



ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

รายงานโครงการหมายเลข EE 2005-57

เครื่องพันขดลวด
Coil Wiring Machine

นายมานิชย์ ทองขาว เลขประจำตัว 463040229-6
นายวรวิทย์ เหล่าพิเชษฐกุล เลขประจำตัว 463040234-3

อาจารย์ที่ปรึกษา
ผศ.ดร.กฤษ เเฉยไสย

อาจารย์ผู้ร่วมประเมินโครงการ
ดร.ชัยภักดิ์ วรรณะสาร
ผศ.อำนาจ สุขศรี

เครื่องพันขดลวด
Coil Wiring Machine

นายมานิชย์ ทองขาว เลขประจำตัว 463040229-6
นายวรวิทย์ เหล่าพิเชษฐกุล เลขประจำตัว 463040234-3

รายงานนี้เป็นรายงาน งานโครงการของนักศึกษาชั้นปีที่4 ซึ่งเสนอเป็นส่วนหนึ่ง
ในหลักสูตรวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

พ.ศ.2548

Coil Wiring Machine

Mr.Manoch Thongkao ID 463040229-6

Mr.Woravit Lheowpichedkul ID 463040234-3

This is the report of fourth year project assignment submitted in partial fulfillment of the requirement for the Degree of Bachelor of Engineering.

Department of Electrical Engineering

Faculty of Engineering , Khon Kaen University

2005

ใบประเมินผลงานโครงการ

ชื่อเรื่องภาษาไทย เครื่องพันขดลวด
ชื่อเรื่องภาษาอังกฤษ Coil Wiring Machine
ผู้เสนอโครงการ
นายมานิช ทองขาว เลขประจำตัว 463040229-6
นายวรวิทย์ เหล่าพิเชษฐกุล เลขประจำตัว 463040234-3

อาจารย์ที่ปรึกษา

(ผศ.ดร.กฤษ เจริญไสย)

อาจารย์ผู้ร่วมประเมินผล

(ดร.ชัยภักดิ์ วรรณะสาร)

(ผศ.อำนาจ สุขศรี)

ประเมินผล ณ วันที่ 10 กุมภาพันธ์ 2549

กิตติกรรมประกาศ

โครงการเรื่องเครื่องพันขดลวด (Coil wiring Machine) จัดทำขึ้นตามเงื่อนไขของการศึกษา
ในหลักสูตรปริญญาตรี คณะวิศวกรรมศาสตร์ โดยในโครงการนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี ผู้จัดทำ
ใคร่ขอแสดงความขอบคุณเป็นอย่างสูงต่อ อ.ดร.กฤษ เฉยใสย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้
รวมทั้งนักศึกษารุ่นพี่ระดับปริญญาโททุกท่านที่ได้ให้คำแนะนำและแนวความคิดต่างๆอันเป็นประโยชน์ต่อ
โครงการนี้และยังเอื้อเฟื้อเครื่องมืออุปกรณ์สถานที่ ตลอดจนช่วยแก้ปัญหาในโครงการนี้เป็นอย่างดี

ขอขอบคุณ เจ้าหน้าที่ภาควิชาวิศวกรรมไฟฟ้าทุกท่าน ที่อำนวยความสะดวกในการยืม การเบิกจ่าย
อุปกรณ์และขอใช้เครื่องมือต่างๆในภาควิชาเป็นอย่างดี

คณะผู้จัดทำ

นายมานิชย์ ทองขาว

นายวรวิทย์ เหล่าพิเชษฐกุล

บทคัดย่อ

เครื่องพินขดลวดในปัจจุบันยังใช้แรงงานคนในการพินขดลวดและเป็นปัญหาสำหรับคนพินขดลวดเองก็คือการนับรอบที่ถูกต้องและต้องใช้แรงงานคนอื่น ดังนั้นเพื่อลดปัญหานี้ จึงได้มีการนำเอาอุปกรณ์อิเล็กทรอนิกส์ที่มีขายอยู่ในประเทศไทยเข้ามาเป็นตัวช่วยในการพินขดลวดและนับรอบที่ถูกต้องได้ ซึ่งได้แก่ วงจร AVR atmega 8 , วงจรตรวจจับรอบการพิน , วงจร Drive สำหรับมอเตอร์ ซึ่งแต่ละวงจรจะมีความสำคัญที่แตกต่างกันไป

เมื่อนำวงจรทั้งหมดมาเชื่อมต่อเข้าด้วยกัน เครื่องพินขดลวดก็จะสามารถทำงานได้ โดยมีลักษณะการทำงาน คือเมื่อทำการเขียนโปรแกรมและบันทึกลงในไอซี ATmega 8 แล้ว ทำการป้อนค่าจำนวนรอบที่ต้องการพิน (input) แล้วกด enter วงจรก็จะบันทึกค่าลงไปแล้วทำการประมวลผลในตัววงจร AVR ATmega8 แล้วส่งสัญญาณไฟดีซีเพื่อให้ออเตอร์ทำงานและเซนเซอร์ที่ติดตั้งอยู่บนแกนข้อต่อมอเตอร์นั้น ก็จะทำการนับและส่งสัญญาณกลับมาให้กับวงจร AVR ATmega8 อีกครั้งหนึ่ง จากนั้นวงจร AVR ก็จะทำการเปรียบเทียบค่าที่ได้รับกับค่าที่นับได้ เมื่อค่าเท่ากันก็จะมอเตอร์ก็หยุดทำงานทันทีแล้ว ส่งผลไปยังหลอด LED seven segment ซึ่งจะแสดงผลของจำนวนรอบให้เห็น

ABSTRACT

At present, Coil wiring machine uses manually controlling which is problems for human such as counting the exact rounds and use much more time. To reduce these problems by using electronic device which is sell in Thailand. Coil wiring machine consist of AVR Atmega 8 circuit , sensor circuit and drive circuit

When we connected all of circuit .Coil wiring machine can running automatically. The principle ,First write the program and save information in IC ATmega8 , Next enter the number of rounds (O/P) for wiring. The program sent signal to drive circuit made motor run and the sensor which is installed on the joint of Coil wiring machine counted the rounds and reverse signal (I/P) to ATmega8 gain. The program compared the I/P and O/P signal. When data of two signal is equal. Motor stoped running immedately and display the number of rounds at the seven segment.

สารบัญ

	หน้า
กิตติกรรมประกาศ	ก
บทคัดย่อภาษาไทย	ข
บทคัดย่อภาษาอังกฤษ	ค
สารบัญ	ง
สารบัญรูป	จ - ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบข่ายของงาน	1 - 2
1.4 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้องและหลักการสร้าง	
2.1 วงจรตรวจวัดรอบการฟัน	3
2.2 วงจร Drive	4
2.3 โครงสร้างและองค์ประกอบภายใน IC ที่ใช้งาน	5 - 8
2.4 หลักการทำงานของระบบ	9 - 10
2.5 วิธีการใช้งาน	11
บทที่ 3 ผลการทดลอง	
3.1 วงจรตรวจจับรอบการฟัน	12
3.2 วงจร Drive	12
3.3 การควบคุมสัญญาณ PWM ด้วยการปรับค่า Duty Cycle	13 - 17
3.4 การเปรียบเทียบความคลาดเคลื่อน (Error)	18 - 22
บทที่ 4 ปัญหาและข้อบกพร่อง	23
บรรณานุกรม	24
ภาคผนวก	2 - 26

สารบัญรูป

	หน้า
รูปที่ 1. วงจรเซนเซอร์	3
รูปที่ 2. ชุดวงจร Drive สำหรับ DC Motor	4
รูปที่ 3. IC ATmega8	5
รูปที่ 4. Timer/Counter, Timing Diagram, Setting of OCF1x with Prescaler (fclk_I/O/8)	7
รูปที่ 5. Fast PWM Mode, Timing Diagram	7
รูปที่ 6. วงจรรวม	9
รูปที่ 7. Block diagram	9
รูปที่ 8. ขั้นตอนการออกแบบของระบบ	10
รูปที่ 9. Digital monitor	11
รูปที่ 10. วงจรตรวจจับรอบการฟัน	12
รูปที่ 11. สัญญาณ Pulse ที่ได้จากตัว Sensor	12
รูปที่ 12. วงจร Drive	12
รูปที่ 13. สัญญาณ Pulse ที่ได้ ของวงจร Drive	12
รูปที่ 14. Duty 0%	13
รูปที่ 15. Duty 7.7 %	13
รูปที่ 16. Duty 15.4%	14
รูปที่ 17. Duty 23.1 %	14
รูปที่ 18. Duty 46.2%	15
รูปที่ 19. Duty 50%	15
รูปที่ 20. Duty 61.54%	16
รูปที่ 21. Duty 70%	16
รูปที่ 22. Duty 81%	17
รูปที่ 23. Duty 100%	17
รูปที่ 24. ความเร็วรอบประมาณ 200 rpm	18
รูปที่ 25. ความเร็วรอบประมาณ 400 rpm	19
รูปที่ 26. ความเร็วรอบประมาณ 700 rpm	20

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 27. เครื่องปั่นขดลวดที่เสร็จสมบูรณ์	21
รูปที่ 28. ผลิตภัณฑ์ที่ได้จากการปั่น	22

สารบัญตาราง

	หน้า
ตาราง 1. ความเร็วรอบประมาณ 200 rpm	18
ตาราง 2. ความเร็วรอบประมาณ 400 rpm	19
ตาราง 3. ความเร็วรอบประมาณ 700 rpm	20

บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

เนื่องจากในปัจจุบันพบว่าตามโรงงานอุตสาหกรรมขนาดเล็กตามร้านรับซ่อมอุปกรณ์ไฟฟ้าที่เกี่ยวข้องกับการพันขดลวดรวมทั้งในสถาบันการศึกษาต่างๆก็ยังต้องใช้เครื่องพันขดลวดด้วยแรงงานคน ข้อดีคือไม่เปลืองทรัพยากร แต่ข้อเสียคือเมื่อพันขดลวดที่รอบมากๆทำให้การนับรอบผิดพลาดและทำให้ร่างกายอ่อนเพลียจากการทำงานและต้องการให้อุปกรณ์สมัยเก่ามาเป็นอุปกรณ์สมัยใหม่ ดังนั้นจึงได้นำเทคโนโลยีจำพวกอิเล็กทรอนิกส์ที่มีอยู่ในประเทศไทยเข้ามาอำนวยความสะดวกสร้างเครื่องพันขดลวด อันได้แก่วงจร AVR ATmega 8 , วงจรตรวจจับรอบการหมุน (Sensor) , วงจรแหล่งจ่าย Supply DC เป็นต้น

เมื่อนำวงจรต่างๆมารวมกันทำให้เครื่องพันขดลวดสามารถทำงานได้ โดยที่ป้อนจำนวนรอบที่ต้องการพันเข้าไป ค่านี้จะถูกเก็บไว้ในวงจร AVR ซึ่งได้มีการเขียนโปรแกรมไว้เรียบร้อยแล้ว จากนั้นเมื่อกดปุ่ม Run มอเตอร์ก็จะหมุน ทำให้เซ็นเซอร์ที่ได้ติดตั้งไว้ นับจำนวนรอบ โดยที่ 1 รอบ เท่ากับ 4 Pulse และนำค่าที่ได้นี้กลับเข้าสู่วงจร AVR อีกครั้งหนึ่ง จากนั้นวงจรก็จะแสดงผลทางหลอด seven segment พร้อมทั้งทำการเปรียบเทียบค่า input ที่ป้อนเข้าไปกับค่าที่เซ็นเซอร์ส่งผลมาจนครบตามจำนวนรอบที่ตั้งไว้

1.2 วัตถุประสงค์

1. ศึกษาวิธีการควบคุมมอเตอร์ด้วยวงจรควบคุมแรงดัน
2. ศึกษาการทำงานของอุปกรณ์เซ็นเซอร์
3. ศึกษาการนำอุปกรณ์เซ็นเซอร์มาสร้างเป็นวงจรตรวจจับจำนวนรอบการหมุน
4. ศึกษา Microcontroller AVR ATmega8 สำหรับการทำงานในส่วนของ Timer/Counter, การสร้างสัญญาณ PWM และ การแสดงผลด้วย seven segment

1.3 ขอบข่ายของงาน

1. ดัดแปลงโครงสร้างของเครื่องพันขดลวด
2. ติดตั้ง DC motor 12 V. เข้ากับเครื่องพันขดลวด
3. สร้างวงจร Supply 12 V และ 5 V.
4. สร้างวงจรตรวจจับรอบการพัน โดยใช้ Photosensor
5. สร้างชุด Monitor Control (Timer/Conter, PWM, Display)
6. สร้างวงจร Drive

1.4 ผลที่คาดว่าจะได้รับ

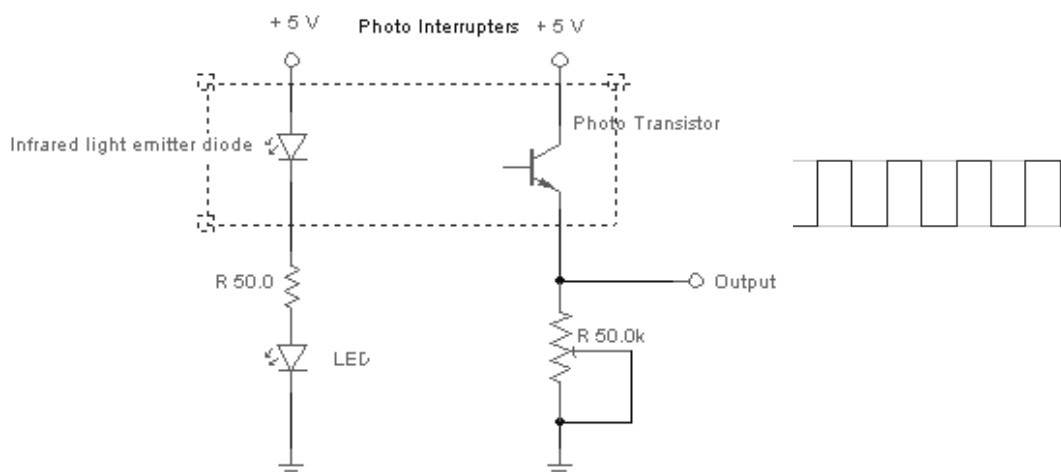
1. สามารถนำอุปกรณ์เซ็นเซอร์มาประยุกต์ใช้งานได้
2. สามารถนำ DC Motor มาประยุกต์ใช้งานได้
3. สามารถควบคุมการทำงานของ DC Motor ได้
4. สามารถนำ Microcontroller ไปประยุกต์ใช้งานได้
5. สามารถนำเครื่องพิมพ์ลงตลับไปใช้ในงานอุตสาหกรรมขนาดเล็กได้

บทที่ 2

ทฤษฎีและหลักการสร้าง

2.1 วงจรตรวจจับการพัน

จากวงจรเมื่อจ่ายแรงดัน DC +5V LED จะติดเพื่อแสดงสถานะของวงจร Infrared จะทำหน้าที่ส่งแสงไปที่ Photo Transistor ทำให้มีกระแสมาทริกที่ขาเบสของ Photo Transistor ทำให้มันทำงาน และจะมีแรงดัน Output ออกมา ส่วน R 50 Ω ทำหน้าที่เป็นตัวปรับระดับแรงดันของ Output ดังรูปวงจรข้างล่าง



รูปที่ 1 วงจรเซนเซอร์

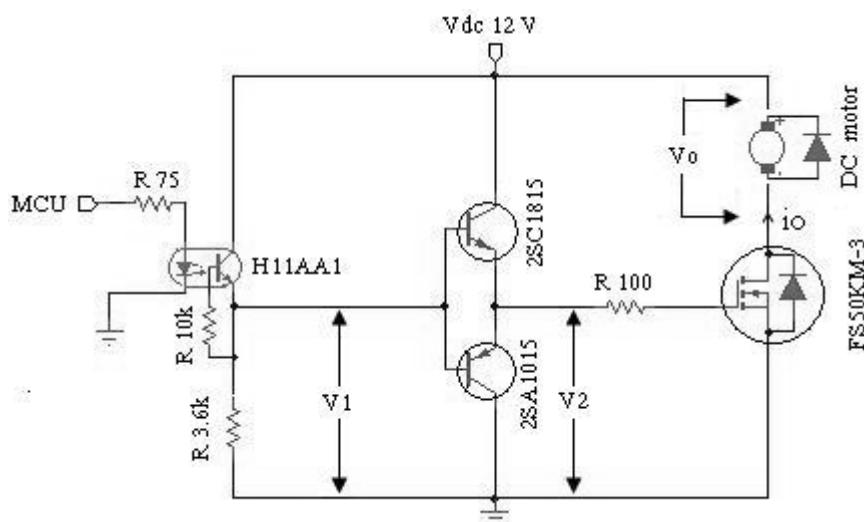
โดยในระบบการพันขดลวดนี้จะต้องมีการนับรอบการหมุนของขดลวดในแต่ละรอบ ซึ่งทำการนับรอบโดยใช้เซนเซอร์เป็นตัวบอกจำนวนรอบของการหมุน เมื่อมีแสงลอดผ่านช่องวงกลมที่ได้เจาะรูเอาไว้สำหรับนับรอบของแกนหมุน ก็จะเกิดการ Trig ของสัญญาณที่เป็นลักษณะการ ON - OFF ของแรงดันขึ้น (เหมือนกับการทำงานของสวิทช์) คือเมื่อ ON จะมีค่าเท่ากับ 5 V และเมื่อ OFF จะมีค่าเท่ากับ 0 V ดังนั้นการนำสัญญาณที่มีการ Trig ขึ้น-ลงนี้ นำไปเป็น input ให้กับวงจร AVR ที่ได้มีการเขียนโปรแกรมรับค่าไว้แล้ว ทำการนับค่าและนำไปแสดงผลต่อไป

