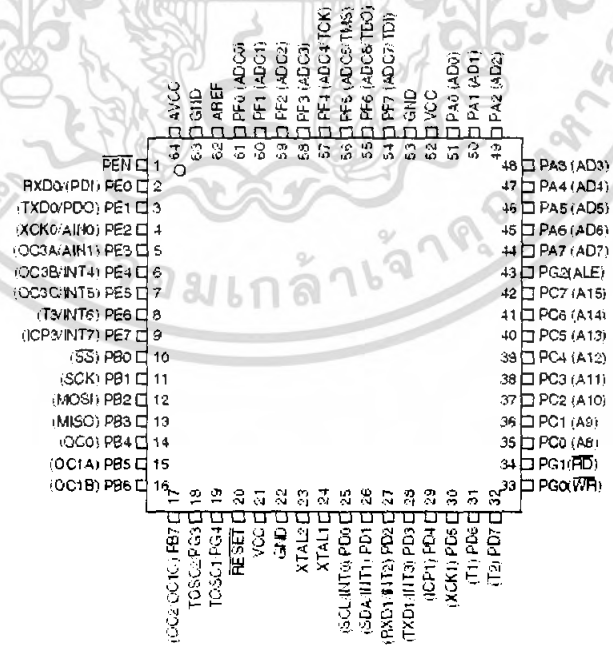
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติที่สำคัญ

- มีหน่วยความจำ flash สำหรับเขียนโปรแกรม 128K Bytes และมี RAM 4K Bytes
- มีหน่วยความจำข้อมูลถาวรแบบ EEPROM ขนาด 4K Bytes และสามารถลบและเขียนซ้ำได้กว่า 100,000 ครั้ง
- จำนวน I/O สูงสุดถึง 53 I/O Pins
- มีวงจรสื่อสาร SPI จำนวน 1 ช่อง , I^2C จำนวน 1 ช่อง Programmable Serial USARTs จำนวน 2 ช่อง
- มี ADC ขนาด 10-Bit จำนวน 8 ช่อง
- มี Timers/Counters 8-Bit จำนวน 2 ช่อง ,Timers/Counters 16-Bit จำนวน 2 ช่อง ,8-Bit PWM 2 ช่อง ,Watchdog Timer ,Real Time Counter

ขาพอร์ตอินพุทเอาต์พุต

ขาพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ ATmega 128 มีจำนวน 64 ขา โดยรายละเอียดขาพอร์ตทั้งหมดแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 ขาพอร์ต AVR (ATmega 128)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียดในแต่ละขาพอร์ตดังนี้

- VCC ขาแรงดันไฟตรง
- GND ขากราวด์
- Port A (PA7-PA0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และสามารถกำหนดใช้งานเป็นพอร์ตอินพุตสัญญาณอนาลอก(A/D Converter)ได้
- Port B (PB7-PB0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาสำหรับโปรแกรมชิพ ขาป้อนสัญญาณนาฬิกาภายนอก เป็นต้น
- Port C (PC7-PC0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) สามารถเป็นอินเตอร์เฟสกับ หน่วยความจำภายนอก
- Port D (PD7-PD0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก เป็นต้น
- Port E (PE7-PE0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และสามารถทำหน้าที่เป็น Timer/counter และ UART
- Port F (PF7-PF0) ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอลกำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษ เช่น ขาเชื่อมต่อกับดีบักและโปรแกรมด้วยการเชื่อมต่อแบบ JTAG เป็นต้น
- Port G (PG4-PG0) ขาพอร์ตเป็นอินพุตเอาท์พุตขนาด 5 บิต มี internal pull-up registers
- RESET ขารีเซตวงจร
- XTAL1 ขาต่อคริสตอลออสซิเลเตอร์ ช่องที่ 1 ด้านอินพุต
- XTAL2 ขาต่อคริสตอลออสซิเลเตอร์ ช่องที่ 2 ด้านเอาต์พุต
- AVCC ขาแรงดันพอร์ต A และโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล
- AREF ขาแรงดันอนาลอกอ้างอิงสำหรับโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล
- PEN ขาสำหรับ enable โหมด SPI Serial Programming โดยปรกติขานี้จะเป็น high แต่เมื่อเป็น low จะไปเปิดการทำงานโหมด SPI Serial Programming

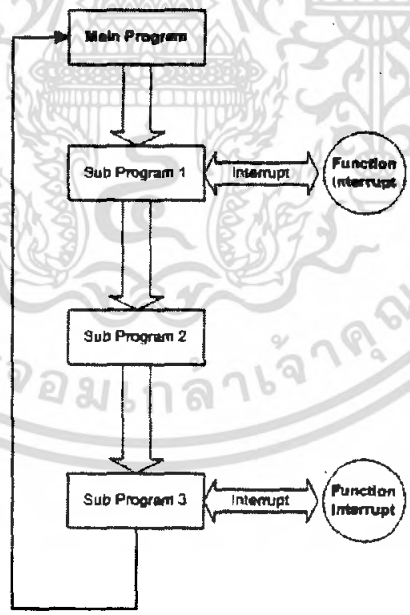
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์รัปต์

การอินเตอร์รัปต์(Interrupt) คือ กระบวนการทำงานของโปรแกรมปกติหรือ โปรแกรมหลักที่กำลังทำงานอยู่เพื่อให้เปลี่ยนมาทำงานในส่วนของโปรแกรมที่ได้กำหนดไว้ในอินเตอร์รัปต์ แสดงดังรูปที่ 2.3 กระบวนการนี้จะช่วยให้ประหยัดเวลาในการทำงานของโปรแกรมหลักและลด การผิดพลาดในการตรวจสอบเงื่อนไขการทำงานของโปรแกรม เพราะไม่ต้องคอยตรวจสอบเงื่อนไขใดเงื่อนไขหนึ่งขณะที่โปรแกรมกำลังทำงานอยู่ตลอด โดยสามารถกำหนดหน้าที่การตรวจสอบนี้ให้กับอินเตอร์รัปต์แทน ประเภทของการอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ จะแบ่งออกเป็น 2 ประเภท คือ

1.การอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก (External Interrupt) เป็นการตรวจสอบสัญญาณที่ รับมา จากภายนอกตัวไมโครคอนโทรลเลอร์

2.การอินเตอร์รัปต์เนื่องจากสัญญาณภายใน (Internal Interrupt) โดยแหล่งกำเนิดสัญญาณนี้จะเกิดจากวงจรภายในของไมโครคอนโทรลเลอร์เอง เช่น การอินเตอร์รัปต์จากสัญญาณการเกิดโอเวอร์โฟลว์ของไทเมอร์ การอินเตอร์รัปต์จากากรรับ/ส่งข้อมูล การอินเตอร์รัปต์เนื่องจากการอ่านเขียนหน่วยความจำ EEPROM เป็นต้น



รูปที่ 2.3 การทำงานของอินเตอร์รัปต์

จากรูปโปรแกรมหลักจะวนทำงานตลอด เมื่อเกิดอินเตอร์รัปต์ที่ส่วนหนึ่งส่วนใดของโปรแกรมย่อยจะกระโดดไปทำงาน ในส่วนของโปรแกรมบริการอินเตอร์รัปต์ก่อน เมื่อทำงานเสร็จแล้วจะกลับมาทำงานที่ตำแหน่งเดิมของโปรแกรมย่อยที่เกิดอินเตอร์รัปต์ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเตอร์รัปต์ไมโครคอนโทรลเลอร์ AVR

ตารางที่ 2.1 รีเซตและอินเตอร์รัปต์เวกเตอร์

หมายเลขเวกเตอร์	แอดเดรสอินเตอร์รัปต์	ที่มาของอินเตอร์รัปต์	รายละเอียดของอินเตอร์รัปต์
1	$000	RESET	• ขารีเซตภายนอก • เพาเวอร์ออนรีเซต,เบราเอาต์รีเซต • วอตด็อกซ์ไทเมอร์รีเซต • JTAG AVR รีเซต
2	$002	INT0	อินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ช่องที่ 0
3	$004	INT1	อินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ช่องที่ 1
4	$006	TIMER2 COMP	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 2 เปรียบเทียบตรงกัน
5	$008	TIMER2 OVF	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 2 เกิดโอเวอร์โฟลว์
6	$00A	TIMER1 CAPT	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 1 เกิดการสัญญาณได้
7	$00C	TIMER1 COMPA	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 1เปรียบเทียบตรงกัน แบบ A
8	$00E	TIMER1 COMPB	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 1เปรียบเทียบตรงกัน แบบ B
9	$010	TIMER1 OVF	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 1 เกิดโอเวอร์โฟลว์
10	$012	TIMER0 OVF	อินเตอร์รัปต์เนื่องจากไทเมอร์/เคาน์เตอร์ 0 เกิดโอเวอร์โฟลว์
11	$014	SPI,STC	การส่งข้อมูลอนุกรมเสร็จสมบูรณ์
12	$016	USART,RXC	การรับข้อมูลจากโมดูล USART เสร็จสมบูรณ์
13	$018	USART,UDRE	รีจิสเตอร์ข้อมูลโมดูล USART ว่าง
14	$01A	USART TXC	การส่งข้อมูลจากโมดูล USART เสร็จสมบูรณ์
15	$01C	ADC	โมดูล ADC แปลงข้อมูลเสร็จสมบูรณ์
16	$01E	EE_RDY	EEPROM ทำงานเสร็จแล้ว
17	$020	ANA_COMP	เกิดการเปรียบเทียบแรงดันอะนาลอก
18	$022	TWI	การรับส่งข้อมูลด้วยสายสัญญาณ 2 เส้น
19	$024	INT2	อินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ช่องที่ 2
20	$026	TIMER0 COMP	อินเตอร์รัปต์เนื่องจาก ไทเมอร์/เคาน์เตอร์ 0 เปรียบเทียบตรงกัน
21	$028	SPM_RDY	หน่วยความจำโปรแกรมบันทึกเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากไมโครคอนโทรลเลอร์ AVR ประกอบด้วยโมดูลการทำงานที่หลากหลาย เช่น ADC, EEPROM, Comparator, Capture, Timer เป็นต้น และโมดูลเหล่านี้สนับสนุนการทำงานกับสัญญาณอินเตอร์รัปต์ด้วย ทำให้มีแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ที่เกี่ยวข้องกับโมดูลดังตารางที่ 2.1

เนื้อหานี้จะแสดงเฉพาะอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ส่วนประเภทของอินเตอร์รัปต์เนื่องจากสัญญาณภายในจะพบรายละเอียดได้ในโมดูลที่เกี่ยวข้องในเนื้อหาถัดไป

การอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก

การอินเตอร์รัปต์เนื่องจากสัญญาณภายนอกเป็นการตรวจสอบสัญญาณที่รับมาจากภายนอกตัวไมโครคอนโทรลเลอร์ที่ขาพอร์ต PD2 (INT0), PA3 (INT1) และ PB2 (INT2) โดยสามารถที่จะกำหนดรูปแบบของสัญญาณการเกิดอินเตอร์รัปต์เนื่องจากสัญญาณภายนอกได้หลายรูปแบบ เช่น

- ขณะที่ระดับสัญญาณได้เปลี่ยนแปลงไปเป็นระดับลอจิกต่ำแล้ว (Low level)
- ขณะที่มีการเปลี่ยนแปลงระดับสัญญาณ (Any Logical change)
- ที่ขอบขาลงของสัญญาณ (Falling edge)
- ที่ขอบขาขึ้นของสัญญาณ (Rising edge)

1.รีจิสเตอร์ SREG (Status Register)

รีจิสเตอร์แสดงสถานการณ์ทำงานของไมโครคอนโทรลเลอร์ ทั้งในส่วนของการประมวลผลคำสั่งทางด้านคณิตศาสตร์และลอจิก เช่น บิตแสดงการทด (Carry Flag) บิตแสดงผลลัพธ์เป็นศูนย์ (Zero Flag) เป็นต้น ในรีจิสเตอร์แสดงสถานะนี้ จะมีบิตที่ใช้ในการเอ็นเอเบิล/ดิสเอเบิลการใช้งานอินเตอร์รัปต์โดยรวม

2.รีจิสเตอร์ MCUCR(MCU Control Rrgister)

รีจิสเตอร์ MCUCR บิตที่ 0 ถึง 3 (บิต ISC00, ISC01, ISC10 และ ISC11) บิตที่กำหนดรูปแบบการเกิดสัญญาณอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก ช่องที่ 0 และช่องที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.รีจิสเตอร์ MCUCSR(MCU Control and Status Register)

