

**PENGENDALIAN KECEPATAN MOTOR *SERVO* UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *RADIAL* DENGAN  
*HOTWIRE* BERBASIS MIKROKONTROLER ATMEGA 328**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



**Disusun Oleh :**

**DONY ADITYA ALBAR**

**NIM. 0610630030 - 63**

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN**

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS TEKNIK**

**JURUSAN TEKNIK ELEKTRO**

**MALANG**

**2012**

**PENGENDALIAN KECEPATAN MOTOR *SERVO* UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *RADIAL* DENGAN  
*HOTWIRE* BERBASIS MIKROKONTROLER ATMEGA 328**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



**Disusun Oleh :**

**DONY ADITYA ALBAR**

**NIM. 0610630030 - 63**

Telah diperiksa dan disetujui,

Dosen Pembimbing :

**Dosen Pembimbing I**

**Ir. Purwanto, MT.**  
**NIP. 19540424 19860 1 001**

**Dosen Pembimbing II**

**Ir. Bambang Siswojo, MT.**  
**NIP. 19621211 198802 1 001**



**PENGENDALIAN KECEPATAN MOTOR *SERVO* UNTUK  
PEMOTONGAN *STYROFOAM* SECARA *RADIAL* DENGAN  
*HOTWIRE* BERBASIS MIKROKONTROLER ATMEGA 328**

**SKRIPSI**

**Disusun Oleh :**

**DONY ADITYA ALBAR**

**NIM. 0610630030 - 63**

Skripsi ini telah diuji dan dinyatakan lulus  
pada tanggal 10 Februari 2012

**Majelis Penguji**

**Goegoes Dwi Nusantoro, ST., MT.**  
**NIP. 19711013 200604 1 001**

**Tri Nurwati, ST., MT.**  
**NIP. 19790615 200812 2 003**

**Muhammad Aziz Muslim, ST., MT.,PhD.**  
**NIP. 19741203 200012 1 001**

**Mengetahui,**  
**Ketua Jurusan Teknik Elektro**

**Dr. Ir. Sholeh Hadi Pramono, MS**  
**NIP. 19580728 198701 1 001**





## KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT atas rahmat, hidayah, dan karunia-Nya sehingga penyusunan skripsi ini dapat terselesaikan dengan lancar dan baik.

Skripsi berjudul “pengendalian kecepatan motor *servo* untuk pemotongan *styrofoam* secara *radial* dengan *hotwire* berbasis mikrokontroler atmega 328” ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Teknik di Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya Malang. Terelesaiakannya skripsi ini tentunya tidak lepas juga dari bantuan berbagai pihak. Oleh karena itu dengan rasa hormat dan segala kerendahan hati, penyusun menyampaikan rasa terima kasih kepada :

1. Ayahku yang terhormat Imam Syai Effendi, Ibuku tercinta Rusmaria Ekowati, kakakku tersayang Arie Ferdiansyah dan Lely dian Amelia beserta segenap keluarga besar yang tiada henti-hentinya selalu memberikan semangat, dukungan serta doa-doa
2. Fenty Catur Herawati yang setia selalu mendampingi aku dalam suka dan duka selama proses penyusunan skripsi ini
3. Bapak Dr. Ir. Sholeh Hadi Pramono, MS selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya dan Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
4. Bapak Ir. Purwanto, MT. selaku Ketua Kelompok Dosen Keahlian Teknik Kontrol Jurusan Teknik Elektro Universitas Brawijaya,
5. Bapak Mochammad Rif'an, ST., MT. selaku Ketua Program Studi Jurusan Teknik Elektro Universitas Brawijaya,
6. Bapak Ir. Purwanto, MT., dan Bapak Ir. Bambang Siswoyo MT., selaku Dosen Pembimbing 1 dan Dosen Pembimbing 2 atas segala bimbingan, nasehat, pengarahan, motivasi, saran dan masukan yang telah diberikan,
7. Bapak Ir. M.Aswin MT., dan Ibu Rusmi Ambarwati ST MT, selaku Dosen Pembimbing Akademik atas nasehat, saran dalam membimbing penulis dalam perkuliahan,

8. 4 teman seperjuangan senasib sepenanggungan I gede D, Rahman Arief, dan Tony Adi Saputro serta saya sendiri. Perjuangan kita tidak sia-sia kawan.tetap semangat pantang menyerah menggapai cita-cita.
9. Teman-teman Teknik Elektro angkatan 2006 dan rekan rekan yang telah banyak membantu, serta semua pihak yang tidak dapat disebutkan satu per satu.

Penyusun menyadari bahwa skripsi ini masih banyak kekurangan, oleh karena itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penyusun mengharapkan semoga skripsi ini dapat bermanfaat bagi rekan-rekan mahasiswa khususnya dan seluruh pembaca pada umumnya.

Malang, januari 2012

Penyusun





## DAFTAR ISI

<b>KATA PENGANTAR</b> .....	i
<b>DAFTAR ISI</b> .....	ii
<b>DAFTAR GAMBAR</b> .....	v
<b>DAFTAR TABEL</b> .....	vii
<b>ABSTRAK</b> .....	viii

## BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Sistematika Penulisan.....	4

## BAB II TINJAUAN PUSTAKA

2.1 Styrofoam .....	6
2.1.1 Pengertian Styrofoam.....	6
2.1.2 Bahan Dasar Styrofoam .....	7
2.1.3 Titik Leleh.....	7
2.1.4 Kelenturan dan Kekuatan .....	8
2.1.5 Jenis-jenis Styrofoam.....	8
2.2 Perkembangan alat potong Styrofoam.....	10
2.3 Mikrokontroler AVR Atmega 328.....	11
2.3.1 Konfigurasi pin.....	13
2.4 Sensor kecepatan radial ( <i>incremental encoder</i> ).....	16
2.5 LCD .....	17
2.6 PWM ( <i>Pulse Width Modulation</i> ).....	20
2.7 Motor DC .....	22
2.8 Kontroler PID ( <i>Proportional Integrator Derivative</i> ).....	24
2.8.1 Kontroler Proporsional.....	25
2.8.2 Kontroler Integral.....	26
2.8.3 Kontroler Differensial .....	27

**BAB III METODOLOGI**

3.1 Studi Literatur ..... 33

3.2 Perancangan Alat ..... 34

3.3 Realisasi Pembuatan Alat ..... 34

3.4 Pengujian Alat..... 35

3.4 Pengambilan Kesimpulan. .... 35

**BAB IV PERANCANGAN DAN PEMBUATAN ALAT**

4.1 Perancangan Sistem..... 34

    4.1.1 Spesifikasi alat ..... 35

    4.1.2 Prinsip kerja alat..... 36

4.2 Perancangan Perangkat Keras ..... 36

    4.2.1 Rangkaian Mikrokontroler AVR ATMEGA 328 ..... 37

    4.2.2 Sensor Incremental Encoder ..... 40

    4.2.3 Driver Motor DC..... 42

    4.2.4 Motor DC ..... 43

4.3 Perancangan Perangkat Lunak ..... 44

    4.3.1 Program Utama ..... 46

    4.3.2 Program Subrutin Kontroler..... 47

**BAB V PENGUJIAN DAN ANALISIS SISTEM**

5.1 Pengujian Rangkaian Sensor Suhu ..... 49

    5.1.1 Tujuan pengujian..... 49

    5.1.2 Peralatan Pengujian..... 49

    5.1.3 Prosedur pengujian ..... 49

    5.1.4 Hasil dan Analisis pengujian..... 50

5.2 Pengujian Mikrokontroler atmega 328..... 51

    5.2.1 Tujuan pengujian..... 51

    5.2.2 Peralatan pengujian ..... 51

    5.2.3 Prosedur pengujian..... 51

    5.2.4 Hasil dan Analisis Pengujian ..... 51

5.3 Pengujian Rangkaian Motor DC NA4056A..... 52

    5.3.1 Tujuan pengujian..... 52

    5.3.2 Peralatan Pengujian..... 52





5.3.3	Prosedur pengujian.....	52
5.3.4	Hasil pengujian .....	53
5.4	Pengujian LCD .....	55
5.4.1	Tujuan pengujian.....	55
5.4.2	Peralatan Pengujian.....	55
5.4.3	Prosedur pengujian.....	55
5.4.4	Hasil pengujian .....	56
5.5	Pengujian Sistem Keseluruhan.....	56
5.5.1	Tujuan pengujian.....	56
5.5.2	Peralatan Pengujian.....	57
5.5.3	Prosedur pengujian.....	57
5.5.4	Hasil pengujian .....	57
<b>BAB VI</b>	<b>PENUTUP.....</b>	<b>59</b>
6.1	Kesimpulan.....	59
6.2	Saran .....	59
<b>DAFTAR PUSTAKA</b>	<b>.....</b>	<b>60</b>
<b>LAMPIRAN 1</b>	<b>Gambar Rangkaian</b>	
<b>LAMPIRAN 2</b>	<b>Listing program mikrokontroler atmega328</b>	
<b>LAMPIRAN 3</b>	<b>Datasheet</b>	

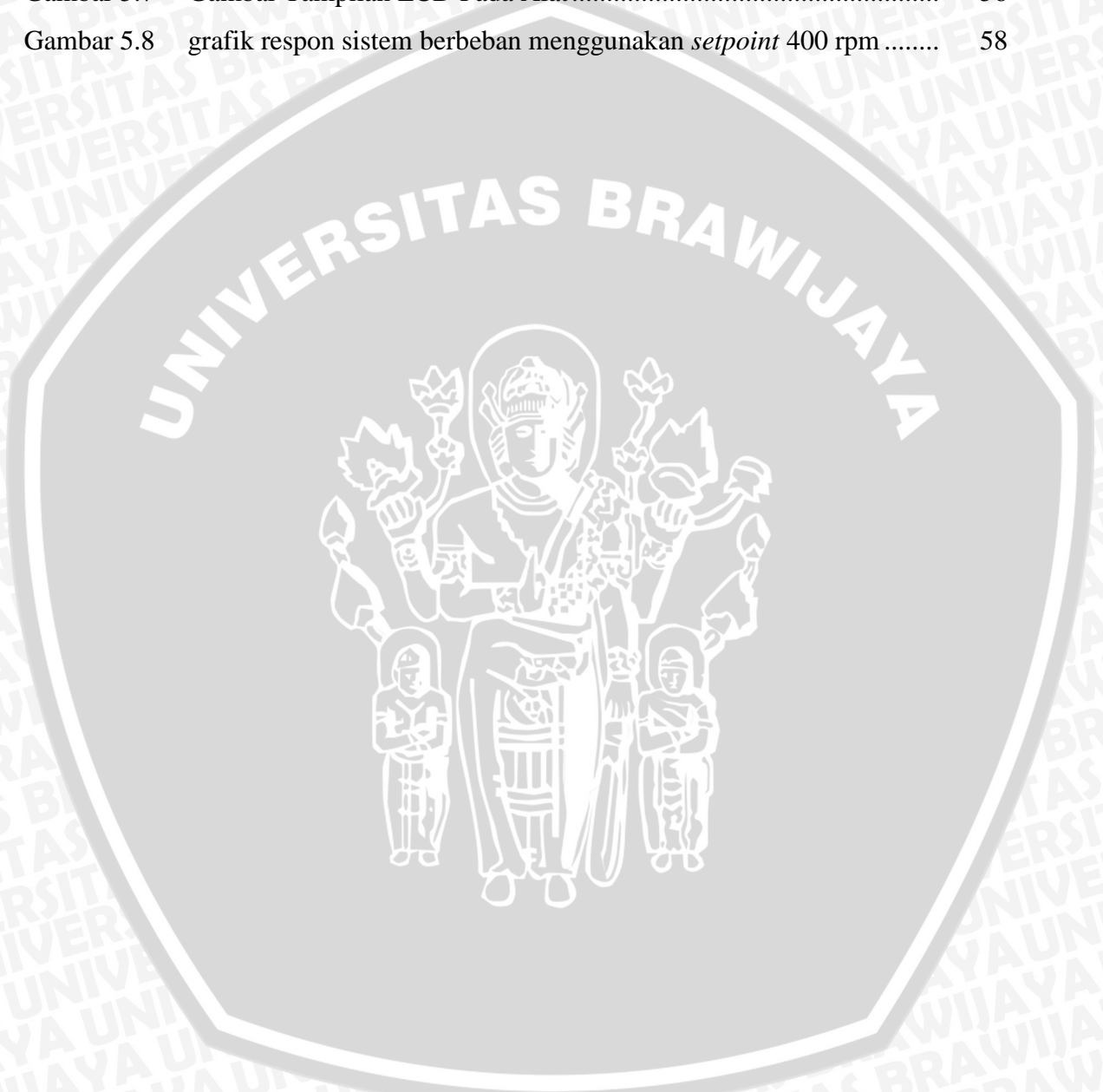


**DAFTAR GAMBAR**

Gambar 2.1	Cutter.....	10
Gambar 2.2	Hotwire tool .....	11
Gambar 2.3	Blok Diagram ATmega328 .....	13
Gambar 2.4	Konfigurasi Pin ATmega328 .....	14
Gambar 2.5	Desain Umum <i>incremental encoder</i> .....	16
Gambar 2.6	Rangkaian Sensor Kecepatan.....	17
Gambar 2.7	Diagram Blok LCD M1632 .....	19
Gambar 2.8	Sinyal Keluaran PWM .....	20
Gambar 2.9	Duty Cycle PWM.....	21
Gambar 2.10	Cara Mendapatkan Sinyal PWM.....	22
Gambar 2.11	Sinyal Sumber yang Dimodulasi dan Sinyal PWM .....	22
Gambar 2.12	Motor DC .....	23
Gambar 2.13	motor dc servo.....	24
Gambar 2.14	Diagram blok kontroler proporsional.....	25
Gambar 2.15	Blok diagram kontroler integral .....	26
Gambar 2.16	Blok Diagram kontroler diferensial .....	27
Gambar 2.17	Blok Diagram kontroler PID .....	29
Gambar 2.18	Hubungan dalam fungsi waktu antara sinyal keluaran dengan masukan untuk kontroler PID.....	29
Gambar 4.1	Diagram blok sistem .....	34
Gambar 4.2	Skema perancangan sistem secara keseluruhan .....	37
Gambar 4.3	Penyemat kaki Mikrokontroler atmega 328.....	38
Gambar 4.4	Blok diagram mikrokontroler atmega 328 .....	40
Gambar 4.5	<i>incremental encoder</i> .....	41
Gambar 4.6	rangkaian <i>incremental encoder</i> .....	41
Gambar 4.7	Rangkaian IC IRF640 .....	42
Gambar 4.8	rangkaian driver motor DC .....	42
Gambar 4.9	flowchart program keseluruhan.....	45
Gambar 4.10	flowchart subrutin kontroler PID .....	46
Gambar 5.1	blok diagram pengujian sensor kecepatan.....	50



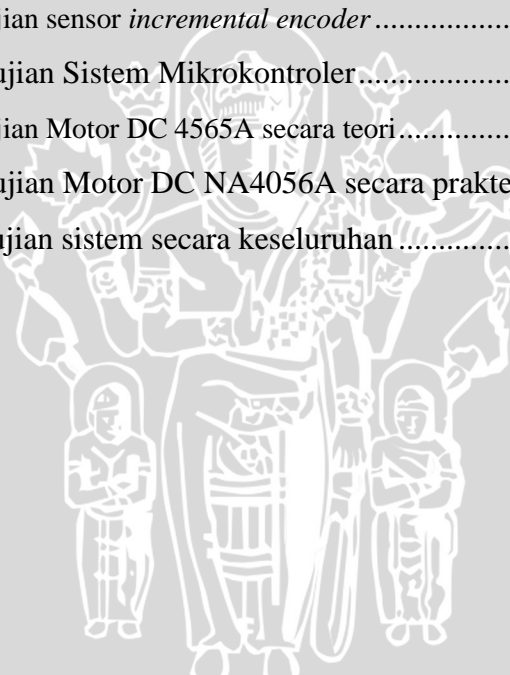
Gambar 5.2	Blok Diagram Pengujian Mikrokontroler .....	51
Gambar 5.3	Blok Diagram Pengujian Motor DC NA4056A.....	52
Gambar 5.4	Grafik respon sistem tanpa beban dengan <i>setpoint</i> 400 rpm.....	54
Gambar 5.5	Diagram alir program pada Mikrokontroler.....	55
Gambar 5.6	Blok diagram pengujian Mikrokontroler dan LCD.....	56
Gambar 5.7	Gambar Tampilan LCD Pada Alat.....	56
Gambar 5.8	grafik respon sistem berbeban menggunakan <i>setpoint</i> 400 rpm .....	58





**DAFTAR TABEL**

Tabel 2.1	Fungsi Pin-Pin pada LCD M1632.....	20
Tabel 4.1	Spesifikasi motor DC NA4056A .....	43
Tabel 5.1.	Hasil pengujian sensor <i>incremental encoder</i> secara teori.....	50
Tabel 5.2	Hasil Pengujian sensor <i>incremental encoder</i> .....	50
Tabel 5.3	Hasil Pengujian Sistem Mikrokontroler.....	51
Tabel 5.4	Hasil Pengujian Motor DC 4565A secara teori.....	53
Tabel 5.5	Hasil Pengujian Motor DC NA4056A secara praktek .....	53
Tabel 5.6	Hasil pengujian sistem secara keseluruhan .....	57



## ABTRAKSI

**Dony Aditya Albar**, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, 2012. “ *Pengendalian Kecepatan Motor Servo Untuk Pemotongan Styrofoam Secara Radial Dengan Hotwire Berbasis Mikrokontroler ATmega 328*”.

Dosen Pembimbing : **Ir. Purwanto, MT** dan **Ir. Bambang siswojo, MT**.

Pesawat aeromodeling saat ini cukup digemari sebagai hobby yang seru dan menantang. Salah satu unsur pembuat badan pesawat model adalah styrofoam. Styrofoam ringan namun kuat.. Untuk membentuk badan pesawat model (fuselage) perlu digunakan alat pemotong styrofoam yang mampu memotong styrofoam secara radial. Oleh karena itu digunakanlah metode pemotongan dengan kawat panas (hotwire) dengan memanfaatkan sifat styrofoam yang akan meleleh jika dipanaskan. Untuk itulah dirancang suatu alat pemotong styrofoam yang menggunakan metode pemotongan dengan hotwire dimana styrofoam diputar dengan motor DC yang telah dikontrol dengan mikrokontroler dengan feedback kecepatan yg berasal dari sensor incremental encoder sehingga diperoleh hasil kecepatan pemotongan styrofoam yang sesuai dengan yang dibutuhkan oleh pemotongan

Motor servo yang digunakan adalah motor DC 24V, menggunakan mikrokontroler atmega328 sebagai kontrolernya. Sistem bekerja sesuai spesifikasi yang diinginkan.. Nilai error pengujian motor DC dengan feedback incremental encoder dan tanpa beban styrofoam yang terkecil adalah 0,54% dan nilai error yang terbesar adalah 2,2 % sehingga diperoleh kesalahan rata-rata sebesar 1,3%. Nilai error pengujian sensor incremental encoder pada kondisi gelap adalah 0,1 % dan pada kondisi terang adalah 0 %. Nilai error pengujian sistem kecepatan motor DC yang telah diberi feedback sensor incremental encoder yang terkecil adalah 1,65 % dan nilai error yang terbesar adalah 2,54% Sehingga diperoleh error rata-rata sebesar 2,1%. Error yang terjadi masih mampu membuat alat menghasilkan pemotongan yang baik. Kontroler sudah dapat bekerja pada sistem dengan baik

Kata kunci: *pengendalian kecepatan, hotwire, mikrokontroler ATmega 328, sensor incremental encoder.*

## BAB 1

### PENDAHULUAN

#### 1.1 Latar Belakang

Pada era saat ini tentu kita tidak asing lagi dengan plastik. Hampir disemua kegiatan kita sehari-hari dapat kita jumpai plastik. Plastik banyak digunakan pada perabotan rumah tangga, sebagai kasing alat-alat elektronik dan lain sebagainya. Selain memiliki sifat yang tidak mudah pecah dan lentur, plastik juga memiliki nilai ekonomis yang tinggi dan mudah diaplikasikan sehingga penggunaannya sangat luas diberbagai bidang. Salah satu jenis plastik yg sering kita jumpai adalah *styrofoam* atau plastik busa.

Kata "*styrofoam*" sebenarnya merupakan nama dagang yang telah dipatenkan oleh perusahaan *dow chemical* untuk salah satu jenis plastik kemasan, yaitu *polystyrene foam*. *Styrofoam* sangat ringan, karena kandungan di dalamnya 90-95% udara dan 5-10% *styrene*. Umumnya digunakan sebagai bahan pelindung dan penahan getaran barang-barang yang mudah pecah atau peka terhadap terhadap guncangan, seperti alat elektronik, jenis plastik ini dinamakan "gabus" oleh beberapa kalangan masyarakat indonesia. Bahan dasar *styrofoam* adalah *polystyrene* yaitu suatu jenis plastik yang sangat ringan, kaku, tembus cahaya, dan murah. Namun, bahan tersebut cepat rapuh karena kelemahannya tersebut, sehingga *polystyrene* di campur seng dan senyawa *butadine*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah warna menjadi putih susu.

Karena *styrofoam* banyak digunakan dalam banyak aplikasi dan diperlukan bentuk tertentu maka diperlukan suatu teknik pemotongan yang bagus. Teknik pemotongan *styrofoam* pada awalnya hanya menggunakan pisau biasa. Namun, ketika ditemukan pisau lipat atau yang saat ini populer disebut *cutter*, maka pisau biasa pun berangsur-angsur ditinggalkan dan jarang digunakan untuk memotong *syrofoam*. Hal ini dipicu karena *cutter* ternyata memiliki banyak keunggulan, misalnya lebih ringan, lebih tajam dalam memotong, berkas potonganya sangat tipis dan halus. Seiring berkembangnya seni rupa dan dekorasi, maka kemampuan *cutter* sebagai pemotong *styrofoam* mulai ada kekurangannya. Terutama untuk memotong



garis melengkung atau bentuk kurva nonlinear. Sehingga diproduksi jenis pemotong *styrofoam* yang lebih modern, yaitu menggunakan elemen kawat yang dipanaskan menggunakan *heater* listrik. Hal ini dilatarbelakangi dari sifat *styrofoam* yang sama seperti plastik lainnya yaitu meleleh apabila dikenai panas. Alat ini hanya mampu digunakan untuk membentuk pola-pola berukuran kecil. Alat ini masih dioperasikan secara manual (menggunakan tangan). Alat ini sangat tergantung dengan keahlian tangan yang mengoperasikan. Selain itu, alat ini menghasilkan panas yang cukup berbahaya dan dapat menimbulkan luka bakar yang serius jika tidak dioperasikan dengan baik.

*Styrofoam* sekarang juga digunakan untuk membuat model pesawat *RC model* yang digunakan pada *aeromodeling*. Misalnya dalam pembuatan *fuselage*, sayap dan bagian-bagian pesawat lainnya. Hal ini dikarenakan sifat dari *styrofoam* yang ringan namun kuat, kokoh, mudah dibentuk dan lentur. Jika menggunakan aluminium, terlalu berat dan mahal dan jika menggunakan kayu ternyata rawan patah, mudah lapuk serta rawan dimakan rayap walau mudah dipahat. Plastik jenis lainnya cukup bagus sebagai bahan dasar pembuatan pesawat *RC model*, namun plastik agak sulit dibentuk dan harus dicetak terlebih dahulu untuk mendapatkan bentuk yang diinginkan.

Dalam teknologi molding dan pembuatan *RC model* pesawat berbahan *styrofoam*, diperlukan teknik pemotongan *styrofoam* menggunakan kawat panas (*hot wire cutting*) untuk masing-masing bagian. Untuk melakukan pemotongan bagian pesawat yang panjang contohnya sayap atau badan pesawat maka diperlukan suhu *hotwire* dan kecepatan gerak *hotwire* yang tepat. Untuk membentuk bagian yang miring diperlukan pemotong yang dapat membentuknya secara simetris dan tidak simetris. Untuk itu diperlukan suatu mesin *servo* yang kecepatannya stabil sesuai dengan suhu dan luas penampang yang diperlukan *styrofoam* sehingga hasil pemotongannya bagus.

Untuk membentuk bagian badan pesawat *RC model* yang berupa tabung yang ujungnya mengerucut diperlukan suatu alat pemotong *styrofoam* yang mampu membentuk *fuselage* tersebut secara simetris dan rapi. *Fuselage* dibuat simetris dan rapi agar badan pesawat memiliki tingkat aerodinamis yang tinggi disamping untuk

menempatkan titik keseimbangan badan pesawat *fuselage* tepat pada porosnya. Agar pemotongan dapat dilakukan cukup rapi dan simetris perlu adanya suatu *hotwire* yang dapat memotong *styrofoam* yg tegak lurus dan *radial*. *Styrofoam* yang akan dipotong ditempatkan di pusat as sumbu putar kemudian diputar sedangkan *hotwire* yang dapat diubah-ubah sudutnya terhadap bidang alas akan memotong bagian-bagian *styrofoam* yang tidak dikehendaki saat *styrofoam* diputar dengan sebuah motor dc *servo* sehingga didapat hasil *styrofoam* membentuk geometri sifatnya tabung simetris dan agar hasil pemotongan lebih rapi. Untuk menghasilkan bentuk *styrofoam* yang rapi perlu diatur kecepatan angular dari sisi *styrofoam* yang akan mengenai *hotwire* pemotong. Jika motor dc *servo* terlalu cepat berputar maka hasil pemotongan menjadi tidak rapi karena *hotwire* belum memiliki cukup suhu untuk memotong bagian *styrofoam* selanjutnya. Namun jika motor dc *servo* terlalu pelan maka dikhawatirkan suhu *hotwire* akan melelehkan *styrofoam* di sekitar *hotwire* hingga kurang rapi permukaan potong *styrofoamnya*. Ketepatan pemotongan *styrofoam* ini sangat diperlukan agar kualitas pemotongan menjadi lebih baik dan memiliki ukuran yang tepat. Panas yang terlalu tinggi menyebabkan kualitas pemotongan menjadi tidak rata dan ukuran menjadi tidak tepat.

Hal ini tergantung pada jenis *styrofoam* yang akan dipotong karena titik leleh masing-masing *styrofoam* berbeda. Adapun parameter yang menentukan kualitas pemotongan adalah :

- Temperatur kawat pemotong
- Jenis *styrofoam*
- Kecepatan pemotongan

Untuk membuat badan pesawat (*fuselage*) bentuk pemotongan umumnya tidak simetris dalam dimensi panjang.

Dalam suatu sistem yang akan dikontrol, tentu terdapat gangguan (*disturbance*) dalam sistem. *Disturbance* dalam sistem pengontrolan kecepatan pemotongan *styrofoam* secara *radial* ini adalah jenis *styrofoam* dan dimensi *styrofoam* yang akan dipotong.

Dalam skripsi ini akan didesain sistem kontrol yang akan mengatur sistem mekanik untuk membuat badan pesawat (*fuselage*) yang berbahan dasar *styrofoam*.



## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas diperoleh perumusan masalah sebagai berikut :

1. Bagaimana merancang mekanikudukan *hotwire cutting* untuk pemotongan *styrofoam* secara *radial* pada badan pesawat model.
2. Bagaimana merancang sistem kontrol *servo* secara digital menggunakan mikrokontroler untuk pemotongan *styrofoam* secara *radial*.
3. Bagaimana mengontrol kecepatan putar motor DC agar hasil pemotongan menjadi lebih rapi.