2.2 วงจร Drive

Mosfet Drive เป็นส่วนที่ขยายสัญญาณ เพื่อที่จะไปควบคุมการ ON-OFF ประกอบไปด้วยส่วนที่สำคัญคือ

Optocouple มีหน้าที่ Isolator ไฟแรงสูงกับไฟแรงต่ำ โดยใช้แสงเป็นตัวกลางในการเชื่อมโยงระบบ โดยแสงที่ได้นี้จะไปขับในส่วนของ Current boot เพื่อขยายกระแสให้มากขึ้น

Mosfet (FS50KM-3 I=50A V= 150V) มีหน้าที่เป็นตัวเปิดปิด ซึ่งในตัว FS50KM-3 มีความเร็วในการเปิดปิดซึ่งในการเปิดปิดของ Mosfet จะอาศัยสัญญาณ PWM จากตัว Microcontroller เข้ามาเพื่อที่จะเป็นตัวเปิดปิดการทำงานของมอเตอร์



รูปที่ 2 ชุดวงจร Drive สำหรับ DC Motor

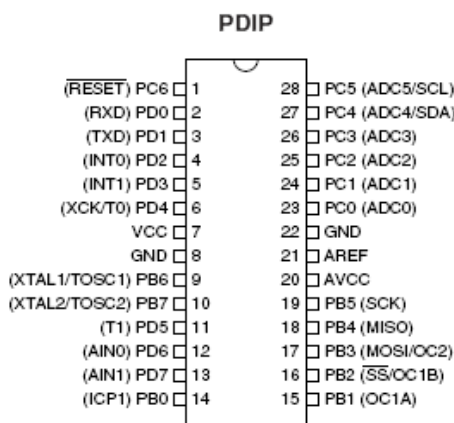
ในการขับมอเตอร์นั้น เราไม่สามารถปรับความเร็วรอบของมอเตอร์ได้โดยตรง ดังนั้นเราจึงสร้างวงจรไทรฟ์ขึ้นมาเพื่อที่สามารถปรับความเร็วรอบของมอเตอร์ได้ โดยการหลักการทำงานจะใช้ตัว Optocoupler (H11AA1) เป็นตัวแยกแรงดันของชุดไมโครคอนโทรลเลอร์กับชุดของวงจรไทรฟ์ออกจากกัน จากนั้นเมื่อมีสัญญาณ PWM เข้ามาทางขา MCU ทำให้ตัว Opto ทำงานเปรียบเสมือนทรานซิสเตอร์ ON ทำให้มีแรงดัน V_1 เท่ากับ 12V แล้วนำแรงดันไปใช้กับชุดเพิ่มกระแส ซึ่งประกอบไปด้วยทรานซิสเตอร์ 2C1815 และ 2SA1015 เข้าทางด้านขาเบสของทรานซิสเตอร์ทั้งสอง แล้วนำแรงดัน Output ทางด้านขาอีมิเตอร์ ที่มีแรงดัน V_2 เท่ากับ 11V ผ่านความต้านทาน R 100 ทำให้มีกระแสไหลไปยังมอสเฟส เมื่อมอสเฟสทำงานทำให้มีแรงดันที่ไหลผ่านมอเตอร์มีค่าแรงดัน V_o เท่ากับ 12V มอเตอร์ก็สามารถทำงานได้

ดังนั้นถ้าปรับสัญญาณ Duty cycle จากตัว MCU เข้ามา แรงดันทุกจุดจะมีค่าเท่าเดิม แต่จะส่งผลให้มอเตอร์สามารถหมุนช้าหรือเร็วได้

2.3 โครงสร้างและองค์ประกอบภายใน IC ที่ใช้งาน

ในการพัฒนาขดลวดที่ต้องการให้หมุนอัตโนมัติ โดยจะต้องใช้ไอซีที่มีความสามารถในการบันทึกและโปรแกรมได้ รวมทั้งยังมีฟังก์ชันต่างๆที่เราต้องการใช้งานซึ่งมีอยู่ในตัวไอซีและเป็นฟังก์ชันที่จำเป็นสำหรับการประดิษฐ์เครื่องพัฒนาขดลวดนี้ เช่น โหมดฟังก์ชัน PWM , โหมดฟังก์ชัน Timer/Counter ดังนั้นภายในตัวไอซีจะมีตำแหน่งขาที่นำไปใช้งานดังนี้

AVR ATmega8



รูปที่ 3. IC ATmega8

Alternate Functions of Port B ลักษณะการทำงานของแต่ละ Pins

- XTAL2/TOSC2 – PB 7
Chip clock Oscillator pin 2. เป็นขาที่ใช้สำหรับต่อ crystal Oscillator หรือ Low-frequency crystal Oscillator เมื่อใช้ขานี้เป็น clock จะไม่สามารถนำไปใช้เป็นขา I/O ได้
- XTAL1/TOSC1 – PB 6
Chip clock Oscillator pin 1. เป็นขาที่ใช้สำหรับใช้งาน RC Oscillator เมื่อต้องการ clock sources ที่นอกเหนือไปจาก clock sources ภายใน
- SCK – PB5 ใช้เป็นขา I/O
- MISO – PB 4 ใช้เป็นขา I/O
- MOSI/OC2 – PB 3 ใช้เป็นขา I/O และใช้เป็นขาสำหรับ output ของสัญญาณ PWM
- SS/OC1B – PB 2 ใช้เป็นขา I/O และใช้เป็นขาสำหรับ output ของสัญญาณ PWM
- OC1A – PB 1 ใช้เป็นขา I/O และใช้เป็นขาสำหรับ output ของสัญญาณ PWM
- ICP1 – PB 0 ใช้เป็นขา I/O

Alternate Functions of Port C ลักษณะการทำงานของแต่ละ Pins

- RESET – PC 6 เป็นขาที่ใช้สำหรับการ Reset การทำงานของไมโครคอนโทรลเลอร์
- SCL/ADC5 – PC 5 ใช้เป็นขา I/O
- SDA/ADC4 – PC 4 ใช้เป็นขา I/O
- ADC3 – PC 3 ใช้เป็นขา I/O
- ADC2 – PC 2 ใช้เป็นขา I/O
- ADC1 – PC 1 ใช้เป็นขา I/O
- ADC0 – PC 0 ใช้เป็นขา I/O

Alternate Functions of Port D ลักษณะการทำงานของแต่ละ Pins

- AIN1 – PD 7 ใช้เป็นขา I/O
- AIN0 – PD 6 ใช้เป็นขา I/O
- T1 – PD 5 เป็นขาที่ใช้สำหรับทำงานในส่วนของ Timer/Counter1
- XCK/T0 – PD 4 เป็นขาที่ใช้สำหรับทำงานในส่วนของ Timer/Counter0
- INT1 – PD 3 ใช้เป็นขา I/O
- INT0 – PD 2 ใช้เป็นขา I/O
- TXD – PD 1 ใช้เป็นขา I/O
- RXD – PD 0 ใช้เป็นขา I/O

โหมด Timer/Counter

16 bit Timer/Counter1

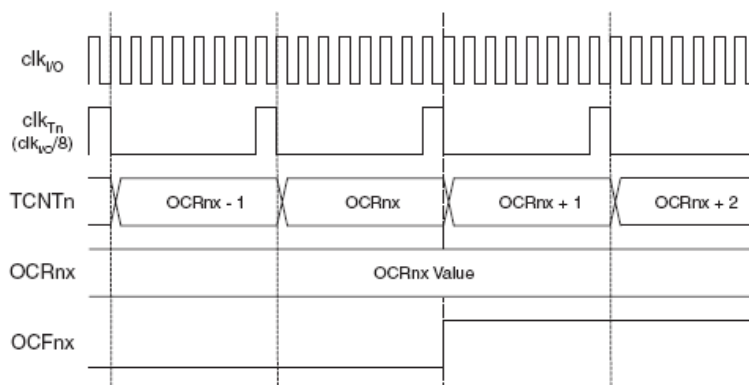
- นำมาใช้สำหรับการสร้างสัญญาณ PWM (16 – bit PWM)
- สามารถ Variable PWM Period
- Frequency Generator

โหมด PWM

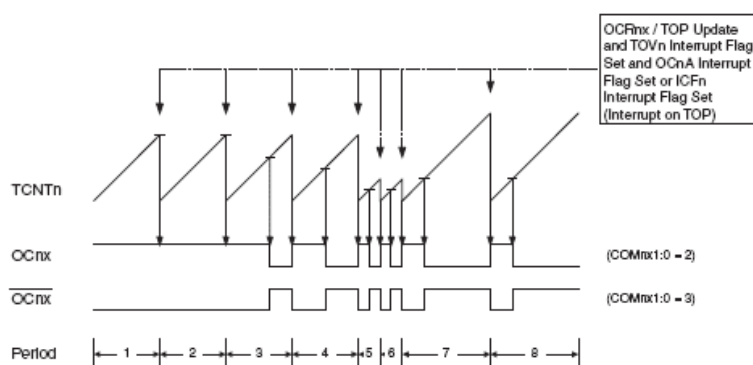
Fast PWM Mode

- The Fast Pulse Width Modulation or fast PWM mode.
- The fast PWM จะมีความแตกต่างจาก PWM แบบอื่นๆ ตรงที่จะสร้างสัญญาณสามเหลี่ยมแบบ single – slope
- Output Compare (OC1x) จะได้จากการเปรียบเทียบกันระหว่างค่าจาก TCNT1 และ OCR1x

ซึ่งโหมดทั้งสองนี้มีจำเป็นมากเพราะจะต้องนำไปปรับความเร็วรอบของมอเตอร์ให้ได้ความเร็วหลาย ๆ ระดับ โดยการปรับค่า Duty cycle ดังในรูปที่ 4 และรูปที่ 5



รูปที่ 4. Timer/Counter, Timing Diagram, Setting of OCF1x, with Prescaler (fclk_{I/O}/8)



รูปที่ 5. Fast PWM Mode, Timing Diagram

ตัวอย่างการเขียนโปรแกรมสำหรับสร้างสัญญาณ PWM

- PWM แบบ Fast mode 16-bit
- Output ออกที่ PORTB, PINB1(OC1A)
- กำหนดความถี่ของสัญญาณ PWM ได้
- สามารถปรับค่า Duty cycle ได้

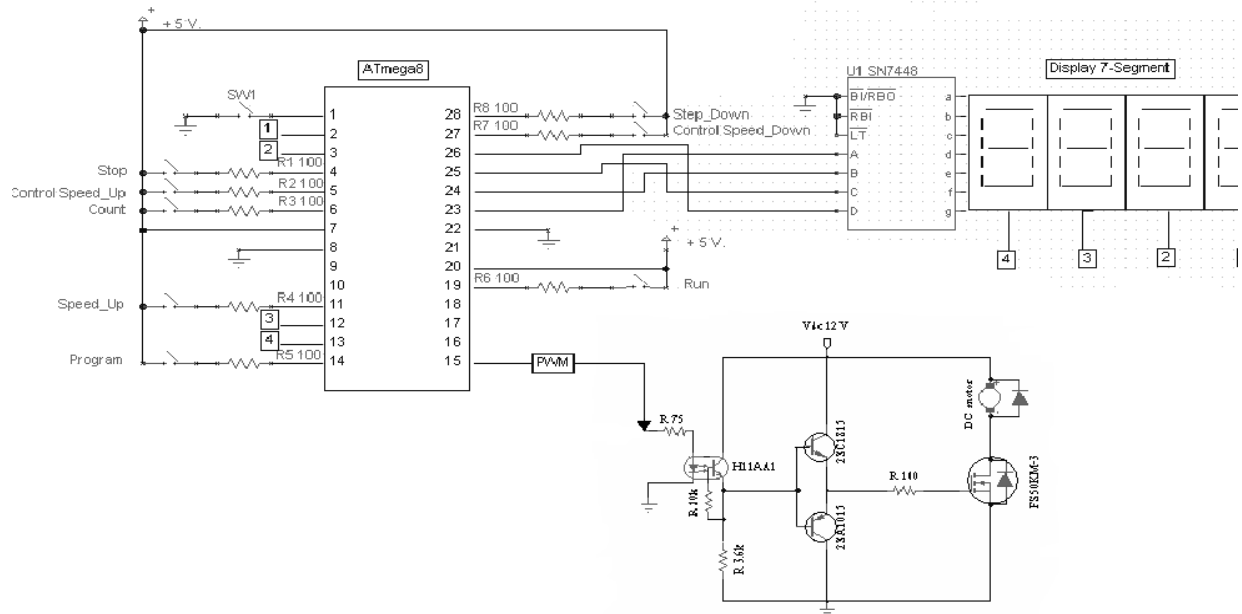
```

/*Chip type           : ATmega8
Program type          : Application
Clock frequency       : 1.000000 MHz
Memory model         : Small
External SRAM size   : 0
Data Stack size      : 256
  
```

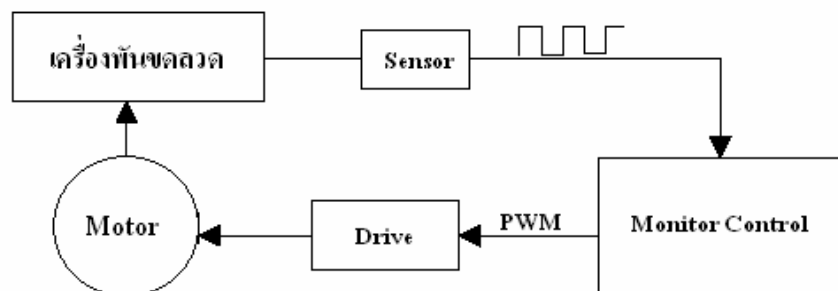
```
***** /
#include <mega8.h>
void main (void)
{
PORTB=0x00;
DDRB=0b00000010;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: ..... kHz
// Mode: Fast PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x81;
TCCR1B=0b00001001;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
    while (1)
    {
        PORTB.1=OCR1A;
    }
}
```

2.4 หลักการทำงานของระบบ

ในการผันขดลวดจะต้องประกอบไปด้วยชุดวงจรไดรฟ์และชุด วงจร microcontroller มารวมเข้าด้วยกันจึงจะสามารถทำการผันขดลวดได้ตามต้องการ ดังรูปที่ 6 เป็นรูปวงจรรวมที่นำไปต่อใช้งานของระบบการผันขดลวด ซึ่งเราสามารถอธิบายระบบการทำงานโดยใช้ Block diagram ดังรูปที่ 7 ข้างล่างดังนี้