รีจิสเตอร์ MCUCSR บิตที่ 6 (บิต ISC2) บิตกำหนดรูปแบบสัญญาณอินเตอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 2

ตารางที่ 2.2 รีจิสเตอร์ MCUCSR

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00
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

4.รีจิสเตอร์ GICR(General Interrupt Register)

รีจิสเตอร์ GICR บิตที่ 5 ถึง 7 (บิต INT2, INT0, INT1) บิตเปิดการใช้งานอินเตอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 0 ถึง 2

ตารางที่ 2.3 รีจิสเตอร์ GICR

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	INTF1	INTF0	INTF2	-	-	-	-	-
Read/Write	R/W	R/W	R/W	R	R	R	R	R
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

5.รีจิสเตอร์ GIFR (General Interrupt Flag Register)

ตารางที่ 2.4 รีจิสเตอร์ GIFR

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	JTD	ISC2	-	JTRF	WDRF	EXTRF	EXTRF	PORF
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	X	X	X	X	X

รีจิสเตอร์ที่ประกอบไปด้วยบิตที่แสดงการเกิดอินเตอร์รัปต์เนื่องจากสัญญาณภายนอกช่องที่ 0,1 และ 2 เมื่อเกิดอินเตอร์รัปต์ขึ้นบิตที่เกี่ยวข้องกับอินเตอร์รัปต์จะถูกเซต (มีสถานะลอจิก "1") และเคลียร์เมื่อฟังก์ชันอินเตอร์รัปต์ที่เกี่ยวข้องถูกเรียกใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ GIFR

ตารางที่ 2.5 รีจิสเตอร์ GIFR

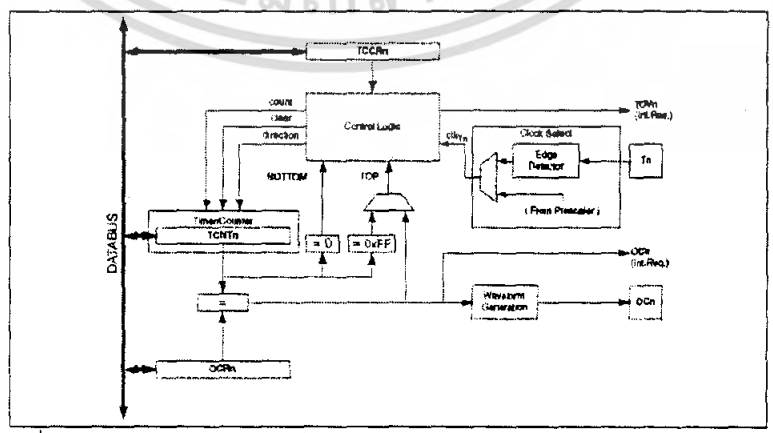
บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	INT1	INT0	INT2	-	-	-	IVSEL	IVCE
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

ไทเมอร์/เคาน์เตอร์0

โมดูล ไทเมอร์ / เคาน์เตอร์ กับสัญญาณ PWM (Pulse Width Modulation) ขนาด 8 บิต (8-bit Timer/Counter0 with PWM) ของ AVR นอกจากคุณสมบัติใช้งานเป็นไทเมอร์หรือเคาน์เตอร์แล้ว ยังมีคุณสมบัติเพิ่มเติมในการสร้างสัญญาณ PWM โดยอาศัยโมดูลการเปรียบเทียบข้อมูลในการสร้างสัญญาณ คุณสมบัติที่สำคัญของไทเมอร์/เคาน์เตอร์0 มีดังนี้

1. ใช้งานเป็นไทเมอร์นับสัญญาณนาฬิกาภายใน หรือเป็นเคาน์เตอร์นับสัญญาณนาฬิกาภายนอกผ่านทางขาพอร์ต PB0 (XCK/T0)
2. ใช้งานเป็นฐานเวลาในโหมดเปรียบเทียบข้อมูลเพื่อสร้างสัญญาณที่ขาพอร์ต PB3 (OC0/AIN 1) เพื่อกำหนดหน้าที่ใดหน้าที่หนึ่ง เช่น เมื่อเปรียบเทียบข้อมูลตรงกันจะสลับสถานะขาพอร์ต (Toggle OC0 on compare match) เมื่อเปรียบเทียบข้อมูลตรงกันจะเซตขาพอร์ต หรือเมื่อเปรียบเทียบข้อมูลตรงกันจะทำการเคลียร์ขาพอร์ต
3. ใช้งานเป็นฐานเวลาในโหมดสร้างสัญญาณ PWM ผ่านทางขาพอร์ต OC0 ในรูปแบบ Fast PWM Mode หรือ Phase Correct PWM Mode

รายละเอียดการทำงานของโมดูล ไทเมอร์/เคาน์เตอร์0 แสดงดังบล็อกไดอะแกรม ดังรูป



รูปที่ 2.4 บล็อกไดอะแกรม 8 บิต ไทเมอร์/เคาน์เตอร์0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 มีรีจิสเตอร์ที่เกี่ยวข้องกับการใช้งานไทเมอร์/เคาน์เตอร์0 ดังนี้

1. รีจิสเตอร์ TCN0 (Timer/Counter0 Register)
   รีจิสเตอร์ที่ใช้การนับขึ้นหรือลงขึ้นอยู่กับการเซตบิตในรีจิสเตอร์ TCCR0
2. รีจิสเตอร์ OCR0 (Output Compare Register)
   รีจิสเตอร์ OCR0 ใช้การกำหนดค่าเพื่อเปรียบเทียบกับค่าในรีจิสเตอร์ TCNT0 เมื่อตรงกันจะให้ผลลัพธ์ตามเงื่อนไขที่กำหนดไว้ในรีจิสเตอร์ TCCR0
3. รีจิสเตอร์ TCCR0 (Timer/Counter Control Register)
   รีจิสเตอร์กำหนดแหล่งสัญญาณนาฬิกาที่ใช้และกำหนดในโหมดทำงาน
4. รีจิสเตอร์ TIMSK (Timer/Counter Interrupt Mark Register)
   รีจิสเตอร์เอ็นเอเบิลการใช้งานอินเตอร์รัปต์ของไทเมอร์/เคาน์เตอร์0 โดยกำหนดเงื่อนไขการเกิดอินเตอร์รัปต์ไว้ 2 รูปแบบ คือ
   - บิต OCIE0 (Timer/Counter0 Output Compare Match Interrupt Enable)
     เอ็นเอเบิลการใช้งานอินเตอร์รัปต์เนื่องจากเปรียบเทียบข้อมูลได้ตรงกัน
   - บิต TOIE0 (Timer/Counter0 Overflow Interrupt Enable)
     เอ็นเอเบิลการใช้งานอินเตอร์รัปต์การเกิดโอเวอร์โฟลว์เนื่องจากการนับ
5. รีจิสเตอร์ TIFR (Timer/Counter Interrupt Flag Register)
   รีจิสเตอร์แสดงการเกิดอินเตอร์รัปต์ของไทเมอร์/เคาน์เตอร์0
   - แฟลก OCF0 (Output Compare Flag 0) ถูกเซตเมื่อเกิดเหตุการณ์ ข้อมูลเปรียบเทียบตรงกัน
   - แฟลก TOV0(Timer/Counter0 Overflow Flag) ถูกเซตเมื่อเกิดข้อมูลเกิดโอเวอร์โฟลว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.โหมดการทำงานของไทเมอร์/เคาน์เตอร์0

ไทเมอร์/เคาน์เตอร์0 กำหนดโหมดการทำงาน 4 โหมด คือ

1. โหมดไทเมอร์/เคาน์เตอร์หรือโหมดปกติ (Normal Mode)
2. โหมดเคลียร์ไทเมอร์เมื่อเปรียบเทียบข้อมูลตรงกัน (Clear Timer on Compare Match(CTC) Mode) หรือที่เรียกว่าโหมด CTC
3. โหมดสร้างสัญญาณ PWM แบบ Fast PWM Mode
4. โหมดสร้างสัญญาณ PWM แบบ Phase Correct PWM Mode

2.การกำหนดโหมดการทำงาน

การกำหนดโหมดการทำงานของไทเมอร์/เคาน์เตอร์0 จะเกี่ยวข้องกับรีจิสเตอร์ TCCR0 โดยมีรายละเอียดดังนี้

.

รีจิสเตอร์ TCCR0

ตารางที่ 2.6 รีจิสเตอร์ TCCR0

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00
Read/write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

ตารางที่ 2.7 บิตที่กำหนดรูปแบบสัญญาณของไทเมอร์/เคาน์เตอร์

โหมด	WGM01 (CTC0)	WGM00 (PWM0)	การทำงานของไทเมอร์/เคาน์เตอร์	ค่าสูงสุด (TOP)	การปรับปรุงค่าของรีจิสเตอร์ OCR0	การเซตแฟลก TOV0
1	0	0	Normal	0xFF	Immediate	MAX
2	0	1	PWM,Phase Correct	0xFF	TOP	BOTTOM
3	1	0	CTC	OCR0	Immediate	MAX
4	1	1	Fast PWM	0xFF	TOP	MAX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ค่า BOTTOM, MAX และ TOP มีรายละเอียดดังนี้

ตารางที่ 2.8 ค่า BOTTOM, MAX และ TOP

BOTTOM	มีค่าเท่ากับ 0x00 (เท่ากับ 0 ในฐานสิบ)
MAX(Maximum)	มีค่าเท่ากับ 0xFF (เท่ากับ 255 ในฐานสิบ)
TOP	ค่าสูงสุดของการนับ โดยกำหนดให้มีค่าสูงสุดเท่ากับ MAX หรือมีค่าเท่ากับที่กำหนดในรีจิสเตอร์ OCR0 (ค่าที่กำหนดนี้จะขึ้นอยู่กับโหมดทำงาน)

ตารางที่ 2.9 บิตกำหนดแหล่งสัญญาณนาฬิกาของไทเมอร์ 0

CS02	CS01	CS00	รายละเอียด
0	0	0	ไม่ใช้สัญญาณนาฬิกา (ไทเมอร์/เคาน์เตอร์หยุดทำงาน)
0	0	1	$Clk_{I/O}$/1 (ไม่ใช้ปรีสเกลเลอร์)
0	1	0	$Clk_{I/O}$/8 (สัญญาณนาฬิกาหารด้วยปรีสเกลเลอร์ 8)
0	1	1	$Clk_{I/O}$/64 (สัญญาณนาฬิกาหารด้วยปรีสเกลเลอร์ 64)
1	0	0	$Clk_{I/O}$/256 (สัญญาณนาฬิกาหารด้วยปรีสเกลเลอร์ 256)
1	0	1	$Clk_{I/O}$/1024 (สัญญาณนาฬิกาหารด้วยปรีสเกลเลอร์ 1024)
1	1	0	ใช้สัญญาณนาฬิกาจากภายนอกที่ขา T0 ทำงานที่ขอบขาลงของสัญญาณ
1	1	1	ใช้สัญญาณนาฬิกาจากภายนอกที่ขา T0 ทำงานที่ขอบขาขึ้นของสัญญาณ

หมายเหตุ: ปรีสเกลเลอร์(Prescaler) คือตัวหารหรือลดทอนสัญญาณนาฬิกา (สัญญาณนาฬิกาจากคริสตอลที่ขา XTAL1 และ XTAL2)

รีจิสเตอร์ TIFR

ตารางที่ 2.10 รีจิสเตอร์ TIFR

บิตที่	7	6	5	4	3	2	1	0
ชื่อบิต	OCF0	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0
Read/write	R/W	R/W	R/W	R	R	R	R/W	R/W
ค่าเริ่มต้น	0	0	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.โหมดไทเมอร์/เคาน์เตอร์0

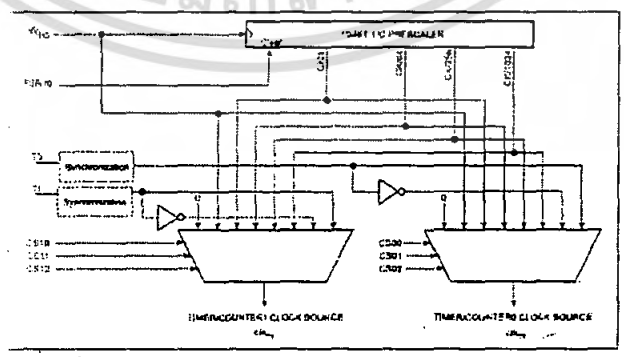
การทำงานในโหมดนี้จะใช้โมดูลไทเมอร์/เคาน์เตอร์0 ทำงานทั้งแบบไทเมอร์และแบบเคาน์เตอร์โดยการเซตบิตต่างๆในรีจิสเตอร์ต่อไปนี้