## 1.3 Batasan Masalah

Adapun batasan masalah dalam skripsi ini adalah:

1. Dititikberatkan pada bagaimana mengontrol kecepatan putar dari motor dc agar didapatkan hasil pemotongan yang sesuai dan rapi.
2. Menggunakan motor DC 24V sebagai aktuator
3. Menggunakan mikrokontroler atmega328 sebagai kontrolernya
4. Menggunakan tegangan DC

## 1.4 Tujuan

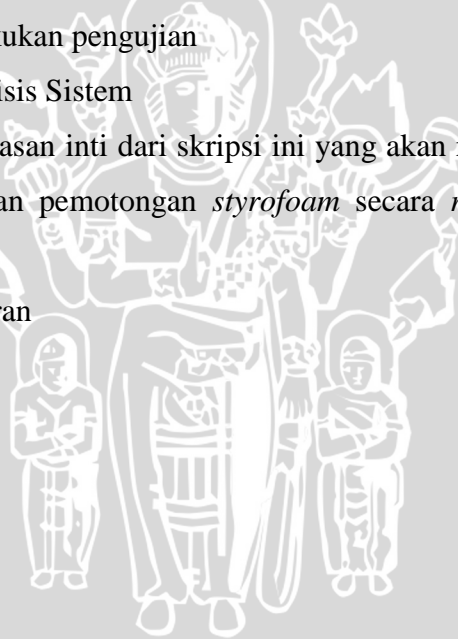
Tujuan yang ingin dicapai dari tugas akhir ini adalah terciptanya *hotwire cutter styrofoam* yang mampu memotong *styrofoam* secara *radial* agar dihasilkan permukaan potong yang rapi. Sehingga nantinya pada masa datang alat ini dapat membantu dalam pembuatan pesawat *RC modeling* dengan bahan baku *styrofoam*.

## 1.5 Sistematika Pembahasan

Sistematika pembahasan tentang hal-hal yang dibahas dalam setiap bab pada penulisan skripsi ini adalah sebagai berikut:



- Bab I      Pendahuluan  
Membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan, dan sistematika pembahasan penelitian.
- Bab II      Dasar Teori  
Membahas tentang dasar teori *styrofoam*, mikrokontroler, motor *servo* dc dan tentang sistem pengendalian kecepatan motor *servo* untuk pemotongan *styrofoam* secara *radial* dengan *hotwire* berbasis mikrokontroler
- Bab III     Metodologi Penelitian  
Berisi mengenai metode penelitian yang dilakukan untuk mencapai sasaran penelitian.
- Bab IV     Perancangan Dan Pembuatan Alat  
Merupakan perancangan sistem baik perangkat keras dan perangkat lunak agar nantinya dapat dilakukan pengujian
- Bab V      Pengujian dan Analisis Sistem  
Merupakan pembahasan inti dari skripsi ini yang akan membahas perancangan pengontrol kecepatan pemotongan *styrofoam* secara *radial* yang diatur oleh mikrokontroler.
- Bab VI     Kesimpulan dan Saran



## BAB II

### TINJAUAN PUSTAKA

Sebelum perancangan dan pembuatan perangkat ini, tentu dibutuhkan pemahaman tentang berbagai hal yang mendukung, maka perlu adanya penjelasan dan uraian teori penunjang yang digunakan dalam penulisan skripsi ini. Teori-teori penunjang yang akan dijelaskan dalam bab ini adalah :

1. *Styrofoam*
2. Perkembangan alat potong *styrofoam*
3. Mikrokontroler ATmega 328
4. Sensor kecepatan *radial (incremental encoder)*
5. LCD (*Liquid Crystal Display*).
6. PWM (*Pulse width Modulation*).
7. Motor DC *servo*
8. Kontroler PID (*Proportional Integrator Derivative*)

#### 2.1 Styrofoam

##### 2.1.1 Pengertian *Styrofoam*

*Styrofoam* atau plastik busa adalah salah satu jenis plastik. Kata ”*styrofoam*” sebenarnya merupakan nama dagang yang telah dipatenkan oleh perusahaan dow chemical untuk salah satu jenis plastik kemasan, yaitu *polystyrene foam*. *polystyrene foam* dikenal luas dengan istilah *styrofoam* yang seringkali digunakan secara tidak tepat oleh publik karena sebenarnya *styrofoam* merupakan nama dagang yang telah dipatenkan oleh perusahaan Dow Chemical. Oleh pembuatnya *Styrofoam* dimaksudkan untuk digunakan sebagai insulator pada bahan konstruksi bangunan, bukan untuk kemasan pangan. 5-10% gas seperti *n-butane* atau *n-pentane*. *Polystyrene* dibuat dari *monomer stirena* melalui proses polimerisasi suspensi pada tekanan dan suhu tertentu, selanjutnya dilakukan pemanasan untuk melunakkan resin dan

mengungkapkan sisa *blowing agent*. *Polystyrene foam* merupakan bahan plastik yang memiliki sifat khusus dengan struktur yang tersusun dari butiran dengan kerapatan rendah, mempunyai bobot ringan, dan terdapat ruang antar butiran yang berisi udara. Kemasan plastik jenis *polistirena* sering menimbulkan masalah pada lingkungan karena bahan ini sulit mengalami peruraian biologis dan sulit didaur ulang.

### 2.1.2 Bahan dasar *Styrofoam*

*Styrofoam* sangat ringan, karena kandungan di dalamnya 90-95% udara dan 5-10% *styrene*. Umumnya digunakan sebagai bahan pelindung dan penahan getaran barang-barang yang mudah pecah atau peka terhadap terhadap guncangan, seperti alat elektronik, jenis plastik ini dinamakan "gabus" oleh beberapa kalangan masyarakat Indonesia. Bahan dasar *styrofoam* adalah *polystyrene*, suatu jenis plastik yang sangat ringan, kaku, tembus cahaya, dan murah. Namun, bahan tersebut cepat rapuh karena kelemahannya tersebut, *polyesterine* di campur seng dan senyawa *butadien*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah menjadi putih susu. Kemudian untuk kelenturannya, ditambahkan zat *plasticier* seperti *dioktilptatat* (DOP), *butil hidroksi tolulena*, atau *n-butyl stearat*. Plastik busa yang mudah terurai menjadi struktur sel-sel kecil merupakan hasil proses peniupan dengan menggunakan gas *chorofluorocarbon* (CFC) dan hasilnya adalah bentuk seperti yang digunakan saat ini. *Polysterine* memiliki massa jenis sampai  $1050 \text{ kg/m}^3$ , kuat tarik sampai  $40 \text{ MN/m}^2$ , modulus geser sampai  $0,990 \text{ GN/m}^2$  dan angka *poisson* 0,330 (Crawford, 1998 dalam Wijaya, 2005).

### 2.1.3 Titik leleh

Karena merupakan salah satu jenis plastik yang lentur, maka *styrofoam* pun dapat meleleh jika dipanaskan. Umumnya *polystyrene* yang merupakan penyusunan utama *styrofoam* meleleh pada suhu  $80^\circ\text{C}$ . ("bahaya dan keuntungan kemasan primer" 2008, July).



#### 2.1.4 Kelenturan dan kekuatan

Pada dasarnya *Styrofoam* merupakan bahan yang mudah rapuh. Karena kelemahannya tersebut, *polysterine* di campur seng dan senyawa *butadien*. Hal ini menyebabkan *polystyrene* kehilangan sifat jernihnya dan berubah menjadi putih susu. Kemudian untuk kelenturannya, ditambahkan zat *plasticier* seperti *dioktilptatat* (DOP), *butil hidroksi tolulena*, atau *n-butyl stearat*. Plastik busa yang mudah terurai menjadi struktur sel-sel kecil merupakan hasil proses peniupan dengan menggunakan gas *chorofluorocarbon* (CFC). Hasilnya adalah bentuk seperti yang dipergunakan saat ini. *Polystyrene* memiliki massa jenis sampai  $1050 \text{ kg/m}^3$ , kuat tarik sampai  $40 \text{ MN/m}^2$ , modulus geser sampai  $0,990 \text{ GN/ m}^2$  dan angka *poisson* 0,330 (Crawford, 1998 dalam Wijaya, 2005). Koefisien gesek *polystyrene* atau *styrofoam* dengan baja, yaitu antara 0,3-0,35 (*coefficient of friction*, 2006). Karena merupakan salah satu jenis plastik yang lentur, maka *styrofoam* pun dapat meleleh jika dipanaskan. Umumnya *polystyrene* yang merupakan penyusunan utama *styrofoam* meleleh pada suhu  $80^\circ\text{C}$ . ("bahaya dan keuntungan kemasan primer" 2008, July).

#### 2.1.5. Jenis-jenis *styrofoam*

- EPO

EPO (*Piocelan*) adalah bahan busa baru yang merupakan campuran dari PE dan PS. bahan EPO mempunyai daya tahan tertinggi, serta berkualitas baik dan jenis busa yang relatif tangguh dan tidak patah seperti beberapa jenis busa lainnya.

- EPS

*Expanded polystyrene* (EPS) busa adalah busa paling umum dipotong untuk bentuk arsitektur, sistem panel struktural terisolasi (SIPS), terisolasi bentuk beton (ICF), cetakan beton pracetak dan model dengan tema, dan tanda pembuatan. EPS secara teknis plastik. plastik 'Kata' berasal dari *plastikos* Yunani, yang berarti mampu menjadi dibentuk. Kebebasan untuk berkreasi

dengan desain khusus adalah alasan utama mengapa EPS hari ini begitu berharga.

- EPP

*Poliolefin - poliolefin* adalah kelompok terbesar termoplastik, sering disebut sebagai termoplastik komoditi, mereka adalah *polimer olefin* sederhana seperti *ethylene, propylene, butenes, isoprenes, dan pentenes, dan kopolimer* dan modifikasi daripadanya. *Poliolefin* hanya terdiri dari atom karbon dan hidrogen dan mereka adalah non-aromatik. *Poliolefin* biasanya diproses oleh *ekstrusi, injection molding, blow molding, dan metode molding* rotasi. *Thermoforming, calender, dan pencetakan kompresi* digunakan untuk tingkat yang lebih rendah. Sebuah karakteristik yang melekat umum untuk semua *poliolefin* adalah, nonpolar tidak keropos, rendah energi permukaan yang tidak menerima tinta, dan lak tanpa *pretreatment* oksidatif khusus. Dua *poliolefin* paling penting dan umum adalah *polyethylene* dan *polypropylene (EPP)* mereka sangat populer karena biaya rendah dan berbagai aplikasi.

- Depron

*Depron* adalah sel busa *styrofoam* tertutup dalam bentuk lembaran, merupakan media yang bagus untuk membangun pesawat model, dan juga media yang populer untuk membangun suatu model arsitektur, serta perahu model dan desain *prototype*. Bahan itu cukup ringan dan kuat untuk membawa motor batere elektrik yang ringan. *Depron* dikembangkan terutama digunakan untuk sebagai isolasi panas dan suara. *Depron* sebenarnya *polystyrene (EPS/XPS)* yang lebih kuat dan padat yang terjadi dalam bentuk sel busa yang tertutup. Buka atau tertutupnya sel mengacu pada struktur dan sel jenis terbuka (jenis spons) dapat menyerap air atau lem sehingga sel jenis terbuka kurang cocok untuk pembuatan pesawat model. Sel busa tertutup lebih unggul karena lebih sedikit menyerap lem dan lebih tahan air. Permukaannya yg tertutup menambahkan kekuatan yang cukup serta sangat mudah untuk mengaplikasikan cat tidak larut *depron* yang ramah lingkungan atau tinta sebagai *finishing* pesawat model.



## 2.2. Perkembangan Alat pemotong *Styrofoam*

Alat pemotong *styrofoam* pada awalnya hanya menggunakan pisau biasa. Namun, ketika ditemukan pisau lipat atau yang saat ini populer disebut *cutter*, maka pisau biasa pun berangsur-angsur ditinggalkan dan jarang digunakan untuk memotong *styrofoam*. Hal ini dipicu karena *cutter* ternyata memiliki banyak keunggulan, misalnya lebih ringan, lebih tajam dalam memotong, berkas potongannya sangat tipis dan halus.



**Gambar 2.1** Cutter

Sumber : <http://www.okokchina.com>

Seiring berkembangnya seni rupa dan dekorasi, maka kemampuan *cutter* sebagai pemotong *styrofoam* mulai ada kekurangannya. Terutama untuk memotong garis melengkung atau bentuk kurva nonlinear. Sehingga diproduksi jenis pemotong *styrofoam* yang lebih modern yang biasa dikenal dengan *hotwire*, yaitu menggunakan elemen kawat yang dipanaskan menggunakan heater listrik. Hal ini dilatarbelakangi dari sifat *styrofoam* yang meleleh apabila dikenai panas. Alat ini hanya mampu digunakan untuk membentuk pola-pola berukuran kecil. Alat ini masih dioperasikan secara manual (menggunakan tangan). Alat ini sangat tergantung dengan keahlian tangan yang mengoperasikan. Selain itu, alat ini menghasilkan panas yang cukup berbahaya dan dapat menimbulkan luka bakar yang serius jika tidak dioperasikan dengan baik.





**Gambar 2.2** Hotwire tool

Sumber : <http://www.anticsonline.co.uk>

*Styrofoam* sekarang juga digunakan untuk membuat model pesawat *remote control* (RC) yang digunakan pada *aeromodeling*. Misalnya dalam pembuatan *fuselage* (bagian badan pesawat), wing sayap dan bagian-bagian pesawat lainnya. Hal ini dikarenakan sifat dari *styrofoam* yang ringan namun kuat dan kokoh, mudah dibentuk dan lentur. Jika menggunakan aluminium, terlalu berat serta sulit dibentuk dan jika menggunakan kayu maka kayu rawan patah. Plastik cukup bagus sebagai bahan dasar pembuatan pesawat RC model, namun plastik agak sulit dibentuk dan harus dicetak terlebih dahulu.

### 2.3 Mikrokontroler ATMEGA 328

Secara umum, mikrokontroler berfungsi sama dengan komputer. Bedanya adalah mikrokontroler memiliki desain dalam sebuah *single chip* (IC). Mikrokontroler terdapat di hampir semua peralatan elektronik di sekeliling kita, didalam tape, TV, radio, telepon genggam (*Hand Phone*) dll. Mikrokontroler memiliki kemampuan yang diperlukan untuk membuat keputusan berdasarkan sinyal dari luar dengan kata lain mikrokontroler merupakan otak dari sebuah perangkat elektronik.

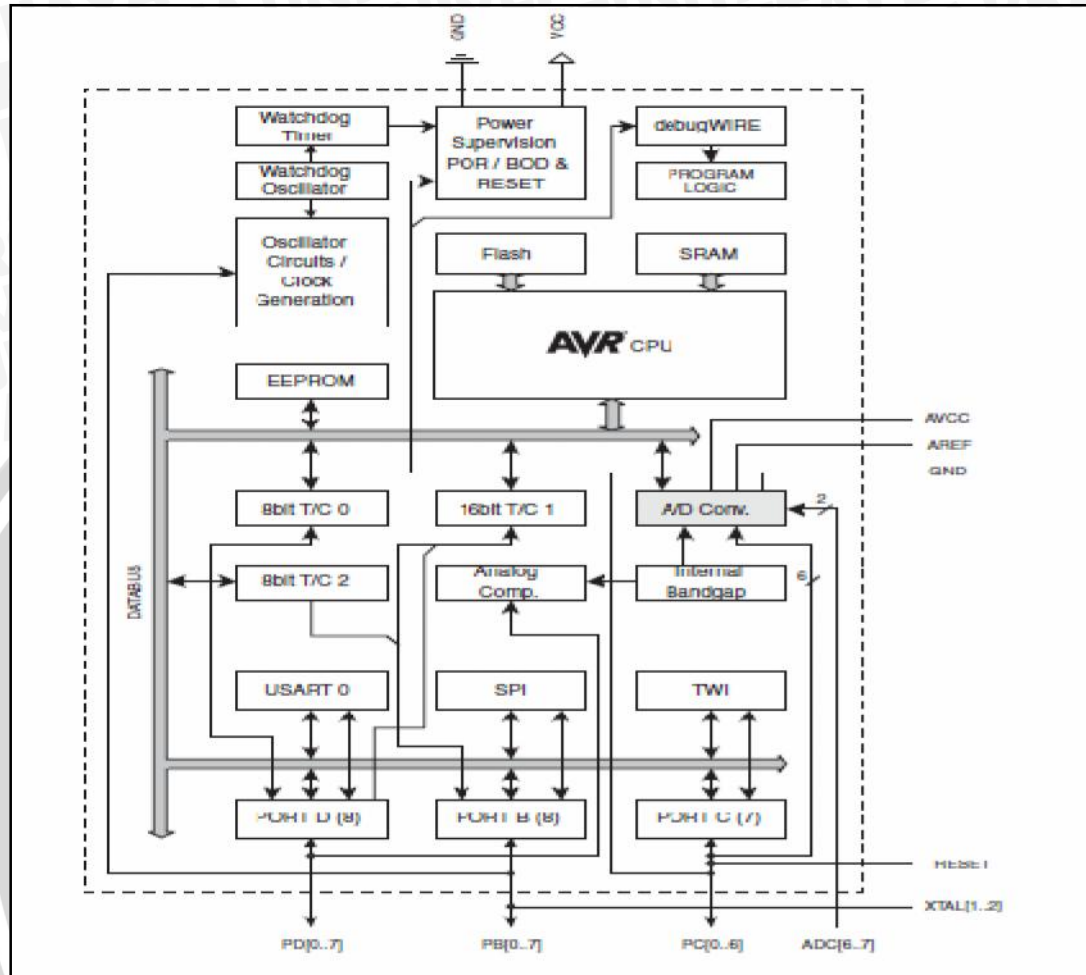
AVR merupakan salah satu jenis mikrokontroler yang didalamnya terdapat berbagai macam fungsi. Perbedaannya dengan mikrokontroler yang pada umumnya digunakan MCS 51 adalah pada AVR tidak perlu menggunakan *oscilator* eksternal karena didalamnya sudah terdapat internal *oscillator*. Selain itu kelebihan dari AVR

adalah memiliki *Power-On Reset*, yaitu tidak perlu adanya tombol *reset* dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC 10- bit, EEPROM sekitar 1 *KBytes* sampai dengan 256 *KBytes*.

Dalam hal ini yang digunakan adalah AVR Atmega328. untuk Atmega328 hanya dapat bekerja pada tegangan 4,5 V- 5,5 V. Berikut adalah gambar dari blok diagram untuk ATmega328. Secara umum konfigurasi yang dimiliki mikrokontroler ATmega328 adalah :

- Catu daya 1,8- 5,5V
- 32 *Kbytes flash memory*
- *Programmable serial USART*
- 1 *KBytes EEPROM*
- 2 *Kbytes RAM*
- Mempunyai rentang temperatur operasi antara  $-40^{\circ}\text{C} - 85^{\circ}\text{C}$
- Frekuensi maksimum 20 MHz
- *In-System Programming by on On- chip Boot*
- Satu buah 16 bit timer, and dua 8 bit timer
- *Built in watchdog timer, PWM 6 channel, SPI, and TWI interface*
- *On-chip komparator analog*
- terdiri 8 channel 10 bit ADC
- Internal RC *osilator* yang sudah dikalibrasi
- *Built-in Real Time Counter* dengan osilator terpisah
- 23 *programmable I/O*
- *Package 32 TQFP*

Blok diagram ATmega328 ditunjukkan dalam gambar 2.3



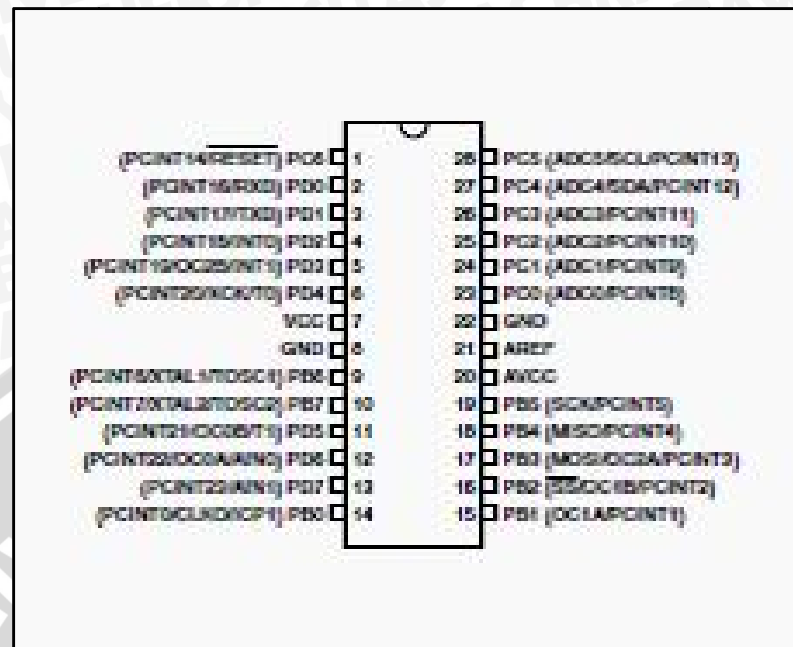
**Gambar 2.3** Blok Diagram ATmega328

Sumber : Atmel, 2011 : p.5

### 2.3.1 Konfigurasi pin

ATmega328 memiliki 28 pin yang masing-masing pin-nya memiliki fungsi yang berbeda-beda baik sebagai port ataupun sebagai fungsi yang lain. Berikut akan dijelaskan tentang kegunaan dari masing-masing kaki pada ATmega. Susunan masing-masing pin dapat dilihat dalam gambar 2.4





**Gambar 2.4** Konfigurasi Pin ATmega328

Sumber : Atmel, 2011 : p.2

Fungsi kaki-kaki Pin ATmega328 adalah :

1. VCC  
Merupakan *supply* tegangan untuk digital.
2. GND  
Merupakan ground untuk semua komponen yang membutuhkan *grounding*.

3. Port B

Di dalam port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah port B adalah 8 buah pin mulai dari pin B.0 sampai dengan pin B.7. Tiap pin dapat digunakan sebagai input dan juga input dan juga output. Port B merupakan sebuah 8-bit *bi-directional* I/O port dengan internal *pull-up resistor*. Sebagai input, pin-pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up resistor* diaktifkan. Jika ingin menggunakan tambahan kristal, maka cukup menghubungkan kaki dari kristal ke kaki pada pin

port B. Namun jika tidak digunakan, maka cukup dibiarkan saja. Penggunaannya kegunaan dari masing-masing kaki ditentukan dari *clock fuse* settingnya.

4. Port C

Port C merupakan sebuah 7-bit *bi-directional I/O* port yang di dalam masing-masing pin terdapat *pull-up resistor*. Jumlah pin-nya hanya 7 buah mulai dari pin C.0 sampai dengan pin C.5 sebagai keluaran/output, port C memiliki karakteristik yang sama dalam hal kemampuan menyerap arus (sink) atau mengeluarkan arus (source).

5. RESET / PC6

Jika *RSTDISBL fuse* diprogram, maka PC6 akan berfungsi sebagai pin I/O. Untuk diperhatikan juga bahwa pin ini memiliki karakteristik yang berbeda dengan pin-pin yang terdapat pada port C. Namun jika *RSTDISBL fuse* tidak diprogram, maka pin ini akan berfungsi sebagai input reset. Dan jika level tegangan yang masuk ke pin ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun clock-nya tidak bekerja.

6. Port D

Port D merupakan 8-bit *bi-directional I/O* dengan internal *pull-up resistor*. Fungsi dari port ini sam dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada port ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

7. AVCC

Pada pin ini memiliki fungsi sebagai *supply* tegangan untuk ADC. Untuk pin ini harus dihubungkan secara terpisah dengan VCC karena pin ini digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan, tetap saja disarankan untuk menghubungkan secara terpisah dengan VCC. Cara menghubungkan AVCC adalah melewati *low-pass filter* setelah itu dihubungkan dengan VCC.

8. AREF

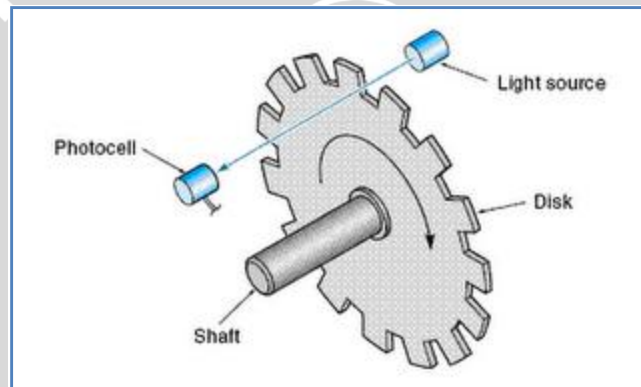
Pin referensi analog jika menggunakan ADC.

#### 9. ADC 7:6

Didalamnya terdapat TQFP dan QFN/MLF. ADC7:6 sebagai input analog ke A/D konverter. Pin ini di supply daya dari tegangan analog dan terdapat 10-bit ADC channel

### 2.4 Sensor kecepatan *radial (incremental encoder)*

Pada umumnya, rotary encoder digunakan sebagai sensor pengukur kecepatan pada sebuah motor. Berikut adalah gambar desain mekanik dari sebuah rotary encoder.

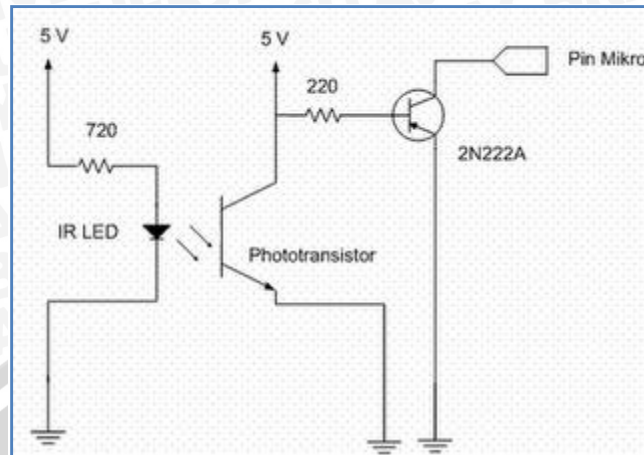


**Gambar 2.5** Desain Umum *incremental encoder*

Sumber : <http://bocah-cakil.blogspot.com/2011/07/rangkaian-sensor-kecepatan.html>

Prinsip kerja dari sensor ini adalah sebagai berikut saat rangkaian sumber cahaya diberikan VCC 5 Volt dan menghasilkan cahaya, cahaya masuk pada photodiode tidak terhalangi maka akan menghasilkan tegangan sekitar 5 V dan begitu juga sebaliknya saat terhalangi maka akan menghasilkan tegangan sekitar 0 V. Dimana tegangan menjadi inputan untuk mikrokontroler. Berikut adalah gambar rangkaian sensor kecepatan yang akan digunakan.





**Gambar 2.6** Rangkaian Sensor Kecepatan

Untuk mengaktifkan inframerah LED dibutuhkan tegangan sebesar 5 volt. Sebagai receiver inframerah adalah *phototransistor*, cara kerja *phototransistor* sebenarnya sama seperti transistor lain. Bila basis tidak dialiri arus yang cukup, maka transistor cut off. Kondisi cut off disebabkan oleh terdapatnya sesuatu yang memotong antara IR LED dan *phototransistor*. Pada perancangan ini sesuatu yang memotong tersebut adalah disk. Bila tidak ada sesuatu yang memotong antara IR led dan *phototransistor*, artinya *phototransistor* akan mendapat sinar inframerah dari IR LED, maka basis *phototransistor* akan teraliri arus. Hal ini menyebabkan arus akan mengalir ke emitter. Kemudian arus akan masuk ke resistor 220 ohm, resistor 220 ohm berfungsi untuk mengurangi arus yang berlebih dari *phototransistor* dan mengaktifkan basis transistor 2N222A, emitter 2N222A disini sebagai pemberi sinyal masukkan ke mikrokontroler. Salah satu Aplikasi incremental encoder sebagai sensor posisi digunakan pada Mouse Tipe Analog (Mouse yang Menggunakan Bola )/

## 2.5 LCD

Untuk menampilkan menu dari suatu peralatan elektronika seperti catu daya *switching* misalnya tentang berapa tegangan keluaran yang diinginkan, maka diperlukan suatu tampilan. Yang lebih sering digunakan adalah LCD.