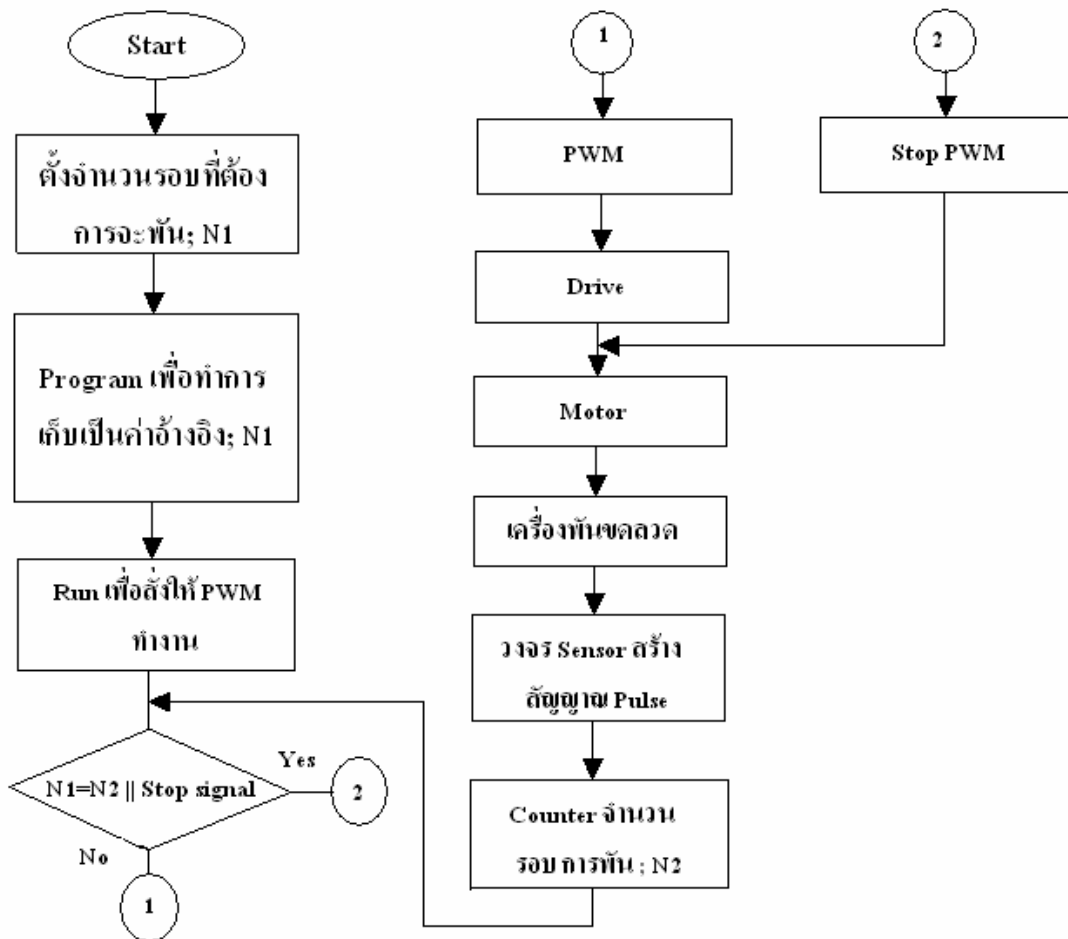
รูปที่ 6. วงจรรวม



รูปที่ 7. Block diagram

การทำงานของระบบจะเริ่มจาก Monitor control เพื่อทำการโปรแกรมค่าที่ต้องการผันเข้าไป จากนั้นตัวMicrocontroller จะสร้างสัญญาณ PWM ออกมา เพื่อที่จะนำสัญญาณนี้เข้าสู่ภาคการขับเคลื่อนซึ่งก็คือวงจรไดรฟ์ เพื่อนำไปขับมอเตอร์ให้หมุน จากนั้นเครื่องผันขดลวดก็จะเริ่มทำการผัน ทำให้ตัวเซนเซอร์ที่ติดอยู่

สร้างเป็นสัญญาณ pulse ขึ้นมาตามจำนวนรอบที่มอเตอร์หมุนและจะเป็น feed back สัญญาณกลับเข้าไป เพื่อที่จะเปรียบเทียบกับค่าที่ต้องการพันเริ่มแรก เมื่อค่าเท่ากันจะทำให้ตัว Microcontroller หยุดจ่ายสัญญาณ PWM ทันที

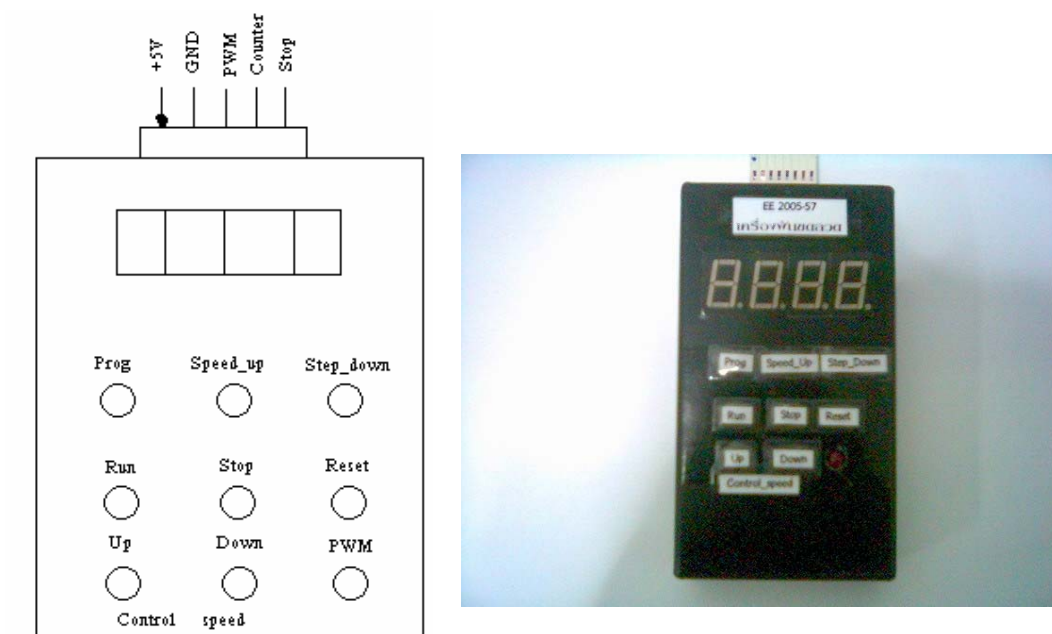


รูปที่ 8. ขั้นตอนการออกแบบของระบบ

หลักการทำงาน

เมื่อเปิดเครื่อง จะเข้าสู่โหมดตั้งจำนวนรอบที่ต้องการพัน จากนั้นเมื่อเครื่องมีการสั่งโปรแกรม ค่าจำนวนรอบที่ตั้งไว้จะถูกเก็บเอาไว้เป็นค่าอ้างอิง (N1) จากนั้นเมื่อมีการสั่ง RUN ตัว Microcontroller ก็จะสร้างสัญญาณ PWM ออกมาซึ่งจะเข้าไปอยู่ในโหมดของการพัน โปรแกรมจะสั่งให้เปรียบเทียบว่า N1=N2 และมีสัญญาณ stop หรือไม่ตลอดเวลา ซึ่งถ้าใช้ก็จะทำการหยุดสัญญาณ PWM แต่ถ้าไม่ก็จะส่งสัญญาณ PWM ออกไปสู่วงจรไฟฟ้าทำการขับมอเตอร์ให้หมุน เครื่องพันเชดลวดก็จะหมุนด้วย ทำให้วงจร sensor ทำการสร้างสัญญาณ Pulse ตามจำนวนรอบที่พัน จากนั้นนำสัญญาณที่ได้ขึ้นไปเป็นสัญญาณ counter (N2) เพื่อไปเปรียบเทียบกับ N1 โดยการทำงานจะวนลูปไปเรื่อย ๆ จนกว่าสมการ N1=N2 ที่เขียนโปรแกรมไว้มีค่าเท่ากันหรือมีสัญญาณ stop เข้ามา

2.5 วิธีใช้งาน



รูปที่ 9. Digital monitor

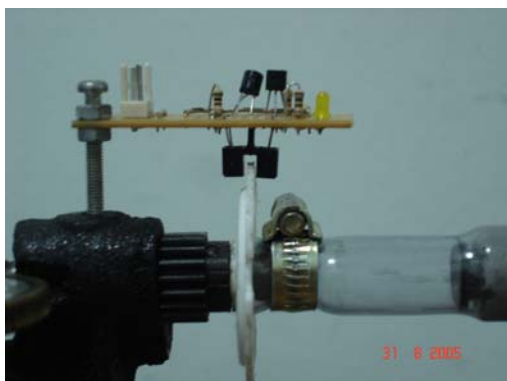
การทำงาน

1. ทำการ Reset เครื่องก่อนการทำงาน เพื่อเตรียมความพร้อมก่อนการทำงาน
2. กำหนดรอบการฟันจากปุ่ม Speed_up , Step_down
3. Program ค่า เพื่อเก็บค่าจำนวนรอบที่ต้องการฟันเอาไว้ จากปุ่ม Prog
4. Run เป็นการสั่งให้ Microcontroller สร้างสัญญาณ PWM ทำงาน จากปุ่ม Run
5. เมื่อเครื่องฟันทำการฟันขดลวดจะมีสัญญาณ Counter เข้ามา monitor ก็จะแสดงผลการฟัน
6. PWM จะหยุดทำงานเมื่อ รอบการฟันเท่ากับรอบที่ตั้งไว้
7. PWM จะหยุดทำงานเมื่อ กดปุ่ม Stop หรือจากการที่มีสัญญาณมาจากภายนอกเช่นกรณีที่ขดลวดขาด
8. การ Control speed จะเป็นการควบคุมการทำงานของสัญญาณ PWM จากการปรับค่า Duty Cycle ทำให้สามารถควบคุมความเร็วของ Motor ได้ตามต้องการ จากปุ่ม Up , Down

บทที่ 3

ผลการทดลอง

3.1 วงจรตรวจจับรอบการฟัน



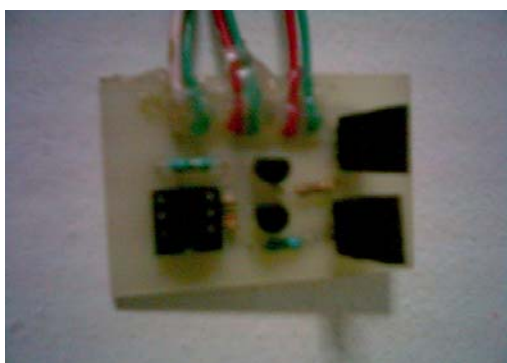
รูปที่ 10. วงจรตรวจจับรอบการฟัน



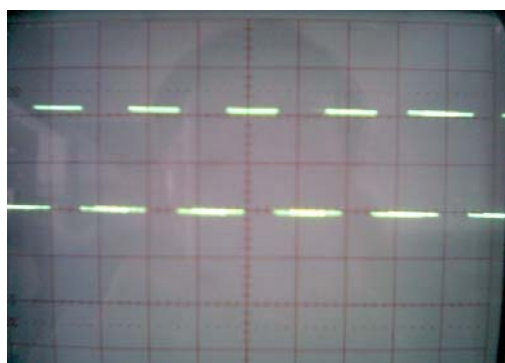
รูปที่ 11. สัญญาณ Pulse ที่ได้จากตัว Sensor
ใช้ Time/DIV=0.1 mS
Volt/DIV= 2.5 V

ทำการวัดสัญญาณ output จากตัว sensor เมื่อมีแสงลอดผ่านแผ่นกั้นและไม่ลอดผ่านแผ่นกั้นจะได้สัญญาณในลักษณะที่เป็นรูป pulse

3.2 วงจร Drive



รูปที่ 12. วงจร Drive

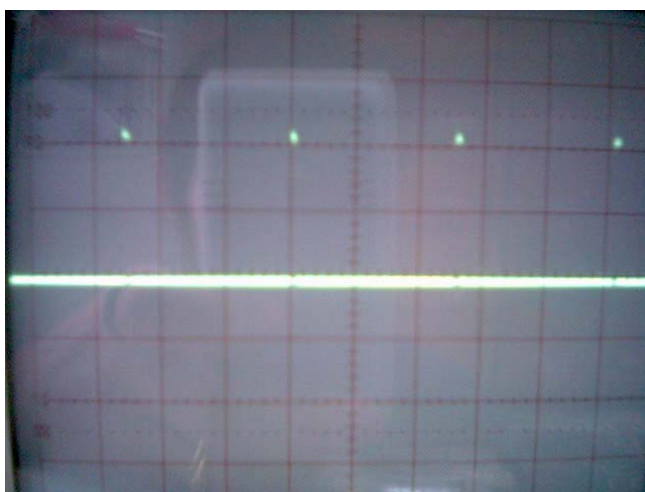


รูปที่ 13. สัญญาณ Pulse ที่ได้จากวงจร
Drive ใช้ Time/DIV= 0.1 mS
Volt/DIV= 5 V

ทำการวัดสัญญาณ output เมื่อสมมุติโดยจำลองการจ่ายสัญญาณ Square wave ที่มีความถี่ 1kHz ไปจนถึง 10kHz ป้อนเป็น input เข้าไปตัว Optocoupler ที่เส้นสีแดงและป้อนแรงดันกระแสตรง 12V ให้กับวงจรขับเคลื่อนที่เส้นสีแดงเขียว1 จากนั้นนำออสซิลโลสโคปวัดสัญญาณ Output ที่เส้นสีแดงเขียว 2 ทำให้ได้สัญญาณและมีแรงดัน 12V เพื่อนำไปใช้งานกับมอเตอร์ของเครื่องฟันชดลวด

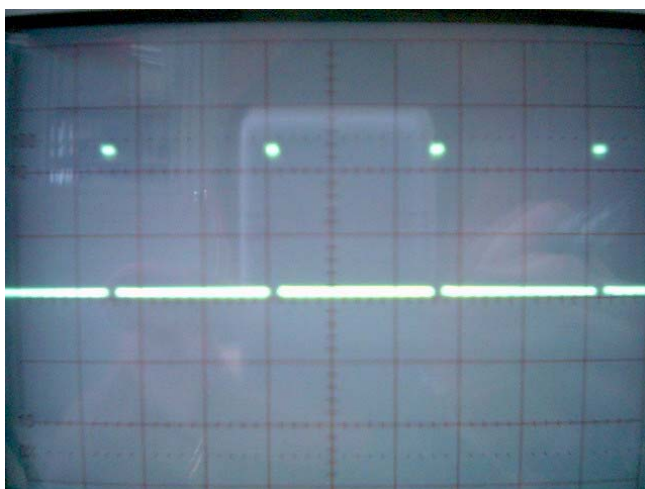
3.3 การควบคุมสัญญาณ PWM ด้วยการปรับค่า Duty Cycle

ทำการวัดสัญญาณที่ขา Pb1 ของ Microcontroller โดยกำหนดให้มีความถี่คงที่ 3.85 kHz จากนั้นทำการเปลี่ยนแปลงค่า Duty cycle



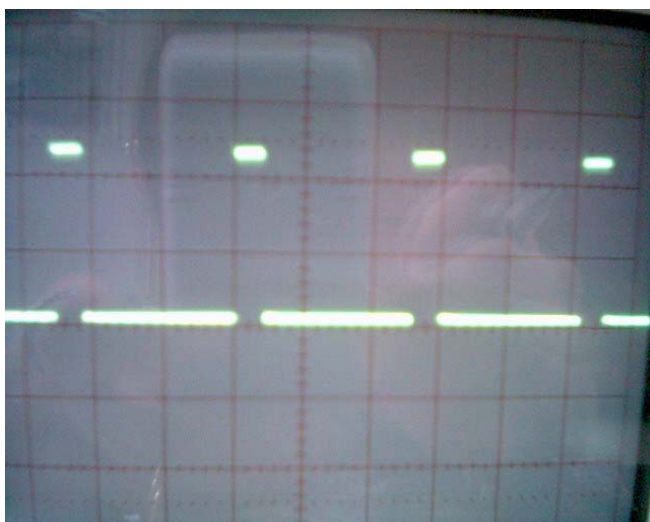
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V.}$

รูปที่ 14. Duty 0%



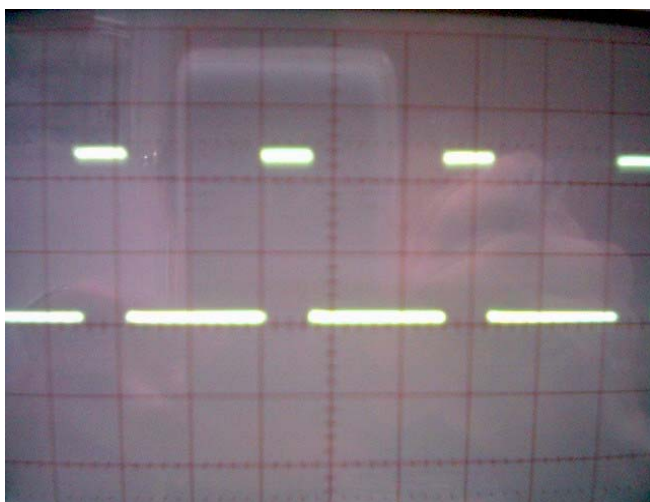
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V.}$

รูปที่ 15. Duty 7.7 %



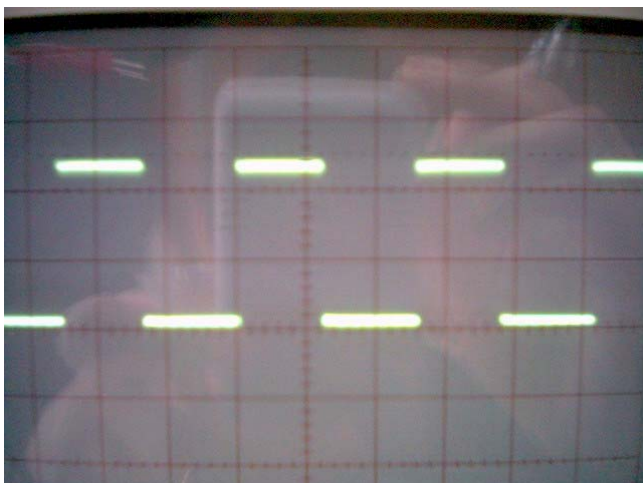
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}$