- รีจิสเตอร์ TCCR0
  - เซตบิต 2-0 : บิต CS02, CS01 และ CS00 เพื่อกำหนดสัญญาณนาฬิกา
  - เคลียร์บิตที่ 3,6 : บิต WGM01 และ WGM00 เพื่อกำหนดรูปแบบสัญญาณปกติ โหมด 1
  - เคลียร์บิตที่ 5,4 : บิต COM01, COM00 เพื่อปิดการใช้งานโหมด CTC และโหมดสร้างสัญญาณ PWM
- รีจิสเตอร์ TIFR

  เซตบิตที่ 0 (TOV0) หรือ 1 (OCF0) เพื่อเอ็นเอเบิลการใช้งานอินเตอร์รัปต์ และต้องเซตบิตอินเตอร์รัปต์โดยรวมบิต I ในรีจิสเตอร์ SREG ด้วย

4.ปรีสเกลเลอร์ไทเมอร์/เคาน์เตอร์ 0 และ 1

ปรีสเกลเลอร์หรือตัวลดทอนสัญญาณนาฬิกาสำหรับไทเมอร์ 0 และ 1 จะใช้งานร่วมกัน เมื่อมีการกำหนดใช้งานปรีสเกลเลอร์ให้กับไทเมอร์0 หรือไทเมอร์1 แล้วไทเมอร์อีกตัวจะมีปรีสเกลเลอร์เหมือนกันโดยสามารถใช้งานปรีสเกลเลอร์ได้ในโหมดของไทเมอร์เท่านั้น หากเป็นโหมดเคาน์เตอร์หรือโหมดนับสัญญาณนาฬิกาจากภายนอกจะไม่มีสัญญาณนาฬิกาจากภายนอกและไม่มีปรีสเกลเลอร์แสดงดังรูปที่ 2.5

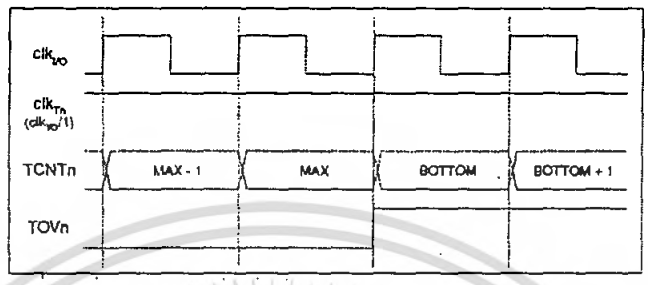


รูปที่ 2.5 ปรีสเกลเลอร์ไทเมอร์/เคาน์เตอร์ 0 และ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

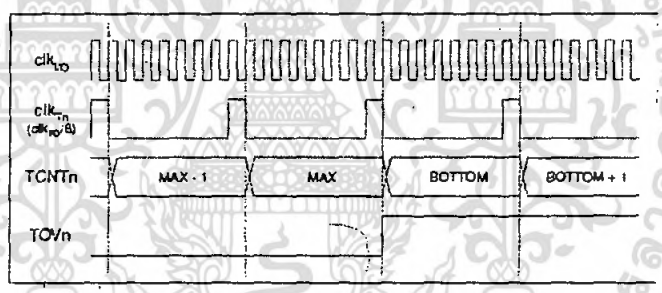
4.1 ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์

ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ ไม่มีกำหนดปรีสเกลเลอร์แสดงดังรูปที่ 2.6 การนับ 1 ครั้งจะขึ้นอยู่กับสัญญาณนาฬิกาโดยตรง หรือมีสัญญาณนาฬิกา 1 ลูกจะเกิดการนับ 1 ครั้ง



รูปที่ 2.6 ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ ไม่มีกำหนดปรีสเกลเลอร์

ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ กำหนดปรีสเกลเลอร์เท่ากับ 8 แสดงดังรูปที่ 2.7 การนับ 1 ครั้งจะขึ้นอยู่กับสัญญาณนาฬิกา 8 ลูก หรือเมื่อมีสัญญาณนาฬิกา 8 ลูกจะเกิดการนับ 1 ครั้ง



รูปที่ 2.7 ไทมิ่งไดอะแกรม ไทเมอร์/เคาน์เตอร์ กำหนดปรีสเกลเลอร์เท่ากับ 8

4.2 บิตปรีสเกลเลอร์

บิตที่ใช้ในการรีเซตการทำงานของปรีสเกลเลอร์ทั้งไทเมอร์0 และ ไทเมอร์1 จะถูกกำหนดอยู่ในบิตที่ 0 บิต PSR10 ในรีจิสเตอร์ SFIOR

รีจิสเตอร์ SFIOR

ตารางที่ 2.11 รีจิสเตอร์ SFIOR

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

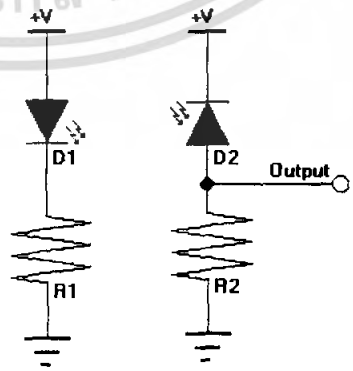
2.2 หลักการทำงานของ IR Sensor

แสงอินฟาเรด คือแสงที่มีความยาวคลื่นต่ำกว่าแสงสีแดงลงไป ดังนั้นจึงไม่สามารถมองเห็นได้ด้วยสายตา ของมนุษย์ ซึ่งคุณสมบัตินี้เอง จึงทำให้ เซ็นเซอร์ชนิดที่ใช้แสงอินฟาเรด เป็นที่นิยมนำมาใช้กันมาก โดยจะอาศัยหลักการของการสะท้อนของแสง กล่าวคือ ใช้อุปกรณ์ส่งแสง เป็นแหล่งกำเนิดปล่อยแสงออกไป และเมื่อแสงกระทบกับวัตถุด้านหน้า มันก็จะสะท้อนแสงกลับมา เข้าที่ตัวรับแสง ส่วนอัตราของการสะท้อนกลับนั้น ขึ้นอยู่กับสี และสภาพความมัน ของวัตถุที่สะท้อน เช่น สีดำ จะมีอัตราการสะท้อนกลับ น้อยกว่าสีขาว , หรือสภาพพื้นผิวที่มีความราบเรียบ เป็นมันวาว จะสามารถสะท้อนแสงได้ดีกว่า พื้นผิวที่มีลักษณะด้าน และขรุขระ เป็นต้น

รูปที่ 2.8 การสะท้อนของ IR Sensor

วงจรภาคส่ง จะประกอบไปด้วย LED infared ตัวส่ง D1 ทำงานร่วมกับค่าความต้านทาน R1 ที่มีหน้าที่จำกัดกระแสให้กับ LED ตัวส่ง ไม่ให้เสียหาย

วงจรภาครับ จะประกอบด้วย LED infared ตัวรับ D2 และค่าความต้านทาน R2 สำหรับการใช้งาน LED ตัวรับ ให้นำกระแส เราจะต้องต่อแบบ Revers เมื่อไม่มีแสงมาตกกระทบที่ตัวมัน จะมีค่าความต้านทานเป็นอนันต์ จึงไม่สามารถนำกระแสได้ แต่เมื่อมีแสงมาตกกระทบ ค่าความต้านทานจะลดลง จนสามารถนำกระแสได้



รูปที่ 2.9 วงจรภาคส่ง - วงจรภาครับ ของ IR Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3 การคิดค่าไฟฟ้า

หลอดประหยัดไฟ 36W

จาก ค่าไฟฟ้า = [(วัตต์ x ชั่วโมง) / 1000] x ค่าไฟฟ้าต่อหน่วย

จำนวนชั่วโมง 12 x 30 = 360 ชั่วโมง

จะได้ หน่วยไฟฟ้าที่ใช้ = [(36W x 360) / 1000]

หน่วยไฟฟ้าที่ใช้ = 12.96 หน่วย / เดือน / หลอด

ถ้า ติดตั้งระบบ Automatic Light Controller

ใน 12 ชั่วโมง เฉลี่ยมีการเข้าออก 10 ครั้ง ครั้งละประมาณ 15 นาที

(เข้าห้องน้ำ ออกไปทำธุระ ซื้อกับข้าว ซื้อของ ฯลฯ)

จะได้ 10 x 15 x 30 = 4500 นาที หรือ 75 ชั่วโมง / เดือน / หลอด

จะได้ หน่วยไฟฟ้าที่ใช้ = [(36W x (360 – 75)) / 1000]

หน่วยไฟฟ้าที่ใช้ = 10.26 หน่วย / เดือน / หลอด

ดังนั้น ประหยัดไป 12.96 – 10.26 = 2.7 หน่วย / เดือน / หลอด

ถ้า มีหลอดไฟ 20 หลอด

จะได้ 2.7 x 20 = 54 หน่วย / เดือน

ตารางที่ 2.12 อัตรารายเดือน : ค่าพลังงานไฟฟ้า

150 หน่วย (กิโลวัตต์ชั่วโมง) แรก (หน่วยที่ 1 - 150)	หน่วยละ	1.8047	บาท
250 หน่วยต่อไป (หน่วยที่ 151 - 400)	หน่วยละ	2.7781	บาท
เกินกว่า 400 หน่วย (หน่วยที่ 401 เป็นต้นไป)	หน่วยละ	2.9780	บาท

81618

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า คิดค่าไฟฟ้าจาก 150 หน่วยแรก จะใช้ค่าไฟฟ้าหน่วยละ 1.8047 บาท ในการคำนวน

จะได้ ประหยัด ค่าไฟฟ้า 2.7 x 1.8047 = 4.87269 บาท / เดือน / หลอด

ถ้า มีหลอดไฟ 20 หลอด

จะได้ ประหยัด ค่าไฟฟ้า 20 x 4.87269 = 97.4538 บาท / เดือน

จะได้ ประหยัด ค่าไฟฟ้า 20 x 4.87269 x 12 = 1169.4456 บาท / ปี

คิดหลายๆห้องรวมกัน ตัวอย่าง เช่น หอพักนักศึกษา

ถ้า จำนวนห้อง 10 ห้อง

จะได้ ประหยัด ค่าไฟฟ้า 97.4538 x 100 = 9745.38 บาท / เดือน

จะได้ ประหยัด ค่าไฟฟ้า 1169.4456 x 100 = 116,944.56 บาท / ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการออกแบบ

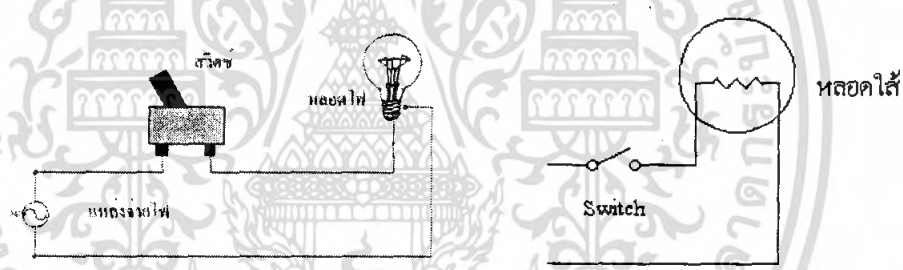
3.1 หลักการอย่างง่าย

การทำงานจะแบ่งเป็นส่วนๆ โดยแยกออกเป็น ส่วนเซนเซอร์ ส่วนควบคุม ส่วนส่งออก

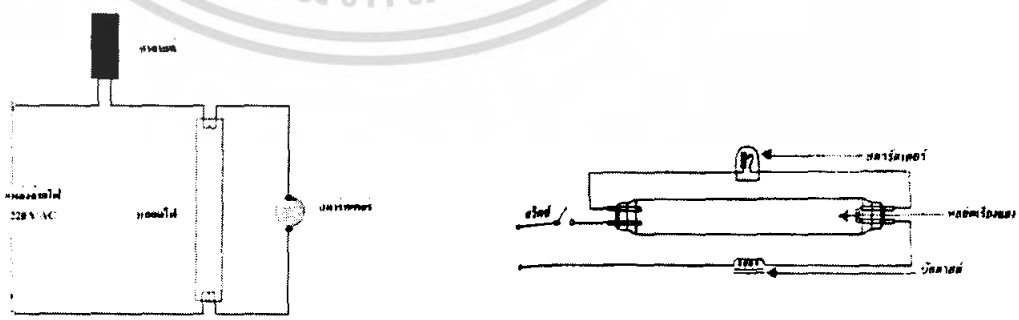
เซนเซอร์ใช้ Infrared และ Infrared Sensor โดยใช้การสะท้อนของ Infrared มาเข้า Infrared Sensor เป็นสัญญานบอกว่ามีวัตถุผ่าน หรือไม่มีวัตถุผ่าน