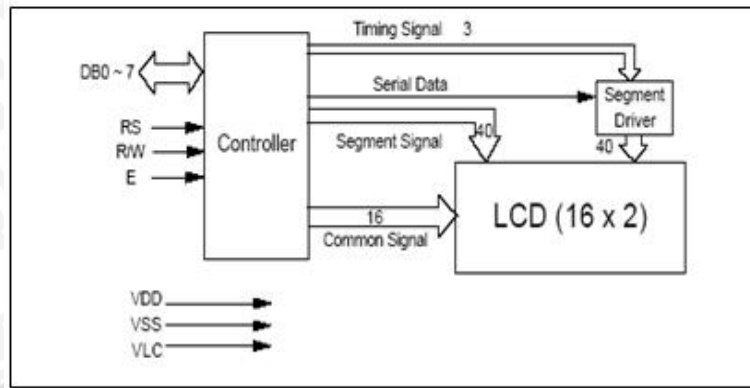
*Liquid crystal display* (LCD) merupakan komponen elektronika yang digunakan untuk menampilkan suatu karakter baik itu angka, huruf atau karakter tertentu, sehingga tampilan tersebut dapat dilihat secara *visual*. Pemakaian LCD sebagai indikator tampilan banyak digunakan disebabkan daya yang dibutuhkan LCD relatif kecil (orde mikrowatt), di samping itu dapat juga menampilkan angka, huruf atau simbol dan karakter tertentu. Meskipun pada komponen ini dibatasi oleh sumber cahaya eksternal/internal, suhu, dan *lifetime*.

LCD terdiri atas tumpukan tipis atau sel dari dua lembar kaca dengan pinggiran tertutup rapat. Antara dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang tembus cahaya. Permukaan luar masing-masing keping kaca mempunyai lapisan tembus cahaya seperti oksida timah (*tin oxide*) atau oksida indium (*indium oxide*). Sel mempunyai ketebalan  $1 \times 10^{-5}$  meter dan diisi dengan kristal cair.

LCD yang digunakan disini untuk menampilkan perintah-perintah dalam tahapan pengisian bahan bakar secara mandiri. LCD yang akan digunakan bertipe M1632 produksi SEIKO instrument inc. corporation. Spesifikasi dari LCD ini adalah sebagai berikut:

- Menampilkan 16 karakter pada tiap baris TN LCD dengan 5 x 7 dot matrik
- Pembangkit karakter ROM untuk 192 jenis karakter
- Pembangkit karakter RAM untuk 8 jenis karakter
- 80 x 8 bit data RAM
- Tegangan catu 5 volt dan temperatur operasi 0–50°C
- Otomatis reset pada saat dihidupkan

Masukan yang diperlukan untuk mengendalikan modul LCD ini berupa bus data yang masih termultipleks dengan bus alamat serta 3 bit sinyal kontrol, yaitu RS, R/W dan E. Sementara pengendali dot matrix LCD dilakukan secara internal oleh kontroler yang sudah terpasang pada modul LCD. Berikut ini adalah diagram blok dari LCD M1632.



**Gambar 2.7** Diagram Blok LCD M1632.

Sumber : Anonymous, 2001:3

Fungsi pin dari LCD M1632 ditunjukkan dalam Tabel 2.1.

No Pin	Nama Pin	Fungsi
16	V – BL	Sebagai <i>ground</i> dari <i>backlight</i>
15	V + BL	Sebagai kutub positif dari <i>backlight</i>
7 – 14	DB0– DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan.
6	E	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca
5	R/W	Sinyal seleksi tulis atau baca 0 = tulis 1 = baca
4	RS	Sinyal pemilih <i>register</i> 0 = register instruksi (tulis) 1 = register data (tulis dan baca)
3	Vlc	Untuk mengendalikan kecerahan LCD dengan mengubah Vlc
2	Vcc	Tegangan catu + 5 volt



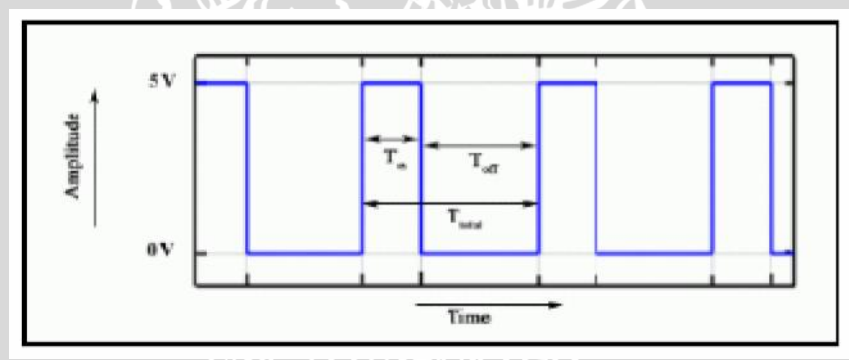
1	Vss	Terminal <i>Ground</i>
---	-----	------------------------

**Tabel 2.1** Fungsi Pin-Pin pada LCD M1632

**Sumber :** Datasheet LCD M1632

**2.6 PWM (Pulse Width Modulation)**

Secara umum *Pulse Width Modulation* atau PWM adalah sebuah cara memanipulasi lebar sinyal atau tegangan yang dinyatakan dengan pulsa dalam satu periode, yang akan digunakan untuk mentransfer data pada telekomunikasi atau mengatur tegangan sumber yang konstan untuk mendapatkan tegangan rata-rata yang berbeda. Penggunaan PWM sangat banyak, antara lain untuk pemodulasian data untuk telekomunikasi, pengontrolan daya atau tegangan yang masuk ke beban regulator tegangan, mengatur kecepatan dari motor DC dan lain-lain. Sinyal atau keluaran PWM ditunjukkan dalam Gambar 2.8.



**Gambar 2.8** Sinyal Keluaran PWM

(insansainsprojects.wordpress.com)

Dalam Gambar 2.8 terlihat bahwa sinyal PWM adalah sinyal yang amplitudonya tetap, namun lebar pulsa yang aktif (*duty cycle*) per periodenya dapat diubah-ubah. Periodenya adalah waktu pulsa *high* ( $T_{on}$ ) ditambah waktu pulsa *low* ( $T_{off}$ ).

$$T_{total} = T_{on} + T_{off} \dots\dots\dots (2-1)$$

Sedangkan *duty cycle* (D) adalah lamanya pulsa high (1)  $T_{on}$  dalam satu periode. Jika  $f(t)$  adalah sinyal PWM, maka besar *duty cycle*-nya adalah:

$$D = \frac{T_{on}}{(T_{on}+T_{off})} = \frac{T_{on}}{T_{total}} \dots\dots\dots (2-2)$$

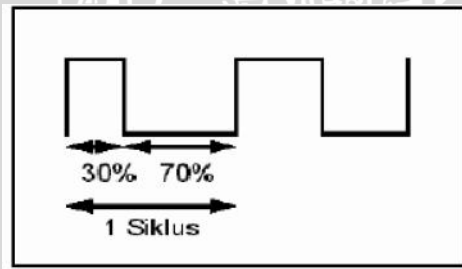
Sedangkan, besarnya tegangan keluaran  $V_{out}$  (volt) adalah perkalian antara *duty cycle* (D)  $V_{in}$  (volt) dan :

$$V_{out} = D \times V_{in} \dots\dots\dots (2-3)$$

Jadi,

$$V_{out} = \frac{T_{on}}{T_{total}} \times V_{in} \dots\dots\dots (2-4)$$

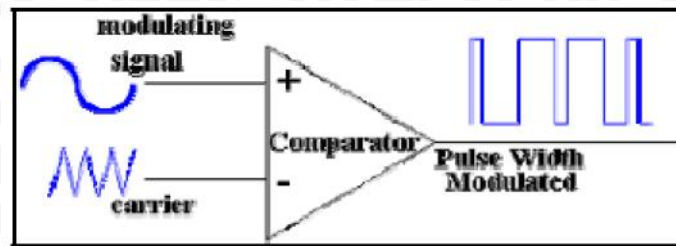
*Duty cycle* menyatakan fraksi waktu sinyal pada keadaan logika *high* dalam satu siklus. Satu siklus diawali oleh transisi *low to high* dari sinyal dan berakhir pada transisi berikutnya. Selama satu siklus, jika waktu sinyal pada keadaan *high* sama dengan *low* maka dikatakan sinyal mempunyai *duty cycle* 50%. *Duty cycle* 30% menyatakan sinyal berada pada logika 1 selama 30% dari dari waktu total. Contoh *duty cycle* ditunjukkan pada gambar 2.9.



**Gambar 2.9.** *Duty Cycle* PWM Sebesar 30%.

**Sumber:** [www.mytutorialcafe.com](http://www.mytutorialcafe.com)

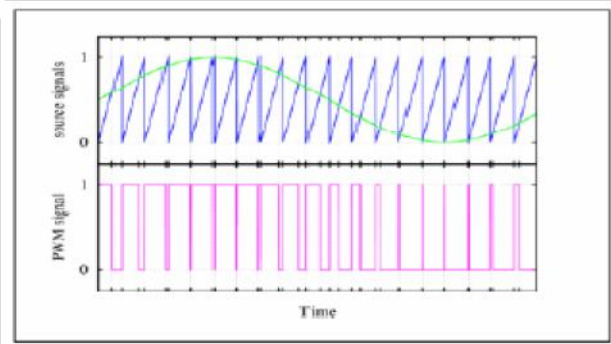
Untuk mendapat sinyal PWM dari masukan berupa sinyal analog, dapat dilakukan dengan membentuk gelombang gigi gergaji atau sinyal segitiga yang diteruskan ke komparator bersama sinyal aslinya. Seperti ditunjukkan pada gambar 2.10.



**Gambar 2.10** Cara Mendapatkan Sinyal PWM.

**Sumber:** [insansainsprojects.wordpress.com](http://insansainsprojects.wordpress.com)

Jika digambarkan dalam bentuk sinyal, maka terlihat seperti pada gambar 2.11 di bawah ini.



**Gambar 2.11.** Sinyal Sumber yang Dimodulasi dan Sinyal PWM.

**Sumber:** [insansainsprojects.wordpress.com](http://insansainsprojects.wordpress.com)

Pada gambar 2.11, sinyal masukan analog dimodulasikan dengan sinyal gigi gergaji, sehingga didapatkan sinyal PWM seperti gambar dibawahnya. Sekarang ini, pada beberapa mikrokontroler sudah dilengkapi dengan fasilitas pembangkit sinyal PWM.

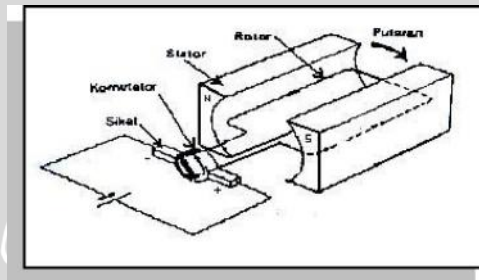
## 2.7 Motor DC

Motor DC merupakan motor arus searah yaitu suatu mesin yang berfungsi mengubah tenaga listrik arus searah menjadi tenaga gerak atau tenaga mekanik, di mana gerak tersebut berupa putaran dari pada rotor.

Motor DC hampir sama konstruksinya dengan motor AC, perbedaannya terletak pada sikat dan cincin belah (*komutator*). Saat siklus pertama, arus



mengalir dari kutub positif ke negatif. Aliran arus yang melewati bagian kabel yang berada didekat kutub N magnet akan menimbulkan gaya Lorentz ke bawah. Sementara itu aliran arus yang melewati kabel yang berada di dekat kutub S magnet akan menyebabkan gaya Lorentz ke atas. Kedua perpaduan gaya Lorentz tersebut akan menyebabkan kawat berputar. Pada siklus berikutnya terjadi hal yang serupa seperti pada siklus sebelumnya. Apabila arus terus-menerus dialirkan, maka kawat akan berputar secara terus menerus pula. Pada aplikasi sesungguhnya, kawat adalah sebuah rotor yang akan dikopel dengan sebuah as dan akan memutar as tersebut terus menerus seiring perputaran motor. Motor DC ditunjukkan dalam Gambar 2.12 di bawah ini:



**Gambar 2.12** Motor DC.

**Sumber :** *Kismet F [1994:98]*

Berbeda dengan motor DC dan motor Stepper, motor DC *servo* adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor *servo*. Motor ini terdiri dari sebuah motor, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran *servo*. Sedangkan sudut dari sumbu motor *servo* diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Tampak pada gambar dengan pulsa 1.5 ms pada periode selebar 2 mS maka sudut dari sumbu motor akan berada pada posisi tengah. Semakin lebar pulsa OFF maka akan semakin besar gerakan sumbu ke arah jarum jam dan semakin kecil pulsa OFF maka akan semakin besar gerakan sumbu ke arah yang berlawanan dengan jarum jam. Motor *servo* biasanya hanya bergerak mencapai sudut tertentu saja

dan tidak kontinu seperti motor DC maupun motor stepper. Walau demikian, untuk beberapa keperluan tertentu, motor *servo* dapat dimodifikasi agar bergerak kontinu.



**Gambar 2.13** motor dc *servo*

Sumber <http://www.automation-drive.com>

### **2.8 Kontroler PID (*Proportional Integrator Derivative*)**

Keberadaan kontroler dalam sebuah sistem kontrol mempunyai kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Artinya, karakteristik plant harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu sub sistem, yaitu kontroler.

Salah satu tugas komponen kontroler adalah mengurangi sinyal kesalahan, yaitu perbedaan antara nilai referensi/nilai yang diinginkan dan nilai aktual. Hal ini sesuai dengan tujuan sistem kontrol adalah mendapatkan nilai sinyal keluaran sama dengan nilai yang diinginkan (referensi). Semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang diterapkan.

Apabila perbedaan antara nilai referensi dengan nilai keluaran relatif besar, maka kontroler yang baik seharusnya mampu mengamati perbedaan ini untuk segera menghasilkan sinyal keluaran untuk mempengaruhi plant. Dengan

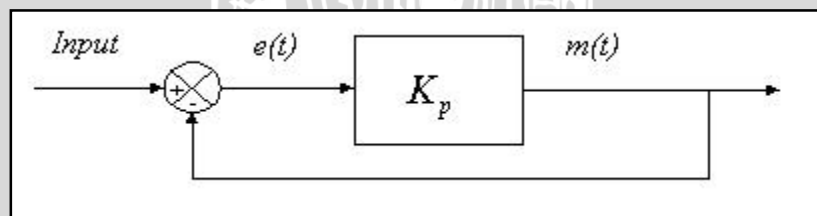
demikian sistem secara cepat mengubah keluaran plant sampai diperoleh selisih dengan nilai referensi sekecil mungkin

Prinsip kerja kontroler adalah membandingkan nilai aktual keluaran plant dengan nilai referensi, kemudian menentukan nilai kesalahan dan akhirnya menghasilkan sinyal kontrol untuk meminimalkan kesalahan (Ogata, 1996: 197-204). adalah sebagai berikut:

### 2.8.1. Kontroler Proporsional

Kontroler proposional memiliki keluaran yang sebanding/proposional dengan besarnya sinyal kesalahan (selisih antara besaran yang diinginkan dengan harga aktualnya) Secara lebih sederhana dapat dikatakan, bahwa keluaran kontroler proporsional merupakan perkalian antara konstanta proporsional dengan masukannya. Perubahan pada sinyal masukan akan segera menyebabkan sistem secara langsung mengubah keluarannya sebesar konstanta pengalinya.

Pada gambar 2.14 menunjukkan blok diagram yang menggambarkan hubungan antara input (besaran referensi yang diinginkan), besaran aktual dengan besaran keluaran kontroler proporsional, dan besaran kesalahan (*error*). Sinyal keasalahan (*error*) merupakan selisih antara besaran setting dengan besaran aktualnya.



**Gambar 2.14.** Diagram blok kontroler proporsional

Pada pengendali proporsional hubungan antara keluaran kontroler  $m(t)$  dan sinyal kesalahan  $e(t)$  adalah

$$m(t) = K_p e(t) \quad (1)$$

dengan  $K_p$  adalah penguatan proporsional. Keluaran  $m(t)$  hanya tergantung pada  $K_p$  dan *error*, semakin besar *error* maka semakin besar koreksi yang dilakukan.



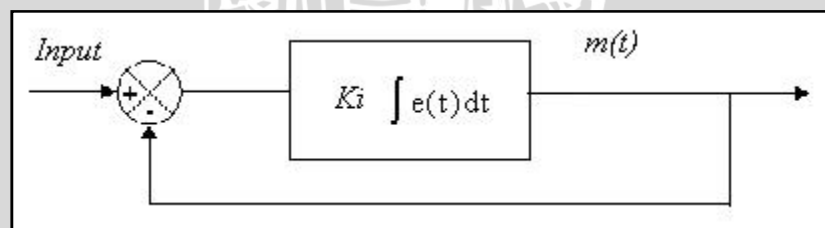
Penambahan  $K_p$  akan menaikkan penguatan sistem sehingga dapat digunakan untuk memperbesar kecepatan respon dan mengurangi kesalahan keadaan mantap. Akan tetapi, penambahan  $K_p$  juga membuat sistem lebih sensitif yang akan juga mengakibatkan ketidakstabilan atau osilasi.

### 2.8.2. Kontroler Integral

Kontroler integral berfungsi menghasilkan respon sistem yang memiliki kesalahan keadaan mantap nol. Kalau sebuah plant tidak memiliki unsur integrator ( $1/s$ ), kontroler proporsional tidak akan mampu menjamin keluaran sistem dengan kesalahan keadaan mantapnya nol. Dengan kontroler integral, respon sistem dapat diperbaiki, yaitu mempunyai kesalahan keadaan mantapnya nol.

Kontroler integral memiliki karakteristik seperti halnya sebuah integral. Keluaran kontroler sangat dipengaruhi oleh perubahan yang sebanding dengan nilai sinyal kesalahan. Keluaran kontroler ini merupakan jumlahan yang terus menerus dari perubahan masukannya. Kalau sinyal kesalahan tidak mengalami perubahan, keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan.

Gambar 2.15. menunjukkan blok diagram antara besaran kesalahan dengan keluaran suatu kontroler integral.



**Gambar 2.15.** Blok diagram kontroler integral

Nilai keluaran kontroler  $m(t)$  sebanding dengan integral sinyal kesalahan  $e(t)$ , sehingga

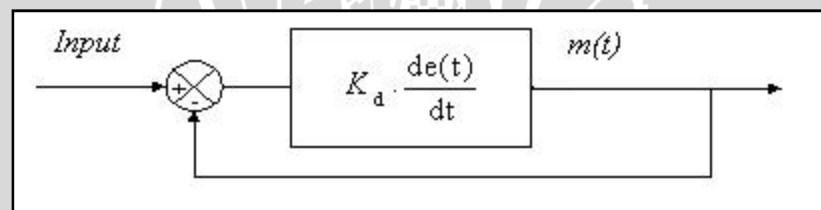
$$\frac{dm(t)}{dt} = K_i e(t) \quad (2)$$

$$m(t) = Ki \int_0^t e(t) dt \quad (3)$$

dengan  $Ki$  adalah konstanta integral. Jika sinyal kesalahan  $e(t) = 0$ , maka laju perubahan sinyal kendali integral  $\frac{dm(t)}{dt} = 0$  atau sinyal keluaran kendali akan tetap berada pada nilai yang dicapai sebelumnya. Aksi kontrol integral digunakan untuk menghilangkan kesalahan posisi dalam keadaan mantap (*error steady state*) tanpa memperhitungkan kecepatan respon.

### 2.8.3. Kontroler Diferensial

Keluaran kontroler diferensial memiliki sifat seperti halnya suatu operasi derivatif. Perubahan yang mendadak pada masukan kontroler, akan mengakibatkan perubahan yang sangat besar dan cepat. Gambar 2.16. menunjukkan blok diagram yang menggambarkan hubungan antara sinyal kesalahan dengan keluaran kontroler.



**Gambar 2.16.** BlokDiagram kontroler diferensial

Nilai keluaran kontroler  $m(t)$  sebanding laju sinyal kesalahan  $\frac{de(t)}{dt}$ .

Hubungan ini dapat ditulis sebagai:

$$m(t) = K_D \frac{de(t)}{dt} \quad (4)$$

Kontroler diferensial akan memberikan sinyal kendali keluaran  $m(t) = 0$ , untuk sinyal kesalahan  $e(t)$  yang konstan sehingga kontroler diferensial tidak mempengaruhi keadaan mantap. Kontroler diferensial digunakan untuk

memperbaiki atau mempercepat respon transien sebuah sistem serta dapat meredam osilasi.

Berdasarkan karakteristik kontroler tersebut, kontroler diferensial umumnya dipakai untuk mempercepat respon awal suatu sistem, tetapi tidak memperkecil kesalahan pada keadaan tunaknya. Kerja kontroler diferensial hanyalah efektif pada lingkup yang sempit, yaitu pada periode peralihan. Oleh sebab itu kontroler diferensial tidak pernah digunakan tanpa ada kontroler lain.

Dari ketiga aksi kontrol dasar di atas dapat dibuat kombinasi dari ketiganya, yaitu:

### **1. Kontroler Proporsional Integral (PI)**

Aksi kontrolnya dinyatakan dalam persamaan:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (5)$$

Kontroler ini menghasilkan sinyal kesalahan  $\int e(t) dt$  kemudian ditambahkan dengan sinyal kesalahan  $e(t)$ .

### **2. Kontroler Proporsional Diferensial (PD)**

Aksi kontrolnya dapat dinyatakan sebagai:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (6)$$

Kontroler PD selalu mengukur kemiringan (*slope*) sinyal kesalahan  $\frac{de(t)}{dt}$  dan memperkirakan besar  $M_p$  yang akan terjadi serta memberikan koreksi sebelum terjadi lewatan sebenarnya sehingga diperoleh *maximum overshoot* yang kecil.

Jika kesalahan keadaan mantap tidak berubah terhadap waktu maka turunannya terhadap waktu sama dengan nol, sehingga kontroler PD tidak mempunyai pengaruh terhadap kesalahan keadaan mantap, tetapi jika terdapat



perubahan kesalahan, kontroler PD akan mengurangi besar kesalahan keadaan mantap. Jadi kontroler PD digunakan untuk memperbaiki suatu sistem pengendalian yang tanggapan peralihannya mempunyai *maximum overshoot* yang berlebihan tanpa memperhitungkan kecepatan responnya.

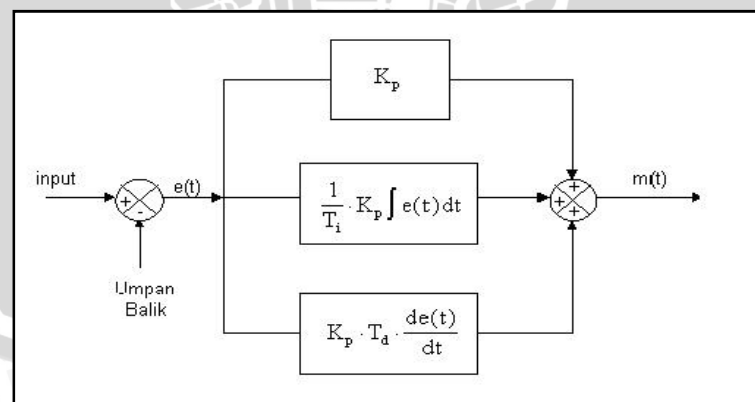
### **3. Kontroler Proporsional Integral Diferensial (PID)**

Setiap kekurangan dan kelebihan dari masing-masing kontroler P, I dan D dapat saling menutupi dengan menggabungkan ketiganya secara paralel menjadi kontroler proporsional integral diferensial (kontroler PID). Elemen-elemen kontroler P, I dan D masing-masing secara keseluruhan bertujuan untuk mempercepat reaksi sebuah sistem, menghilangkan *offset* dan menghasilkan perubahan awal yang besar (Gunterus, 1994, 8-10).

Aksi kontrolnya dinyatakan sebagai:

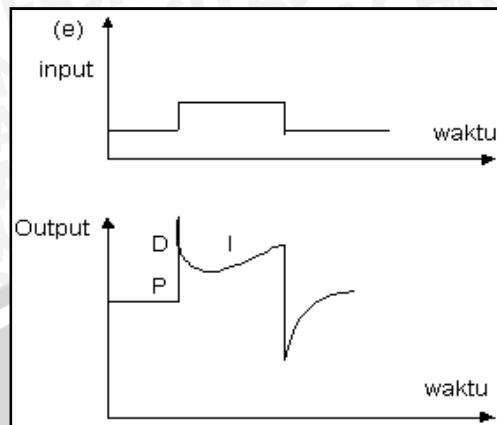
$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (7)$$

Jenis kontroler ini digunakan untuk memperbaiki kecepatan respon, mencegah terjadinya kesalahan keadaan mantap serta mempertahankan kestabilan.



**Gambar 2.17.** Diagram blok kontroler PID

Sumber: Philip, 1996: 220



**Gambar 2.18.** Hubungan dalam fungsi waktu antara sinyal keluaran dengan masukan untuk kontroler PID

Sumber: Gunterus, 1994: 8-11

Keluaran controller PID merupakan jumlahan dari keluaran kontroler proporsional, keluaran kontroler integral. Gambar 2.18 menunjukkan hubungan tersebut. Karakteristik kontroler PID sangat dipengaruhi oleh kontribusi besar dari ketiga parameter P, I dan D. Penyetelan konstanta  $K_p$ ,  $T_i$ , dan  $T_d$  akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat disetel lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan (Gunterus, 1994, 8-10).

## BAB III METODOLOGI

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif yang diwujudkan dalam bentuk *prototype*, yaitu perencanaan dan perealisasiian alat agar dapat menampilkan unjuk kerja sesuai dengan yang direncanakan dengan mengacu pada rumusan masalah.

Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat secara umum adalah sebagai berikut:

- Studi literatur
- Perancangan alat
- Realisasi pembuatan alat
- Pengujian alat dan analisa data
- Pengambilan kesimpulan

### 3.1 Studi Literatur

Studi literatur mengacu pada prinsip kerja alat yang dirancang meliputi sistem Pengendalian Kecepatan Motor *Servo* untuk Pemotongan *Styrofoam* Secara *Radial* dengan *Hotwire* Berbasis Mikrokontroler, karakteristik dari komponen yang digunakan serta studi tentang pengendalinya yaitu mikrokontroler atmega328.