รูปที่ 16. Duty 15.4%



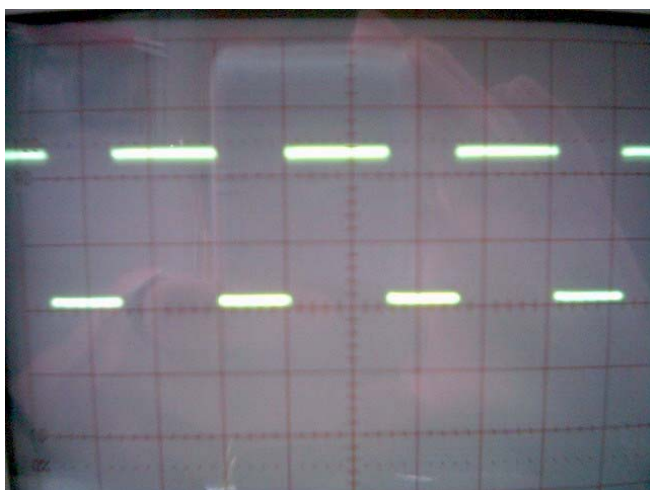
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}$

รูปที่ 17. Duty 23.1 %



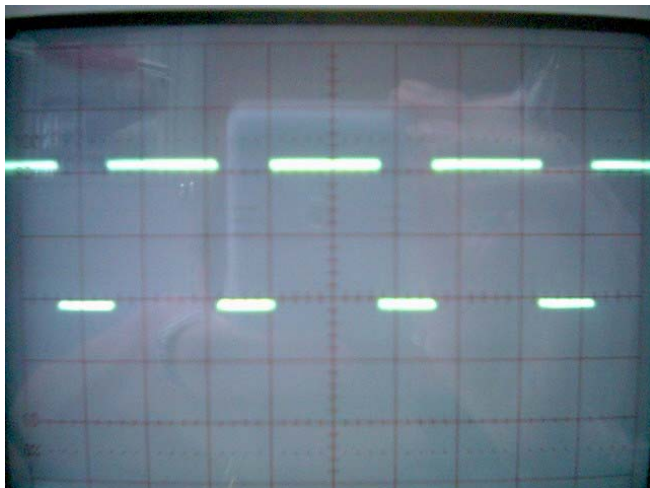
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}.$

รูปที่ 18. Duty 46.2%



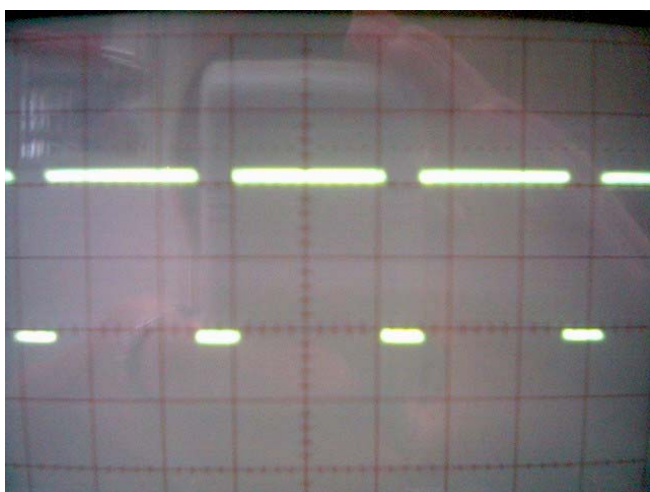
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}.$

รูปที่ 19. Duty 50%



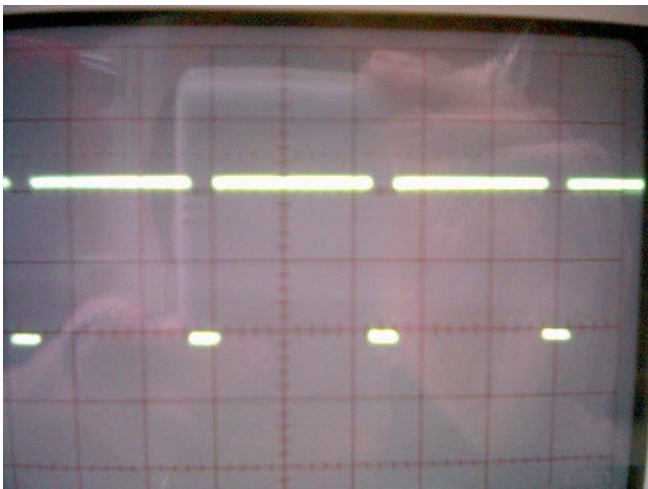
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V.}$

รูปที่ 20 Duty 61.54%



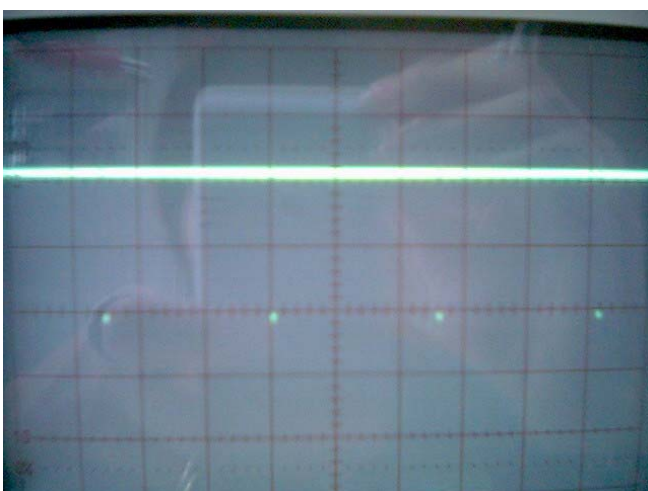
เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V.}$

รูปที่ 21. Duty 70%



เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}.$

รูปที่ 22. Duty 81%



เมื่อใช้ $\text{Time/DIV} = 0.1 \text{ mS}$
 $\text{Volt/DIV} = 2.5 \text{ V}$
 $V_{p-p} = 5.00 \text{ V}.$

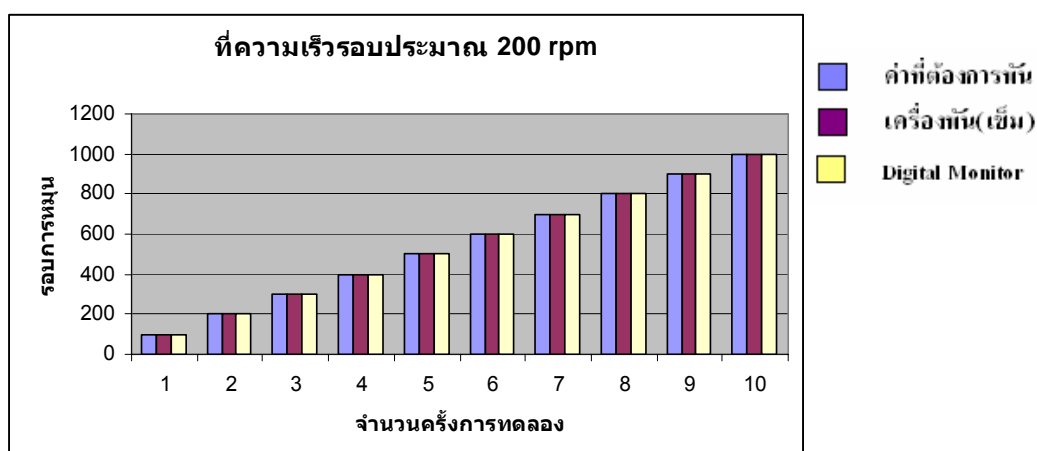
รูปที่ 23. Duty 100%

3.4 การเปรียบเทียบความคลาดเคลื่อน (Error)

จากการทดลองที่เป็นการทดลองใช้งานจริงที่ความเร็วรอบต่างๆ โดยเลือกพิจารณาอย่างคร่าวๆ 3 ค่า คือ 200 , 400 และ 700 rpm ในแต่ละความเร็วจะทดลองการหมุน 10 ครั้ง โดยการโปรแกรมจำนวนรอบที่ต้องการ จากนั้นทำการ run จนครบรอบที่ตั้งไว้และพิจารณาที่เข็มบอกจำนวนรอบของเครื่องพั่นกับพิจารณาที่ Digital monitor แล้วนำค่าทั้ง 3 มาหาความคลาดเคลื่อน (Error)

ที่ความเร็วรอบประมาณ 200 rpm		
ค่าที่ต้องการพั่น	เครื่องพั่น(เข็ม)	Digital Monitor
100	100	100
200	200	200
300	300	300
400	400	400
500	500	500
600	600	600
700	700	700
800	800	800
900	900	900
1000	1000	1000
Error ระหว่างค่าที่ต้องการพั่นกับเครื่องพั่น(เข็ม) ประมาณ 0%		
Error ระหว่างค่าที่ต้องการพั่นกับ Digital Monitor ประมาณ 0%		
Error ระหว่างเครื่องพั่น(เข็ม)กับ Digital Monitor ประมาณ 0%		

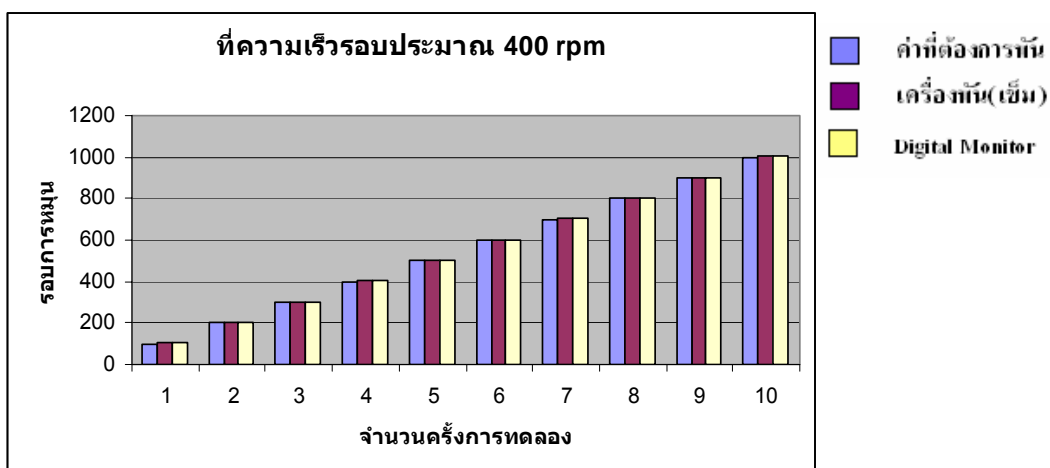
Table 1. ความเร็วรอบประมาณ 200 rpm



รูปที่ 24. ความเร็วรอบประมาณ 200 rpm

ที่ความเร็วรอบประมาณ 400 rpm		
ค่าที่ต้องการฟัน	เครื่องฟัน(เข็ม)	Digital Monitor
100	102	102
200	202	202
300	302	302
400	402	402
500	502	502
600	602	602
700	702	702
800	802	802
900	902	902
1000	1003	1003
Error ระหว่างค่าที่ต้องการฟันกับเครื่องฟัน(เข็ม) ประมาณ 2%		
Error ระหว่างค่าที่ต้องการฟันกับ Digital Monitor ประมาณ 2%		
Error ระหว่างเครื่องฟัน(เข็ม)กับ Digital Monitor ประมาณ 0%		

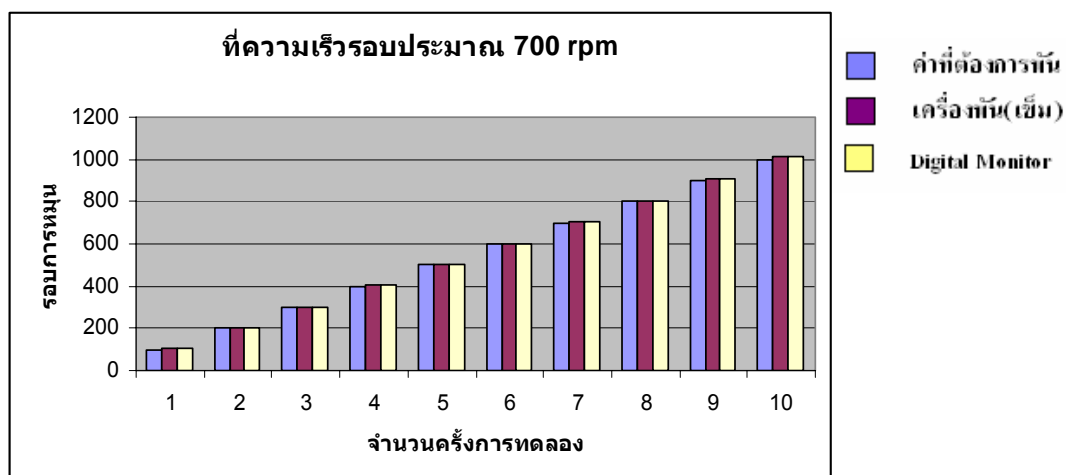
Table 2. ความเร็วรอบประมาณ 400 rpm



รูปที่ 25. ความเร็วรอบประมาณ 400 rpm

ที่ความเร็วรอบประมาณ 700 rpm		
ค่าที่ต้องการฟัน	เครื่องฟัน(เข็ม)	Digital Monitor
100	102	102
200	204	204
300	304	304
400	404	404
500	504	504
600	604	604
700	704	704
800	805	805
900	908	908
1000	1010	1010
Error ระหว่างค่าที่ต้องการฟันกับเครื่องฟัน(เข็ม) ประมาณ 5.8%		
Error ระหว่างค่าที่ต้องการฟันกับ Digital Monitor ประมาณ 5.8%		
Error ระหว่างเครื่องฟัน(เข็ม)กับ Digital Monitor ประมาณ 0%		

Table 3. ความเร็วรอบประมาณ 700 rpm



รูปที่ 26. ความเร็วรอบประมาณ 700 rpm

หมายเหตุ วัดความเร็วของมอเตอร์ด้วย Digital Tachometer ของทางภาควิชาไฟฟ้า จึงถือเป็นความเร็วมาตรฐานที่ใช้อ้างอิง

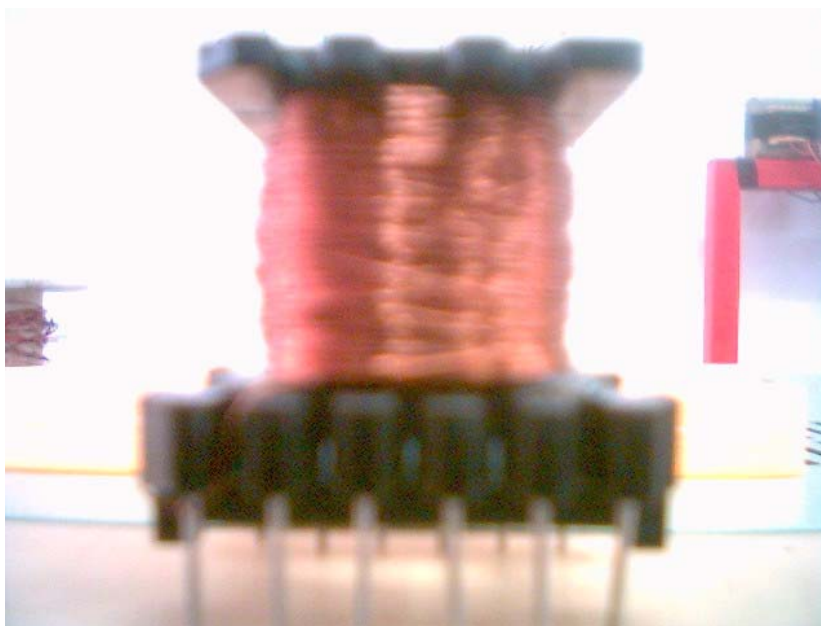
รูปเครื่องผันขดลวด



รูปที่ 27. เครื่องผันขดลวดที่เสร็จสมบูรณ์

เป็นรูปโดยรวมของเครื่องผันขดลวดที่ประกอบเข้ากับมอเตอร์ มีทั้ง Supply และตัวตัดการทำงาน เมื่อขดลวดขาดกะทันหัน