ส่วนควบคุม ใช้ AVR Controller ในการควบคุม เขียนโปรแกรมด้วยภาษา C โดยกำหนดเงื่อนไขต่างๆตามความเหมาะสม

ส่วนส่งออก ใช้ Relay ในการส่งไปที่เอาพุท โดยรับสัญญาณจาก AVR Controller แล้ว Relay ทำให้วงจรไฟ 220V ทำงาน

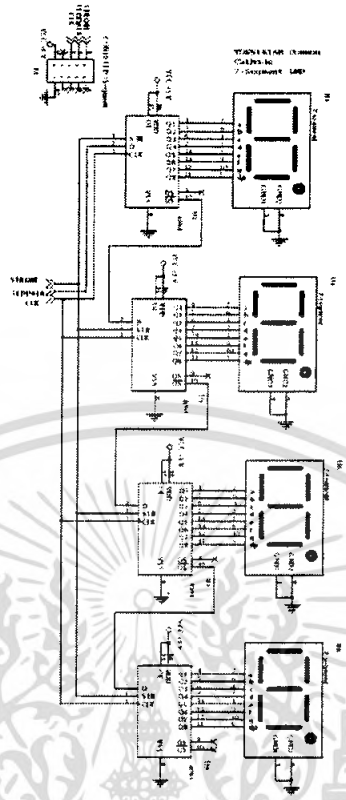


รูปที่ 3.1 วงจรแสงสว่างแบบหลอดไส้

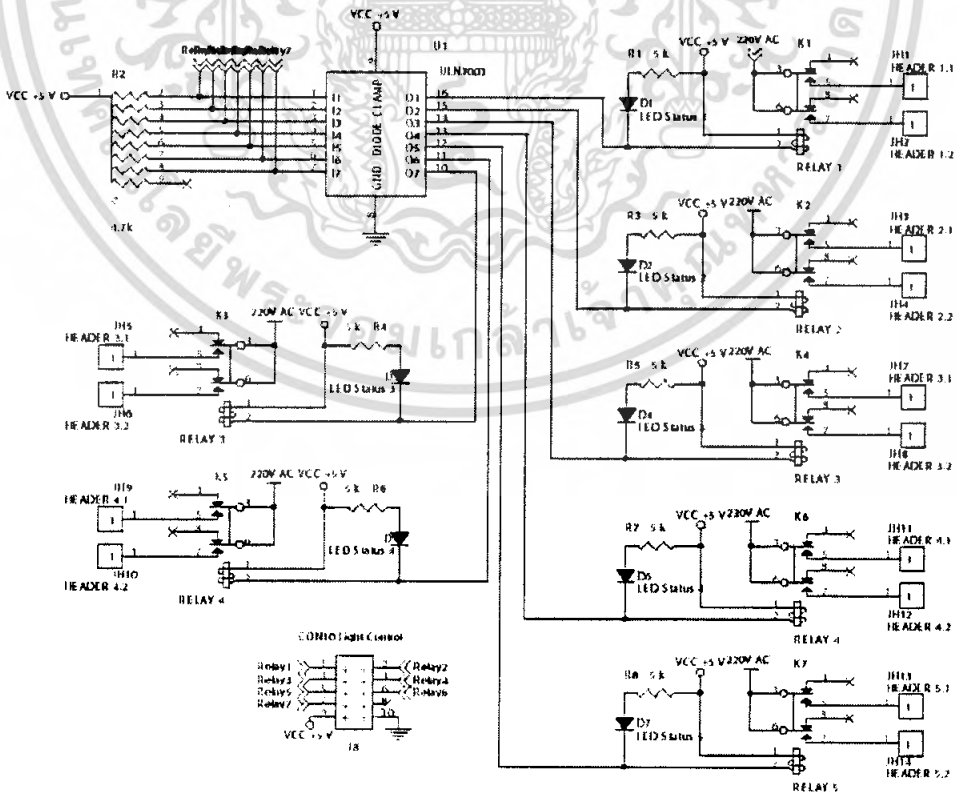


รูปที่ 3.2 วงจรแสงสว่างแบบหลอดฟูออเรสเซ็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

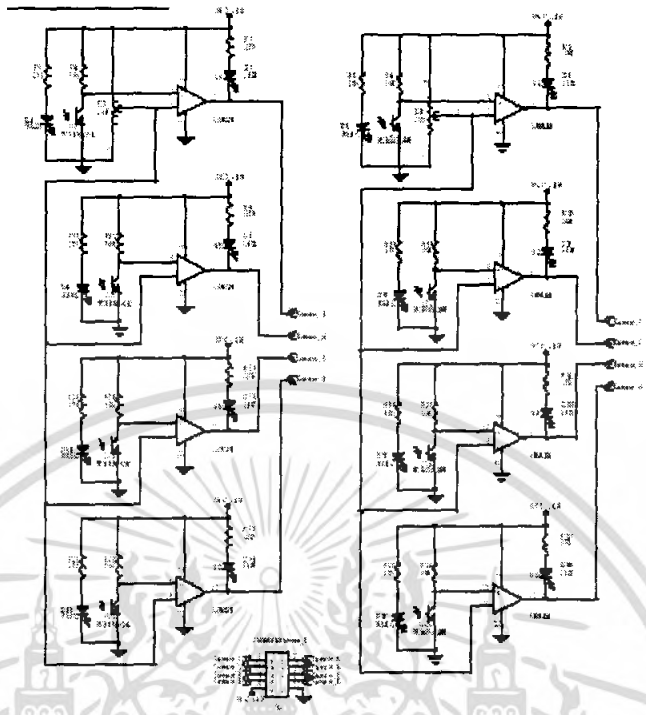


รูปที่ 3.3 วงจร 7Segment LED Display

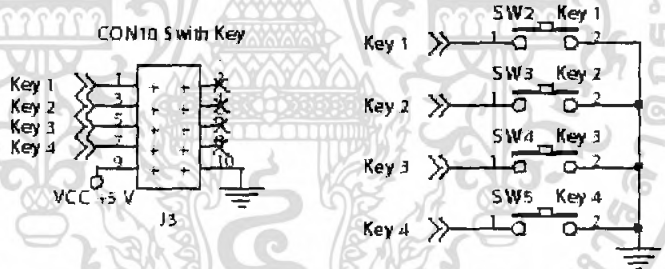


รูปที่ 3.4 วงจร Relay

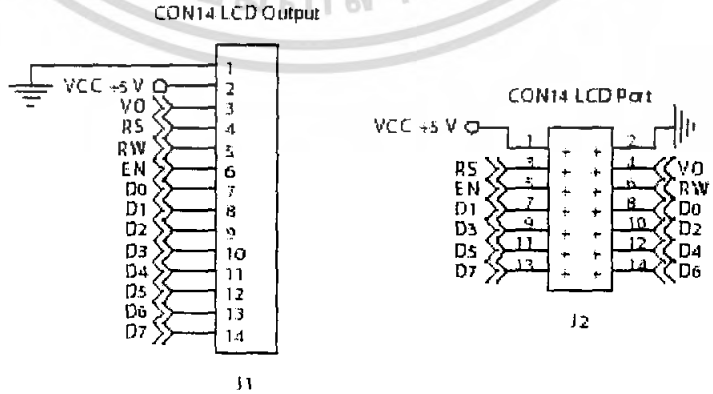
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจร รับ/ส่ง Infrared



รูปที่ 3.6 วงจร Switch



รูปที่ 3.7 วงจร LCD Connector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และ ผลการทดลอง

4.1 การทดลอง ทดสอบโครงงานพิเศษควบคุมไฟอัตโนมัติ

อุปกรณ์

1. AVR Controller Board
2. Automatic Light Controller Board
3. IR Sensor
4. ชุดไฟทดสอบ
5. Scope
6. Digital Multimeter
7. Power Supply
8. สายไฟ
9. ปากคีบสายไฟ

ขั้นตอนการทดลอง

1.เขียน Programs รับคำสั่งจาก Sensor แล้วสั่งให้วงจรทำงานตามกำหนด

2.ทำการต่อ AVR Controller Board เข้ากับ Automatic Light Controller Board

3.ต่อ IR Sensor เข้ากับ Automatic Light Controller Board

4.ต่อชุดไฟทดสอบเข้ากับ Output ที่ Relay ของวงจรบน Automatic Light Controller Board

5.ต่อ Power Supply เข้ากับวงจรทั้งหมด

6.ทำการทดสอบวงจร และ ระบบโดยการนำวัตถุมาบังที่ Sensor แทนการเดินผ่านของคน

7.เมื่อระบบไม่มีปัญหาจึงทำการวัดค่าด้วย Scope และ Multimeter

- Scope วัดที่ IR Sensor ตัวที่ 1 และ IR Sensor ตัวที่ 2
- Multimeter วัดที่ Input ที่ Relay ของวงจรบน Automatic Light Controller Board

8.นำวัตถุมาบังที่ Sensor แทนการเดินผ่านของคน และบันทึกผลการเปลี่ยนแปลงของแต่ละจุด

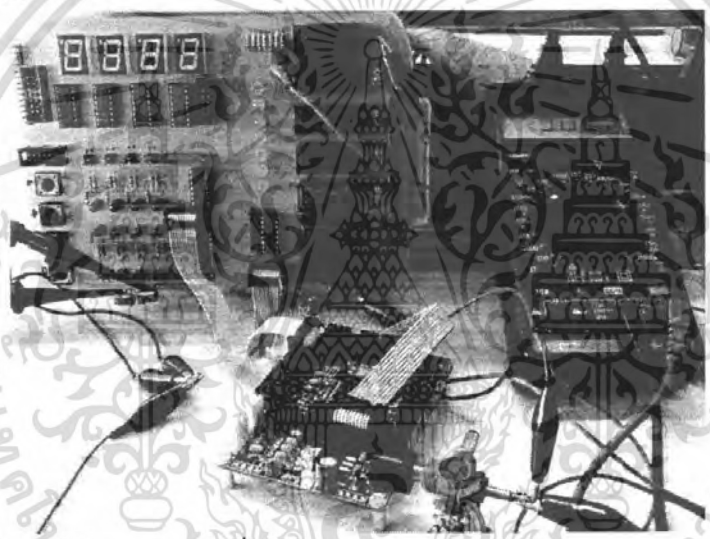
9.วิเคราะห์ และวิจารณ์ ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

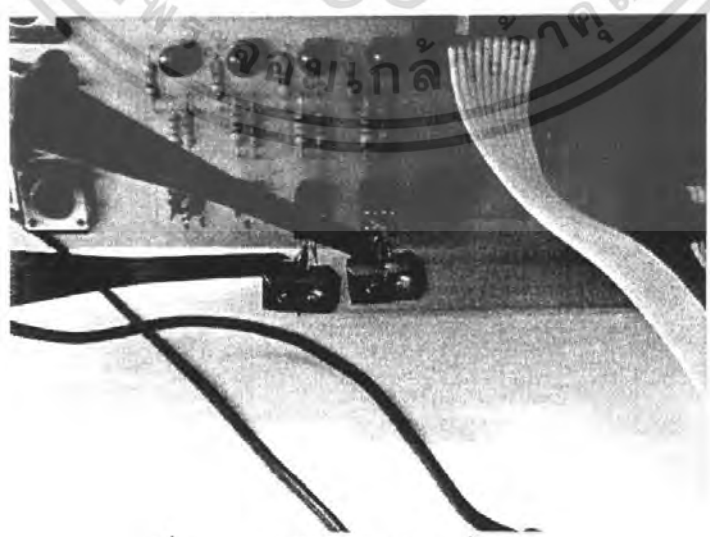
ผลการทดลอง

การต่อวงจรทดลอง โดย

- Scope วัดที่ IR Sensor ตัวที่ 1 และ IR Sensor ตัวที่ 2
- Multimeter วัดที่ Input ที่ Relay ของวงจรบน Automatic Light Controller Board
- Logic High มี Voltage ประมาณ 4 – 5 Volt
- Logic Low มี Voltage ประมาณ 0 – 1 Volt
- Max (1) = IR Sensor ตัวที่ 1
- Max (2) = IR Sensor ตัวที่ 2