Adapun hal-hal yang berhubungan dengan hal tersebut adalah :

- a. Studi tentang teori dasar pemotongan *styrofoam*
  - Temperatur kawat pemotong
  - Jenis *styrofoam*
  - Kecepatan pemotongan
- b. Studi tentang rangkaian sensor
  - Karakteristik spesifikasi sensor kecepatan *incremental encoder*.
  - Penerapan sensor kecepatan sesuai sistem.
  - Karakteristik komponen elektronik yang digunakan diantaranya meliputi driver motor DC, relay, LCD M1632, , Mikrokontroler atmega328, IC IRF640 sebagai saklar elektrik dan motor DC.
- c. Mikrokontroller Atmega328



- Karakteristik umum mikrokontroler atmega328.
- Cara memprogram mikrokontroler atmega328 sebagai pusat pengontrol kecepatan putaran motor DC

### 3.2 Perancangan Alat

Perancangan sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem beserta rangkaian elektronik pendukungnya, hal ini dimaksudkan agar sistem pengendalian pemotongan *Styrofoam* secara radial menggunakan motor *servo* tersebut dapat berjalan sesuai dengan deskripsi awal yang telah direncanakan. Perancangan sistem yang dilakukan meliputi:

- Penentuan dimensi dan bahan yang digunakan dalam pembuatan alat pemotong *Styrofoam* secara *radial*.
- Penentuan rangkaian elektronik yang digunakan.
- Perancangan program pada mikrokontroler atmega328 sebagai pusat pengontrol.
- Penentuan spesifikasi sistem yang akan dibuat, meliputi :
  - a. Penentuan deskripsi kerja sistem secara keseluruhan.
  - b. Mikrokontroler atmega328 dan rangkaian elektronik pendukung

### 3.3 Realisasi Pembuatan Alat

#### 3.3.1 Perancangan Perangkat Keras dan Realisasi Tiap Blok

- a. Pembuatan blok diagram lengkap sistem
- b. Penentuan dan perhitungan komponen yang akan digunakan
- c. Merakit perangkat keras masing-masing blok

#### 3.3.2 Perancangan dan Penyusunan Perangkat Lunak

Setelah mengetahui perangkat keras yang dirancang, maka dibutuhkan perangkat lunak untuk mengendalikan dan mengatur kerja dari alat ini. Parameter yang diperoleh dari hasil perhitungan kemudian diterapkan kedalam mikrokontroler Atmega 328 dengan menggunakan bahasa C.

### 3.4 Pengujian Alat

Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka dilakukan pengujian alat meliputi perangkat keras (*hardware*) yang dilakukan baik per blok rangkaian maupun keseluruhan sistem.

#### 3.4.1 Pengujian Perangkat Keras

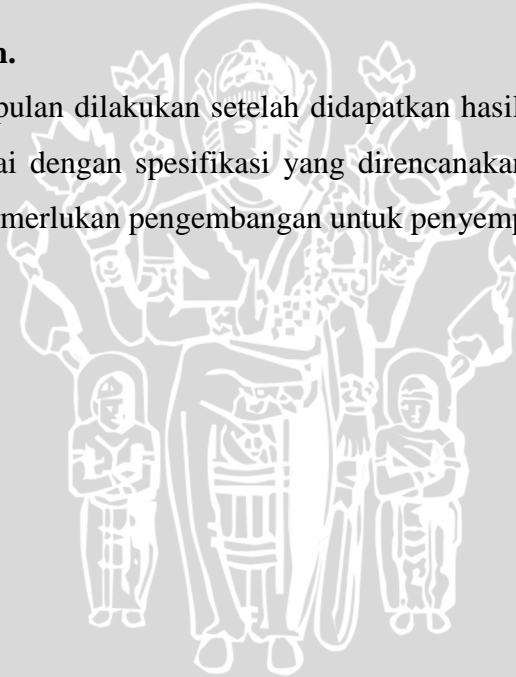
Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus yang diijinkan bekerja dalam komponen berdasarkan data sekunder komponen yang diambil dari buku data komponen elektronika maupun dari *datasheet*.

#### 3.4.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

### 3.5 Pengambilan Kesimpulan.

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.



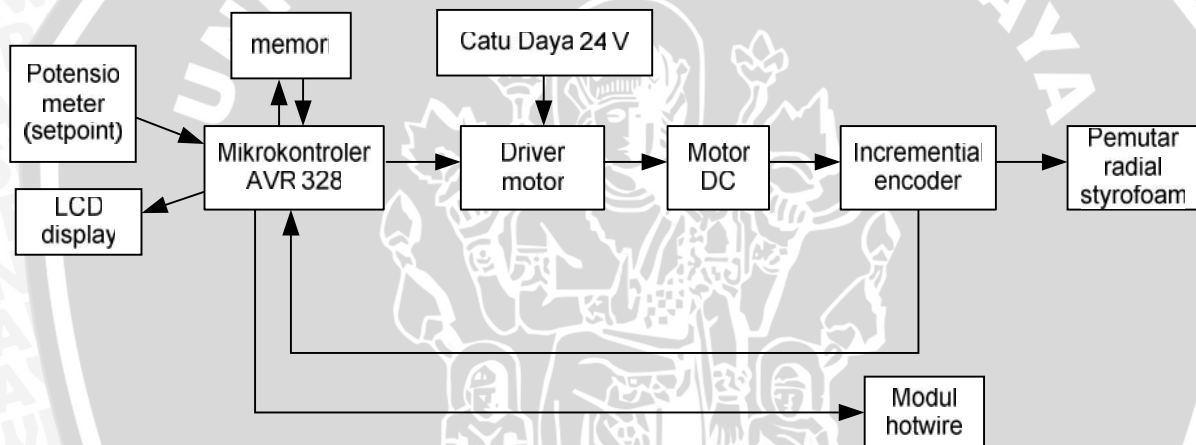
## BAB IV

### PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas tentang perencanaan dan pembuatan prototipe alat untuk pemotongan *Styrofoam* secara radial dengan *hotwire* menggunakan mikrokontroler AVR sebagai pengendali utama yang meliputi perencanaan sistem, penentuan spesifikasi alat, perencanaan masing-masing blok rangkaian serta perencanaan sistem secara keseluruhan.

#### 4.1. Perancangan sistem

Perancangan sistem perangkat keras ini digambarkan secara diagram blok seperti yang diperlihatkan pada gambar 4.1 berikut :



**Gambar 4.1** Diagram Blok Sistem

**Sumber:** Perancangan

Gambar diatas secara keseluruhan terdiri dari bagian-bagian sebagai berikut :

1. **Mikrokontroler AVR ATMEGA 328** sebagai kontroler kecepatan pemotongan *fuselage styrofoam* secara digital terprogram.
2. **Driver Motor DC** sebagai kendali untuk menyediakan variasi tegangan dari motor DC yang akan dikontrol oleh mikrokontroler.
3. **Motor DC** sebagai penggerak mekanik dudukan pemutar radial *styrofoam* yang akan dipotong.



4. **Incremental encoder** sebagai sensor kecepatan radial dari mekanikudukan pemutar radial *styrofoam*.
5. **Mekanik Dudukan Pemutar radial styrofoam** sebagai tempat menahan *styrofoam* yang akan dipotong secara radial.

#### 4.1.1. Spesifikasi alat

Penentuan spesifikasi alat yang akan dirancang adalah sebagai berikut :

1. **Mikrokontroler**

Mikrokontroler yang digunakan adalah jenis AVR atmega 328 sebagai pengontrol utama.

2. **Sensor**

Sensor kecepatan berupa sensor *incremental encoder* berfungsi untuk jumlah pulsa yang dihasilkan dalam periode tertentu.

3. **Periode sampling**

Periode penghitungan jumlah pulsa adalah 200 milidetik. Periode ini merupakan periode sampling.

4. **Driver Motor**

Untuk mengatur tegangan motor DC digunakan metode PWM dengan menggunakan MOSFET IRF640.

5. **Motor DC**

Motor DC yang digunakan adalah motor DC 24 Volt, 3500 rpm

6. **Modul hotwire**

Modul hot wire adalah hasil tugas akhir I Gede D. ( 0610630050).

Hubungan dengan sistem disini adalah hanya untuk mengaktifkan atau menonaktifkan modul tersebut.

7. **Dudukan Styrofoam**

Dudukan *styrofoam* berbentuk baut yang memiliki tempat menahan *styrofoam* sehingga mampu diputar. Jarak antara poros dudukan *styrofoam* dengan *hotwire* adalah 10 cm. Dengan kata lain, diameter maksimal *fuselage* yang mampu dibentuk oleh alat ini adalah 20cm.

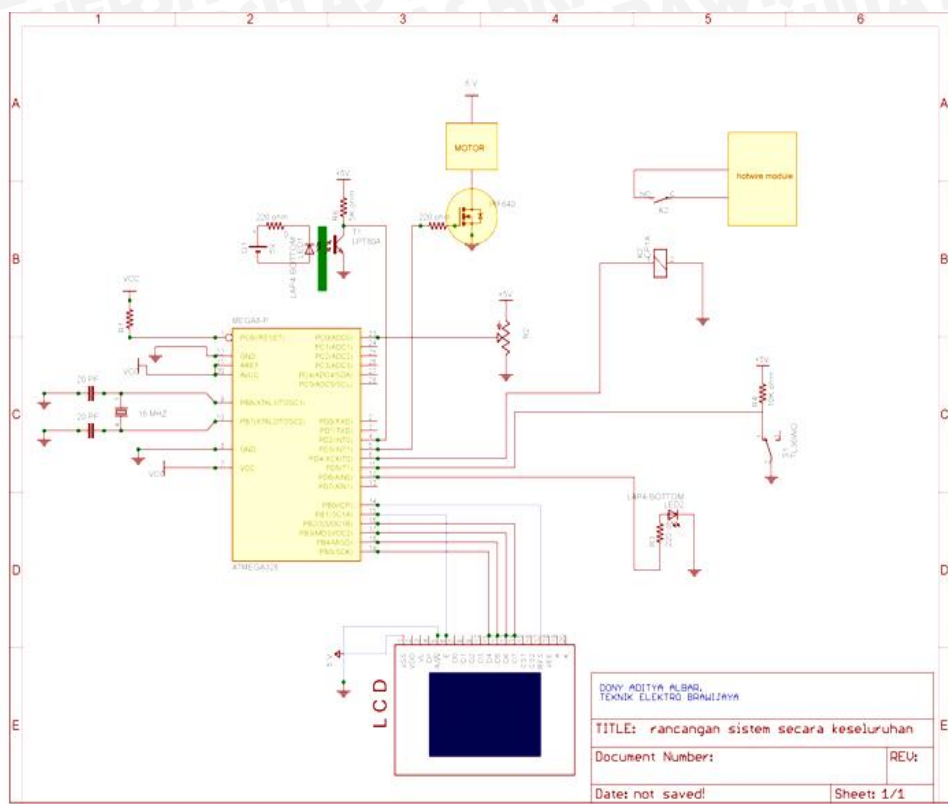
#### 4.1.2. Prinsip Kerja alat

Alat yang akan dikerjakan memiliki kerja sebagai berikut :

1. Pertama user memberikan masukan setpoint kecepatan yang diinginkan dengan memutar potensiometer .
2. Data masukan setpoint tersebut digunakan oleh mikrokontroler untuk mengatur kecepatan motor DC sebagai setpoint untuk memperoleh *error*.
3. Mikrokontroler mengaktifkan modul hot wire dan piringan pemutar radial *styrofoam*.
4. Kecepatan piringan *styrofoam* akan dihitung oleh mikrokontroler berdasarkan lebar pulsa yang dihasilkan oleh sensor incremental encoder.
5. Mikrokontroler menghitung besarnya *error* berdasarkan perbedaan antara besarnya setpoint dengan besarnya kecepatan sesuai hasil pada poin 4.
6. Berdasarkan algoritma kontroler PID, keluaran dari PID dihitung berdasarkan besarnya *error* yang telah dihitung dalam poin 5. Selanjutnya hasil perhitungannya dikeluarkan ke PWM mikrokontroler untuk mengatur besarnya tegangan motor DC.
7. Pada saat sistem bekerja, operator bisa mengaktifkan atau menonaktifkan kontroler sesuai dengan kondisi yang diinginkan. Pada saat kontroler aktif sistem bekerja dan modul *hotwire* diaktifkan. Sebaliknya, pada saat kontroler nonaktif modul dinonaktifkan.

#### 4.2 Perancangan perangkat keras

Berdasar pada blok diagram perancangan alat yang telah disusun, perancangan perangkat keras meliputi perancangan rangkaian sensor kecepatan (*incremental encoder*), rangkaian driver motor, rangkaian LCD, motor DC dan modul mikrokontroler. Di bawah ini adalah rangkaian keseluruhan alat.



**Gambar 4.2** skema perancangan sistem secara keseluruhan  
sumber : perancangan

**4.2.1. Mikrokontroler AVR ATMEGA 328**

Mikrokontroler berfungsi sebagai unit pengolah utama mengendalikan kecepatan motor DC. Selain itu mikrokontroler berfungsi untuk menampilkan data, status pengontrolan, dan membaca setpoint yang diinginkan.

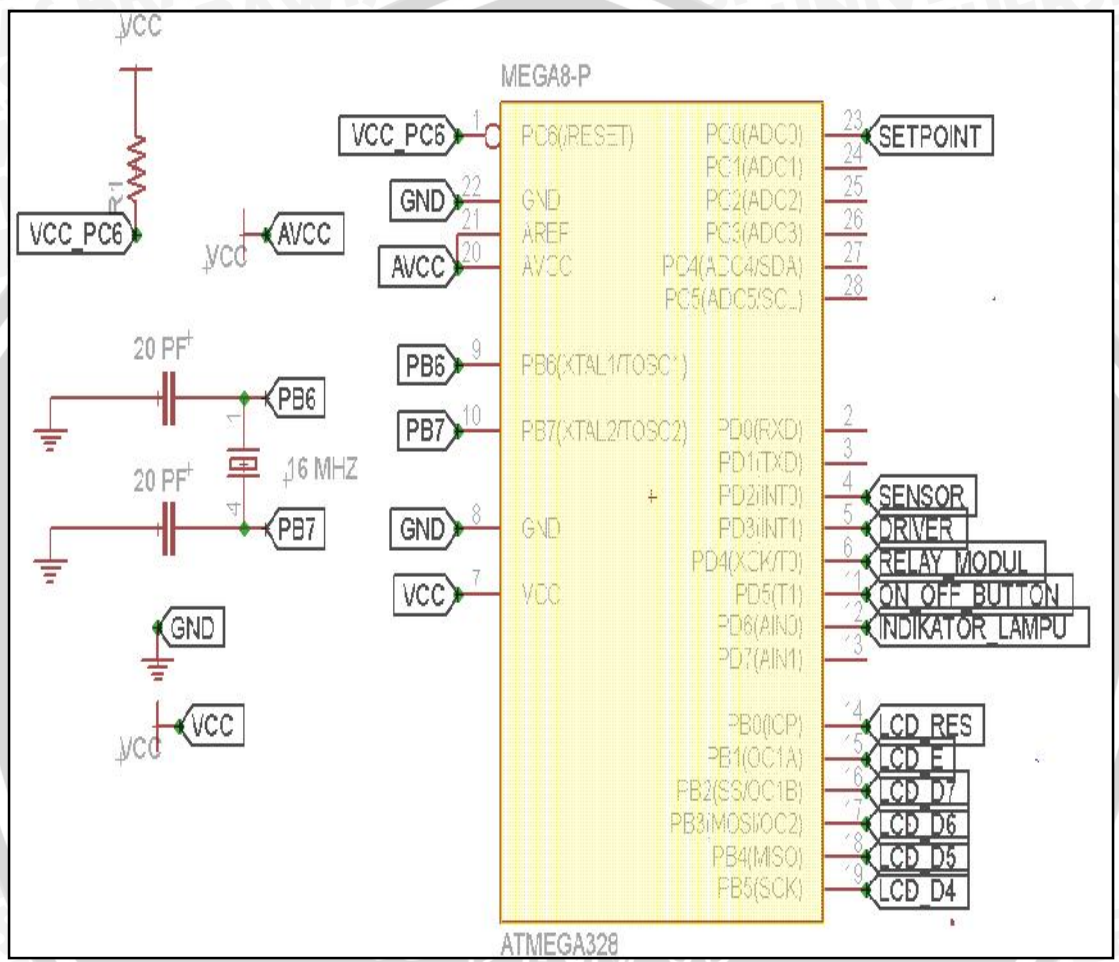
Mikrokontroler yang digunakan ialah AVR ATMEGA 328. Adapun spesifikasi yang dimiliki :

1. Memiliki kapasitas hingga 32K Bytes *In-System Self-Programmable Flash program memory*
2. Memiliki 1Kbytes EEPROM dan 2Kbytes *Internal SRAM*
3. Terdapat dua buah 8-bit *timer/counter*
4. Memilik 6 channel PWM (*Pulse Width Modulation*)
5. 23 saluran I/O yang dapat diprogram



6. Tegangan pengoperasian antara -1,8 V hingga 5,5 V.

Pada perancangan ini, menggunakan mikrokontroler ATmega328 sebagai pengatur aliran data dari ADC menuju LCD dan konverter. Dalam Gambar 4.8 ditunjukkan pin-pin mikrokontroler ATmega328 yang terhubung ke Serial Sensor arus, LCD M1632, tombol.



Gambar 4.3 Penyemat kaki Mikrokontroler atmega 328

Sumber : perancangan

Agar sebuah mikrokontroler dapat bekerja sebagai pengontrol, kaki-kaki/port mikrokontroler dihubungkan dalam rangkaian-rangkaian eksternal. Dalam perancangan ini, port yang digunakan adalah sebagai berikut :

1. Port 0

Port 0 merupakan port yang terletak pada pin 32-39, dalam perancangannya P0.0-0.7 digunakan sebagai inputan LCD M1632.

## 2. Port 1

Port 1 disediakan sebagai port I/O dan menempati pin 1-8, dalam perancangan ini P1.0-1.7 digunakan sebagai saluran keypad.

## 3. Port 2

Port 2 disediakan sebagai port I/O dan menempati pin 21-28, dalam perancangannya memiliki fungsi sebagai berikut :

- P2.0-P2.1 sebagai inputan ke RTC DS1307 dan ADC PCF8591
- P2.2-P2.5 sebagai inputan driver motor
- 2.6 digunakan sebagai control RS dan E

## 4. Port 3

Port 3 menempati pin 10-17, dalam perancangannya memiliki fungsi sebagai output untuk driver relay, driver motor dan jendela.

## 5. XTAL1 dan XTAL2

Digunakan sebagai input dari rangkaian osilator kristal. Rangkaian osilator kristal terdiri dari kristal osilator 11.0592 MHz, kapasitor C1 dan C2 yang masing-masing bernilai 33 pF, akan membangkitkan pulsa *clock* yang menjadi penggerak bagi seluruh operasi internal mikrokontroler.

## 6. VCC

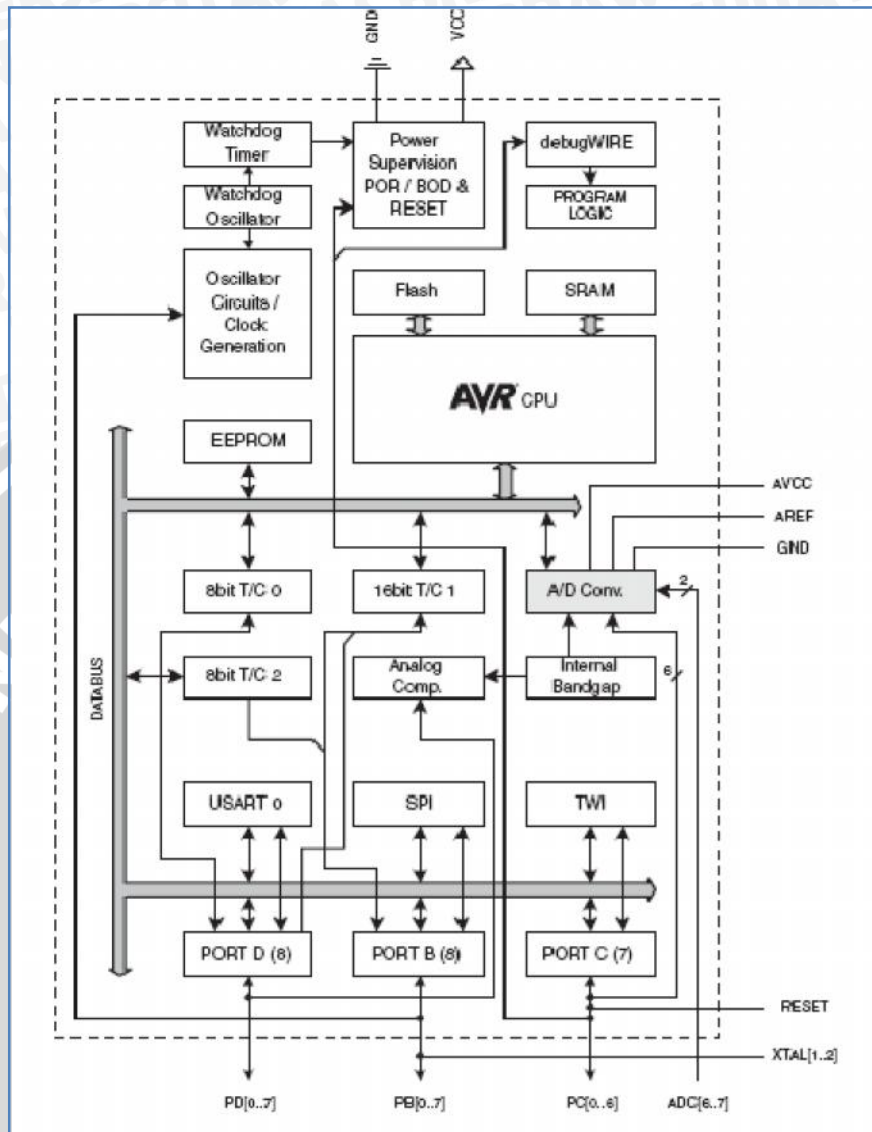
VCC dihubungkan dengan tegangan sebesar +5V sesuai dengan tegangan operasi *chip* tunggal yang diijinkan dalam *datasheet*.

## 7. GND

GND dihubungkan ke *ground* catu daya.

## 8. Reset

Digunakan untuk mereset program kontrol mikrokontroler, maka pin reset diberi logika tinggi selama sekurang-kurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset, kapasitor dihubungkan dengan VCC dan sebuah resistor yang dihubungkan ke *ground*



**Gambar 4.4** blok diagram mikrokontroler atmega 328

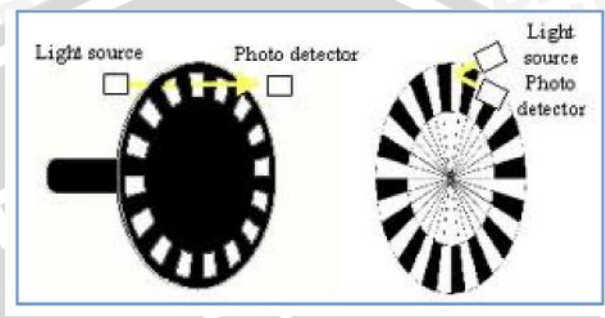
Sumber : datasheet mikrokontroler atmega 328

#### 4.2.2 Sensor Incremental Encoder

Encoder adalah perangkat elektrik-mekanik yang mengubah perpindahan linier atau putar menjadi sinyal digital atau pulsa. Pada perancangan *incremental encoder* terdiri dari satu piringan yang memiliki lubang-lubang pada bagian tepinya, 1 buah LED dan 1 buah *phototransistor*. Metode yang digunakan dalam pembacaan oleh sensor *incremental encoder* ini adalah dengan cara menghitung banyaknya pulsa yg dihasilkan persatuan waktu dimana 40 pulsa yang dihasilkan = 1 putaran. Jika dalam 1 sekon terdapat 250 pulsa maka :

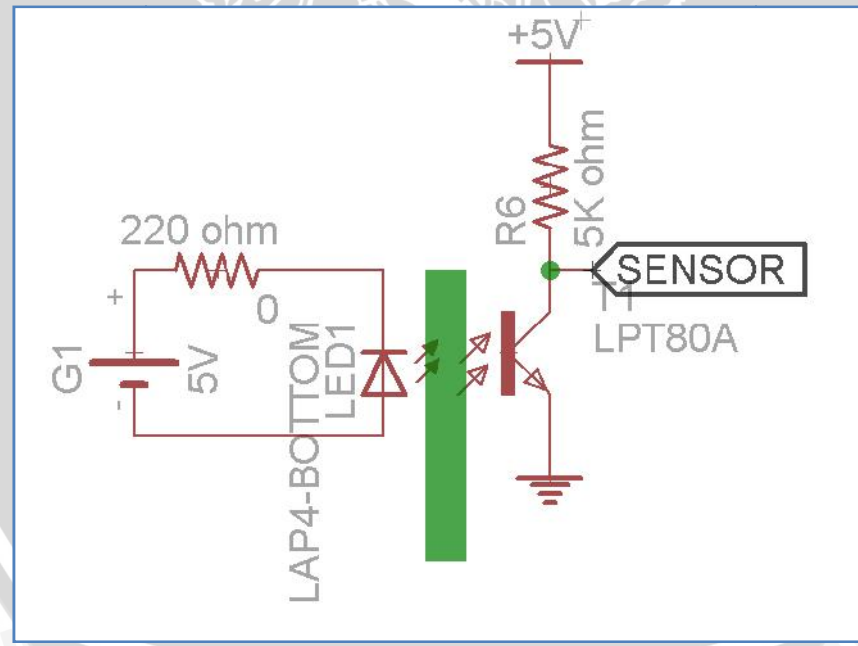


Maka,



Gambar 4.5 incremental encoder

Sumber : perancangan



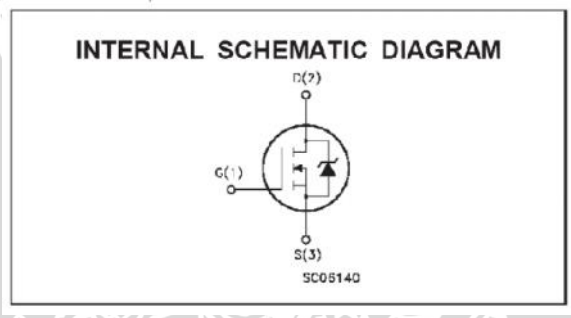
Gambar 4.6 rangkaian incremental encoder

sumber : perancangan

Incremental encoder ini selanjutnya berfungsi sebagai umpan balik bagi mikrokontroler berupa pulsa yang selanjutnya dihitung menjadi kecepatan putar motor.

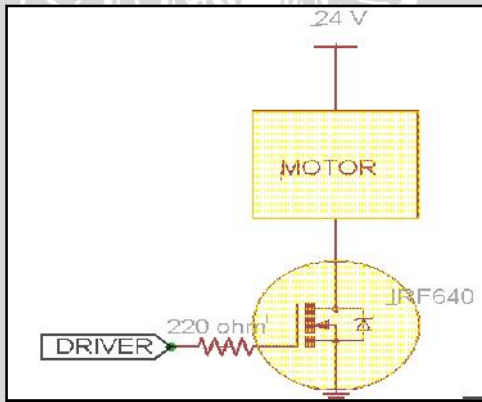
**4.2.3. Driver Motor DC**

Driver motor digunakan sebagai penghubung antar mikrokontroler ke motor DC. Digunakan driver motor karena arus yang keluar dari mikrokontroler tidak mampu mencukupi kebutuhan dari motor DC. Rangkaian driver motor dengan utama transistor, mosfet dan relay. Relay digunakan untuk membalik polaritas motor yang diaktifkan oleh transistor. Sedangkan mosfet digunakan untuk mengatur kecepatan motor DC. Nilai yang diberikan pada input motor digunakan untuk menentukan arah putaran motor. Nilai yang diberikan pada input PWM digunakan untuk mengatur kecepatan motor. Rangkaian driver motor DC dalam Gambar 10.



**Gambar 4.7** Rangkaian IC IRF640  
 sumber : datasheet IRF640

Berikut gambar rangkaian driver motor DC menggunakan IC IRF 640.