รูปของผลิตภัณฑ์ที่ได้จากการพัน



รูปที่ 28. ผลิตภัณฑ์ที่ได้จากการพัน

จากรูปเป็นแกนหม้อแปลงที่ได้จากการพันด้วยเครื่องพันขดลวด จะเห็นว่าขดลวดไม่เป็นระเบียบนัก เนื่องจากไม่มีสลิปสำหรับเลื่อนขดลวดให้เป็นระเบียบ

สรุปผลการทดลอง

จากการทดลองนำเครื่องพันขดลวดมาใช้งานจริง เครื่องสามารถพันขดลวดได้สูงสุด 9999 รอบ ขดลวดที่พันต้องเป็นขดลวดขนาดเล็กเท่านั้น ถ้าเป็นขดลวดขนาดใหญ่จะไม่สามารถพันได้เนื่องจาก motor มี torque ต่ำ ในการนับรอบจะนับได้แม่นยำ 100% ที่ความเร็วรอบประมาณ 200 rpm ซึ่งถ้าเร็วมากเวลาหยุดจะทำให้รอบเกิน ซึ่งเกิดจากแรงเฉื่อยของ motor และผลิตภัณฑ์ที่ได้จะไม่ค่อยมีระเบียบมาก

บทที่ 4

ปัญหาและแนวทางการแก้ไข

ปัญหา

1. เครื่องไม่สามารถทำการนับได้ตรงกับรอบที่พันได้จริง ถ้าสั่งให้เครื่องทำการพันด้วยความเร็วรอบที่สูง คาดว่าน่าจะเกิดจากการเขียนโปรแกรม เนื่องจากโปรแกรมมีความยาวมาก ทำให้ต้องใช้เวลาในการวนลูปในส่วนของการแสดงผลและเมื่อมอเตอร์หมุนเร็วมาก การวนลูปเพื่อการแสดงผลจึงทำไม่ทัน
2. ในการทดสอบใช้งานจริง การใช้งานยังไม่สะดวกเท่าที่ควร
3. ลักษณะของเส้นลวดที่ได้ทำการพันดูไม่ค่อยเป็นระเบียบนัก เนื่องจากไม่มีตัวจัดระเบียบของขดลวด
4. ความสามารถในการทำงานของเครื่องค่อนข้างมีขีดจำกัด เช่น ไม่สามารถพันขดลวดที่มีขนาดใหญ่ได้ เนื่องจาก Torque ของมอเตอร์ต่ำ

แนวทางการแก้ไข

1. แนวทางแก้ไขจะต้องศึกษาการเขียนโปรแกรมให้มากกว่านี้
2. ควรเพิ่มตัว Sensor เพื่อวัดแรงตึง (Tention) ของขดลวด

บรรณานุกรม

1. ผศ.วิชัย ประเสริฐเจริญสุข ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น , “Digital Electronics”
2. หนังสือคู่มือไอซี TTL
3. รศ.ยีน ภู่วรรณ ภาควิชาวิศวกรรมไฟฟ้า, “อิเล็กทรอนิกส์อุตสาหกรรม”
4. ผศ.ดร.กฤษ เฉยไสย ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น, “Power Electronic”
5. <http://www.es.co.th>
6. <http://www.datasheet4u.co.th>

ภาคผนวก

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 8K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 1K Byte Internal SRAM
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and MLF package
 - Six Channels 10-bit Accuracy
 - Two Channels 8-bit Accuracy
 - 6-channel ADC in PDIP package
 - Four Channels 10-bit Accuracy
 - Two Channels 8-bit Accuracy
 - 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
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-lead PDIP, 32-lead TQFP, and 32-pad MLF
- Operating Voltages
 - 2.7 - 5.5V (ATmega8L)
 - 4.5 - 5.5V (ATmega8)
- Speed Grades
 - 0 - 8 MHz (ATmega8L)
 - 0 - 16 MHz (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C
 - Active: 3.6 mA
 - Idle Mode: 1.0 mA
 - Power-down Mode: 0.5 µA



**8-bit AVR[®]
with 8K Bytes
In-System
Programmable
Flash**

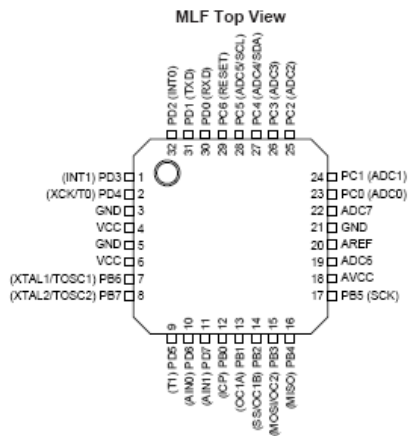
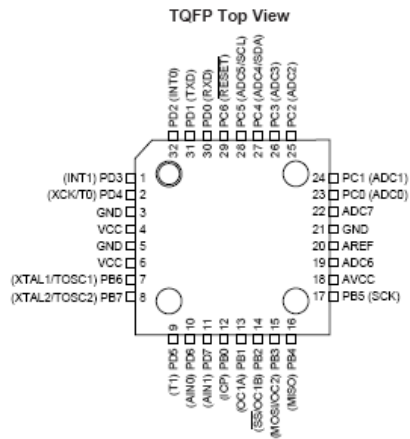
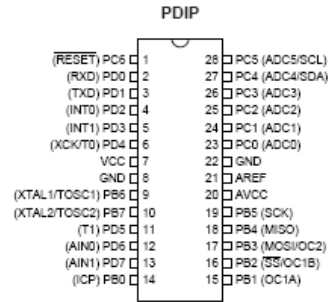
**ATmega8
ATmega8L**

Preliminary

Rev. 2486I-AVR-12/02



Pin Configurations

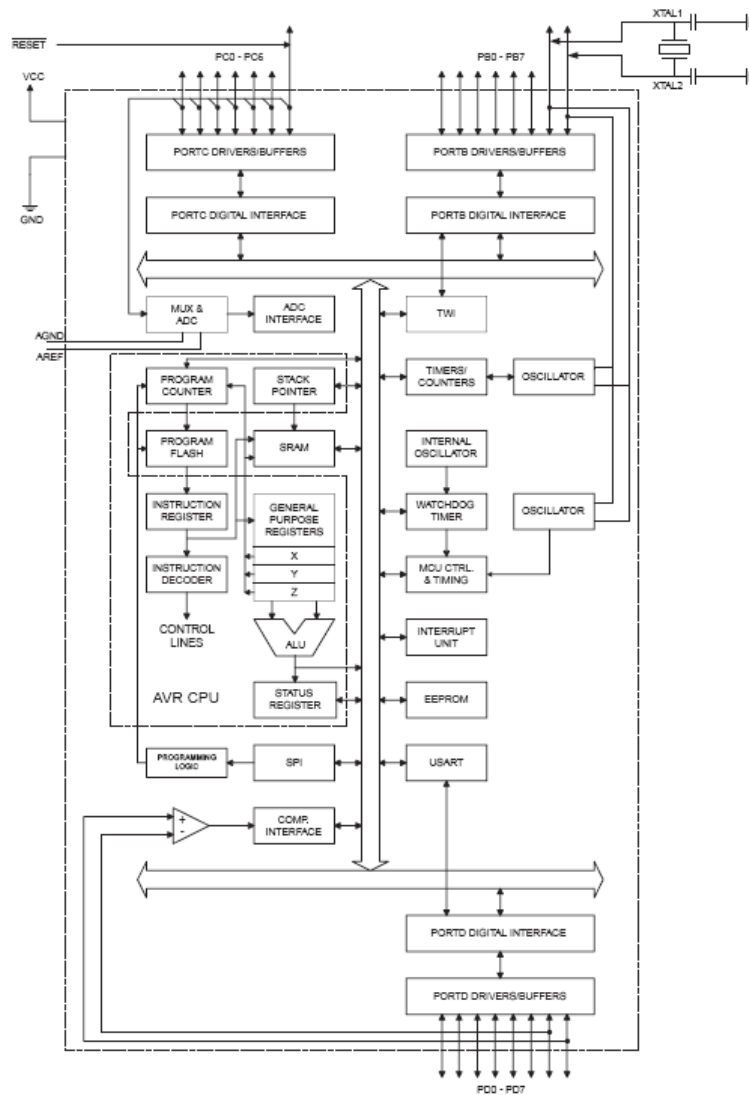


Overview

The ATmega8 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega8 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 1. Block Diagram



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

The ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and MLF packages) where four (six) channels have 10-bit accuracy and two channels have 8-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five 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.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile 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 ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 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.

Disclaimer

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

Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port B (PB7..PB0)/XTAL1/ XTAL2/TOSC1/TOSC2	<p>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.</p> <p>Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p> <p>Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.</p> <p>If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.</p> <p>The various special features of Port B are elaborated on page 56.</p>
Port C (PC5..PC0)	<p>Port C is an 7-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.</p>
PC6/ $\overline{\text{RESET}}$	<p>If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.</p> <p>If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 36. Shorter pulses are not guaranteed to generate a Reset.</p> <p>The various special features of Port C are elaborated on page 59.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega8 as listed on page 61.</p>
$\overline{\text{RESET}}$	<p>Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 36. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.

AVCC	AVCC is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that Port C (5..4) use digital supply voltage, V_{CC} .
AREF	AREF is the analog reference pin for the A/D Converter.
ADC7..6 (TQFP and MLF Package Only)	In the TQFP and MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.
About Code Examples	This datasheet contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

Segment identification

Resulting display

Description:
This device converts BCD input data into control signals for 7-segment displays.

Mode of operation:
The BCD code is fed to inputs A through D, and after decoding in the IC, provides 7-segment display (a - f) segment control data. The outputs are open-collector, but with an internal 2kΩ pull-up resistor. The decoder outputs are active-high and have a maximum low-level output sink current of 6 mA. If higher currents are required, especially for multiplex operation, additional external transistors are required. There is no internal latch.
The top horizontal segment (a) of the number 6 and the bottom horizontal segment (d) of the number 9 are not displayed. For normal operation, the pins LT (lamp test, pin 3) and BI/RBO (ripple blanking output, pin 4) are pulled high (RBI = ripple blanking input, can be either level).
All segments can be checked by taking LT low. This should activate all segments, i.e. a figure 8 should be displayed. Leading zeros in multi-digit displays are suppressed by linking the BI/RBO output of one digit with the RBI input of the place below it. As suppression of the zero in the least significant digit is not normally desirable, the RBI of this stage is left open. Trailing zeros after the decimal point can be suppressed in a similar manner. As all segments are switched off when BI/RBO is low, a display intensity control can be implemented by applying a pulse-modulated signal to this pin.

Application:
Control of 7-segment displays, especially in multiplexing.

Data:									
Propagation delay	ns	100						100	
Maximum collector current	mA	55						25	
Families:		Std	ALS	AS	F	H	L	LS	S
		●						●	

BCD-TO-7-SEGMENT DECODER/DRIVER

7448

CNA1014H (ON1387)

Photo Interrupters

■ Outline

CNA1014H is a transmissive photosensor series in which a high efficiency GaAs infrared light emitting diode is used as the light emitting element, and a high sensitivity phototransistor is used as the light detecting element. The two elements are arranged so as to face each other, and objects passing between them are detected.

■ Features

- Position detection accuracy: 0.3 mm
- With attachment positioning boss
- Fast response: t_r , $t_f = 5 \mu\text{s}$ (typ.)

■ Absolute Maximum Ratings $T_a = 25^\circ\text{C}$

	Parameter	Symbol	Rating	Unit
Input (Light emitting diode)	Reverse voltage (DC)	V_R	3	V
	Forward current (DC)	I_F	50	mA
	Power dissipation *1	P_D	75	mW
Output (Photo transistor)	Collector current	I_C	20	mA
	Collector to emitter voltage	V_{CE0}	30	V
	Emitter to collector voltage	V_{ECO}	5	V
	Collector power dissipation *2	P_C	100	mW
Temperature	Operating ambient temperature	T_{opr}	-25 to +85	$^\circ\text{C}$
	Storage temperature	T_{stg}	-40 to +100	$^\circ\text{C}$

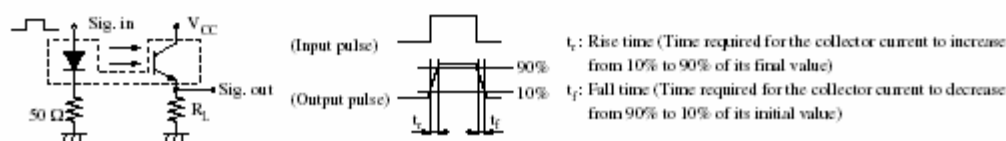
Note) *1: Input power derating ratio is 1.0 mW/ $^\circ\text{C}$ at $T_a = 25^\circ\text{C}$.

*2: Output power derating ratio is 1.33 mW/ $^\circ\text{C}$ at $T_a = 25^\circ\text{C}$.

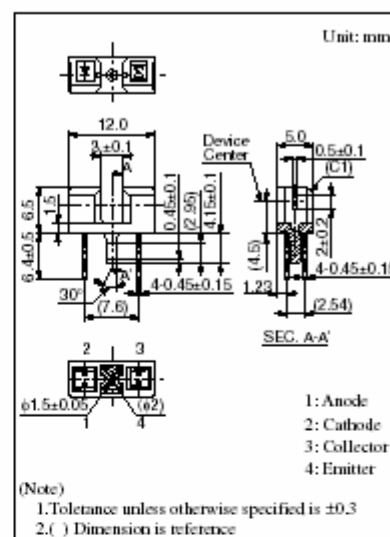
■ Electrical Characteristics $T_a = 25^\circ\text{C} \pm 3^\circ\text{C}$

	Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Input characteristics	Forward voltage (DC)	V_F	$I_F = 20\text{mA}$		1.25	1.4	V
	Reverse current (DC)	I_R	$V_R = 3\text{V}$			10	μA
Output characteristics	Collector cutoff current	I_{CEO}	$V_{CE} = 10\text{V}$		10	200	nA
Transfer characteristics	Collector current	I_C	$V_{CE} = 5\text{V}$, $I_F = 20\text{mA}$, $R_L = 100\ \Omega$	1.5		15	mA
	Collector to emitter saturation voltage	$V_{CE(sat)}$	$I_F = 40\text{mA}$, $I_C = 1\text{mA}$			0.4	V
	Response time *	t_r, t_f	$V_{CC} = 5\text{V}$, $I_C = 1\text{mA}$, $R_L = 100\ \Omega$		5		μs

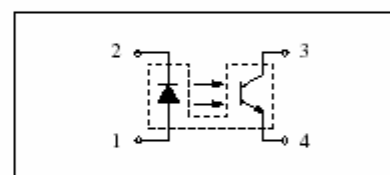
Note) *: Switching time measurement circuit

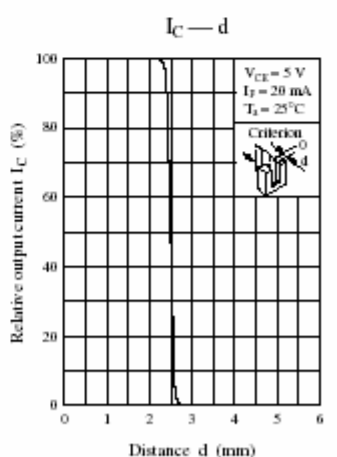
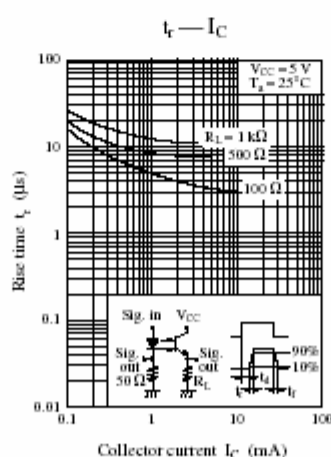
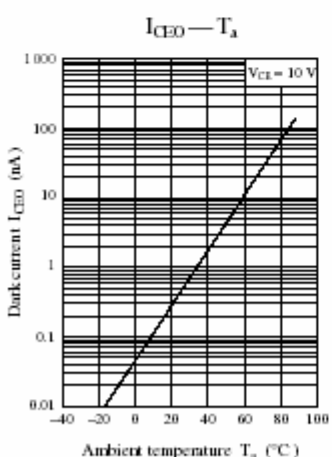
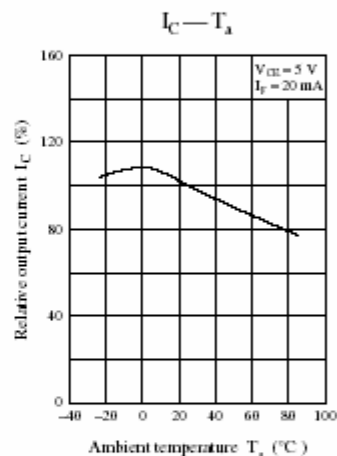
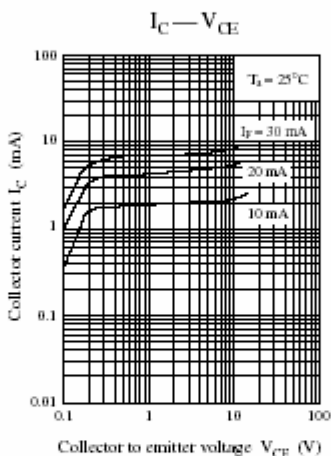
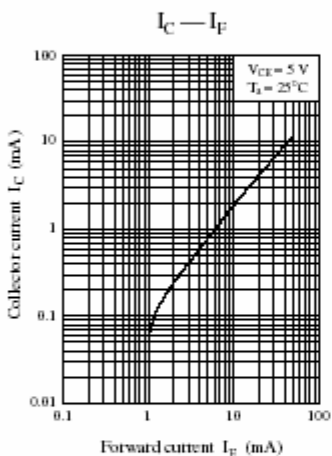
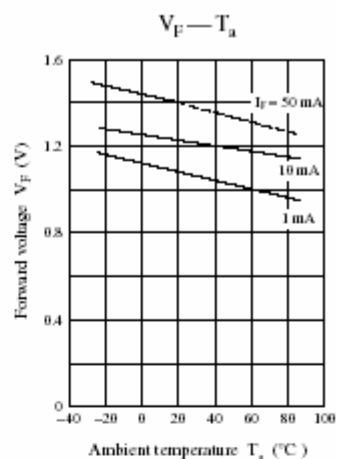
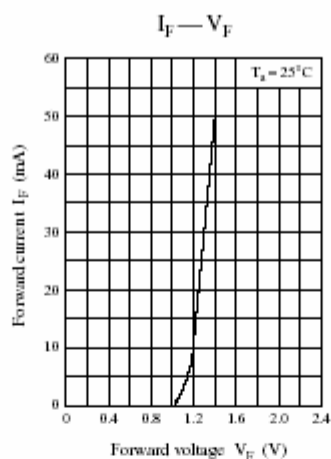
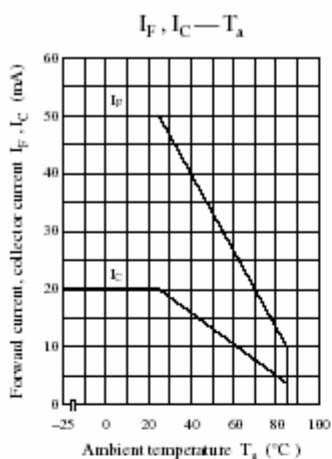


Note) The part number in the parenthesis shows conventional part number.