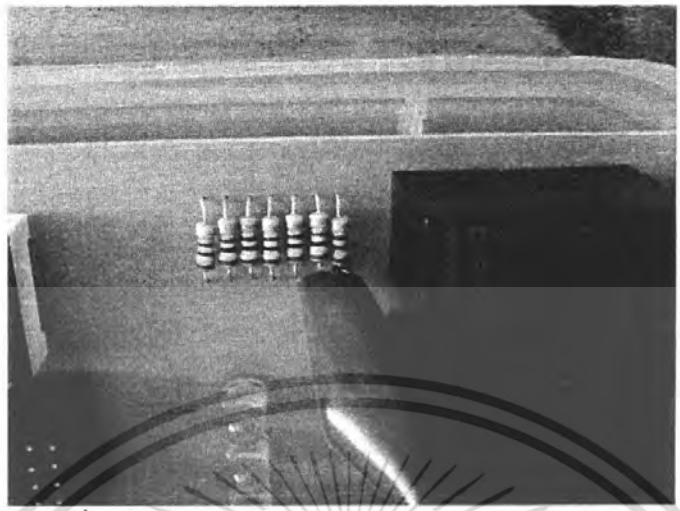


รูปที่ 4.1 การต่อวงจรทดลอง



รูปที่ 4.2 การใช้ Scope จับวัดที่ IR Sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

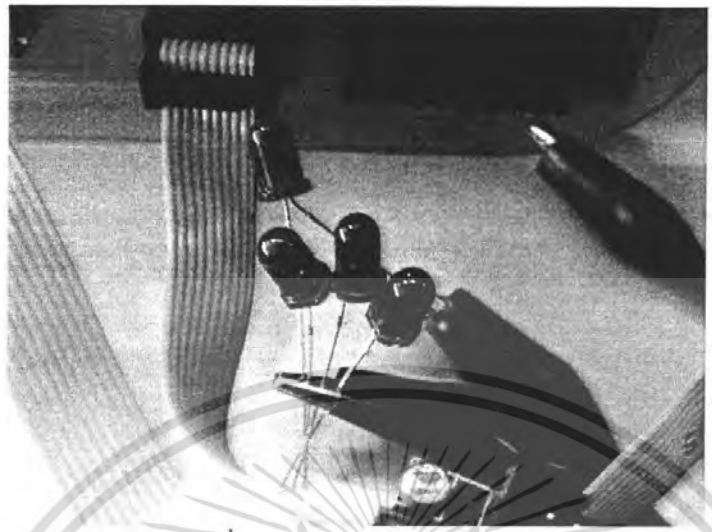


รูปที่ 4.3 การใช้ Multimeter วัดส่วน Input ของ Relay

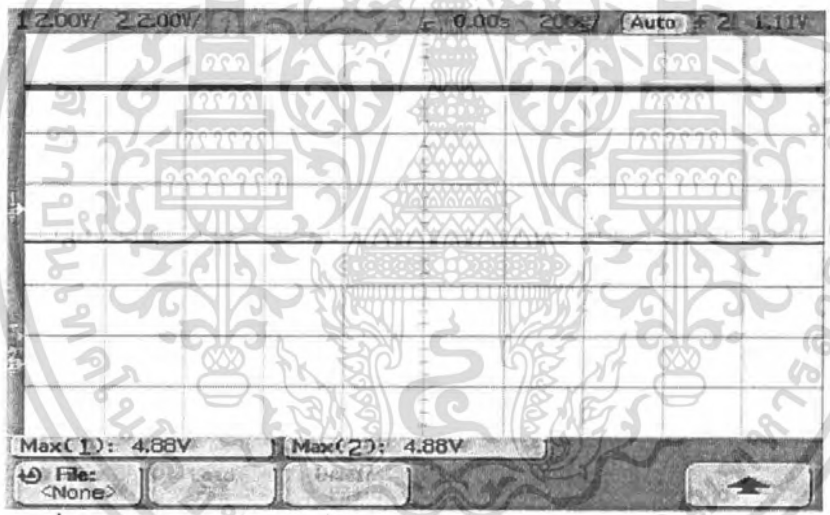
รูปที่ 4.4 การต่อไฟบวกเข้าที่ Input ของ Relay

รูปที่ 4.5 การต่อไฟบวกเข้าที่ Input ของ Relay

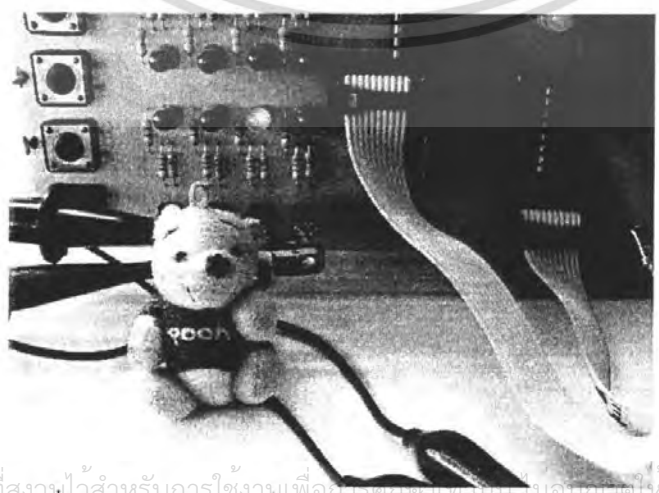
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 การต่อชุดไฟทดสอบ

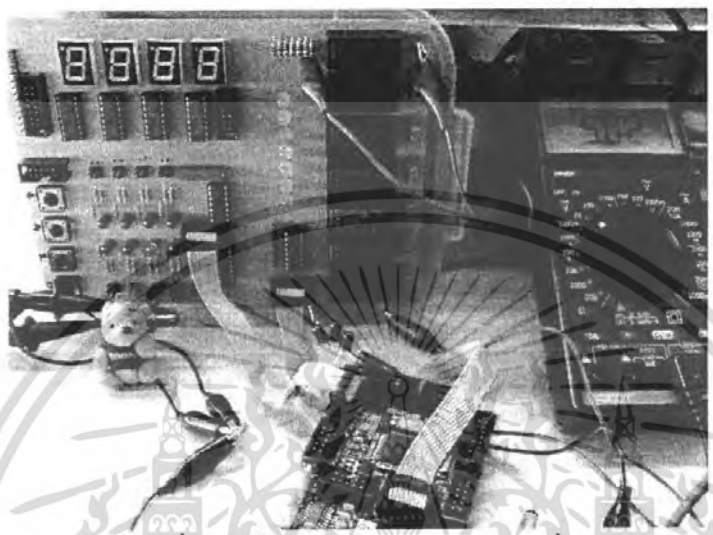


รูปที่ 4.7 ผลจาก Scope จับที่ Sensor ตัวที่ 1 และ 2 ขณะยังไม่มีวัตถุผ่าน

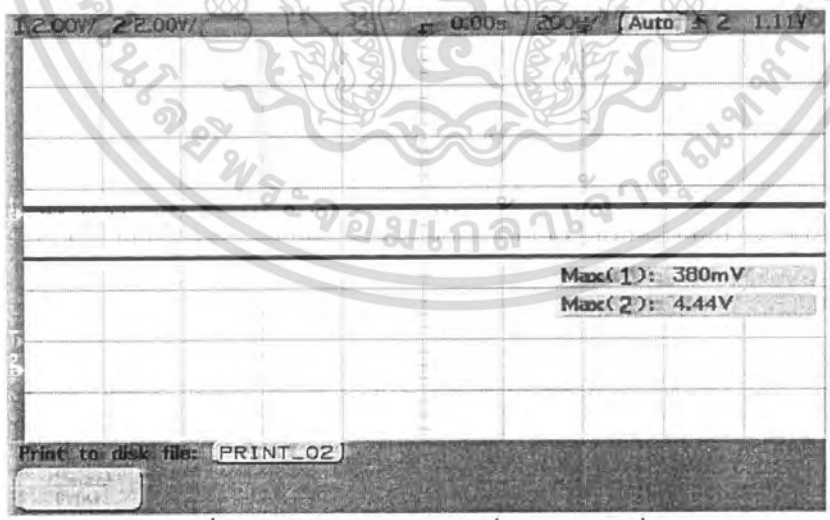


รูปที่ 4.8 นำวัตถุมาบังที่ Sensor แทนการเดินผ่านของคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

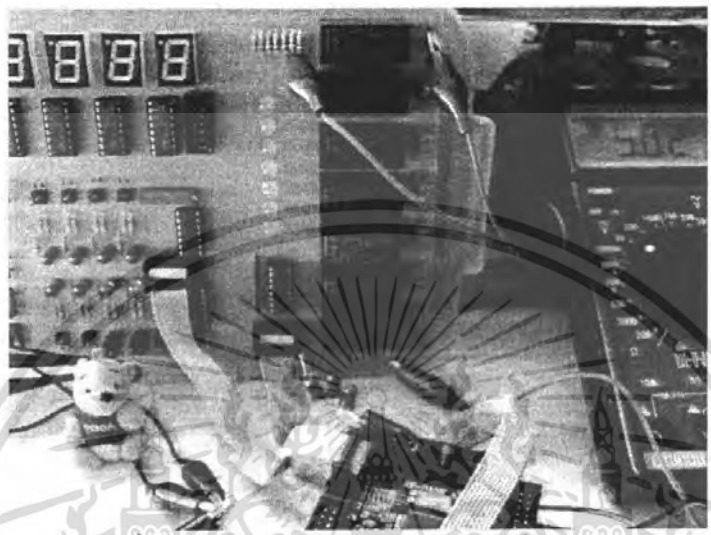


รูปที่ 4.9 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1

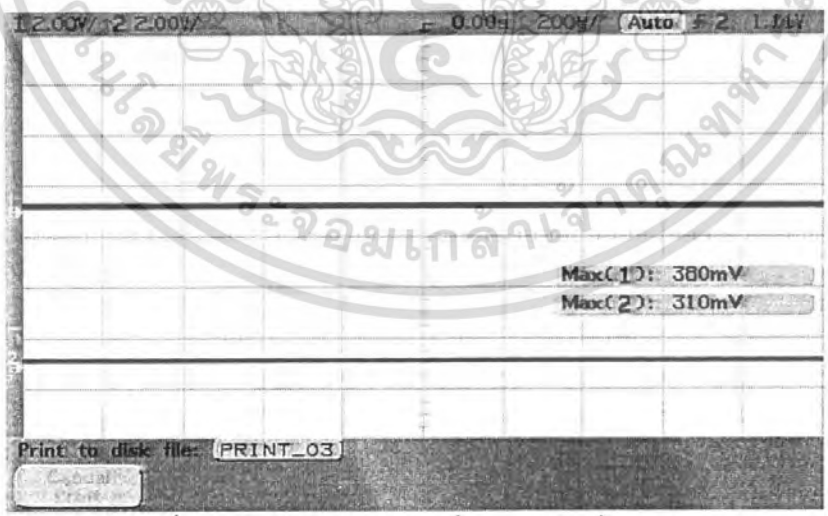


รูปที่ 4.10 ผลจาก Scope จับที่ Sensor ตัวที่ 1
เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