**Gambar 4.8** rangkaian driver motor DC  
 Sumber : Perancangan

#### 4.2.5 Motor DC

Sebagai penggerakudukan *styrofoam* menggunakan motor DC buatan nisca corp. tipe NA4056A dengan spesifikasi yang diperlihatkan pada table 4.1 sebagai berikut :

Voltage (V)	At Maximum Efficiency					Continuo us Operatio n(at 50°C)	No Load	Starting		
	Torqu e [mN · m]	Speed [rpm]	Curre nt [A]	Outp ut [W]	Efficien cy [%]	Possible Max Output [W]	Spee d [rpm]	Curre nt [A]	Torque [mN · m]	Curre nt [A]
24	32.86	3735	0.783	12.8 5	68.4	21.7	4380	0,135	223.02	4.529

**Tabel 4.1** Spesifikasi motor DC NA4056A

sumber : <http://www.nisca.co.jp/english/e-mos02.html>

Dalam perancangan, kecepatan potong *styrofoam* maksimal yang di rancang adalah 4 cm/s. Sedangkan diameter *fuselage* maksimal yg dipotong ditentukan sebesar 20 cm. Kecepatan putar maksimal motor ini adalah 3505 rpm atau dipakai 3500 rpm dengan demikian diperlukan penurunan putaran sehingga kecepatan linier pemotongan *styrofoam* menjadi 4cm/s.

Dalam memperhitungkan kecepatan putar, perlu diketahui dahulu keliling bidang potong yang dalam hal ini berbentuk lingkaran. Seperti yang telah dijelaskan pada sub bab sebelumnya, alat ini mampu memotong *Styrofoam* hingga didapatkan diameter maksimal dari *fuselage* ( $D_{st}$ ) sebesar 20 cm. Oleh karena itu, maka didapatkan panjang lintasan potong maksimal ( $K$ ) alat yaitu :

$$K = \pi D_{st}$$

$$K = 3,14 \times 20\text{cm}$$

$$K = 62,8\text{cm/putaran}$$



Kecepatan potong *Styrofoam* default ( $V_d$ ) yang diinginkan adalah 2 cm/s, maka lamanya waktu yang dibutuhkan untuk mencapai 1 putaran ( $t$ ) dan kecepatan pemotongan maksimal per detik ( $V_{Rps}$ ) yang diperlukan adalah :

$$t = \frac{K}{V_d}$$

$$t = \frac{62,8}{2}$$

$$t = 31,4 \text{ detik}$$

$$V_{Rps} = \frac{1}{31,4}$$

$$V_{Rps} = 0,031 \text{ rps}$$

$$V_{rpm} = V_{rps} \times 60 \text{ s}$$

$$= 0,031 \times 60$$

$$= 1,91 \text{ rpm}$$

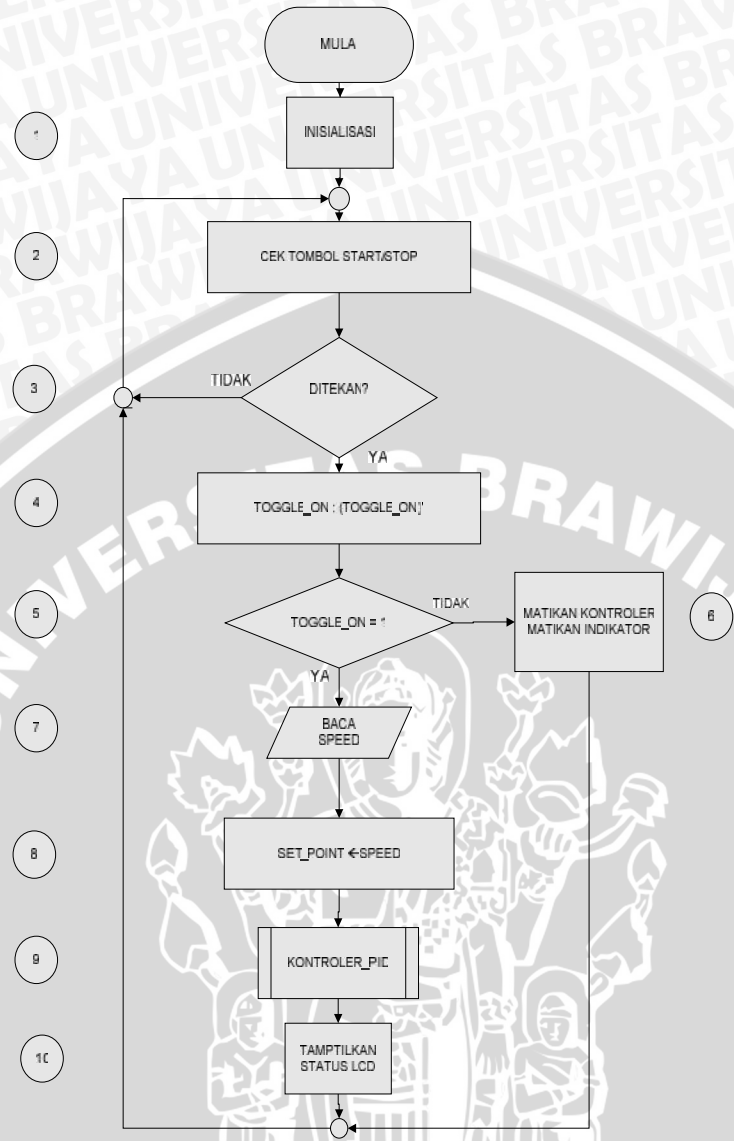
Pada perancangan motor DC dengan menggunakan gearbox rasio 1 : 200, untuk menghitung besarnya kecepatan motor DC secara default yaitu dengan mengalikan besarnya kecepatan potong *Styrofoam* dengan ratio gearbox sebesar 200.

$$\text{Kecepatan motor default} = V_{rpm} \cdot 200$$

$$= 382 \text{ rpm}$$

#### 4.3 Perancangan perangkat lunak

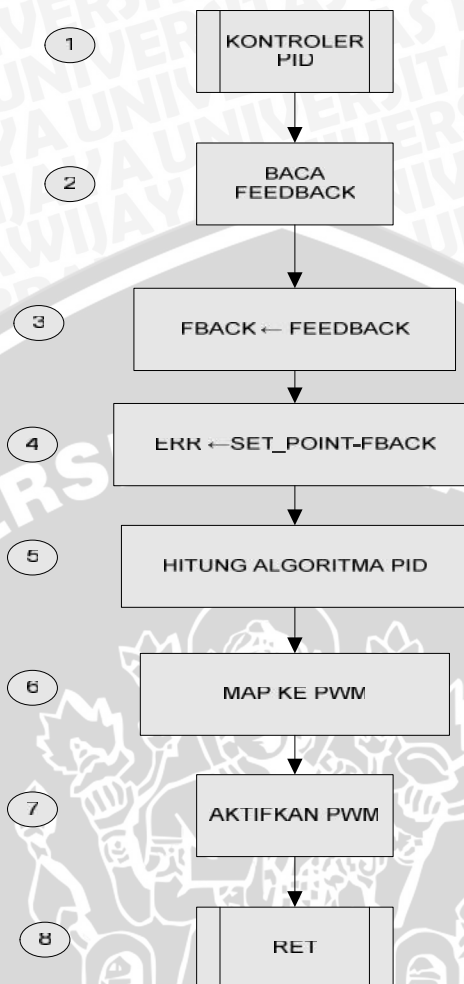
Perancangan perangkat lunak pada sistem pengendalian motor dc untuk pemotongan *Styrofoam* secara radial dengan *howire* berbasis mikrokontroler ini direncanakan dengan menggunakan pemrograman dengan bahasa C. Untuk mempermudah penjelasan isi program dituliskan dalam bentuk diagram alir yang terdiri dari satu program utama dan satu subrutin. Mengenai diagram alir tersebut diperlihatkan dalam gambar berikut :



Gambar 4.9 flowchart program keseluruhan

Sumber : perancangan

Berikut diagram alir program subrutin diperlihatkan pada gambar sebagai berikut :



**Gambar 4.10** flowchart subrutin kontroler PID

Sumber : perancangan

Adapun penjelasan dalam diagram alir dapat diuraikan sebagai berikut :

#### 4.3.1 Program utama

Program utama secara *flowchart* diperlihatkan dalam gambar 4.9 dengan uraian sebagai berikut :

1. Inisialisasi

Inisialisasi ini berfungsi sebagai untuk menginisialisasi LCD serta PWM kontroler dimatikan, set point diset pada default sebesar 382 rpm.

2. Cek tombol start/stop

Digunakan untuk mengecek adanya penekanan tombol yang berfungsi untuk menghidupkan atau mematikan motor.



3. Ditekan?  
Digunakan untuk menguji tombol ditekan apa tidak. Jika ditekan menuju ke nomor 4 jika tidak ditekan menuju ke nomor 2.
4. Toggle\_ON = (Toggle\_ON)'  
Merupakan flag bersifat toggle untuk menyimpan kondisi kontroler saat ini. Kondisi kontroler hidup jika toggle\_ON =1 sebaliknya jika toggle\_ON = 0 menunjukkan bahwa kondisi kontroler mati.
5. Toggle\_ON = 1 ?  
Digunakan untuk menguji kondisi flag toggle\_ON apakah berlogika 1, jika ya menuju ke nomor 7 dan jika tidak menuju ke nomor 6.
6. Matikan kontroler dan matikan Indikator  
Berfungsi untuk mematikan kontroler dan mematikan indicator.
7. Baca speed  
Berfungsi untuk membaca set point yang diperoleh dari pemutaran potensiometer, dimana tegangannya dikonversi oleh ADC mikrokontroler atmega328.
8. SET\_POINT SPEED  
Menyimpan hasil pembacaan set point ke variable set\_point
9. Kontroler\_PID  
Merupakan proses pelaksanaan algoritma kontroler PID dengan memanggil subrutin kontrol PID.
10. Tampilkan status LCD  
Berfungsi untuk menampilkan status LCD.

#### 4.3.2 Program subrutin kontroler PID

Program subrutin diperlihatkan dalam gambar 4.10 dengan uraian sebagai berikut :

1. Kontroler\_PID  
Merupakan algoritma kontroler secara PID
2. Baca feedback  
Berfungsi untuk membaca sensor dari *plant* berupa pulsa yg dihasilkan *incremental encoder* dalam periode 200ms
3. FBACK FEEDBACK  
Memindahkan isi variable feedback ke variable fback.

4. ERR SET\_POINT-FBACK  
Menghitung besarnya *error* yang diperoleh berdasarkan pengurangan nilai setpoint dengan fback.
5. HITUNG ALGORITMA PID  
Berfungsi untuk menghitung algoritma PID berdasarkan konstanta  $K_p$ ,  $K_d$ , dan  $K_i$
6. MAP KE PWM  
Memproporsionalkan nilai dari hasil perhitungan kontroler ke PWM 8bit internal mikrokontroler.
7. AKTIFKAN PWM  
Mengaktifkan PWM
8. RET  
Kembali ke pemanggil subrutin.

UNIVERSITAS BRAWIJAYA



## BAB V

### PENGUJIAN DAN ANALISIS SISTEM

Tujuan pengujian sistem ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perancangan. Pengujian pada sistem ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk menemukan letak kesalahan dan mempermudah analisis pada sistem apabila alat tidak bekerja sesuai dengan perancangan. Pengujian pada sistem ini dibagi menjadi tujuh bagian, yaitu:

- Pengujian rangkaian sensor kecepatan (*incremental encoder*)
- Pengujian mikrokontroler atmega 328
- Pengujian rangkaian driver motor DC
- Pengujian LCD
- Pengujian sistem secara keseluruhan

#### 5.1 Pengujian rangkaian sensor kecepatan (*incremental encoder*)

##### 5.1.1 Tujuan Pengujian

Pengujian rangkaian sensor kecepatan (*incremental encoder*) bertujuan untuk mengetahui rangkaian berfungsi dengan baik dan mampu memberikan *feedback* yang dibutuhkan mikrokontroler

##### 5.1.2 Peralatan Pengujian

Peralatan yang dibutuhkan antara lain:

- Voltmeter
- catu daya +5 V
- rangkaian sensor kecepatan (*incremental encoder*)

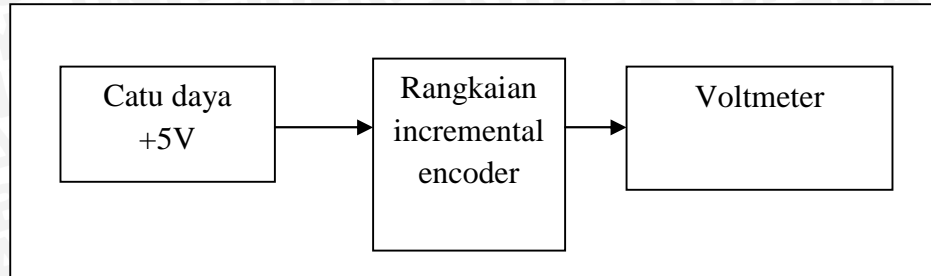
##### 5.1.3 Langkah Pengujian

Langkah-langkah pengujiannya adalah:

- merangkai peralatan seperti dalam gambar 5.1 keluaran rangkaian terhubung dengan voltmeter digital.
- mengaktifkan catu daya.



- mencatat tegangan yang tampak pada voltmeter digital saat kondisi incremental encoder gelap dan terang



**Gambar 5.1** : blok diagram pengujian sensor kecepatan

**5.1.4 Hasil Pengujian dan Analisis**

**Tabel 5.1.** hasil pengujian sensor *incremental encoder* secara teori

R Phototransistor	Kondisi	Vout
10	terang	0,0002
200k	gelap	4,999

Untuk mengetahui Vout maka dapat dihitung:

$$\begin{aligned}
 \text{Gelap} &= \frac{R_1}{R_1 + R_2} \times V \\
 &= \frac{200k}{200 + 0,01} \times 5 \\
 &= 4,999 \text{ Volt.}
 \end{aligned}$$

$$\begin{aligned}
 \text{Terang} &= \frac{R_2}{R_1 + R_2} \times V \\
 &= \frac{0,01}{200 + 0,01} \times 5 \\
 &= 0,0002 \text{ Volt.}
 \end{aligned}$$

**Tabel 5.2.** Hasil Pengujian sensor *incremental encoder*

Vout (teori)	Vout (praktek)	Error	Persentase (%)
4,999 V	4,994 V	0,005	0,1
0,0002 V	0,0007 V	0,0005	0

## 5.2 Pengujian Mikrokontroler atmega 328

### 5.2.1 Tujuan Pengujian

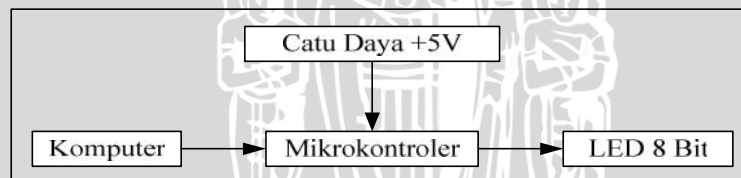
Pengujian mikrokontroler bertujuan untuk mengetahui kondisi awal dari sistem mikrokontroler agar sesuai dengan yang diharapkan.

### 5.2.2 Peralatan Pengujian

- komputer dengan kabel penghubungnya
- minimum sistem mikrokontroler atmega 328
- lampu LED mewakili keluaran 8 bit
- catu daya +5 volt

### 5.2.3 Langkah Pengujian

- merangkai peralatan seperti dalam gambar 5.2 keluaran terhubung dengan lampu LED yang mewakili keluaran 8 bit.
- mengisi mikrokontroler dengan program sederhana yang meletakkan 0FH dan F0H pada Accumulator secara bergantian kemudian memindahkannya ke port 1. kemudian download pada mikrokontroler atmega 328
- mengaktifkan catu daya.
- mereset program untuk mengeluarkan data mikrokontroler.
- mencatat data keluaran yang diwakili oleh lampu LED 8 bit ke dalam bentuk biner.



Gambar 5.2. Blok Diagram Pengujian Mikrokontroler

### 5.2.4 Hasil Pengujian dan Analisis

Tabel 5.3. Hasil Pengujian Sistem Mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
I	1	1	1	1	0	0	0	0
II	0	0	0	0	1	1	1	1

Dari tabel 5.3 terlihat bahwa port 1 memberikan logika OFH dan FOH secara bergantian sesuai dengan isi program.

### 5.3 Pengujian Rangkaian Motor DC NA4056A

#### 5.3.1 Tujuan Pengujian

Pengujian Motor DC NA4056A bertujuan untuk mengetahui apakah motor DC dan drivernya berfungsi dengan baik tanpa memotong *styrofoam*.

#### 5.3.2 Peralatan Pengujian

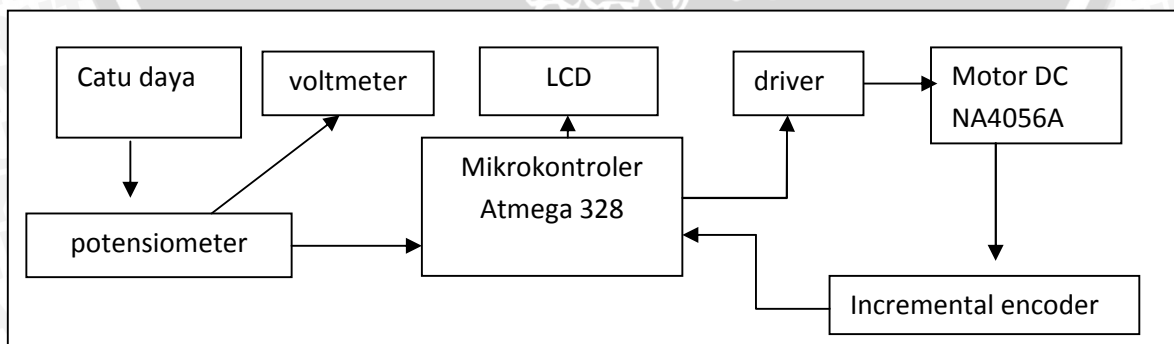
Peralatan yang dibutuhkan antara lain:

- Catu daya +5 V
- Rangkaian driver dan Motor DC NA4056A
- Voltmeter
- Incremental encoder
- LCD
- Mikrokontroler atmega 328
- Potensiometer

#### 5.3.3 Langkah Pengujian

Langkah-langkah pengujiannya adalah:

- merangkai peralatan seperti dalam gambar 5.3. Pada potensiometer diukur tegangan masukan (*setpoint*)
- mengaktifkan catu daya.
- Potensiometer diset dengan range tertentu untuk menghasilkan tegangan *setpoint*
- Mencatat hasil pembacaan LCD saat motor diberi tegangan masukan



**Gambar 5.3.** Blok Diagram Pengujian Motor DC NA4056A



5.3.4 Hasil Pengujian dan Analisis

Tabel 5.4. Hasil Pengujian Motor DC NA4056A secara teori

$V_{setpoint}$ (V)	Kecepatan ( Rpm)
0,5	350
1	700
1,5	1050
2	1400
2,5	1750
3	2100
3,5	2450
4	2800
5	3500

Perhitungan secara teori adalah sebagai berikut :

Dengan  $V_{setpoint} = 2,5$  V, maka kecepatan motor DC ( $v$ ) :

$$\frac{V_{setpoint}}{V_{maks}} = \frac{v}{v_{maks}}$$

$$v = \frac{V_{setpoint} \cdot v_{maks}}{V_{maks}}$$

$$V_{setpoint} = \frac{2,5 \cdot 3500}{5}$$

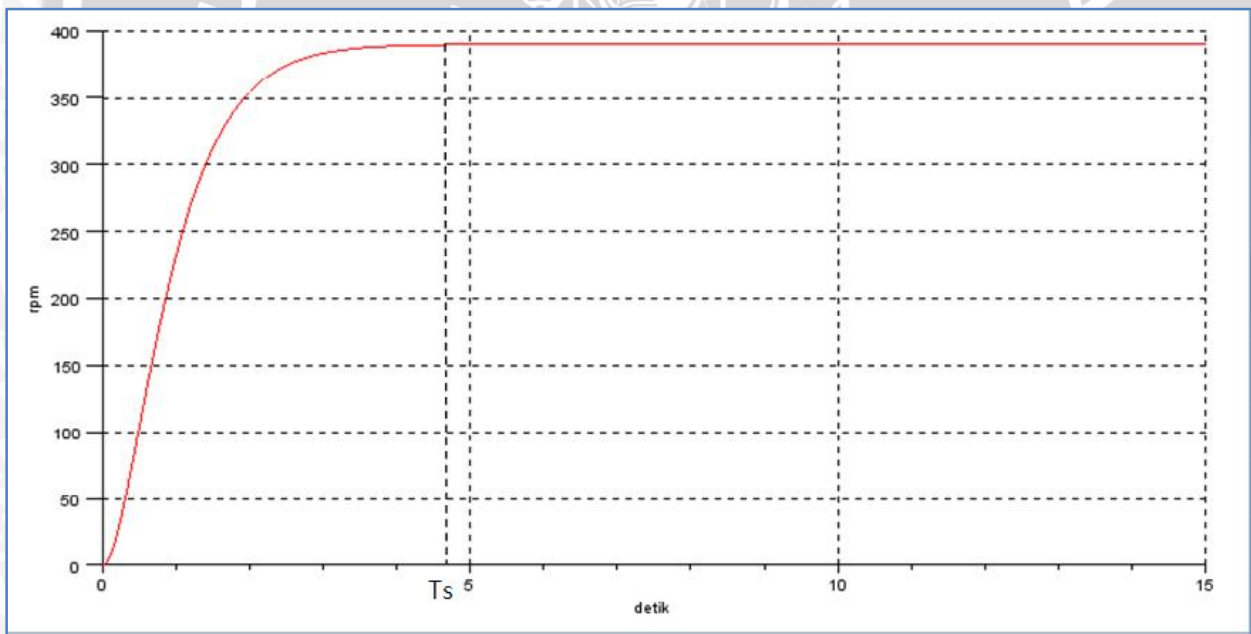
$$V_{setpoint} = 1750 \text{ Rpm}$$

Tabel 5.5 Hasil Pengujian Motor DC NA4056A secara praktek

$V_{setpoint}$ (V)	Kecepatan ( Rpm)				Error Rata-rata	% Error
	Teori	Praktek				
		Percobaan 1	Percobaan 2	Percobaan 3		
0.5	350	358	340	363	3,67	1,04

1	700	714	710	709	11	1,57
1,5	1050	1056	1065	1086	19	1,8
2	1400	1425	1421	1411	19	1,36
2,5	1750	1757	1768	1771	15,33	0,9
3	2100	2083	2076	2107	11,33	0,54
3,5	2450	2412	2498	2482	14	0,57
4	2800	2818	2770	2668	48	1,7
5	3500	3450	3405	3415	76,67	2,2
%Error rata-rata						1,3

Berikut adalah gambar respon dan performansi dengan tanpa beban styrofoam menggunakan *setpoint* yang diset sebesar 400 rpm :



**Gambar 5.4** grafik respon sistem tanpa beban dengan *setpoint* 400 rpm

Dimana performansi sistem dengan *setpoint* 400 rpm sebagai berikut:

Mp = tidak ada karena tidak terjadi overshoot

Ts = 4,75 s

Tp = tidak ada karena tidak terdapat overshoot

Td = tidak ada karena tidak terdapat overshoot

$$\text{Error Steady state} = \frac{7}{400} \times 100\% = 1,75\%$$

## 5.4 Pengujian LCD

### 5.4.1 Tujuan Pengujian :

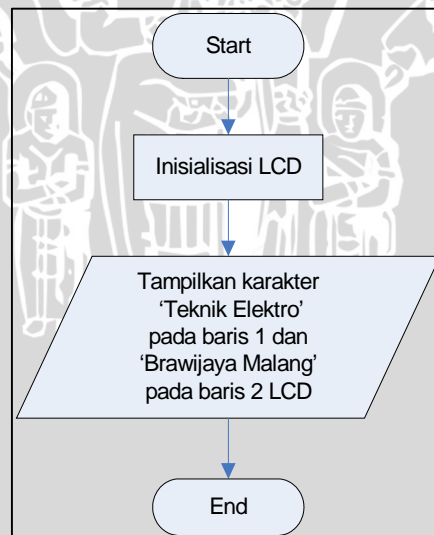
Pengujian LCD bertujuan untuk mengetahui LCD masih dapat berfungsi dengan baik sesuai dengan program yang diberikan oleh mikrokontroler.

### 5.4.2 Peralatan yang digunakan :

1. Mikrokontroller
2. LCD
3. Catu Daya 5 V

### 5.4.3 Prosedur Pengujian :

1. Membuat program pada mikrokontroler dengan menggunakan bahasa *assembly* untuk menampilkan data karakter jurusan “Teknik Elektro” pada baris pertama dan universitas “Brawijaya Malang” pada baris kedua di LCD. Diagram alir program pada mikrokontroler ditunjukkan dalam Gambar 5.5.

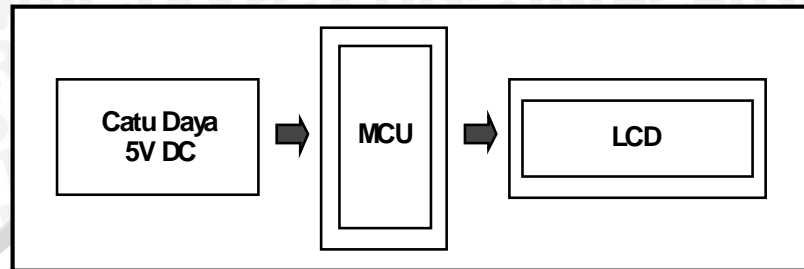


**Gambar 5.5.** Diagram alir program pada Mikrokontroler

2. Menyusun rangkaian pengujian mikrokontroler dan LCD seperti yang ditunjukkan dalam Gambar 5.6.
3. Memastikan rangkaian sudah tersusun dengan benar sebelum terhubung ke catu daya.



- 4 Menghubungkan catu daya ke rangkaian pengujian.
- 5 Melihat data karakter yang ditampilkan di LCD.



**Gambar 5.6.** Blok diagram pengujian Mikrokontroler dan LCD

#### 5.4.4 Hasil dan Analisis Pengujian :

Data karakter yang diprogram pada mikrokontroler dapat ditampilkan dengan baik pada LCD, dibuktikan dengan ditampilkannya karakter jurusan dan universitas seperti yang ditunjukkan dalam Gambar 5.7.



**Gambar 5.7** Gambar Tampilan LCD Pada Alat

### 5.5 Pengujian Sistem Secara Keseluruhan

#### 5.5.1 Tujuan Pengujian

Pengujian alat secara keseluruhan bertujuan untuk mengetahui apakah perangkat keras dan perangkat lunak berjalan dengan baik. Hal ini dimaksudkan untuk melihat apakah sistem pengendalian berjalan sesuai dengan perencanaan dan sebagai indikasi penelitian ini berhasil.