Internal connection





MITSUBISHI Nch POWER MOSFET

FS50KM-3

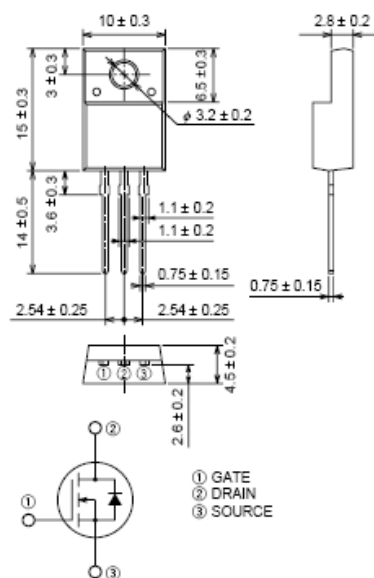
HIGH-SPEED SWITCHING USE

FS50KM-3

- 10V DRIVE
- V_{DSS} 150V
- $r_{DS(ON)}$ (MAX) $31m\Omega$
- I_D 50A
- Integrated Fast Recovery Diode (TYP.) 130ns
- V_{iso} 2000V

OUTLINE DRAWING

Dimensions in mm



TO-220FN

APPLICATION

Motor control, Lamp control, Solenoid control
DC-DC converter, etc.

MAXIMUM RATINGS ($T_c = 25^\circ\text{C}$)

Symbol	Parameter	Conditions	Ratings	Unit
V_{DSS}	Drain-source voltage	$V_{GS} = 0V$	150	V
V_{GSS}	Gate-source voltage	$V_{DS} = 0V$	± 20	V
I_D	Drain current		50	A
I_{DM}	Drain current (Pulsed)		200	A
I_{DA}	Avalanche drain current (Pulsed)	$L = 100\mu H$	50	A
I_S	Source current		50	A
I_{SM}	Source current (Pulsed)		200	A
P_D	Maximum power dissipation		35	W
T_{ch}	Channel temperature		$-55 \sim +150$	$^\circ\text{C}$
T_{stg}	Storage temperature		$-55 \sim +150$	$^\circ\text{C}$
V_{iso}	Isolation voltage	AC for 1minute, Terminal to case	2000	V
—	Weight	Typical value	2.0	g

Feb.1999

MITSUBISHI Nch POWER MOSFET

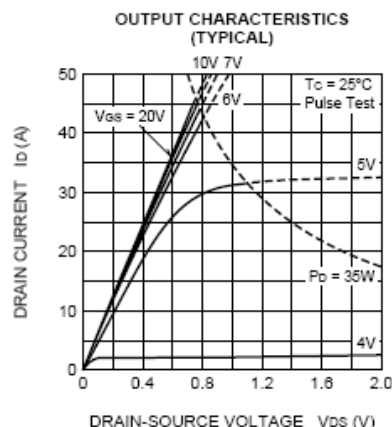
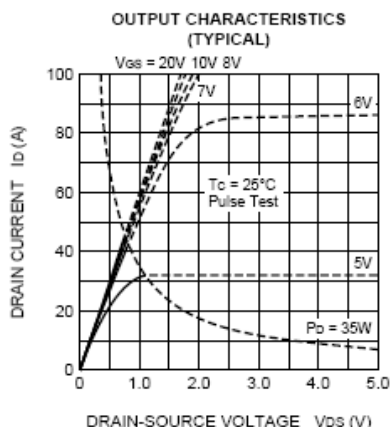
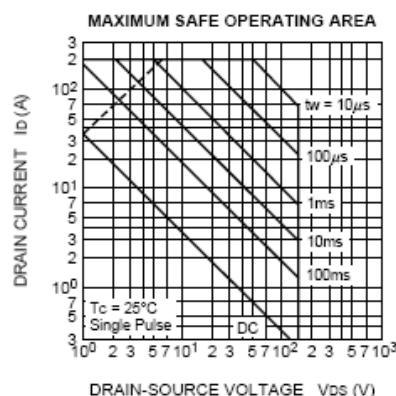
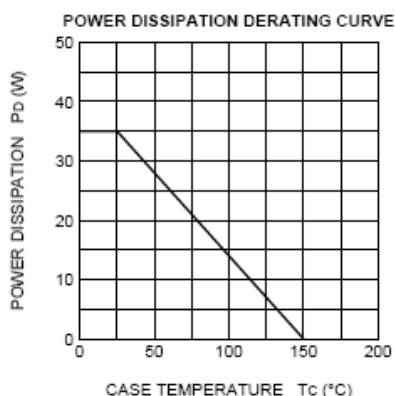
FS50KM-3

HIGH-SPEED SWITCHING USE

ELECTRICAL CHARACTERISTICS (Tch = 25°C)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V(BR)DSS	Drain-source breakdown voltage	ID = 1mA, VGS = 0V	150	—	—	V
IGSS	Gate-source leakage current	VGS = ±20V, VDS = 0V	—	—	±0.1	µA
IDSS	Drain-source leakage current	VDS = 150V, VGS = 0V	—	—	0.1	mA
VGS(th)	Gate-source threshold voltage	ID = 1mA, VDS = 10V	2.0	3.0	4.0	V
rDS(ON)	Drain-source on-state resistance	ID = 25A, VGS = 10V	—	24	31	mΩ
VDS(ON)	Drain-source on-state voltage	ID = 25A, VGS = 10V	—	0.600	0.775	V
yfs	Forward transfer admittance	ID = 25A, VDS = 10V	—	55	—	S
Ciss	Input capacitance	VDS = 10V, VGS = 0V, f = 1MHz	—	6540	—	pF
Coss	Output capacitance		—	880	—	pF
Crss	Reverse transfer capacitance		—	380	—	pF
td(on)	Turn-on delay time	VDD = 80V, ID = 25A, VGS = 10V, RGEN = RGS = 50Ω	—	95	—	ns
tr	Rise time		—	155	—	ns
td(off)	Turn-off delay time		—	380	—	ns
tf	Fall time		—	180	—	ns
VSD	Source-drain voltage	IS = 25A, VGS = 0V	—	1.0	1.5	V
Rth(ch-c)	Thermal resistance	Channel to case	—	—	3.57	°C/W
trr	Reverse recovery time	IS = 50A, dis/dt = -100A/µs	—	130	—	ns

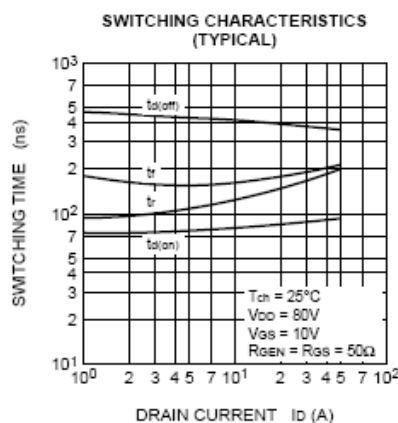
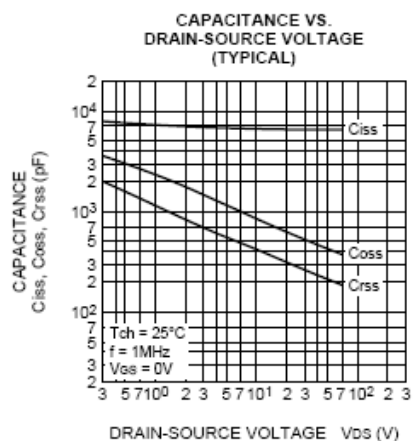
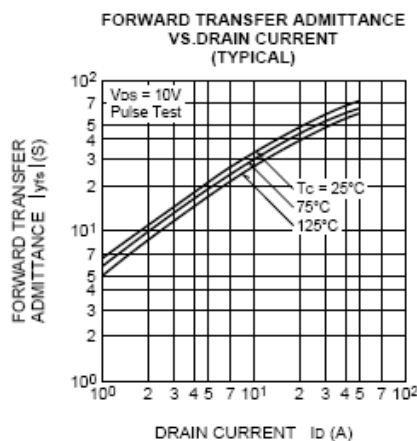
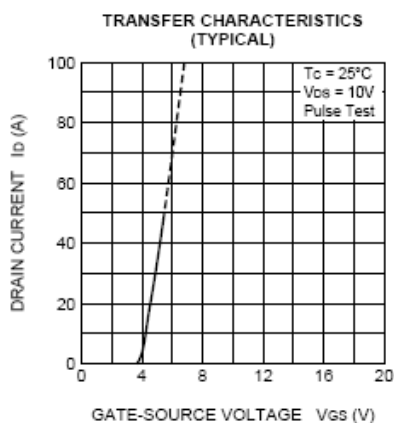
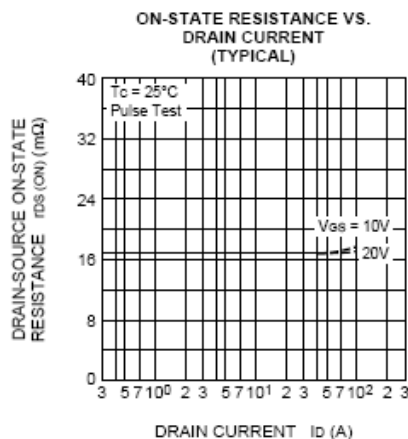
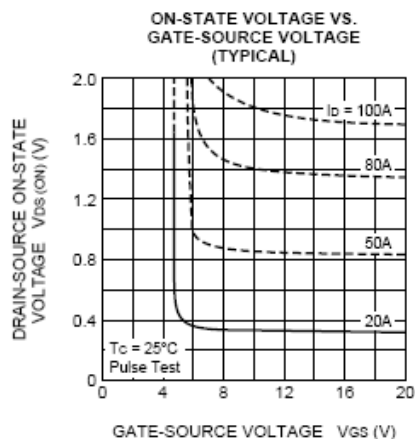
PERFORMANCE CURVES



MITSUBISHI Nch POWER MOSFET

FS50KM-3

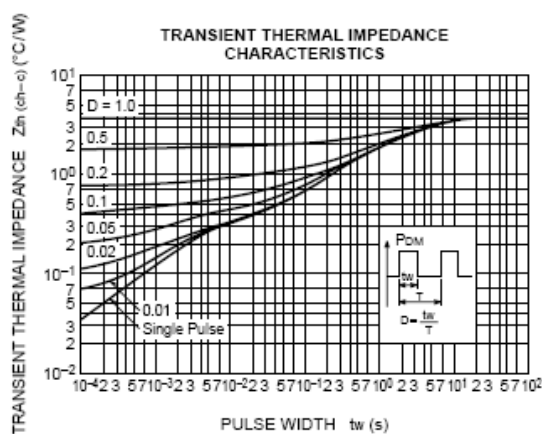
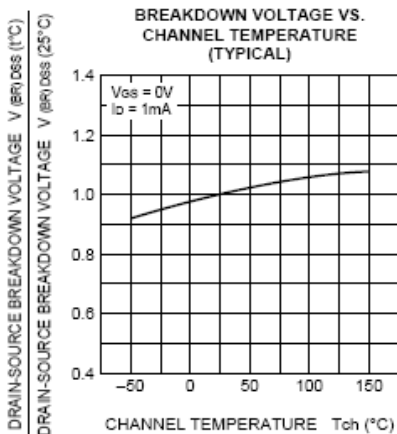
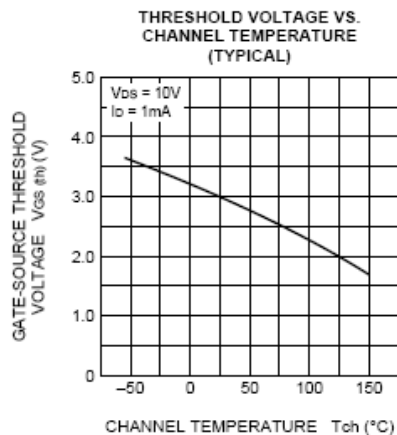
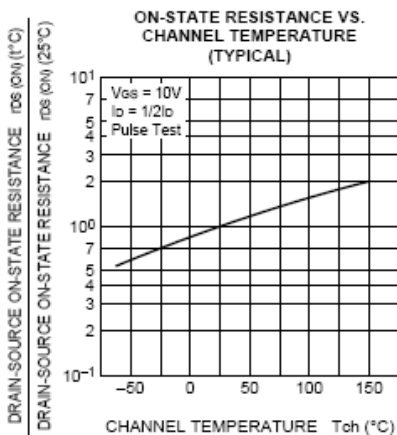
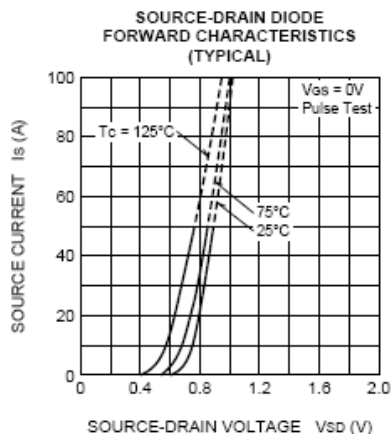
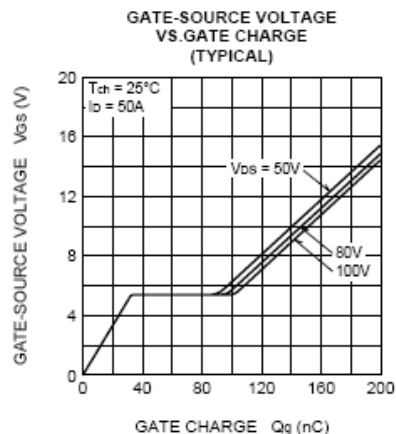
HIGH-SPEED SWITCHING USE



MITSUBISHI Nch POWER MOSFET

FS50KM-3

HIGH-SPEED SWITCHING USE





October 2005

H11AA1, H11AA3, H11AA2, H11AA4 AC Input/Phototransistor Optocouplers

Features

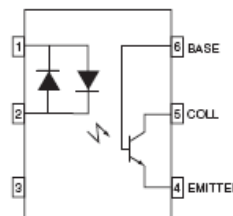
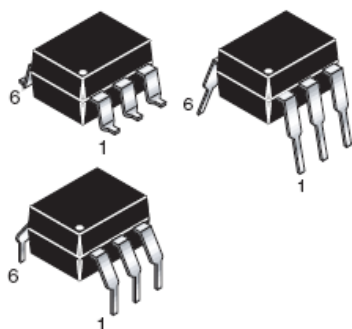
- Bi-polar emitter input
- Built-in reverse polarity input protection
- Underwriters Laboratory (UL) recognized – File #E90700
- VDE approved – File #E94766 (ordering option '300')

Description

The H11AAX series consists of two gallium-arsenide infrared emitting diodes connected in inverse parallel driving a single silicon phototransistor output.