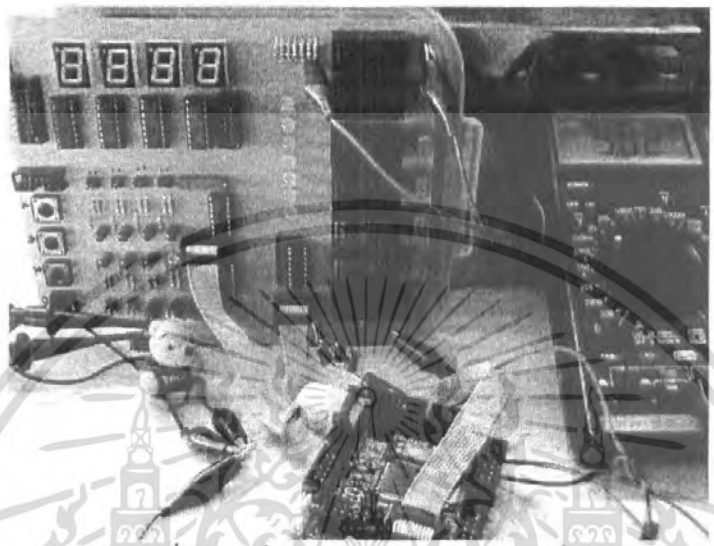


รูปที่ 4.11 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1 และ 2

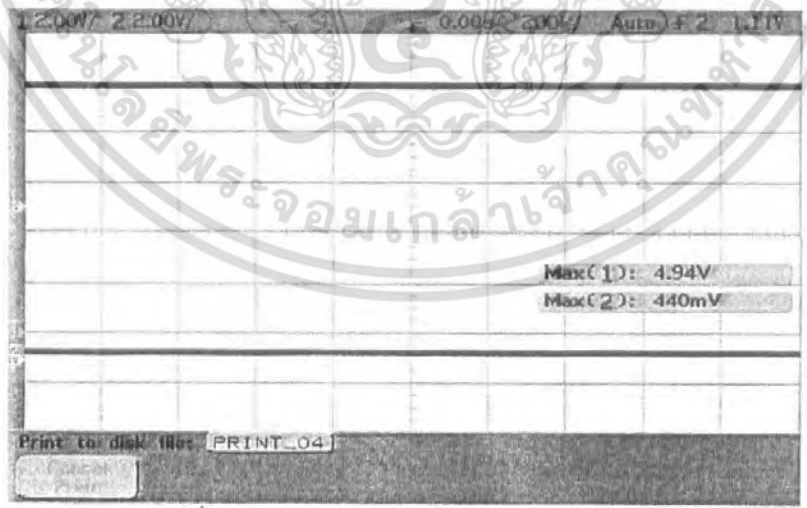


รูปที่ 4.12 ผลจาก Scope จับที่ Sensor ตัวที่ 1 และ 2
เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



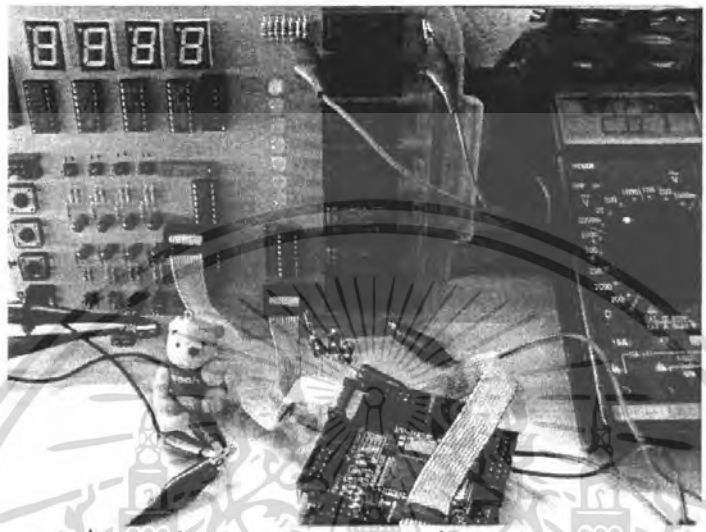
รูปที่ 4.13 เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 2



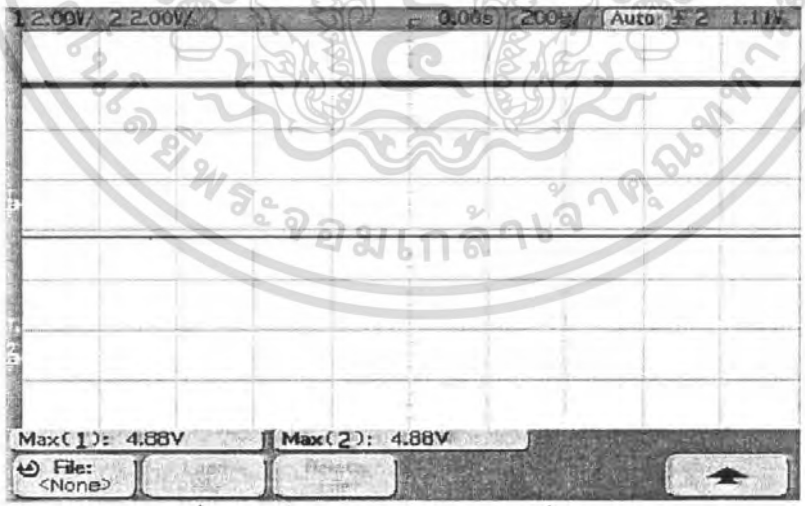
รูปที่ 4.14 ผลจาก Scope จับที่ Sensor ตัวที่ 2

เมื่อวัตถุอยู่หน้า Sensor ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

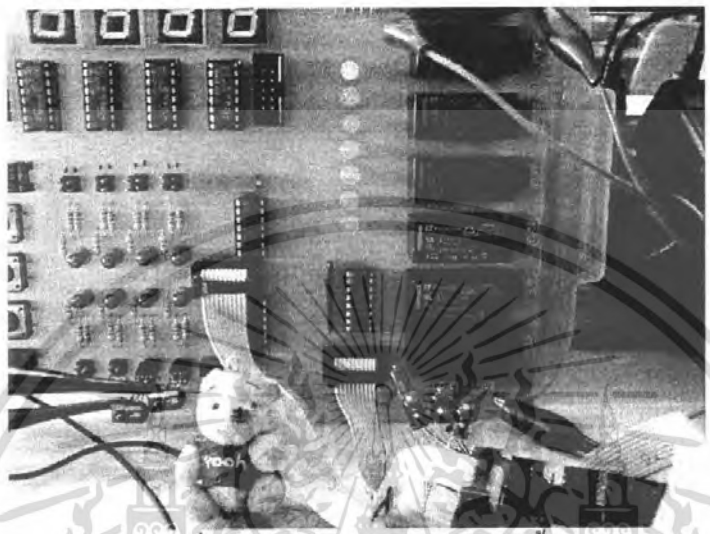


รูปที่ 4.15 เมื่อวัตถุผ่าน Sensor ตัวที่ 1 และ 2 ตามลำดับ

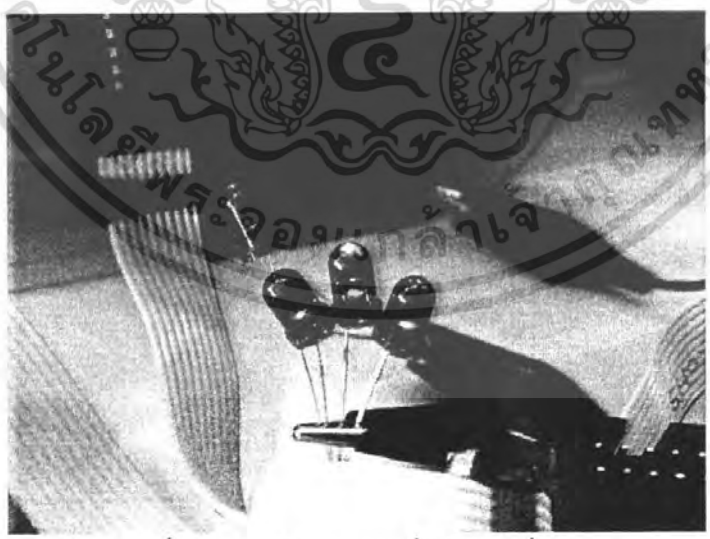


รูปที่ 4.16 เมื่อวัตถุผ่าน Sensor ตัวที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ชุดไฟทดสอบจะสว่างขึ้น



รูปที่ 4.18 ชุดไฟทดสอบที่ได้รับคำสั่งเปิด
จากโครงงานพิเศษควบคุมไฟอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป และข้อเสนอแนะ

5.1 สรุป

จากการออกแบบ ประดิษฐ์ และทดลอง โครงงานพิเศษ ระบบควบคุมไฟอัตโนมัติ โดยใช้อินฟาเรทในการตรวจจับวัตถุที่ผ่านเข้ามา และทำการนับจำนวนวัตถุที่ผ่านเข้ามาด้วยการโปรแกรมในคอนโทรเลอร์ จากนั้นสั่งการให้เปิด หรือปิดไฟตามเงื่อนไขที่กำหนด ระบบทั้งหมดนี้สามารถช่วยประหยัดพลังงานไฟฟ้าได้จากการคำนวณ และเมื่อการใช้งานของอุปกรณ์น้อยลง ก็สามารถยืดอายุของอุปกรณ์ไฟฟ้าได้อีกด้วย

สิ่งที่สำคัญที่สุด คือ ระบบนี้สามารถนำไปประยุกต์ใช้ได้อย่างกว้างขวาง และจะทำให้ช่วยประหยัดพลังงานในภาพรวมได้อีกมากมาย

5.2 ข้อเสนอแนะ และแนวทางในการพัฒนา

1. ตัวเซนเซอร์ ที่ใช้ อินฟาเรท นั้นยังมีระยะการรับส่งที่ไม่มากนัก สามารถนำเซนเซอร์ประเภทอื่นมาใช้กับระบบ เพื่อเพิ่มความสามารถในการตรวจจับวัตถุ ให้มีประสิทธิภาพมากขึ้นได้

2. ตัวโปรแกรม สามารถเพิ่มเงื่อนไข รูปแบบ หรือแม้กระทั้งการเชื่อมต่อกับระบบอื่น ได้อีกมากมายแล้วแต่ความต้องการของผู้พัฒนาโปรแกรม เพื่อการใช้ประโยชน์สูงสุดของระบบ

3. อาจพัฒนาโดยการเปลี่ยนแปลงเพิ่ม หรือลด อุปกรณ์บางอย่างเพื่อให้ต้นทุนในการผลิตเหมาะสมกับความต้องการมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

http://www.Google.co.th

http://en.wikipedia.org/wiki/Main_Page

http://www.mea.or.th/internet/index_mea1.html

http://www.ett.co.th/article/avr.html

http://www.atmel.com/dyn/general/advanced_search_results.asp?device=1&tools=1&faqs=1&datasheets=1&appNotes=1&userGuides=1&software=1&press=1&articles=1&flyers=1&checkAll=1&checkAllReference=1&target=avr

http://www.electoday.com/modules.php?name=News&file=article&sid=137

http://www.wara.com/modules.php?name=News&file=article&sid=567

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <mega128.h>
#include <stdio.h>
#include <delay.h>

unsigned int count=0;
unsigned int sec,min,hour,t;

interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
      if (count++>488)
      {count = 0;

         if (sec++>= 59)
           {sec = 0;


            if (min++>= 59)
              {min = 0;
                 if (hour++>= 23)
                   {hour = 0;
                   }
              }
            }
      }
}

void main(void)

{

PORTA=0x00;
DDRA=0xff;

PORTB=0x00;
DDRB=0x10;

PORTC=0x00;
DDRC=0x00;

PORTD=0x00;
DDRD=0x00;

PORTE=0x00;
DDRE=0x00;

PORTF=0x00;
DDRF=0x00;