### 5.5.2 Peralatan Pengujian

Peralatan yang digunakan antara lain:

- alat yang telah selesai dibuat dan diprogram
- *styrofoam*
- Modul *hotwire*

### 5.5.3 Langkah Pengujian

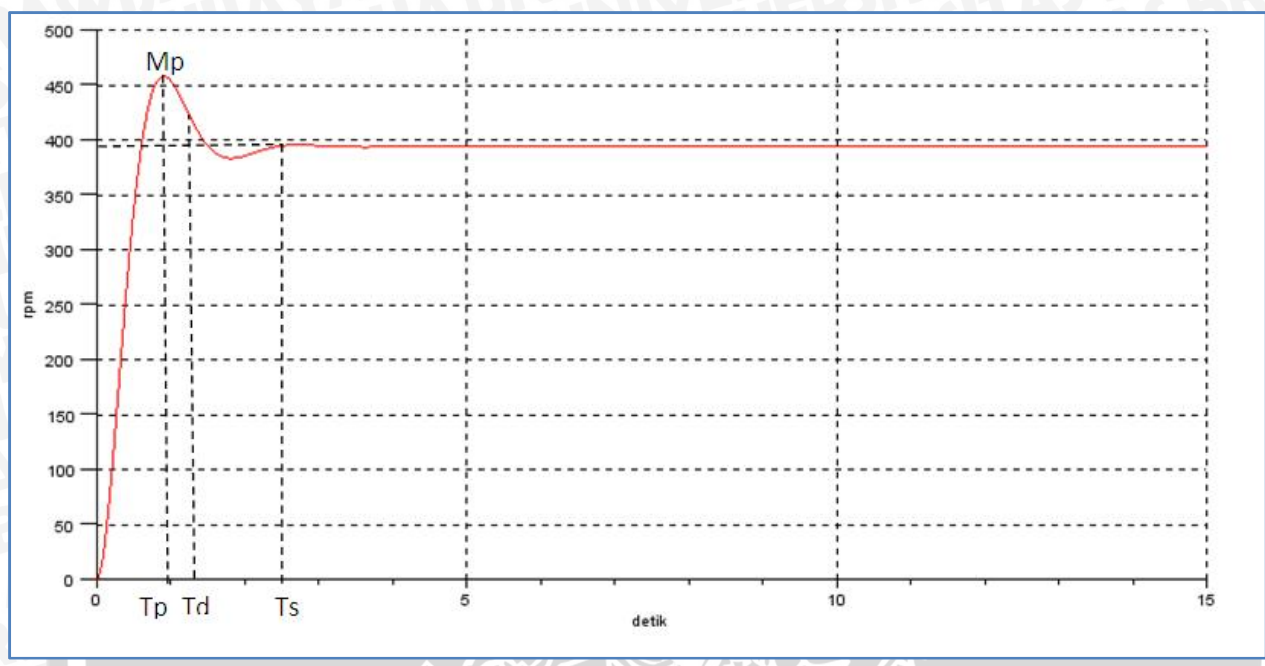
- Tombol start ditekan maka modul *hotwire* ON dan alat siap digunakan
- *Setpoint* diset pada nilai tertentu, kemudian kecepatan *steady state* motor DC yang terbaca di LCD dicatat. Hasil pencatatan dibandingkan dengan hasil perhitungan secara teori. Dimana kecepatan *steady state* motor DC yg ditentukan adalah 382 rpm untuk *setpoint* yang diset sebesar 15%.

### 5.5.4 Hasil pengujian dan Analisis

**Tabel 5.6 Hasil pengujian sistem secara keseluruhan**

<i>Setpoint</i> (%)	Kecepatan steady state motor DC (Rpm)				Error rata-rata	%Error
	teori	praktek				
		Percobaan 1	Percobaan 2	Percobaan 3		
50	875	845	859	863	19	2.17
60	1050	1036	1011	1023	26,67	2,54
70	1225	1180	1208	1206	27	2,2
80	1400	1367	1370	1386	25,67	1.83
90	1575	1557	1589	1585	26	1.65
100	1750	1713	1759	1760	30	1.71
% Error Rata-rata						2,1

Berikut adalah gambar respon dan performansi dengan beban styrofoam menggunakan *setpoint* yang diset sebesar 400 rpm :



**Gambar 5.8** grafik respon sistem berbeban menggunakan *setpoint* 400 rpm

Dengan performansi sistem dimana *setpoint* diset sebesar 400 rpm sebagai berikut :

$$Mp = \frac{55}{400} \times 100\% = 13,75\%$$

$$Ts = 2,5 \text{ s}$$

$$Tp = 1 \text{ s}$$

$$Td = 1,25 \text{ s}$$

$$\text{Error Steady state} = \frac{5}{400} \times 100\% = 1,25\%$$



## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan pada pengujian alat dan analisis pengendalian kecepatan motor *servo styrofoam* secara *radial* dengan *hotwire* berbasis mikrokontroler atmega 328 maka dapat diambil kesimpulan sebagai berikut:

1. Nilai *error* pengujian motor DC dengan *feedback incremental encoder* dan tanpa beban *styrofoam* yang terkecil adalah 0,54% dan nilai *error* yang terbesar adalah 2,2 % sehingga diperoleh kesalahan rata-rata sebesar 1,3%. Jadi *error* yang terjadi pada pengujian motor DC cukup kecil dikarenakan motor DC tidak berbeban karena tidak memutar *styrofoam*.
2. Nilai *error* pengujian sensor *incremental encoder* pada kondisi gelap adalah 0,1 % dan pada kondisi terang adalah 0 %.
3. Nilai *error* pengujian sistem kecepatan motor DC yang telah diberi *feedback* sensor *incremental encoder* yang terkecil adalah 1,65 % dan nilai *error* yang terbesar adalah 2,54% Sehingga diperoleh *error* rata-rata sebesar 2,1%. *Error* yang terjadi masih mampu membuat alat menghasilkan pemotongan dengan permukaan potong yang rata dan rapi. Kontroler sudah dapat bekerja pada sistem dengan baik

#### 6.2 Saran

Dalam perancangan dan pembuatan alat ini, masih terdapat kekurangan dan kelemahan, sehingga perlu adanya penyempurnaan dalam rangka pengembangan selanjutnya. Hal yang dapat disempurnakan, diantaranya:

1. Penyempurnaan dalam hal mekanik yaitu perlu ditambahkan sensor kemiringan pemotongan dengan tampilan derajat kemiringan pada LCD.
2. Untuk pemotongan *styrofoam* dengan dimensi yang lebih besar perlu dibutuhkan motor DC dengan torsi yang lebih besar serta penyempurnaan mekanik berupa pelebaran diameter pemotongan maksimal.
3. Menggunakan sensor kecepatan yang lebih baik seperti *speedometer* digital agar lebih praktis dan lebih tepat.

## DAFTAR PUSTAKA

- Coughlin, Robert F. 1982. *Penguat Operasional dan Rangkaian Terpadu Linear*, Jakarta: Erlangga.
- Elektuur.1985. *Data Sheet Book I*, Jakarta : Elex Media Komutindo.
- Malvino, Albert Paul. 1987. *Prinsip-prinsip Elektronika, Jilid 2, Cetakan ketiga*, terjemahan Prof. M. Barmawi, Ph.D, M.O Tjia, Ph.D. Jakarta : Erlangga.
- Ogata, Katsuhiko. 1997. *Teknik Kontrol Automatik*. Jakarta. Penerbit Erlangga.
- Tooley, Mike. 2002. *Rangkaian Elektronik Prinsip dan Aplikasi*. Jakarta. Penerbit Erlangga.
- Atmel. 2011, *8-bit Microcontroller with 32K Bytes Flash ATMEGA 328*.  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc8271.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc8271.pdf)
- Nisca corp. 2011 *DC Motor spec list nisca corp*. <http://www.nisca.co.jp/english/e-mos02.html>
- Seiko Instruments Inc, 1987, *Liquid Crystal Display Module M1632 User Manual*, Japan
- Mellen, Bob 2011. *Precision CNC foam cores*. <http://www.flyingfoam.com>
- Dow.2012. *The Dow Chemical Company*, <http://www.dow.com/>

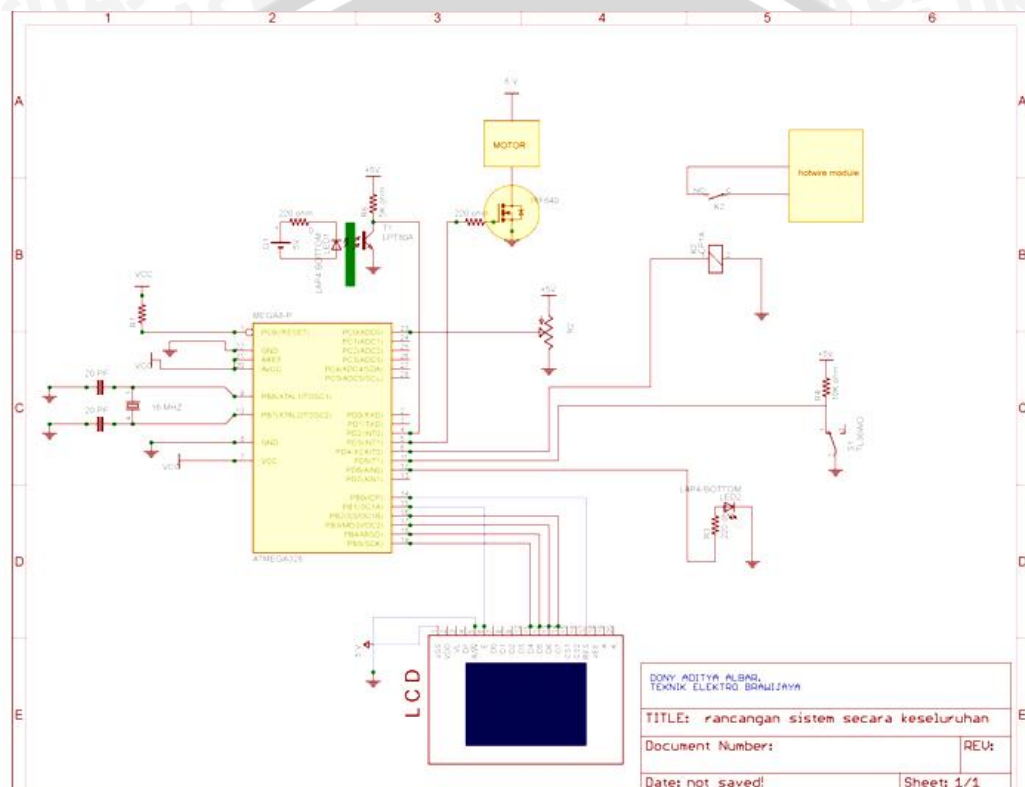




LAMPIRAN I

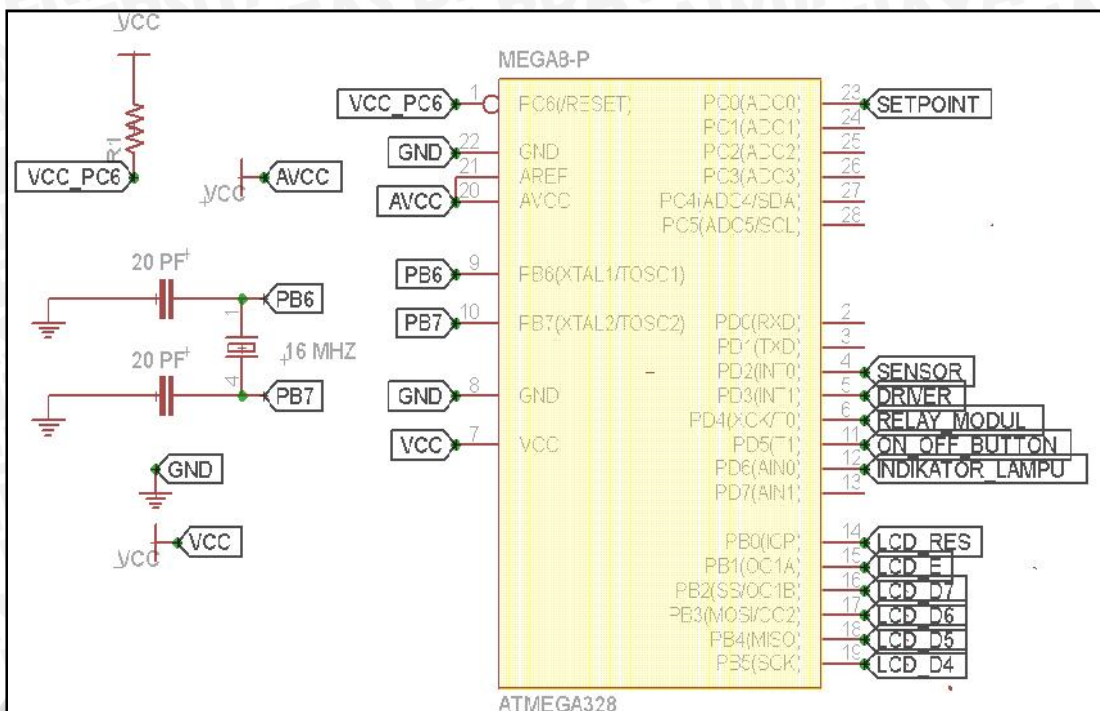
GAMBAR RANGKAIAN



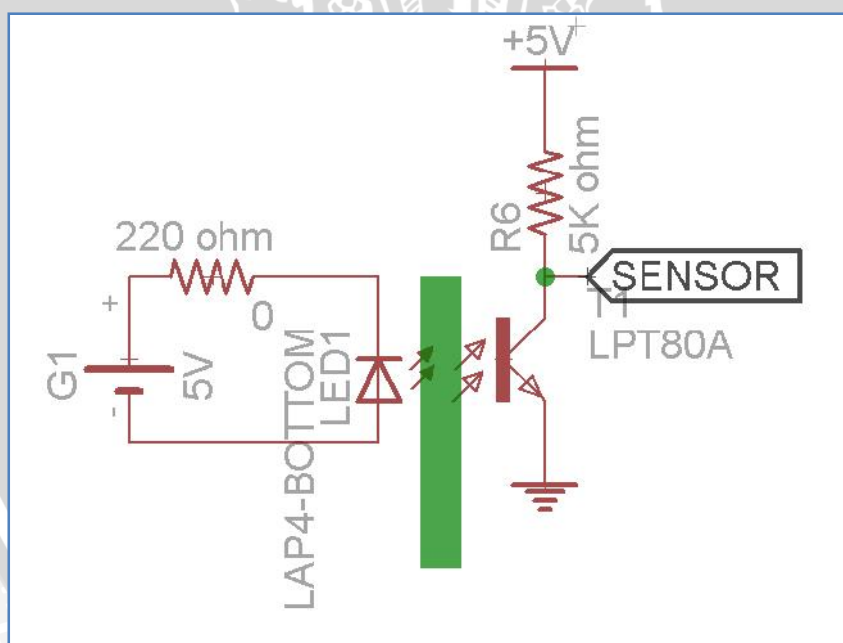


Rangkaian Sistem Mikrokontroler



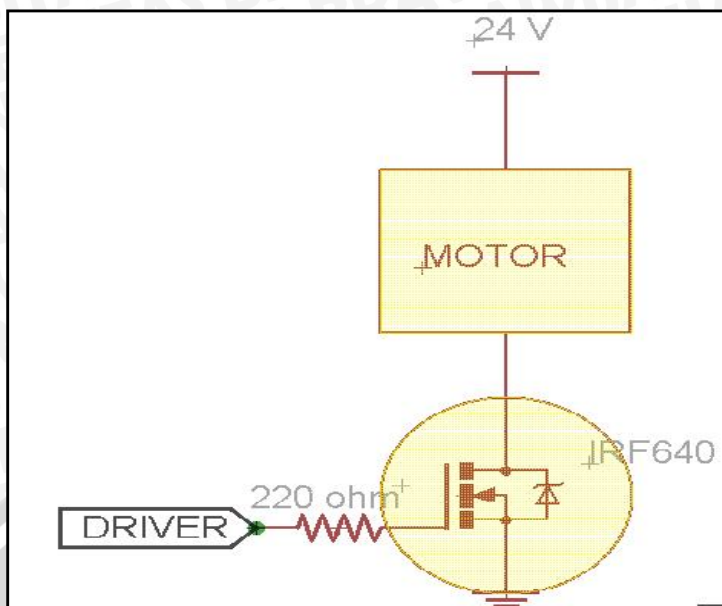


Penyemat kaki Mikrokontroler atmega 328

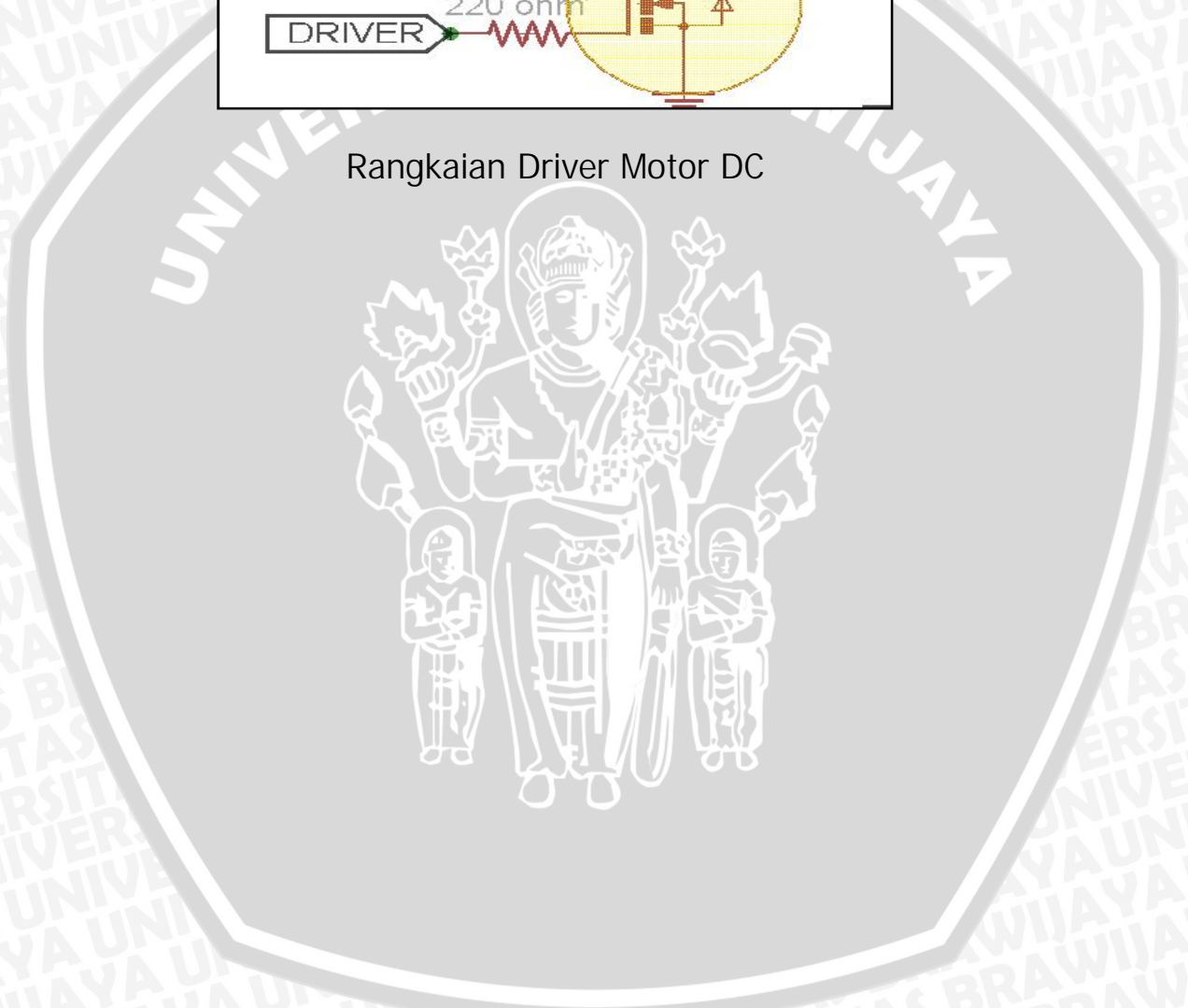


Rangkaian Sensor Incremental Encoder





Rangkaian Driver Motor DC



---

LISTING PROGRAM  
MIKROKONTROLER ATMEGA328



```

/*****/
/* SKRIPSI */
/* JUDUL: Pengendalian kecepatan motor servo untuk pemotongan */
/* styrofoam secara radial dengan hotwire berbasis */
/* mikrokontroler ATMEGA-328 */
/* NAMA : Dony Aditya Albar */
/* NIM : 0610630030 */
/*****/

```

```

#include <LiquidCrystal.h>
#include <PID_v1.h>
#include <FreqCounter.h>

```

```
LiquidCrystal lcd(7,8,9,10,11,12,13);
```

```

const int outputPWM = 6;
const int pinSetPoint = A1;
char strRPM[5];

```

```

// variabel kontroler
double Setpoint, Input, Output;

```

```
PID myPID(&Input, &Output, &Setpoint,25,6,0.5, DIRECT);
```

```

/*****/
/* initLCD() */
/* inialisasi lcd dan tampilan copyright */
/*****/
void initLCD()
{
// inialisasi lcd
lcd.begin(16,2);
lcd.clear();
lcd.print("Pemotong Radial ");
lcd.setCursor(0,1);
lcd.print("-- Styro Foam --");
delay(4000);

for (int positionCounter = 0; positionCounter < 17; positionCounter++) {
lcd.scrollDisplayLeft();
delay(150);
}
delay(300);
lcd.clear();
lcd.print("Dony Aditya A.");
lcd.setCursor(0,1);
lcd.print("NIM: 0610630030");
}
/*****/

```

```

/*****/
/* initStateDisp() */
/* inialisasi form tampilan status kontroler */
/*****/
void initStateDisp()
{
lcd.clear();
lcd.print("SPoint.");
lcd.setCursor(13,0);

```



```

lcd.print("RPM");
lcd.setCursor(0,1);
lcd.print("Output:");
lcd.setCursor(13,1);
lcd.print("RPM");
}
/*****/

```

```

void readRPM()
{
// membaca umpan balik RPM
FreqCounter::f_comp=10;
FreqCounter::start(250);
while (FreqCounter::f_ready == 0);
Input=FreqCounter::f_freq;
Input = (Input*4*60)/8;
}

```

```

void dispRPM()
{
unsigned long t;
t = millis()-dispCurrentTime;
if (t<250)
return;
else
dispCurrentTime = millis();

```

```

// indikator
lcd.setCursor(7,1);
if (flagRUN == false)
{
lcd.print("");
flagRUN = true;
}
else
{
lcd.print(" ");
flagRUN = false;
}

```

```

lcd.setCursor(8,1);
lcd.print(strRPM);
}

```

```

void dispsetpointRPM()
{
lcd.setCursor(10,0);
desiredSetPoint();
dtostrf(setpointCurrent,4,2,strCurrent);
lcd.print(strCurrent);
}

```

```

void setup()
{

```



```
analogWrite(pinPWM,0);  
initLCD();  
delay(2000);  
initStatusDisp();  
  
readRPM();  
Setpoint = analogRead(pinSetPoint);  
// hidupkan pid  
myPID.SetMode(AUTOMATIC);  
}
```

```
void loop()  
{  
  // baca feedback rpm  
  readRPM();  
  // display rpm  
  dispRPM();  
  dispsetpointRPM();  
  // hitung pid  
  myPID.Compute();  
  analogWrite(outputPWM,Output);  
}
```



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

\* PID Library

\* by Brett Beauregard <br3ttb@gmail.com> brettbeauregard.com

\*

\* This Code is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

\*\*\*\*\*/

#include <WProgram.h>

#include <PID\_v1.h>

/\*Constructor (...)\*\*\*\*\*\*

\* The parameters specified here are those for for which we can't set up

\* reliable defaults, so we need to have the user set them.

\*\*\*\*\*/

PID::PID(double\* Input, double\* Output, double\* Setpoint,  
double Kp, double Ki, double Kd, int ControllerDirection)

{

    PID::SetOutputLimits(0, 255); //default output limit  
corresponds to

    //the pwm limits

    SampleTime = 100; //default Controller  
    Sample Time is 0.1 seconds

    PID::SetControllerDirection(ControllerDirection);

    PID::SetTunings(Kp, Ki, Kd);

    lastTime = millis()-SampleTime;

    inAuto = false;

    myOutput = Output;

    myInput = Input;

    mySetpoint = Setpoint;



}

/\* Compute() \*\*\*\*\*/

- \* This, as they say, is where the magic happens. this function should be called
- \* every time "void loop()" executes. the function will decide for itself whether a new
- \* pid Output needs to be computed

\*\*\*\*\*/

void PID::Compute()

{

if(!inAuto) return;

unsigned long now = millis();

int timeChange = (now - lastTime);

if(timeChange>=SampleTime)

{

/\*Compute all the working error variables\*/

double input = \*myInput;

double error = \*mySetpoint - input;

ITerm+= (ki \* error);

if(ITerm > outMax) ITerm= outMax;

else if(ITerm < outMin) ITerm= outMin;

double dInput = (input - lastInput);

/\*Compute PID Output\*/

double output = kp \* error + ITerm- kd \* dInput;

if(output > outMax) output = outMax;

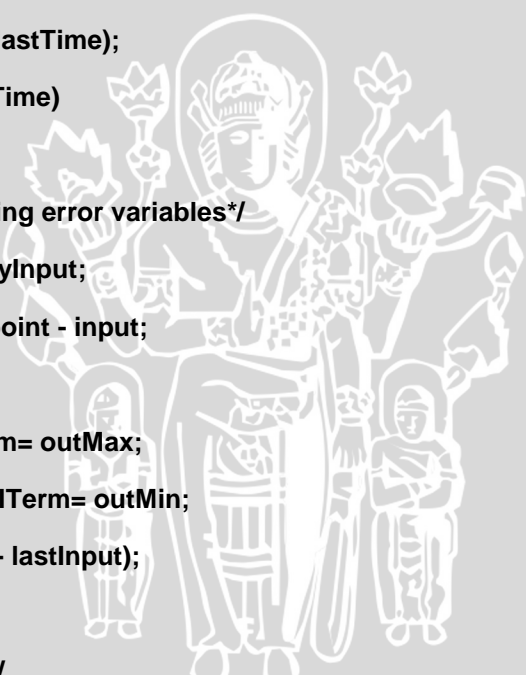
else if(output < outMin) output = outMin;

\*myOutput = output;

/\*Remember some variables for next time\*/

lastInput = input;

lastTime = now;



```

}
}

/* SetTunings(...)*****
* This function allows the controller's dynamic performance to be adjusted.
* it's called automatically from the constructor, but tunings can also
* be adjusted on the fly during normal operation
*****/

```

```

void PID::SetTunings(double Kp, double Ki, double Kd)
{
    if (Kp<0 || Ki<0 || Kd<0) return;

    dispKp = Kp; dispKi = Ki; dispKd = Kd;

    double SampleTimeInSec = ((double)SampleTime)/1000;
    kp = Kp;
    ki = Ki * SampleTimeInSec;
    kd = Kd / SampleTimeInSec;

    if(controllerDirection ==REVERSE)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
}

```

```

/* SetSampleTime(...) *****
* sets the period, in Milliseconds, at which the calculation is performed
*****/

void PID::SetSampleTime(int NewSampleTime)
{
    if (NewSampleTime > 0)

```



```

{
    double ratio = (double)NewSampleTime
        / (double)SampleTime;
    ki *= ratio;
    kd /= ratio;
    SampleTime = (unsigned long)NewSampleTime;
}
}

```

```

/* SetOutputLimits(...)*****

```

```

* This function will be used far more often than SetInputLimits. while
* the input to the controller will generally be in the 0-1023 range (which is
* the default already,) the output will be a little different. maybe they'll
* be doing a time window and will need 0-8000 or something. or maybe they'll
* want to clamp it from 0-125. who knows. at any rate, that can all be done
* here.