Applications

- AC line monitor
- Unknown polarity DC sensor
- Telephone line interface



Parameter	Symbol	Device	Value	Units
TOTAL DEVICE				
Storage Temperature	T _{STG}	All	-55 to +150	°C
Operating Temperature	T _{OPR}	All	-55 to +100	°C
Lead Solder Temperature	T _{SOL}	All	260 for 10 sec	°C
Total Device Power Dissipation	P _D	All	350	mW
Derate Linearly From 25°C			4.6	mW/°C
EMITTER				
Continuous Forward Current	I _F	All	100	mA
Forward Current - Peak (1 μs pulse, 300 pps)	I _{F(pk)}	All	±1.0	A
LED Power Dissipation	P _D	All	200	mW
Derate Linearly From 25°C			2.6	mW/°C
DETECTOR				
Detector Power Dissipation	P _D	All	300	mW
Derate above 25°C			4.0	mW/°C

Electrical Characteristics ($T_A = 25^\circ\text{C}$ Unless otherwise specified.)**Individual Component Characteristics**

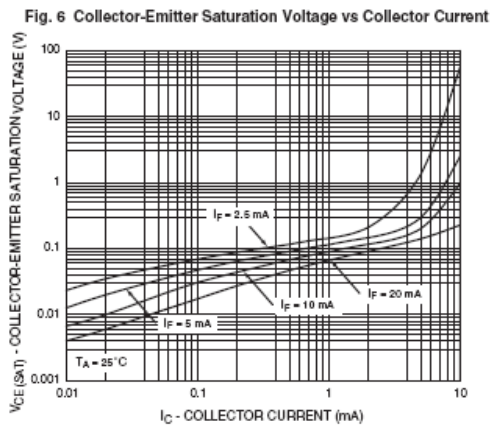
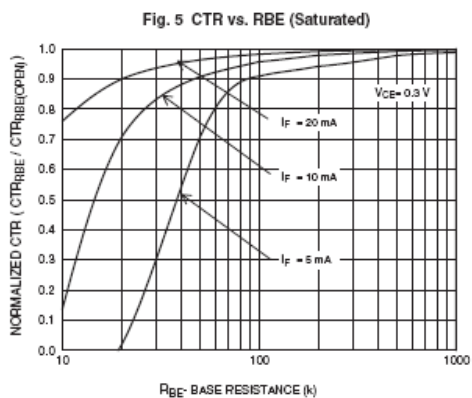
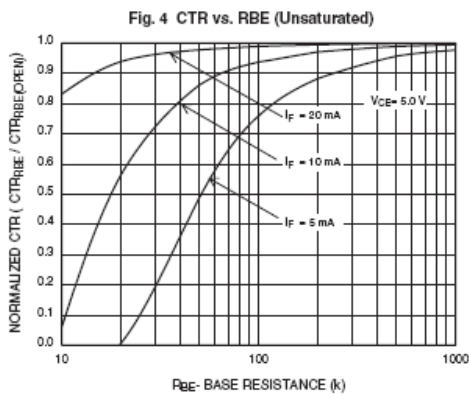
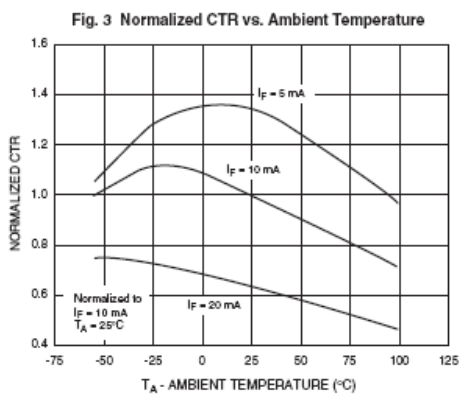
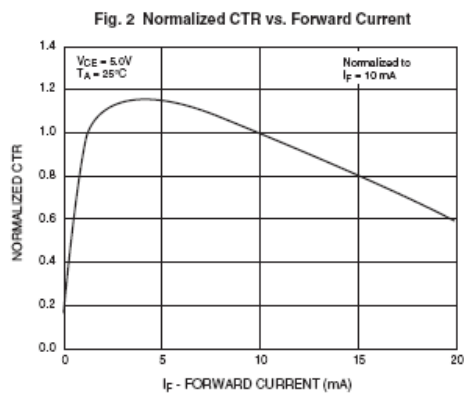
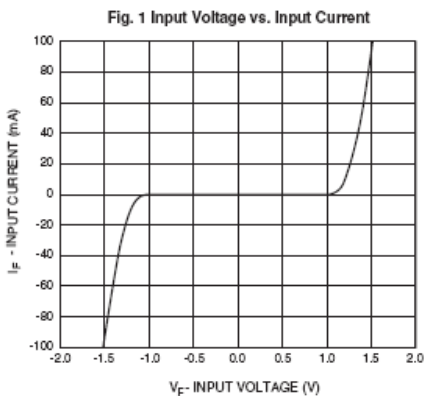
Parameter	Test Conditions	Symbol	Device	Min	Typ	Max	Unit
EMITTER							
Input Forward Voltage	$I_F = \pm 10 \text{ mA}$	V_F	All		1.2	1.5	V
Capacitance	$V_F = 0 \text{ V}, f = 1.0 \text{ MHz}$	C_J	All		80		pF
DETECTOR							
Breakdown Voltage Collector to Emitter	$I_C = 1.0 \text{ mA}, I_F = 0$	BV_{CEO}	All	30			V
Collector to Base	$I_C = 100 \mu\text{A}, I_F = 0$	BV_{CBO}	All	70			V
Emitter to Base	$I_E = 100 \mu\text{A}, I_F = 0$	BV_{EBO}	All	5			V
Emitter to Collector	$I_E = 100 \mu\text{A}, I_F = 0$	BV_{ECO}	All	7			V
Leakage Current Collector to Emitter	$V_{CE} = 10 \text{ V}, I_F = 0$	I_{CEO}	H11AA1,3,4 H11AA2			50 200	nA
Capacitance Collector to Emitter	$V_{CE} = 0, f = 1 \text{ MHz}$	C_{CE}	All		10		pF
Collector to Base	$V_{CE} = 0, f = 1 \text{ MHz}$	C_{CB}	All		80		pF
Emitter to Base	$V_{CE} = 0, f = 1 \text{ MHz}$	C_{EB}	All		15		pF

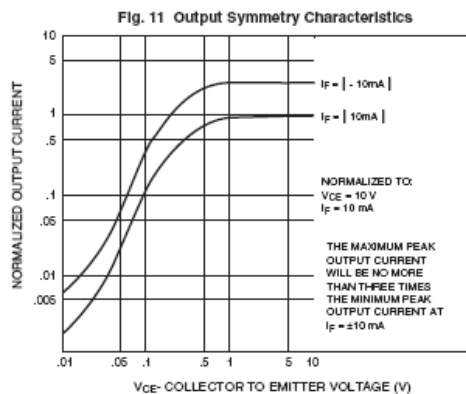
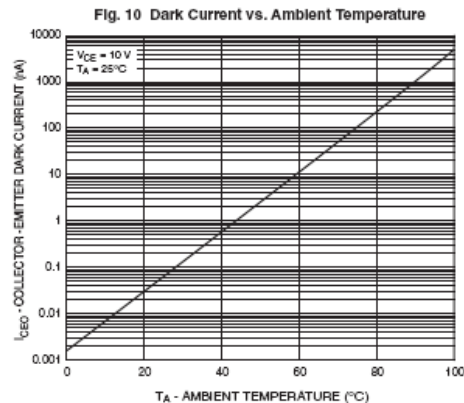
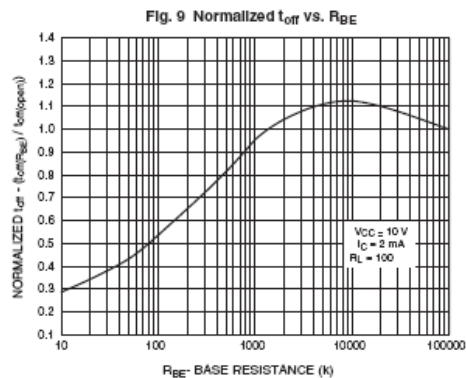
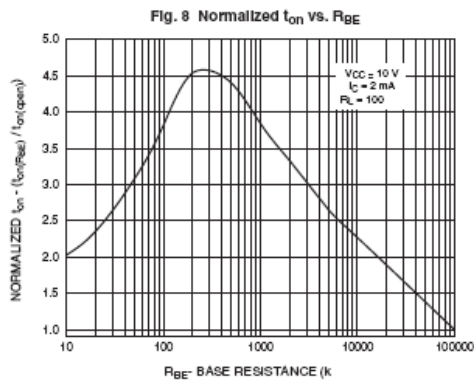
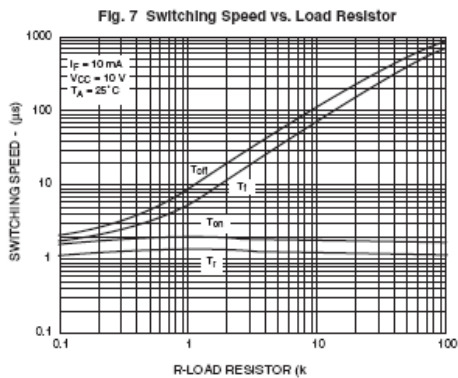
Transfer Characteristics ($T_A = 25^\circ\text{C}$ Unless otherwise specified.)

Characteristics	Test Conditions	Symbol	Device	Min	Typ	Max	Units
Current Transfer Ratio, Collector to Emitter	$I_F = \pm 10 \text{ mA}, V_{CE} = 10 \text{ V}$	CTR_{CE}	H11AA4	100			%
			H11AA3	50			
			H11AA1	20			
			H11AA2	10			
Current Transfer Ratio, Symmetry	$I_F = \pm 10 \text{ mA}, V_{CE} = 10 \text{ V}$ (Figure.8)		All	.33		3.0	
Saturation Voltage Collector to Emitter	$I_F = \pm 10 \text{ mA}, I_{CE} = 0.5 \text{ mA}$	$V_{CE(SAT)}$	All			.40	V

Isolation Characteristics

Characteristic	Test Conditions	Symbol	Min	Typ	Max	Units
Package Capacitance input/output	$V_{I-O} = 0, f = 1 \text{ MHz}$	C_{I-O}		0.7		pF
Isolation Voltage	$f = 60 \text{ Hz}, t = 1 \text{ min.}$	V_{ISO}	5300			VAC(RMS)
Isolation Resistance	$V_{I-O} = 500 \text{ VDC}$	R_{ISO}	1011			Ω





TOSHIBA**2SC1815**

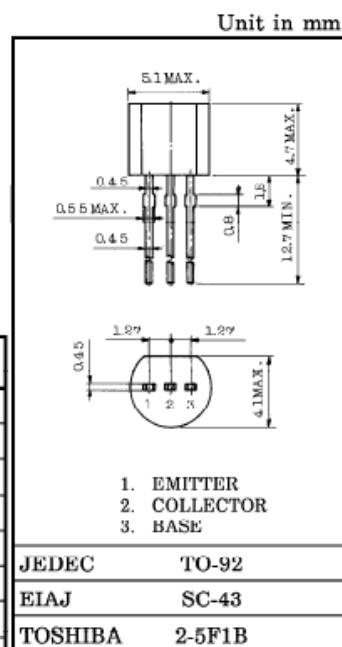
TOSHIBA TRANSISTOR SILICON NPN EPITAXIAL TYPE (PCT PROCESS)

2SC1815AUDIO FREQUENCY GENERAL PURPOSE AMPLIFIER APPLICATIONS.
DRIVER STAGE AMPLIFIER APPLICATIONS.

- High Voltage and High Current
: $V_{CE0} = 50V$ (Min.), $I_C = 150mA$ (Max.)
- Excellent h_{FE} Linearity
: $h_{FE(2)} = 100$ (Typ.) at $V_{CE} = 6V$, $I_C = 150mA$
: $h_{FE}(I_C = 0.1mA) / h_{FE}(I_C = 2mA) = 0.95$ (Typ.)
- Low Noise : $NF = 1dB$ (Typ.) at $f = 1kHz$
- Complementary to 2SA1015 (O, Y, GR class)

MAXIMUM RATINGS ($T_a = 25^\circ C$)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Collector-Base Voltage	V_{CBO}	60	V
Collector-Emitter Voltage	V_{CEO}	50	V
Emitter-Base Voltage	V_{EBO}	5	V
Collector Current	I_C	150	mA
Base Current	I_B	50	mA
Collector Power Dissipation	P_C	400	mW
Junction Temperature	T_j	125	$^\circ C$
Storage Temperature Range	T_{stg}	-55~125	$^\circ C$



Weight : 0.21g

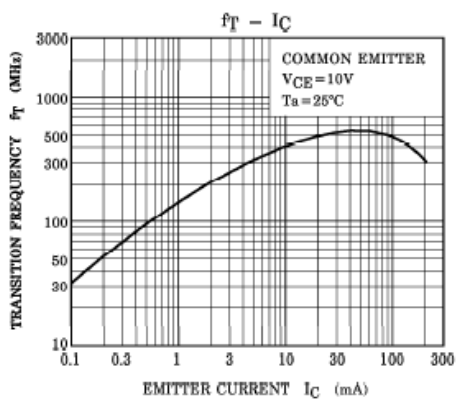
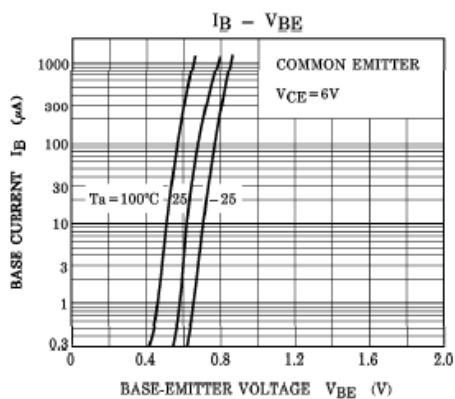
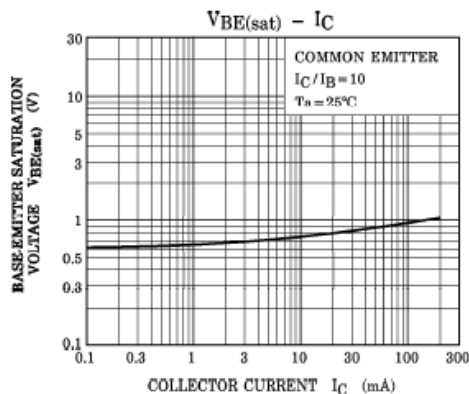
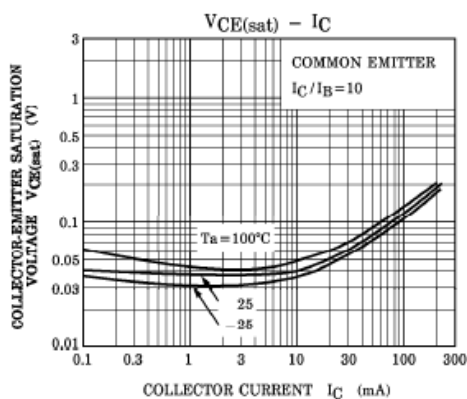
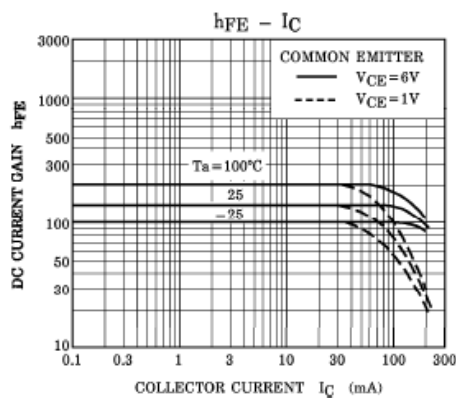
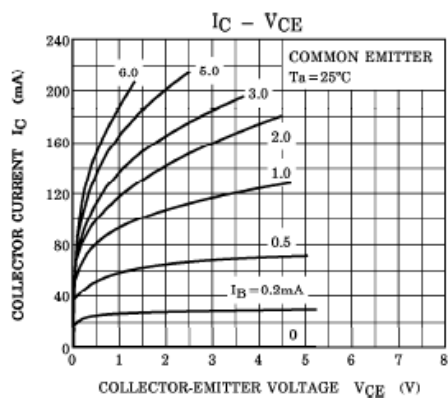
ELECTRICAL CHARACTERISTICS ($T_a = 25^\circ C$)