PORTG=0x00;
```

เอกสารนี้จัดทำขึ้นเพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DDRG=0x00;

ASSR=0x00;
TCCR0=0x15;
TCNT0=0x00;
OCR0=0x00;

TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

TIMSK=0x01;
ETIMSK=0x00;

UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACSR=0x80;
SFIOR=0x00;

      sec = 58;
      min = 59;
      hour = 23;


#asm("sei")

while (1)
    {  while(count==488  ){
      printf("\n\r %d:%d:%d",hour,min,sec);     }
      t=0;
      if(sec==5)
      {PORTA=0xff;
      }
      if(sec==10)
      {PORTA=0x00;
      }

       if(sec==15)
      {PORTA=0xff;
      }
      if(sec==20)
      {PORTA=0x00;
      }

      if(sec==25)
      {PORTA=0xff;
      }
      if(sec==30)
      {PORTA=0x00;
      }
    };
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <mega128.h>
#include <delay.h>
// Declare your global variables here
unsigned int count=0;
unsigned int sec,min,hour;

interrupt [8] void timer0_overflow(void)
{
 if (count++>29)
 {count = 0;
 if (sec++>= 59)
 {sec = 0;
 if (min++>= 59)
 {min = 0;
 if (hour++>= 24)
 {hour = 0;
 }
 }
 }
 }
}

void main(void)
{
unsigned char a1,a2,a3 , b1,b2,b3 , z ;
          a1=0;
          a2=0;
          a3=0;
          b1=0;
          b2=0;
          b3=0;
          z=0;

PORTA=0xff;
DDRA=0x00;

PORTB=0x00;
DDRB=0xff;

sec = 0;
min = 10;
hour = 10;



while (count==0) {
     //open
     if(PINA.4==0){
     a1=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        while(a1==1){
        if(PINA.6==0){
         a1=0;
         a2=1;
        }
        }
        while(a2==1){
        if(PINA.4==1){
         b1=1;
         a2=0;
        }
        }
        while(b1==1){
        if(PINA.6==1){
        b1=0;
        z++;
        PORTB.0=1;
        }
        }
        //close
        if(PINA.6==0){
         a1=1;
        }
        while(a1==1){
        if(PINA.4==0){
         a1=0;
         a2=1;
        }
        }
        while(a2==1){
        if(PINA.6==1){
         b1=1;
         a2=0;
        }
        }
        while(b1==1){
        if(PINA.4==1){
        b1=0;
        z--;
        //PORTB.1=1;
        if (z==0) {
        PORTB.0=0;
        }
        }
        }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 133 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 128K Bytes of In-System Reprogrammable Flash
    Endurance: 1,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- 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
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 1 to 16 Bits
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels
    2 Differential Channels with Programmable Gain (1x, 10x, 200x)
  - Byte-oriented 2-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with 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
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP
- Operating Voltages
  - 2.7 - 5.5V (ATmega128L)
  - 4.5 - 5.5V (ATmega128)
- Speed Grades
  - 0 - 8 MHz (ATmega128L)
  - 0 - 16 MHz (ATmega128)

8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

ATmega128
ATmega128L

Preliminary

Summary

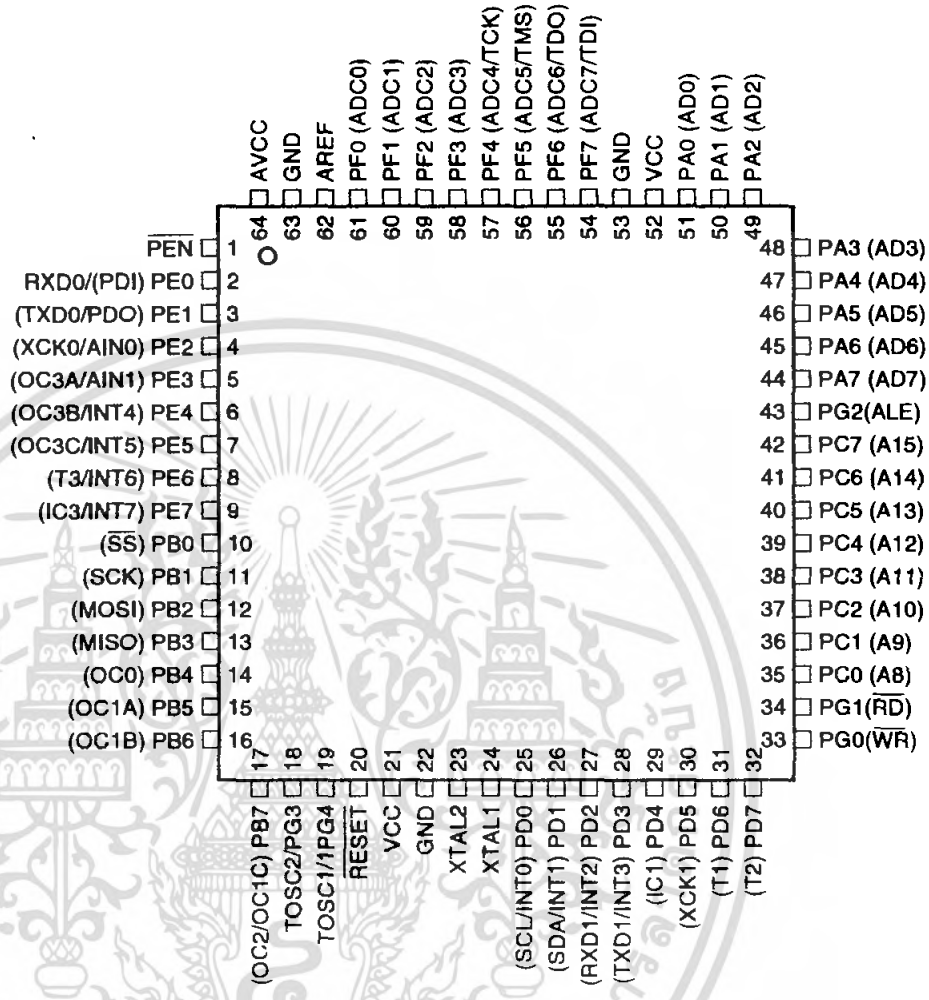
Rev. 2467AS-08/01

Note: This is a summary document. A complete document is available on our web site at *www.atmel.com*.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

Figure 1. Pinout ATmega128



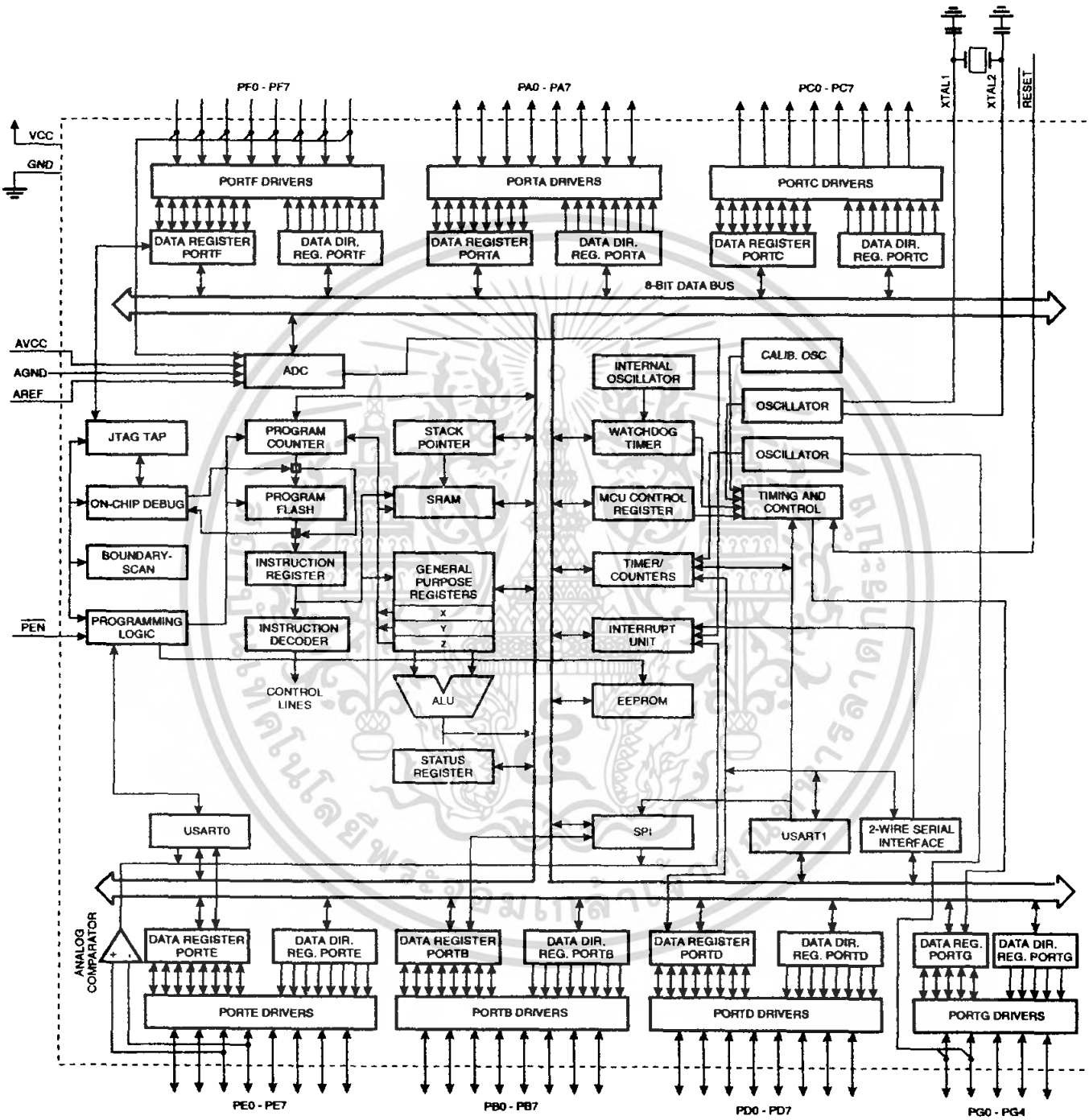
Overview

The ATmega128 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 ATmega128 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram

Figure 2. Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

The ATmega128 provides the following features: 128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 4K bytes SRAM, 53 general-purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible timer/counters with compare modes and PWM, 2 USARTs, a byte oriented 2-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with internal oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, timer/counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset. In 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 ATmega128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128 AVR 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.

ATmega103 and ATmega128 Compatibility

The ATmega128 is a highly complex microcontroller where the number of I/O locations supersedes the 64 I/O location reserved in the AVR instruction set. To ensure backward compatibility with the ATmega103, all I/O locations present in ATmega103 have the same location in ATmega128. Most additional I/O locations are added in an Extended I/O space starting from $60 to $FF, (i.e., in the ATmega103 internal RAM space). These location can be reached by using LD/LDS/LDD and ST/STS/STD instruction only, not by using IN and OUT instruction. The relocation of the internal RAM space may still be a problem for ATmega103 users. Also, the increased number of interrupt vectors might be a problem if the code uses absolute addresses. To solve these problems, an ATmega103 compatibility mode can be selected by programming the fuse M103C. In this mode, none of the functions in the Extended I/O space are in use, so the internal RAM is located as in ATmega103. Also, the extended interrupt vectors are removed.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The ATmega128 is 100% pin compatible with ATmega103, and can replace the ATmega103 on current Printed Circuit Boards. The application note "Replacing ATmega103 by ATmega128" describes what the user should be aware of replacing the ATmega103 by an ATmega128.

ATmega103 Compatibility Mode

By programming the M103C fuse, the ATmega128 will be compatible with the ATmega103 regards to RAM, I/O pins and interrupt vectors as described above. However, some new features in ATmega128 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, asynchronous mode only. Only the 8 least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with 2 compare registers instead of two 16-bit Timer/Counters with 3 compare registers.
- 2-wire serial interface is not supported.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait-states to different External Memory Address sections.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

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

Port A also serves the functions of various special features of the ATmega128 as listed on page 68.

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 ATmega128 as listed on page 69.

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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port C also serves the functions of special features of the ATmega128 as listed on page 72. In ATmega103 compatibility mode, Port C is output only, and the port C pins are **not** tri-stated when a reset condition becomes active.

Port D (PD7..PD0)

Port D is an 8-bit bidirectional 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 ATmega128 as listed on page 73.

Port E (PE7..PE0)

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

Port E also serves the functions of various special features of the ATmega128 as listed on page 76.

Port F (PF7..PF0)

Port F serves as the analog inputs to the A/D Converter.

Port F 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 F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F 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 PF7(TDI), PF5(TMS) and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input Port only.

Port G (PG4..PG0)

Port G is a 5-bit bidirectional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

The port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32 kHz oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are oscillator pins.

$\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 46. Shorter pulses are not guaranteed to generate a reset.

XTAL1

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XTAL2	Output from the inverting oscillator amplifier.
AVCC	This is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter.
AREF	This is the analog reference pin for the A/D Converter.
PEN	This is a programming enable pin for the serial programming mode. By holding this pin low during a power-on reset, the device will enter the serial programming mode. PEN has no function during normal operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($FF)	Reserved	-	-	-	-	-	-	-	-	
..	Reserved	-	-	-	-	-	-	-	-	
($9E)	Reserved	-	-	-	-	-	-	-	-	
($9D)	UCSR1C	-	UMSEL1	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	UCPOL1	185
($9C)	UDR1	USART1 I/O Data Register								182
($9B)	UCSR1A	RXC1	TXC1	UDRE1	FE1	DOR1	UPE1	U2X1	MPCM1	183
($9A)	UCSR1B	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB81	TXB81	184
($99)	UBRR1L	USART1 Baud Rate Register Low								186
($98)	UBRR1H	-	-	-	-	USART1 Baud Rate Register High				186
($97)	Reserved	-	-	-	-	-	-	-	-	
($96)	Reserved	-	-	-	-	-	-	-	-	
($95)	UCSR0C	-	UMSEL0	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	UCPOL0	185
($94)	Reserved	-	-	-	-	-	-	-	-	
($93)	Reserved	-	-	-	-	-	-	-	-	
($92)	Reserved	-	-	-	-	-	-	-	-	
($91)	Reserved	-	-	-	-	-	-	-	-	
($90)	UBRR0H	-	-	-	-	USART0 Baud Rate Register High				186
($8F)	Reserved	-	-	-	-	-	-	-	-	
($8E)	Reserved	-	-	-	-	-	-	-	-	
($8D)	Reserved	-	-	-	-	-	-	-	-	
($8C)	TCCR3C	FOC3A	FOC3B	FOC3C	-	-	-	-	-	132
($8B)	TCCR3A	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	127
($8A)	TCCR3B	ICNC3	ICES3	-	WGM33	WGM32	CS32	CS31	CS30	130
($89)	TCNT3H	Timer/Counter3 - Counter Register High Byte								132
($88)	TCNT3L	Timer/Counter3 - Counter Register Low Byte								132
($87)	OCR3AH	Timer/Counter3 - Output Compare Register A High Byte								133
($86)	OCR3AL	Timer/Counter3 - Output Compare Register A Low Byte								133
($85)	OCR3BH	Timer/Counter3 - Output Compare Register B High Byte								133
($84)	OCR3BL	Timer/Counter3 - Output Compare Register B Low Byte								133
($83)	OCR3CH	Timer/Counter3 - Output Compare Register C High Byte								133
($82)	OCR3CL	Timer/Counter3 - Output Compare Register C Low Byte								133
($81)	ICR3H	Timer/Counter3 - Input Capture Register High Byte								134
($80)	ICR3L	Timer/Counter3 - Input Capture Register Low Byte								134
($7F)	Reserved	-	-	-	-	-	-	-	-	
($7E)	Reserved	-	-	-	-	-	-	-	-	
($7D)	ETIMSK	-	-	TICIE3	OCIE3A	OCIE3B	TOIE3	OCIE3C	OCIE1C	135
($7C)	ETIFR	-	-	ICF3	OCF3A	OCF3B	TOV3	OCF3C	OCF1C	136
($7B)	Reserved	-	-	-	-	-	-	-	-	
($7A)	TCCR1C	FOC1A	FOC1B	FOC1C	-	-	-	-	-	131
($79)	OCR1CH	Timer/Counter1 - Output Compare Register C High Byte								133
($78)	OCR1CL	Timer/Counter1 - Output Compare Register C Low Byte								133
($77)	Reserved	-	-	-	-	-	-	-	-	
($76)	Reserved	-	-	-	-	-	-	-	-	
($75)	Reserved	-	-	-	-	-	-	-	-	
($74)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	198
($73)	TWDR	2-wire Serial Interface Data Register								199
($72)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	200
($71)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	199
($70)	TWBR	2-wire Serial Interface Bit Rate Register								197
($6F)	OSCCAL	Oscillator Calibration Register								38
($6E)	Reserved	-	-	-	-	-	-	-	-	
($6D)	XMCRA	-	SRL2	SRL1	SRL0	SRW01	SRW00	SRW11		29
($6C)	XMCRB	XMBK	-	-	-	-	XMM2	XMM1	XMM0	31
($6B)	Reserved	-	-	-	-	-	-	-	-	
($6A)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	84
($69)	Reserved	-	-	-	-	-	-	-	-	
($68)	SPMCSR	SPMIE	RWWSB	-	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	270
($67)	Reserved	-	-	-	-	-	-	-	-	
($66)	Reserved	-	-	-	-	-	-	-	-	
($65)	PORTG	-	-	-	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	83
($64)	DDRG	-	-	-	DDG4	DDG3	DDG2	DDG1	DDG0	83
($63)	PING	-	-	-	PING4	PING3	PING2	PING1	PING0	83
($62)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
($61)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	83
($60)	Reserved	-	-	-	-	-	-	-	-	
$3F ($5F)	SREG	I	T	H	S	V	N	Z	C	9
$3E ($5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	12
$3D ($5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	12
$3C ($5C)	XDIV	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	39
$3B ($5B)	RAMPZ	-	-	-	-	-	-	-	RAMPZ0	12
$3A ($5A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	85
$39 ($59)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	86
$38 ($58)	EIFR	INTF7	INTF6	INTF5	INTF4	INTF3	INTF	INTF1	INTF0	86
$37 ($57)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	103, 134, 153
$36 ($56)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	103, 136, 154
$35 ($55)	MCUCR	SRE	SRW10	SE	SM1	SM0	SM2	IVSEL	IVCE	29, 41, 58
$34 ($54)	MCUCSR	JTD	-	-	JTRF	WDRF	BORF	EXTRF	PORF	49, 246
$33 ($53)	TCCR0	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	98
$32 ($52)	TCNT0	Timer/Counter0 (8 Bit)								100
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register								100
$30 ($50)	ASSR	-	-	-	-	AS0	TCN0UB	OCR0UB	TCR0UB	101
$2F ($4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	127
$2E ($4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	130
$2D ($4D)	TCNT1H	Timer/Counter1 - Counter Register High Byte								132
$2C ($4C)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								132
$2B ($4B)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								133
$2A ($4A)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								133
$29 ($49)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								133
$28 ($48)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								133
$27 ($47)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								134
$26 ($46)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								134
$25 ($45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	151
$24 ($44)	TCNT2	Timer/Counter2 (8 Bit)								153
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register								153
$22 ($42)	OCDR	IDRD/ OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	242
$21 ($41)	WDTCR	-	-	-	WDCE	WDE	WDP2	WDP1	WDP0	51
$20 ($40)	SFIOR	TSM	-	-	ADHSM	ACME	PUD	PSR0	PSR321	67, 104, 139, 237
$1F ($3F)	EEARH	-	-	-	-	EEPROM Address Register High				19
$1E ($3E)	EEARL	EEPROM Address Register Low Byte								19
$1D ($3D)	EEDR	EEPROM Data Register								20
$1C ($3C)	EECR	-	-	-	-	EERIE	EEMWE	EEWE	EERE	20
$1B ($3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	81
$1A ($3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	81
$19 ($39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	81
$18 ($38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	81
$17 ($37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	81
$16 ($36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	82
$15 ($35)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	82
$14 ($34)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	82
$13 ($33)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	82
$12 ($32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	82
$11 ($31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	82
$10 ($30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	82
$0F ($2F)	SPDR	SPI Data Register								163
$0E ($2E)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	162
$0D ($2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	161
$0C ($2C)	UDR0	USART0 I/O Data Register								182
$0B ($2B)	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	183
$0A ($2A)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	184
$09 ($29)	UBRR0L	USART0 Baud Rate Register Low								186
$08 ($28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	218
$07 ($27)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	233
$06 ($26)	ADCSRA	ADEN	ADSC	ADRF	ADIF	ADIE	ADPS2	ADPS1	ADPS0	235
$05 ($25)	ADCH	ADC Data Register High Byte								236
$04 ($24)	ADCL	ADC Data Register Low byte								236
$03 ($23)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	82
$02 ($22)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$01 ($21)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	83
$00 ($20)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	83

Notes: 1. *For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.*

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF − Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 − Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd − 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd − Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd − Rr − C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd − K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
ELPM		Extended Load Program Memory	R0 ← (RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd ← (RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Inc	Rd ← (RAMPZ:Z), RAMPZ:Z ← RAMPZ:Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
8	2.7 - 5.5V	ATmega128-8AC	64A	Commercial (0°C to 70°C)
		ATmega128-8AI	64A	Industrial (-40°C to 85°C)
16	4.5 - 5.5V	ATmega128-16AC	64A	Commercial (0°C to 70°C)
		ATmega128-16AI	64A	Industrial (-40°C to 85°C)

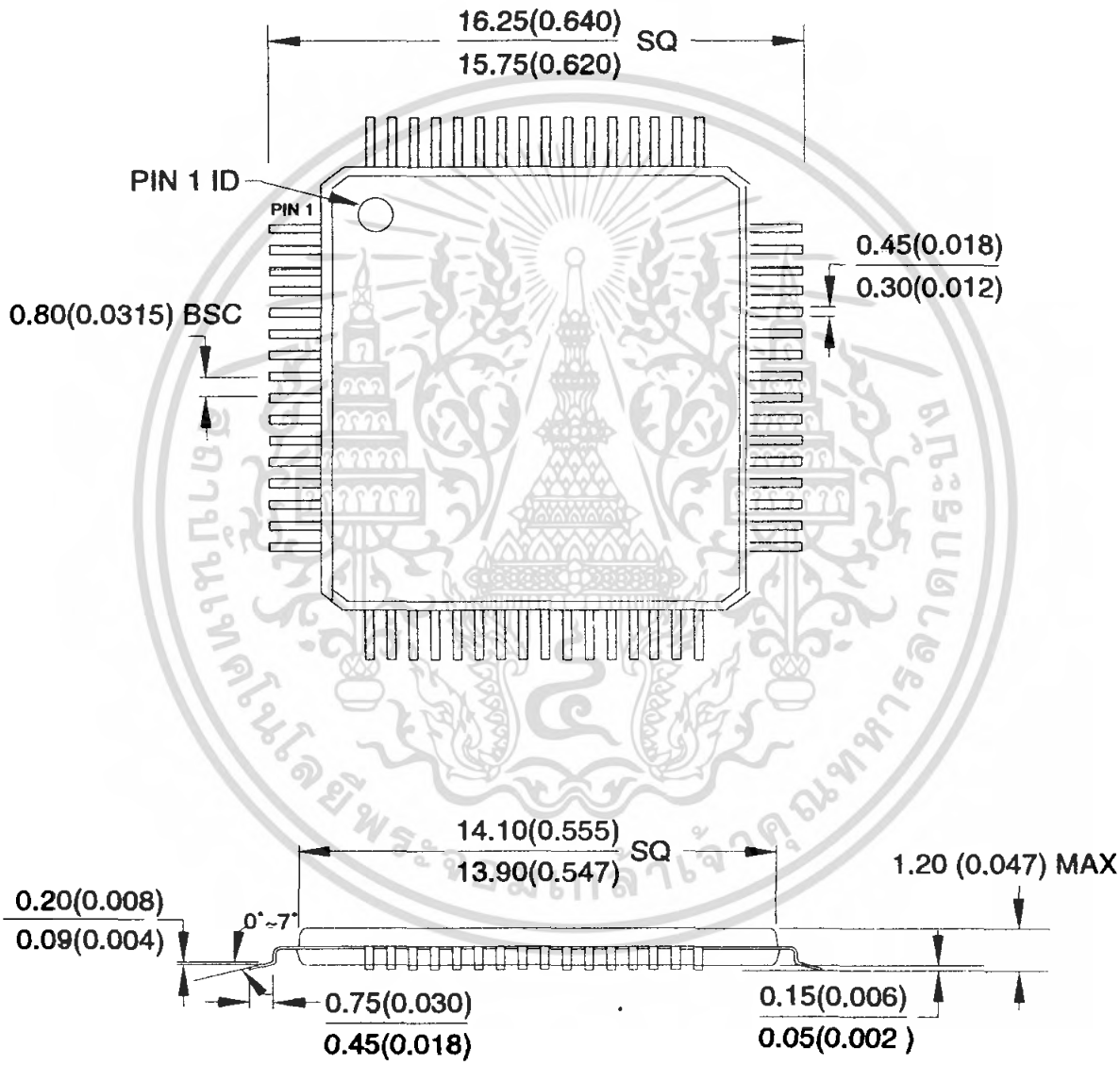
Package Type	
64A	64-Lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

64A

64-lead, Thin (1.0 mm) Plastic Quad Flat Package
(TQFP), 14x14mm body, 2.0mm footprint, 0.8mm pitch.
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 AEB



*Controlliing dimension: millimeter

REV. A 04/11/2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Product Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2001.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL®, AVR® and AVR Studio® are the registered trademarks of Atmel.

Microsoft®, Windows® and Windows NT® are the registered trademarks of Microsoft Corporation.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.

2467AS–08/01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้