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Collector Cut-off Current	I_{CBO}	$V_{CB} = 60V$, $I_E = 0$	—	—	0.1	μA
Emitter Cut-off Current	I_{EBO}	$V_{EB} = 5V$, $I_C = 0$	—	—	0.1	μA
DC Current Gain	$h_{FE(1)}$ (Note)	$V_{CE} = 6V$, $I_C = 2mA$	70	—	700	
	$h_{FE(2)}$	$V_{CE} = 6V$, $I_C = 150mA$	25	100	—	
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	$I_C = 100mA$, $I_B = 10mA$	—	0.1	0.25	V
Base-Emitter Saturation Voltage	$V_{BE(sat)}$	$I_C = 100mA$, $I_B = 10mA$	—	—	1.0	V
Transition Frequency	f_T	$V_{CE} = 10V$, $I_C = 1mA$	80	—	—	MHz
Collector Output Capacitance	C_{ob}	$V_{CB} = 10V$, $I_E = 0$, $f = 1MHz$	—	2.0	3.5	pF
Base Intrinsic Resistance	$r_{bb'}$	$V_{CE} = 10V$, $I_E = -1mA$ $f = 30MHz$	—	50	—	Ω
Noise Figure	NF	$V_{CE} = 6V$, $I_C = 0.1mA$ $f = 1kHz$, $R_G = 10k\Omega$	—	1.0	10	dB

Note : h_{FE} Classification 0 : 70~140 Y : 120~240 GR : 200~400 BL : 350~700

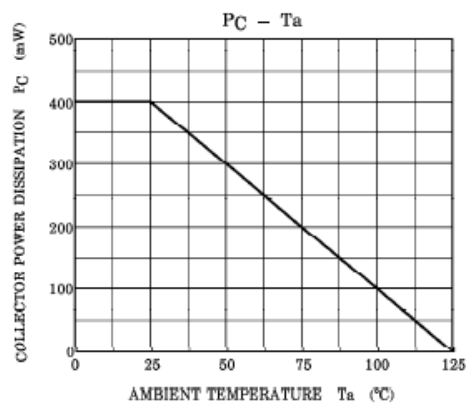
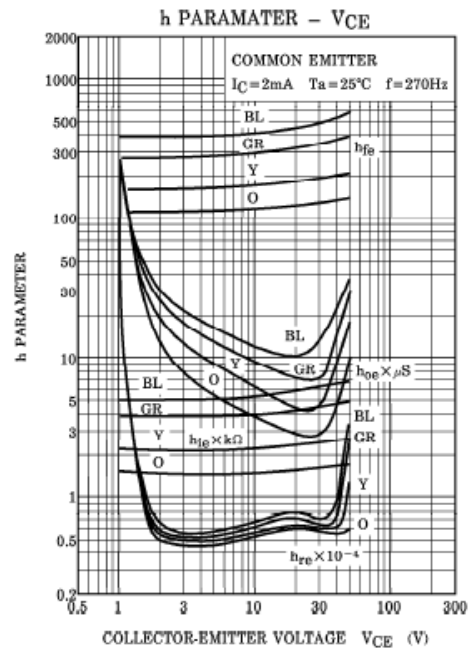
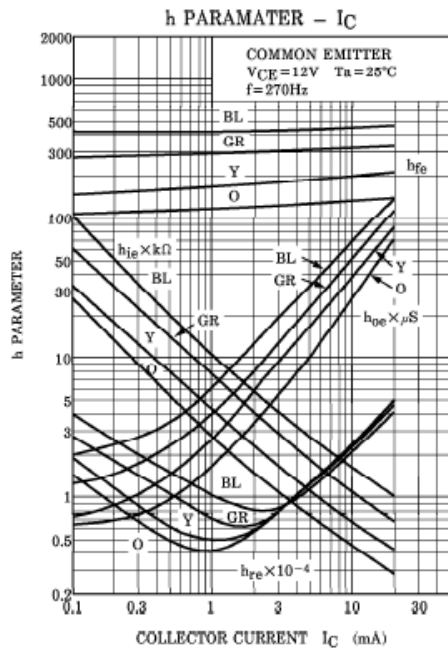
961001FAA2

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.



961001EAA2

● The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
● The information contained herein is subject to change without notice.



TOSHIBA**2SA1015(L)**

TOSHIBA Transistor Silicon PNP Epitaxial Type (PCT process)

2SA1015(L)

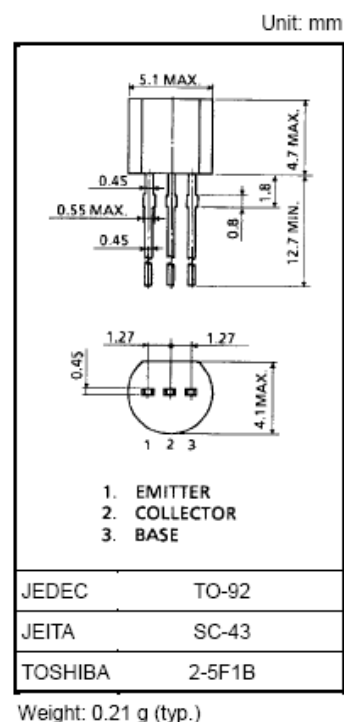
Audio Frequency Amplifier Applications

Low Noise Amplifier Applications

- High voltage and high current: $V_{CE0} = -50$ V (min),
 $I_C = -150$ mA (max)
- Excellent h_{FE} linearity: $h_{FE} (2) = 80$ (typ.) at $V_{CE} = -6$ V, $I_C = -150$ mA
: $h_{FE} (I_C = -0.1$ mA)/ $h_{FE} (I_C = -2$ mA) = 0.95 (typ.)
- Low noise: $NF = 0.2$ dB (typ.) ($f = 1$ kHz)
- Complementary to 2SC1815 (L)

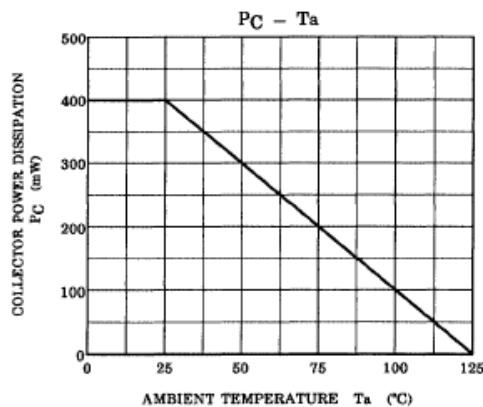
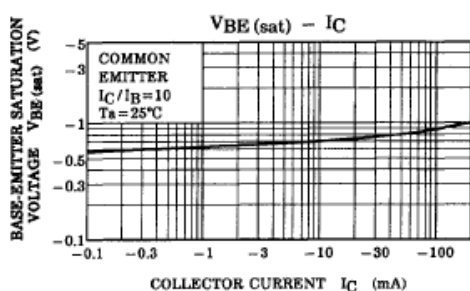
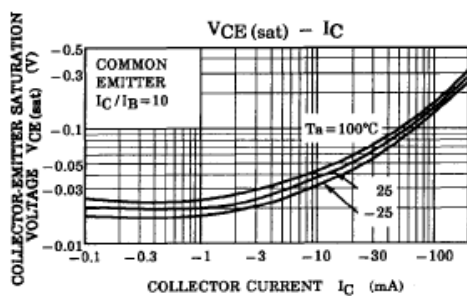
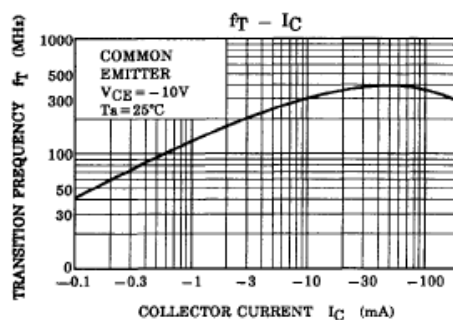
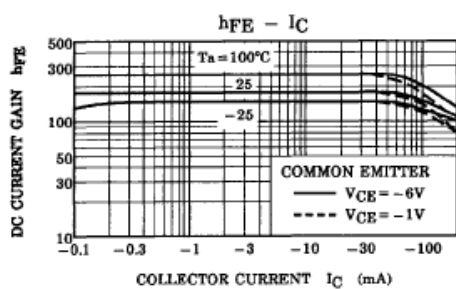
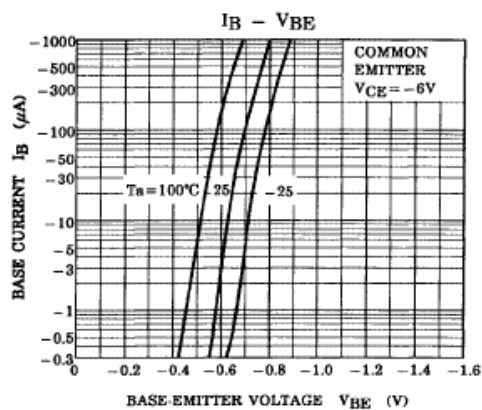
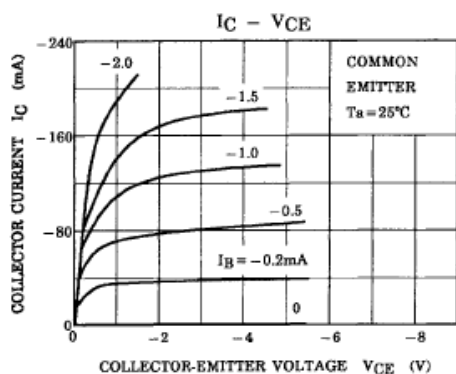
Maximum Ratings ($T_a = 25^\circ\text{C}$)

Characteristics	Symbol	Rating	Unit
Collector-base voltage	V_{CB0}	-50	V
Collector-emitter voltage	V_{CE0}	-50	V
Emitter-base voltage	V_{EB0}	-5	V
Collector current	I_C	-150	mA
Base current	I_B	-50	mA
Collector power dissipation	P_C	400	mW
Junction temperature	T_j	125	$^\circ\text{C}$
Storage temperature range	T_{stg}	-55~125	$^\circ\text{C}$

**Electrical Characteristics ($T_a = 25^\circ\text{C}$)**

Characteristics	Symbol	Test Condition	Min	Typ.	Max	Unit
Collector cut-off current	I_{CBO}	$V_{CB} = -50$ V, $I_E = 0$	—	—	-0.1	μA
Emitter cut-off current	I_{EBO}	$V_{EB} = -5$ V, $I_C = 0$	—	—	-0.1	μA
DC current gain	$h_{FE} (1)$ (Note)	$V_{CE} = -6$ V, $I_C = -2$ mA	70	—	400	
	$h_{FE} (2)$	$V_{CE} = -6$ V, $I_C = -150$ mA	25	80	—	
Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_C = -100$ mA, $I_B = -10$ mA	—	-0.1	-0.3	V
Base-emitter saturation voltage	$V_{BE(sat)}$	$I_C = -100$ mA, $I_B = -10$ mA	—	—	-1.1	V
Transition frequency	f_T	$V_{CE} = -10$ V, $I_C = -1$ mA	80	—	—	MHz
Collector output capacitance	C_{ob}	$V_{CB} = -10$ V, $I_E = 0$ $f = 1$ MHz	—	4	7	pF
Base intrinsic resistance	$r_{bb'}$	$V_{CB} = -10$ V, $I_E = 1$ mA $f = 30$ MHz	—	30	—	Ω
Noise figure	NF (1)	$V_{CE} = -6$ V, $I_C = -0.1$ mA $f = 100$ Hz, $R_G = 10$ k Ω	—	0.5	6	dB
	NF (2)	$V_{CE} = -6$ V, $I_C = -0.1$ mA $f = 1$ kHz, $R_G = 10$ k Ω	—	0.2	3	

Note: $h_{FE} (1)$ classification O: 70~140, Y: 120~240, GR: 200~400



การเขียน Program

```

/*Chip type      : ATmega8
Program type     : Application
Clock frequency  : 1.000000 MHz
Memory model    : Small
External SRAM size : 0
Data Stack size  : 256
*****/

#include <mega8.h>
void Delay(int time);
void Delay2(int time2);
void scan_1(void);
void scan_2(void);
void num_to_disp_1(int num_1);
void num_to_disp_2(int num_2);
unsigned char disp_1[4];
unsigned char disp_2[4];
unsigned char pat_data[10] = {0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09};
unsigned char pat_scan [] = {0xC2,0xC1,0x83,0x43};
static unsigned int counter_1=0;
static unsigned int counter_2=0;
void main(void)
{
num_to_disp_1 (counter_1);
num_to_disp_2 (counter_2);
PORTB=0x00;
DDRB=0b00000010;
PORTC=0x00;
DDRC=0x0F;
PORTD=0x00;
DDRD=0xC3;

```

```
// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x06;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: ..... kHz
// Mode: Fast PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0x81;
TCCR1B=0b00001001;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;
```

```

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
MCUCR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

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

        while(PINB.0==0)                                //Prog
        {
            if (PIND.5==1)                                //Speed_Up
            {
                counter_1++;
                num_to_disp_1 (counter_1);
                scan_1 ();
            }
            if (PINC.5==1)                                //Step_Down
            {
                counter_1--;
                num_to_disp_1 (counter_1);
                while (PINC.5==1)
                {
                    scan_1 ();
                }
            }
        }

```

```

        scan_1 ();
    }
    while(PINB.5==0)
    {
        num_to_disp_1 (0);
        scan_1 ();
    }
    while (1)
    {
        if (PIND.4==1)
        {
            ++counter_2;                //Count_In

            num_to_disp_2 (counter_2/4);
            while (PIND.4==1)
            {
                scan_2 ();
            }
        }
        else if (PINB.5==1)            //Run
        {
            static int Duty=3;        //Start PWM

            while(PIND.2==0&&counter_1!=(counter_2/4))
            {
                if (PIND.3==1)        //Duty_Up
                {
                    Duty++;
                    if (PIND.4==1)
                    {
                        ++counter_2;    //Count_In
                        num_to_disp_2 (counter_2/4);
                        while (PIND.4==1)
                        {
                            scan_2 ();

```

```

    }
}
}
if (PINC.4==1) //Duty_Down
{
    Duty--;
    if (PIND.4==1)
    {
        ++counter_2; //Count_In
        num_to_disp_2 (counter_2/4);
        while (PIND.4==1)
        {
            scan_2 ();
        }
    }
}
}
while(PIND.3==0&&PINC.4==0&&PIND.2==0&&counter_1!=(counter_2/4))
{
    OCR1A=Duty; //PWM_Out, Count_In
    scan_2 ();
    if (PIND.4==1)
    {
        ++counter_2; //Count_In
        num_to_disp_2 (counter_2/4);
        while (PIND.4==1)
        {
            scan_2 ();
        }
    }
}
scan_2();
}
}

```

```
                else
                    {
                        OCR1A=0x00;
                        scan_2 ();
                    }
            }
        scan_2 ();
    }
void scan_1 (void)
{
    char i;
    for (i=0; i<4; i++)
    {
        PORTC = disp_1 [i];
        PORTD = pat_scan [i];
        Delay (5);
    }
}
void scan_2 (void)
{
    char j;
    for (j=0; j<4; j++)
    {
        PORTC = disp_2 [j];
        PORTD = pat_scan [j];
        Delay2 (2);
    }
}
void num_to_disp_1 (int num_1)
{
    int to_disp_1;
    to_disp_1 = num_1/1000;
```



```

    disp_1 [3] = pat_data [to_disp_1];
    to_disp_1 = ((num_1%1000)/100);
    disp_1 [2] = pat_data [to_disp_1];
    to_disp_1 = ((num_1%100)/10);
    disp_1 [1] = pat_data [to_disp_1];
    to_disp_1 = num_1%10;
    disp_1 [0] = pat_data [to_disp_1];
}
void num_to_disp_2 (int num_2)
{
    int to_disp_2;
    to_disp_2 = num_2/1000;
    disp_2 [3] = pat_data [to_disp_2];
    to_disp_2 = ((num_2%1000)/100);
    disp_2 [2] = pat_data [to_disp_2];
    to_disp_2 = ((num_2%100)/10);
    disp_2 [1] = pat_data [to_disp_2];
    to_disp_2 = num_2%10;
    disp_2 [0] = pat_data [to_disp_2];
}
void Delay (int time)
{
    unsigned int loops,inloops;
    for (loops = 0; loops<time; loops++)
    {
        for (inloops = 0; inloops <100; inloops++);
    }
}
void Delay2 (int time2)
{
    unsigned int loops2,inloops2;
    for (loops2 = 0; loops2<time2; loops2++)

```

```
    {  
        for (inloops2 = 0; inloops2 <50; inloops2++);  
    }  
}
```