```

```

***** /

```

```

void PID::SetOutputLimits(double Min, double Max)

```

```

{
    if(Min >= Max) return;
    outMin = Min;
    outMax = Max;

    if(inAuto)
    {
        if(*myOutput > outMax) *myOutput = outMax;
        else if(*myOutput < outMin) *myOutput = outMin;

        if(ITerm > outMax) ITerm= outMax;
        else if(ITerm < outMin) ITerm= outMin;
    }
}

```



```

/* SetMode(...)*****
 * Allows the controller Mode to be set to manual (0) or Automatic (non-zero)
 * when the transition from manual to auto occurs, the controller is
 * automatically initialized
*****/

void PID::SetMode(int Mode)
{
    bool newAuto = (Mode == AUTOMATIC);
    if(newAuto == !inAuto)
    { /*we just went from manual to auto*/
        PID::Initialize();
    }
    inAuto = newAuto;
}

/* Initialize()*****
 * does all the things that need to happen to ensure a bumpless transfer
 * from manual to automatic mode.
*****/

void PID::Initialize()
{
    ITerm = *myOutput;
    lastInput = *myInput;
    if(ITerm > outMax) ITerm = outMax;
    else if(ITerm < outMin) ITerm = outMin;
}

/* SetControllerDirection(...)*****
 * The PID will either be connected to a DIRECT acting process (+Output leads
 * to +Input) or a REVERSE acting process(+Output leads to -Input.) we need to
 * know which one, because otherwise we may increase the output when we should
 * be decreasing. This is called from the constructor.
*****/

```

```

void PID::SetControllerDirection(int Direction)
{
    if(inAuto && Direction !=controllerDirection)
    {
        kp = (0 - kp);
        ki = (0 - ki);
        kd = (0 - kd);
    }
    controllerDirection = Direction;
}

```

```

/* Status Funcions*****

```

```

* Just because you set the Kp=-1 doesn't mean it actually happened. these
* functions query the internal state of the PID. they're here for display
* purposes. this are the functions the PID Front-end uses for example

```

```

*****/

```

```

double PID::GetKp(){ return dispKp; }

```

```

double PID::GetKi(){ return dispKi;}

```

```

double PID::GetKd(){ return dispKd;}

```

```

int PID::GetMode(){ return inAuto ? AUTOMATIC : MANUAL;}

```

```

int PID::GetDirection(){ return controllerDirection;}

```

LAMPIRAN III

FOTO ALAT





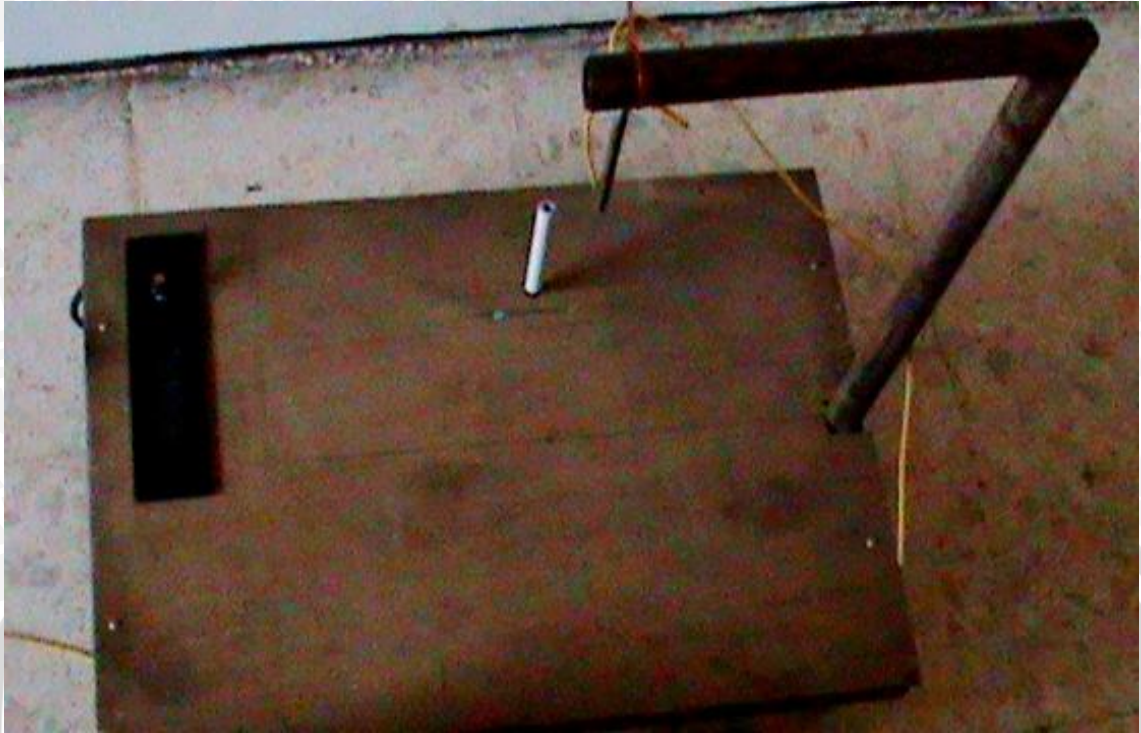
FOTO MOTOR DC SERVO DAN GEARBOX DENGAN SENSOR INCREMENTAL ENCODER



FOTO ALAT SECARA KESELURUHAN



FOTO ALAT TAMPAK ATAS





LAMPIRAN IV

DATASHEET





## Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32KBytes of In-System Self-Programmable Flash program memory
  - 256/512/512/1KBytes EEPROM
  - 512/1K/1K/2KBytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Atmel® QTouch® library support
  - Capacitive touch buttons, sliders and wheels
  - QTouch and QMatrix® acquisition
  - Up to 64 sense channels
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    - Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    - Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Byte-oriented 2-wire Serial Interface (Philips I<sup>2</sup>C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 4MHz@1.8 - 5.5V, 0 - 10MHz@2.7 - 5.5.V, 0 - 20MHz @ 4.5 - 5.5V
- Power Consumption at 1MHz, 1.8V, 25°C
  - Active Mode: 0.2mA
  - Power-down Mode: 0.1µA
  - Power-save Mode: 0.75µA (Including 32kHz RTC)



## 8-bit Atmel Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48A  
ATmega48PA  
ATmega88A  
ATmega88PA  
ATmega168A  
ATmega168PA  
ATmega328  
ATmega328P

## Summary

Rev. 8271DS-AVR-05/11



# ATmega48A/PA/88A/PA/168A/PA/328/P

## 1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

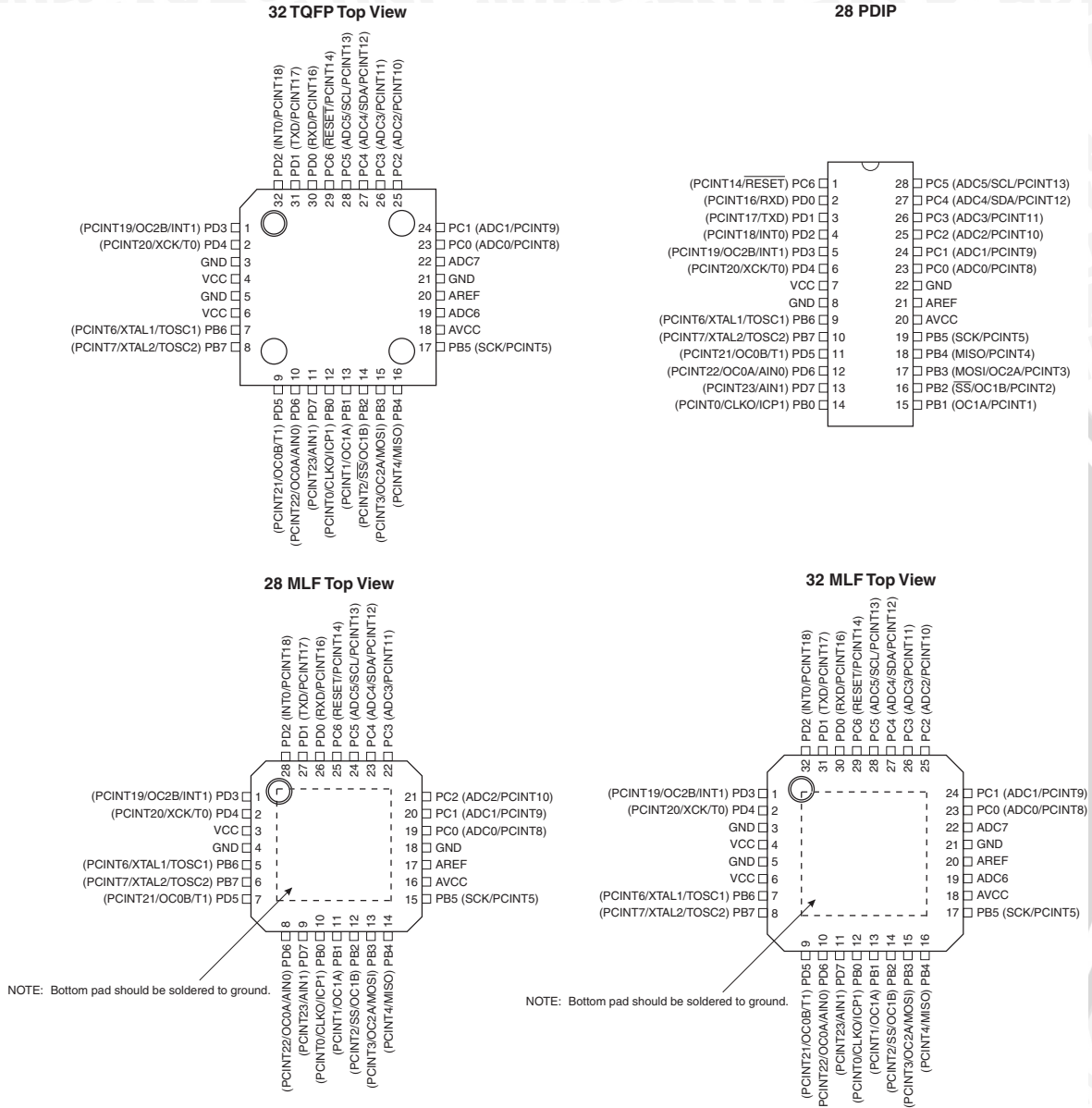


Table 1-1. 32UBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4



## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

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.

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.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

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.

The various special features of Port B are elaborated in ["Alternate Functions of Port B" on page 84](#) and ["System Clock and Clock Options" on page 27](#).

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5...0 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.

### 1.1.5 PC6/RESET

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.

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 29-12 on page 324](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 87](#).

### 1.1.6 Port D (PD7:0)

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.



# ATmega48A/PA/88A/PA/168A/PA/328/P

The various special features of Port D are elaborated in ["Alternate Functions of Port D"](#) on page 90.

## 1.1.7 AV<sub>CC</sub>

AV<sub>CC</sub> is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V<sub>CC</sub>, even if the ADC is not used. If the ADC is used, it should be connected to V<sub>CC</sub> through a low-pass filter. Note that PC6...4 use digital supply voltage, V<sub>CC</sub>.

## 1.1.8 AREF

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

## 1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/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.

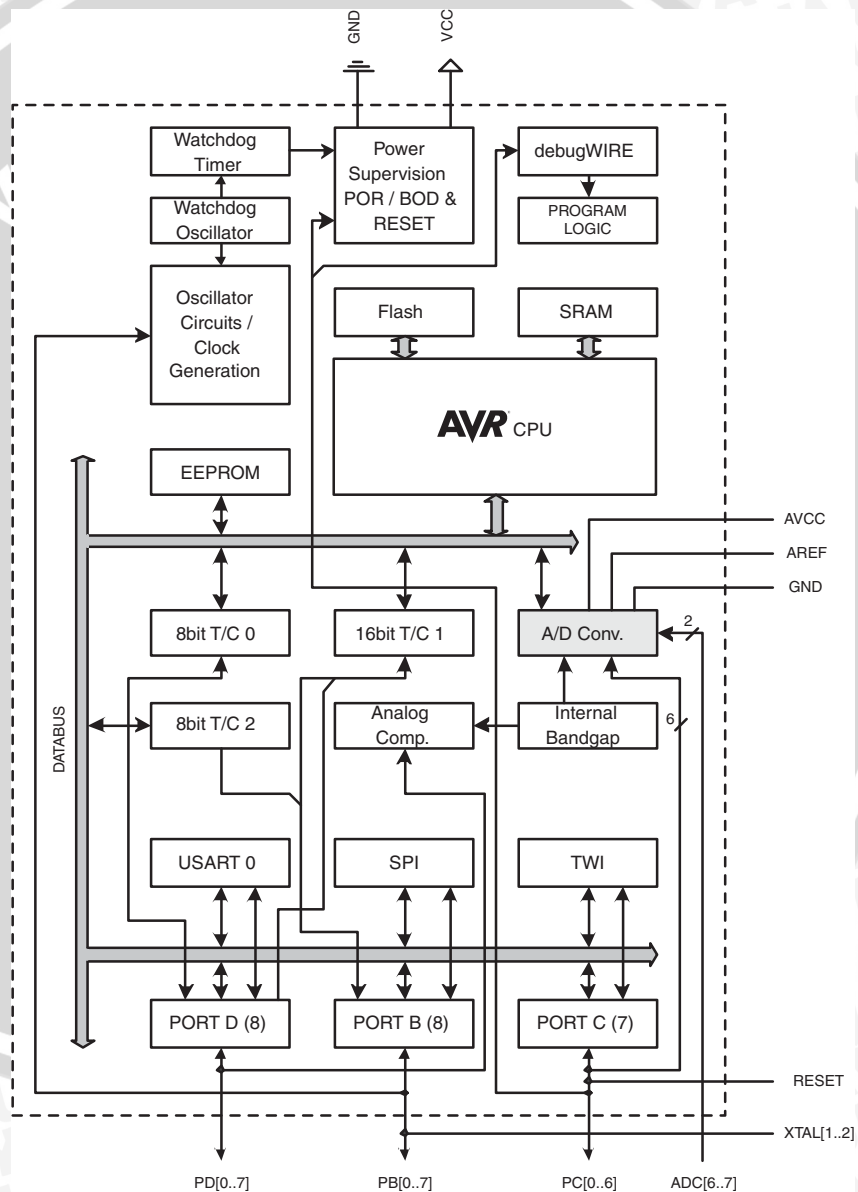


## 2. Overview

The ATmega48A/PA/88A/PA/168A/PA/328/P 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 ATmega48A/PA/88A/PA/168A/PA/328/P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

### 2.1 Block Diagram

Figure 2-1. 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 ATmega48A/PA/88A/PA/168A/PA/328/P provides the following features: 4K/8Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1Kbytes EEPROM, 512/1K/1K/2Kbytes 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 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, 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.

Atmel® offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR® microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use QTouch Suite toolchain allows you to explore, develop and debug your own touch applications.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to 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 ATmega48A/PA/88A/PA/168A/PA/328/P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48A/PA/88A/PA/168A/PA/328/P 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.

## 2.2 Comparison Between Processors

The ATmega48A/PA/88A/PA/168A/PA/328/P differ only in memory sizes, boot loader support, and interrupt vector sizes. [Table 2-1](#) summarizes the different memory and interrupt vector sizes for the devices.

**Table 2-1.** Memory Size Summary

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48A	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega48PA	4KBytes	256Bytes	512Bytes	1 instruction word/vector
ATmega88A	8KBytes	512Bytes	1KBytes	1 instruction word/vector



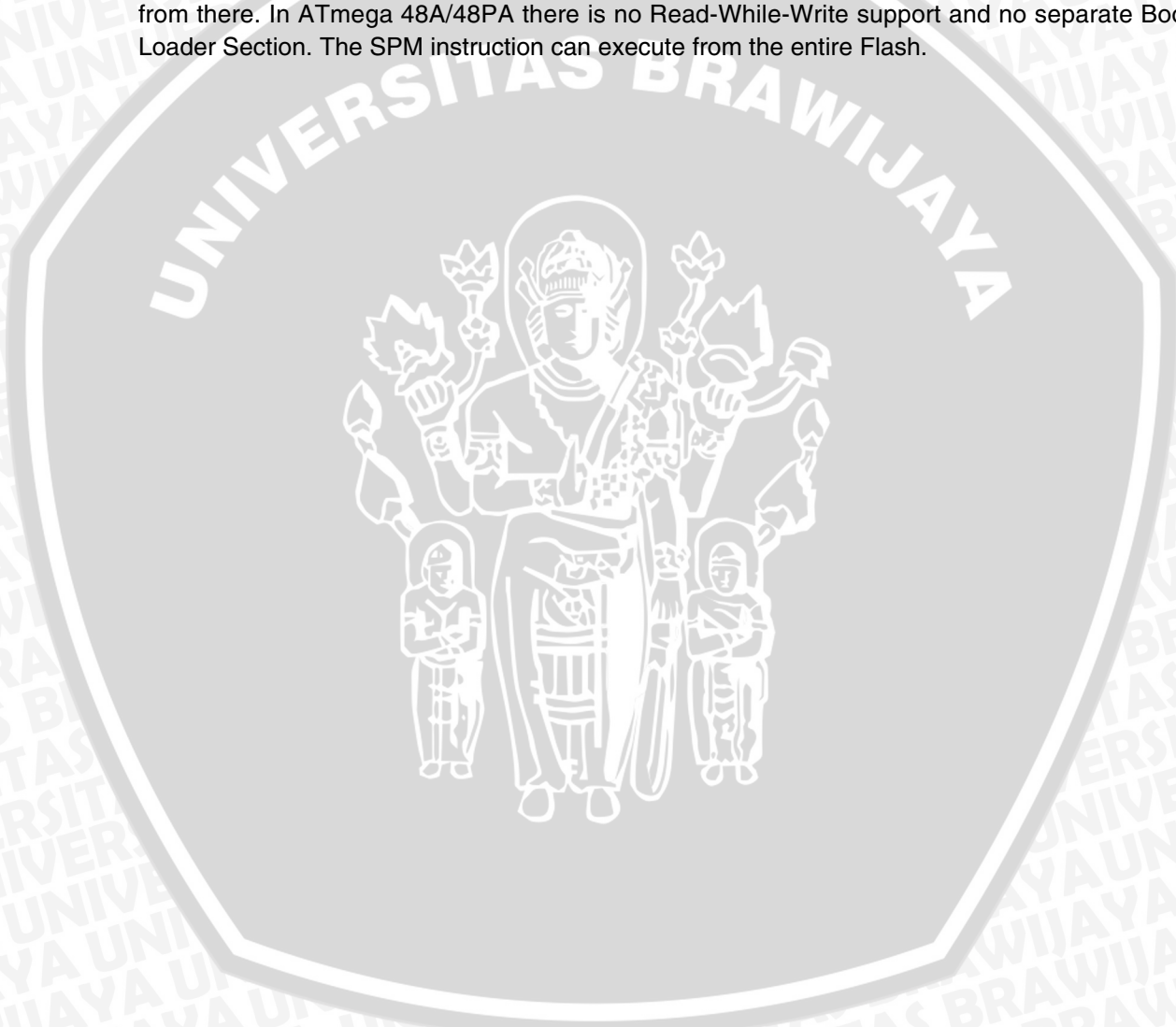


# ATmega48A/PA/88A/PA/168A/PA/328/P

**Table 2-1.** Memory Size Summary (Continued)

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega88PA	8KBytes	512Bytes	1KBytes	1 instruction word/vector
ATmega168A	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega168PA	16KBytes	512Bytes	1KBytes	2 instruction words/vector
ATmega328	32KBytes	1KBytes	2KBytes	2 instruction words/vector
ATmega328P	32KBytes	1KBytes	2KBytes	2 instruction words/vector

ATmega48A/PA/88A/PA/168A/PA/328/P support a real Read-While-Write Self-Programming mechanism. There is a separate Boot Loader Section, and the SPM instruction can only execute from there. In ATmega 48A/48PA there is no Read-While-Write support and no separate Boot Loader Section. The SPM instruction can execute from the entire Flash.



## 3. Resources

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

## 4. Data Retention

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

## 5. About Code Examples

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

For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

## 6. Capacitive Touch Sensing

The Atmel® QTouch® Library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR® microcontrollers. The QTouch Library includes support for the Atmel QTouch and Atmel QMatrix® acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch Library for the AVR Microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch Library is FREE and downloadable from the Atmel website at the following location: [www.atmel.com/qtouchlibrary](http://www.atmel.com/qtouchlibrary). For implementation details and other information, refer to the [Atmel QTouch Library User Guide](#) - also available for download from Atmel website.

## 7. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0xFF	Reserved	-	-	-	-	-	-	-	-	
0xFE	Reserved	-	-	-	-	-	-	-	-	
0xFD	Reserved	-	-	-	-	-	-	-	-	
0xFC	Reserved	-	-	-	-	-	-	-	-	
0xFB	Reserved	-	-	-	-	-	-	-	-	
0xFA	Reserved	-	-	-	-	-	-	-	-	
0xF9	Reserved	-	-	-	-	-	-	-	-	
0xF8	Reserved	-	-	-	-	-	-	-	-	
0xF7	Reserved	-	-	-	-	-	-	-	-	
0xF6	Reserved	-	-	-	-	-	-	-	-	
0xF5	Reserved	-	-	-	-	-	-	-	-	
0xF4	Reserved	-	-	-	-	-	-	-	-	
0xF3	Reserved	-	-	-	-	-	-	-	-	
0xF2	Reserved	-	-	-	-	-	-	-	-	
0xF1	Reserved	-	-	-	-	-	-	-	-	
0xF0	Reserved	-	-	-	-	-	-	-	-	
0xEF	Reserved	-	-	-	-	-	-	-	-	
0xEE	Reserved	-	-	-	-	-	-	-	-	
0xED	Reserved	-	-	-	-	-	-	-	-	
0xEC	Reserved	-	-	-	-	-	-	-	-	
0xEB	Reserved	-	-	-	-	-	-	-	-	
0xEA	Reserved	-	-	-	-	-	-	-	-	
0xE9	Reserved	-	-	-	-	-	-	-	-	
0xE8	Reserved	-	-	-	-	-	-	-	-	
0xE7	Reserved	-	-	-	-	-	-	-	-	
0xE6	Reserved	-	-	-	-	-	-	-	-	
0xE5	Reserved	-	-	-	-	-	-	-	-	
0xE4	Reserved	-	-	-	-	-	-	-	-	
0xE3	Reserved	-	-	-	-	-	-	-	-	
0xE2	Reserved	-	-	-	-	-	-	-	-	
0xE1	Reserved	-	-	-	-	-	-	-	-	
0xE0	Reserved	-	-	-	-	-	-	-	-	
0xDF	Reserved	-	-	-	-	-	-	-	-	
0xDE	Reserved	-	-	-	-	-	-	-	-	
0xDD	Reserved	-	-	-	-	-	-	-	-	
0xDC	Reserved	-	-	-	-	-	-	-	-	
0xDB	Reserved	-	-	-	-	-	-	-	-	
0xDA	Reserved	-	-	-	-	-	-	-	-	
0xD9	Reserved	-	-	-	-	-	-	-	-	
0xD8	Reserved	-	-	-	-	-	-	-	-	
0xD7	Reserved	-	-	-	-	-	-	-	-	
0xD6	Reserved	-	-	-	-	-	-	-	-	
0xD5	Reserved	-	-	-	-	-	-	-	-	
0xD4	Reserved	-	-	-	-	-	-	-	-	
0xD3	Reserved	-	-	-	-	-	-	-	-	
0xD2	Reserved	-	-	-	-	-	-	-	-	
0xD1	Reserved	-	-	-	-	-	-	-	-	
0xD0	Reserved	-	-	-	-	-	-	-	-	
0xCF	Reserved	-	-	-	-	-	-	-	-	
0xCE	Reserved	-	-	-	-	-	-	-	-	
0xCD	Reserved	-	-	-	-	-	-	-	-	
0xCC	Reserved	-	-	-	-	-	-	-	-	
0xCB	Reserved	-	-	-	-	-	-	-	-	
0xCA	Reserved	-	-	-	-	-	-	-	-	
0xC9	Reserved	-	-	-	-	-	-	-	-	
0xC8	Reserved	-	-	-	-	-	-	-	-	
0xC7	Reserved	-	-	-	-	-	-	-	-	
0xC6	UDR0	USART I/O Data Register								201
0xC5	UBRR0H					USART Baud Rate Register High				205
0xC4	UBRR0L	USART Baud Rate Register Low								205
0xC3	Reserved	-	-	-	-	-	-	-	-	
0xC2	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0	203/214
0xC1	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	202
0xC0	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0	201



# ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xBF)	Reserved	–	–	–	–	–	–	–	–	
(0xBE)	Reserved	–	–	–	–	–	–	–	–	
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	–	246
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	243
(0xBB)	TWDR	2-wire Serial Interface Data Register								245
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	246
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	245
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register								243
(0xB7)	Reserved	–	–	–	–	–	–	–	–	
(0xB6)	ASSR	–	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB	166
(0xB5)	Reserved	–	–	–	–	–	–	–	–	
(0xB4)	OCR2B	Timer/Counter2 Output Compare Register B								164
(0xB3)	OCR2A	Timer/Counter2 Output Compare Register A								164
(0xB2)	TCNT2	Timer/Counter2 (8-bit)								164
(0xB1)	TCCR2B	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	163
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	–	–	WGM21	WGM20	160
(0xAF)	Reserved	–	–	–	–	–	–	–	–	
(0xAE)	Reserved	–	–	–	–	–	–	–	–	
(0xAD)	Reserved	–	–	–	–	–	–	–	–	
(0xAC)	Reserved	–	–	–	–	–	–	–	–	
(0xAB)	Reserved	–	–	–	–	–	–	–	–	
(0xAA)	Reserved	–	–	–	–	–	–	–	–	
(0xA9)	Reserved	–	–	–	–	–	–	–	–	
(0xA8)	Reserved	–	–	–	–	–	–	–	–	
(0xA7)	Reserved	–	–	–	–	–	–	–	–	
(0xA6)	Reserved	–	–	–	–	–	–	–	–	
(0xA5)	Reserved	–	–	–	–	–	–	–	–	
(0xA4)	Reserved	–	–	–	–	–	–	–	–	
(0xA3)	Reserved	–	–	–	–	–	–	–	–	
(0xA2)	Reserved	–	–	–	–	–	–	–	–	
(0xA1)	Reserved	–	–	–	–	–	–	–	–	
(0xA0)	Reserved	–	–	–	–	–	–	–	–	
(0x9F)	Reserved	–	–	–	–	–	–	–	–	
(0x9E)	Reserved	–	–	–	–	–	–	–	–	
(0x9D)	Reserved	–	–	–	–	–	–	–	–	
(0x9C)	Reserved	–	–	–	–	–	–	–	–	
(0x9B)	Reserved	–	–	–	–	–	–	–	–	
(0x9A)	Reserved	–	–	–	–	–	–	–	–	
(0x99)	Reserved	–	–	–	–	–	–	–	–	
(0x98)	Reserved	–	–	–	–	–	–	–	–	
(0x97)	Reserved	–	–	–	–	–	–	–	–	
(0x96)	Reserved	–	–	–	–	–	–	–	–	
(0x95)	Reserved	–	–	–	–	–	–	–	–	
(0x94)	Reserved	–	–	–	–	–	–	–	–	
(0x93)	Reserved	–	–	–	–	–	–	–	–	
(0x92)	Reserved	–	–	–	–	–	–	–	–	
(0x91)	Reserved	–	–	–	–	–	–	–	–	
(0x90)	Reserved	–	–	–	–	–	–	–	–	
(0x8F)	Reserved	–	–	–	–	–	–	–	–	
(0x8E)	Reserved	–	–	–	–	–	–	–	–	
(0x8D)	Reserved	–	–	–	–	–	–	–	–	
(0x8C)	Reserved	–	–	–	–	–	–	–	–	
(0x8B)	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								140
(0x8A)	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								140
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								140
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								140
(0x87)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								140
(0x86)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								140
(0x85)	TCNT1H	Timer/Counter1 - Counter Register High Byte								140
(0x84)	TCNT1L	Timer/Counter1 - Counter Register Low Byte								140
(0x83)	Reserved	–	–	–	–	–	–	–	–	
(0x82)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	139
(0x81)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	138
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	136
(0x7F)	DIDR1	–	–	–	–	–	–	AIN1D	AIN0D	251
(0x7E)	DIDR0	–	–	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	268



# ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7D)	Reserved	–	–	–	–	–	–	–	–	
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	264
(0x7B)	ADCSRB	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	267
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	265
(0x79)	ADCH	ADC Data Register High byte								267
(0x78)	ADCL	ADC Data Register Low byte								267
(0x77)	Reserved	–	–	–	–	–	–	–	–	
(0x76)	Reserved	–	–	–	–	–	–	–	–	
(0x75)	Reserved	–	–	–	–	–	–	–	–	
(0x74)	Reserved	–	–	–	–	–	–	–	–	
(0x73)	Reserved	–	–	–	–	–	–	–	–	
(0x72)	Reserved	–	–	–	–	–	–	–	–	
(0x71)	Reserved	–	–	–	–	–	–	–	–	
(0x70)	TIMSK2	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	165
(0x6F)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	141
(0x6E)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	113
(0x6D)	PCMSK2	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	76
(0x6C)	PCMSK1	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	76
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	76
(0x6A)	Reserved	–	–	–	–	–	–	–	–	
(0x69)	EICRA	–	–	–	–	ISC11	ISC10	ISC01	ISC00	73
(0x68)	PCICR	–	–	–	–	–	PCIE2	PCIE1	PCIE0	
(0x67)	Reserved	–	–	–	–	–	–	–	–	
(0x66)	OSCCAL	Oscillator Calibration Register								38
(0x65)	Reserved	–	–	–	–	–	–	–	–	
(0x64)	PRR	PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC	43
(0x63)	Reserved	–	–	–	–	–	–	–	–	
(0x62)	Reserved	–	–	–	–	–	–	–	–	
(0x61)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	38
(0x60)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	56
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	10
0x3E (0x5E)	SPH	–	–	–	–	–	(SP10) <sup>5</sup>	SP9	SP8	13
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	13
0x3C (0x5C)	Reserved	–	–	–	–	–	–	–	–	
0x3B (0x5B)	Reserved	–	–	–	–	–	–	–	–	
0x3A (0x5A)	Reserved	–	–	–	–	–	–	–	–	
0x39 (0x59)	Reserved	–	–	–	–	–	–	–	–	
0x38 (0x58)	Reserved	–	–	–	–	–	–	–	–	
0x37 (0x57)	SPMCSR	SPMIE	(RWWSB) <sup>5</sup>	–	(RWWRE) <sup>5</sup>	BLBSET	PGWRT	PGERS	SELFPRGEN	295
0x36 (0x56)	Reserved	–	–	–	–	–	–	–	–	
0x35 (0x55)	MCUCR	–	BODS <sup>(6)</sup>	BODSE <sup>(6)</sup>	PUD	–	–	IVSEL	IVCE	46/70/94
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	56
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	41
0x32 (0x52)	Reserved	–	–	–	–	–	–	–	–	
0x31 (0x51)	Reserved	–	–	–	–	–	–	–	–	
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	249
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPI Data Register								177
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	176
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	175
0x2B (0x4B)	GPOR2	General Purpose I/O Register 2								26
0x2A (0x4A)	GPOR1	General Purpose I/O Register 1								26
0x29 (0x49)	Reserved	–	–	–	–	–	–	–	–	
0x28 (0x48)	OCR0B	Timer/Counter0 Output Compare Register B								
0x27 (0x47)	OCR0A	Timer/Counter0 Output Compare Register A								
0x26 (0x46)	TCNT0	Timer/Counter0 (8-bit)								
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	PSRASY	PSRSYNC	145/167
0x22 (0x42)	EEARH	(EEPROM Address Register High Byte) <sup>5</sup>								22
0x21 (0x41)	EEARL	EEPROM Address Register Low Byte								22
0x20 (0x40)	EEDR	EEPROM Data Register								22
0x1F (0x3F)	EEDR	–	–	EEDR1	EEDR0	EERIE	EEMPE	EEPE	EERE	22
0x1E (0x3E)	GPOR0	General Purpose I/O Register 0								26
0x1D (0x3D)	EIMSK	–	–	–	–	–	–	INT1	INT0	74
0x1C (0x3C)	EIFR	–	–	–	–	–	–	INTF1	INTF0	74



# ATmega48A/PA/88A/PA/168A/PA/328/P

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	–	–	–	–	–	PCIF2	PCIF1	PCIF0	
0x1A (0x3A)	Reserved	–	–	–	–	–	–	–	–	
0x19 (0x39)	Reserved	–	–	–	–	–	–	–	–	
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	TIFR2	–	–	–	–	–	OCF2B	OCF2A	TOV2	165
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	141
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	
0x10 (0x30)	Reserved	–	–	–	–	–	–	–	–	
0x0F (0x2F)	Reserved	–	–	–	–	–	–	–	–	
0x0E (0x2E)	Reserved	–	–	–	–	–	–	–	–	
0x0D (0x2D)	Reserved	–	–	–	–	–	–	–	–	
0x0C (0x2C)	Reserved	–	–	–	–	–	–	–	–	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	95
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	95
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	95
0x08 (0x28)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	94
0x07 (0x27)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	94
0x06 (0x26)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	94
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	94
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	94
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	94
0x02 (0x22)	Reserved	–	–	–	–	–	–	–	–	
0x01 (0x21)	Reserved	–	–	–	–	–	–	–	–	
0x0 (0x20)	Reserved	–	–	–	–	–	–	–	–	

- Note:
- For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  - I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
  - Some of the Status Flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such Status Flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
  - When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega48A/PA/88A/PA/168A/PA/328/P is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
  - Only valid for ATmega88A/88PA/168A/168PA/328/328P.
  - BODS and BODSE only available for picoPower devices ATmega48PA/88PA/168PA/328P





## 8. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
<b>ARITHMETIC AND LOGIC INSTRUCTIONS</b>					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	RdI,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	RdI,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
<b>BRANCH INSTRUCTIONS</b>					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
JMP <sup>(1)</sup>	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
CALL <sup>(1)</sup>	k	Direct Subroutine Call	$PC \leftarrow k$	None	4
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ 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) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N $\oplus$ V = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N $\oplus$ V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then $PC \leftarrow PC + k + 1$	None	1/2

# ATmega48A/PA/88A/PA/168A/PA/328/P

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
<b>BIT AND BIT-TEST INSTRUCTIONS</b>					
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
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
<b>DATA TRANSFER INSTRUCTIONS</b>					
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
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



# ATmega48A/PA/88A/PA/168A/PA/328/P

Mnemonics	Operands	Description	Operation	Flags	#Clocks
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
<b>MCU CONTROL INSTRUCTIONS</b>					
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

Note: 1. These instructions are only available in ATmega168PA and ATmega328P.





## 9. Ordering Information

### 9.1 ATmega48A

Speed (MHz)	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega48A-AU ATmega48A-AUR <sup>(5)</sup> ATmega48A-CCU ATmega48A-CCUR <sup>(5)</sup> ATmega48A-MMH <sup>(4)</sup> ATmega48A-MMHR <sup>(4)(5)</sup> ATmega48A-MU ATmega48A-MUR <sup>(5)</sup> ATmega48A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "[Speed Grades](#)" on page 322.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



Package Type	
<b>32A</b>	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6 mm package, ball pitch 0.5 mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

# ATmega48A/PA/88A/PA/168A/PA/328/P

## 9.2 ATmega48PA

Speed (MHz) <sup>(3)</sup>	Power Supply	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range	
20	1.8 - 5.5	ATmega48PA-AU	32A	Industrial (-40°C to 85°C)	
		ATmega48PA-AUR <sup>(5)</sup>	32A		
		ATmega48PA-CCU	32CC1		
		ATmega48PA-CCUR <sup>(5)</sup>	32CC1		
		ATmega48PA-MMH <sup>(4)</sup>	28M1		
		ATmega48PA-MMHR <sup>(4)(5)</sup>	28M1		
		ATmega48PA-MU	32M1-A		
		ATmega48PA-MUR <sup>(5)</sup>	32M1-A		
	ATmega48PA-PU	28P3			
			ATmega48PA-AN	32A	Industrial (-40°C to 105°C)
			ATmega48PA-ANR <sup>(4)</sup>	32A	
			ATmega48PA-MMN	28M1	
			ATmega48PA-MMNR <sup>(4)</sup>	28M1	
			ATmega48PA-MN	32M1-A	
ATmega48PA-MNR <sup>(4)</sup>			32M1-A		
ATmega48PA-PN	28P3				

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See ["Speed Grades" on page 322](#).
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)



## 9.3 ATmega88A

Speed (MHz)	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega88A-AU ATmega88A-AUR <sup>(5)</sup> ATmega88A-CCU ATmega88A-CCUR <sup>(5)</sup> ATmega88A-MMH <sup>(4)</sup> ATmega88A-MMHR <sup>(4)(5)</sup> ATmega88A-MU ATmega88A-MUR <sup>(5)</sup> ATmega88A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "Speed Grades" on page 322.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



### Package Type

<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)



# ATmega48A/PA/88A/PA/168A/PA/328/P

## 9.4 ATmega88PA

Speed (MHz) <sup>(3)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega88PA-AU	32A	Industrial (-40°C to 85°C)
		ATmega88PA-AUR <sup>(5)</sup>	32A	
		ATmega88PA-CCU	32CC1	
		ATmega88PA-CCUR <sup>(5)</sup>	32CC1	
		ATmega88PA-MMH <sup>(4)</sup>	28M1	
		ATmega88PA-MMHR <sup>(4)(5)</sup>	28M1	
		ATmega88PA-MU	32M1-A	
	ATmega88PA-MUR <sup>(5)</sup>	32M1-A		
	ATmega88PA-PU	28P3		
	1.8 - 5.5	ATmega88PA-AN	32A	Industrial (-40°C to 105°C)
		ATmega88PA-ANR <sup>(5)</sup>	32A	
		ATmega88PA-MMN	28M1	
		ATmega88PA-MMNR <sup>(5)</sup>	28M1	
		ATmega88PA-MN	32M1-A	
ATmega88PA-MNR <sup>(5)</sup>		32M1-A		
ATmega88PA-PN		28P3		

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See ["Speed Grades" on page 322](#).
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



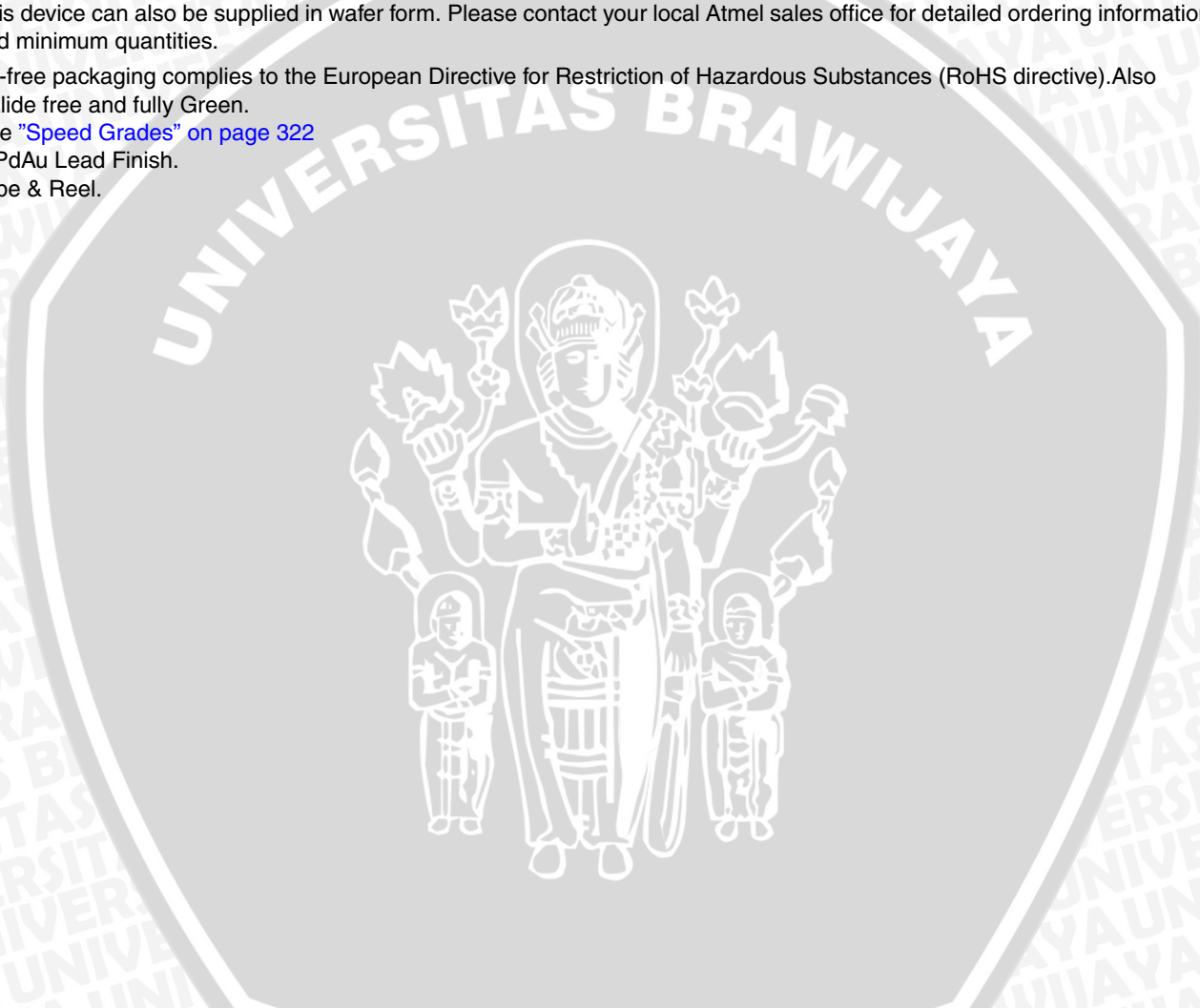
Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5 mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50 mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

# ATmega48A/PA/88A/PA/168A/PA/328/P

## 9.5 ATmega168A

Speed (MHz) <sup>(3)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega168A-AU ATmega168A-AUR <sup>(5)</sup> ATmega168A-CCU ATmega168A-CCUR <sup>(5)</sup> ATmega168A-MMH <sup>(4)</sup> ATmega168A-MMHR <sup>(4)(5)</sup> ATmega168A-MU ATmega168A-MUR <sup>(5)</sup> ATmega168A-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "[Speed Grades](#)" on page 322
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6 mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)

## 9.6 ATmega168PA

Speed (MHz) <sup>(3)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega168PA-AU ATmega168PA-AUR <sup>(5)</sup> ATmega168PA-CCU ATmega168PA-CCUR <sup>(5)</sup> ATmega168PA-MMH <sup>(4)</sup> ATmega168PA-MMHR <sup>(4)(5)</sup> ATmega168PA-MU ATmega168PA-MUR <sup>(5)</sup> ATmega168PA-PU	32A 32A 32CC1 32CC1 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)
20	1.8 - 5.5	ATmega168PA-AN ATmega168PA-ANR <sup>(5)</sup> ATmega168PA-MN ATmega168PA-MNR <sup>(5)</sup> ATmega168PA-PN	32A 32A 32M1-A 32M1-A 28P3	Industrial (-40°C to 105°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See "[Speed Grades](#)" on page 322.
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>32CC1</b>	32-ball, 4 x 4 x 0.6mm package, ball pitch 0.5mm, Ultra Thin, Fine-Pitch Ball Grill Array (UFBGA)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)



# ATmega48A/PA/88A/PA/168A/PA/328/P

## 9.7 ATmega328

Speed (MHz)	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20 <sup>(3)</sup>	1.8 - 5.5	ATmega328-AU ATmega328-AUR <sup>(5)</sup> ATmega328-MMH <sup>(4)</sup> ATmega328-MMHR <sup>(4)(5)</sup> ATmega328-MU ATmega328-MUR <sup>(5)</sup> ATmega328-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)

- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See [Figure 29-1 on page 322](#).
  4. NiPdAu Lead Finish.
  5. Tape & Reel



Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)



## 9.8 ATmega328P

Speed (MHz) <sup>(3)</sup>	Power Supply (V)	Ordering Code <sup>(2)</sup>	Package <sup>(1)</sup>	Operational Range
20	1.8 - 5.5	ATmega328P-AU ATmega328P-AUR <sup>(5)</sup> ATmega328P-MMH <sup>(4)</sup> ATmega328P-MMHR <sup>(4)(5)</sup> ATmega328P-MU ATmega328P-MUR <sup>(5)</sup> ATmega328P-PU	32A 32A 28M1 28M1 32M1-A 32M1-A 28P3	Industrial (-40°C to 85°C)
		ATmega328P-AN ATmega328P-ANR <sup>(5)</sup> ATmega328P-MN ATmega328P-MNR <sup>(5)</sup> ATmega328P-PN	32A 32A 32M1-A 32M1-A 28P3	Industrial (-40°C to 105°C)

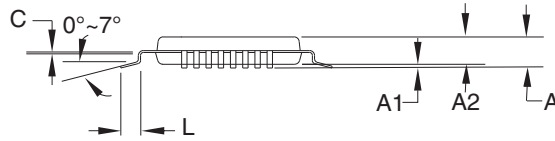
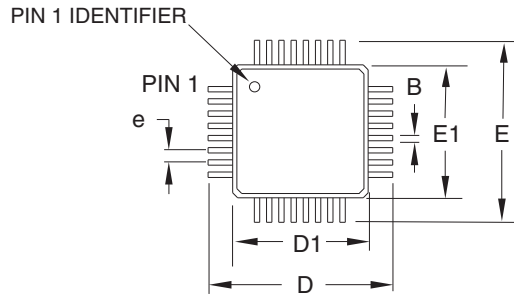
- Note:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information and minimum quantities.
  2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
  3. See [Figure 29-1 on page 322](#).
  4. NiPdAu Lead Finish.
  5. Tape & Reel.



Package Type	
<b>32A</b>	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
<b>28M1</b>	28-pad, 4 x 4 x 1.0 body, Lead Pitch 0.45mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)
<b>28P3</b>	28-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
<b>32M1-A</b>	32-pad, 5 x 5 x 1.0 body, Lead Pitch 0.50mm Quad Flat No-Lead/Micro Lead Frame Package (QFN/MLF)

## 10. Packaging Information

### 10.1 32A



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

**Notes:**

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

2010-10-20



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**32A**, 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness,  
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

**DRAWING NO.**

32A

**REV.**

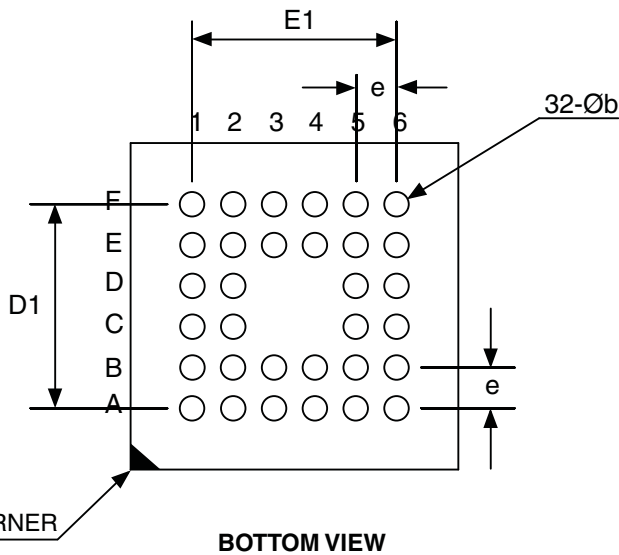
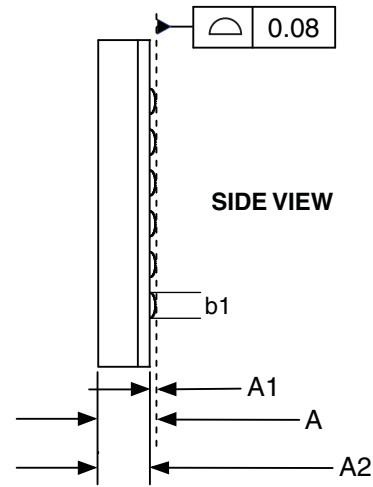
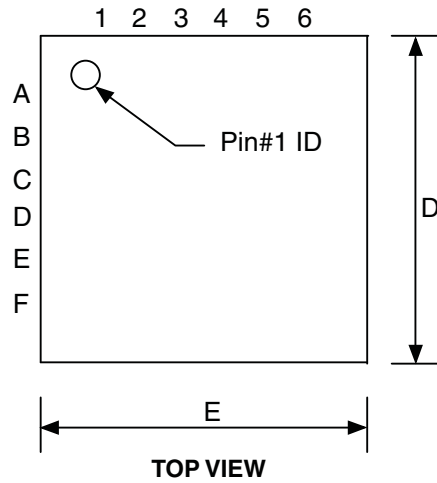
C





# ATmega48A/PA/88A/PA/168A/PA/328/P

## 10.2 32CC1



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	0.60	
A1	0.12	–	–	
A2	0.38 REF			
b	0.25	0.30	0.35	1
b1	0.25	–	–	2
D	3.90	4.00	4.10	
D1	2.50 BSC			
E	3.90	4.00	4.10	
E1	2.50 BSC			
e	0.50 BSC			

Note1: Dimension "b" is measured at the maximum ball dia. in a plane parallel to the seating plane.

Note2: Dimension "b1" is the solderable surface defined by the opening of the solder resist layer.

07/06/10



**Package Drawing Contact:**  
packagedrawings@atmel.com

**TITLE**

**32CC1**, 32-ball (6 x 6 Array), 4 x 4 x 0.6 mm package, ball pitch 0.50 mm, Ultra Thin, Fine-Pitch Ball Grid Array (UFBGA)

**GPC**

CAG

**DRAWING NO.**

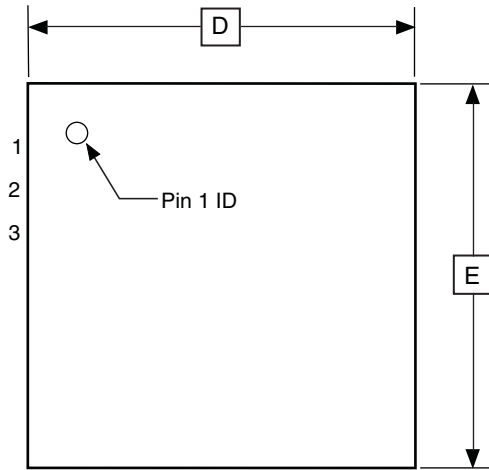
32CC1

**REV.**

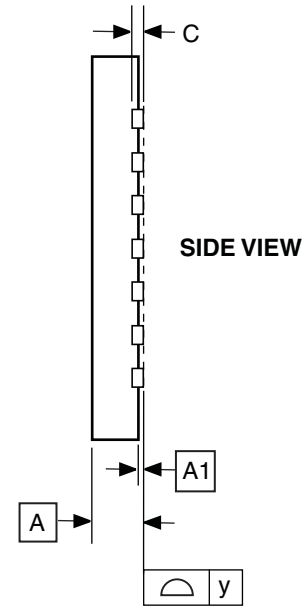
B



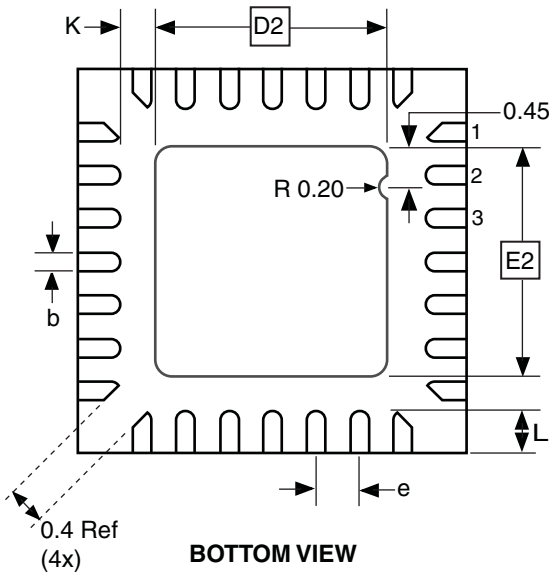
## 10.3 28M1



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	0.00	0.02	0.05	
b	0.17	0.22	0.27	
C	0.20 REF			
D	3.95	4.00	4.05	
D2	2.35	2.40	2.45	
E	3.95	4.00	4.05	
E2	2.35	2.40	2.45	
e	0.45			
L	0.35	0.40	0.45	
y	0.00	-	0.08	
K	0.20	-	-	

Note: The terminal #1 ID is a Laser-marked Feature.

10/24/08



Package Drawing Contact:  
packagedrawings@atmel.com

**TITLE**

**28M1**, 28-pad, 4 x 4 x 1.0 mm Body, Lead Pitch 0.45 mm, 2.4 x 2.4 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad Flat No Lead Package (VQFN)

**GPC**

ZBV

**DRAWING NO.**

28M1

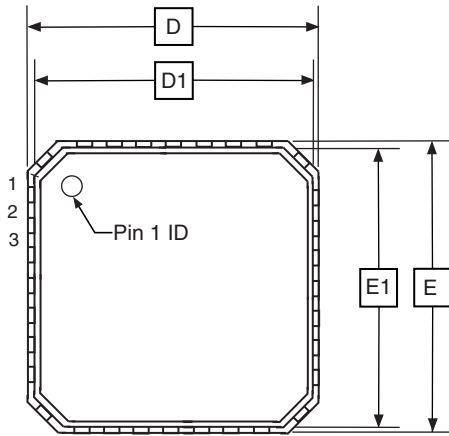
**REV.**

B

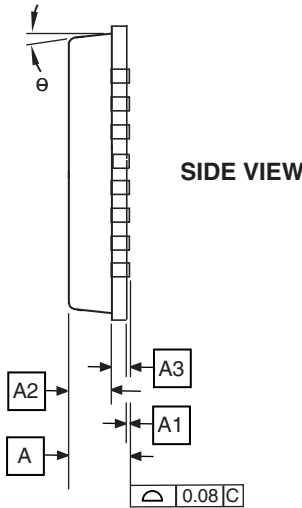


# ATmega48A/PA/88A/PA/168A/PA/328/P

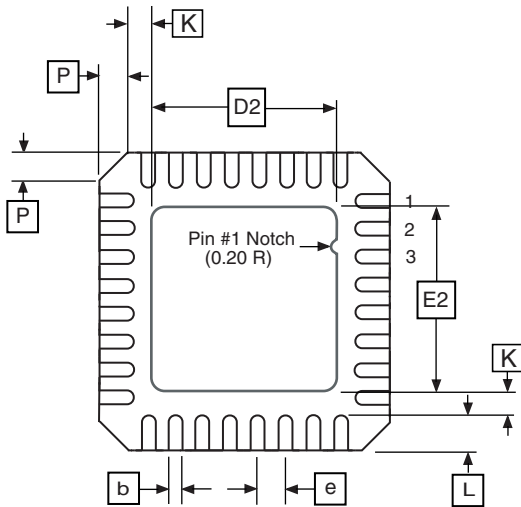
## 10.4 32M1-A



TOP VIEW



SIDE VIEW



BOTTOM VIEW

COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	0.80	0.90	1.00	
A1	–	0.02	0.05	
A2	–	0.65	1.00	
A3	0.20 REF			
b	0.18	0.23	0.30	
D	4.90	5.00	5.10	
D1	4.70	4.75	4.80	
D2	2.95	3.10	3.25	
E	4.90	5.00	5.10	
E1	4.70	4.75	4.80	
E2	2.95	3.10	3.25	
e	0.50 BSC			
L	0.30	0.40	0.50	
P	–	–	0.60	
θ	–	–	12°	
K	0.20	–	–	

Note: JEDEC Standard MO-220, Fig. 2 (Anvil Singulation), VHHD-2.

5/25/06



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**32M1-A**, 32-pad, 5 x 5 x 1.0 mm Body, Lead Pitch 0.50 mm,  
3.10 mm Exposed Pad, Micro Lead Frame Package (MLF)

**DRAWING NO.**

32M1-A

**REV.**

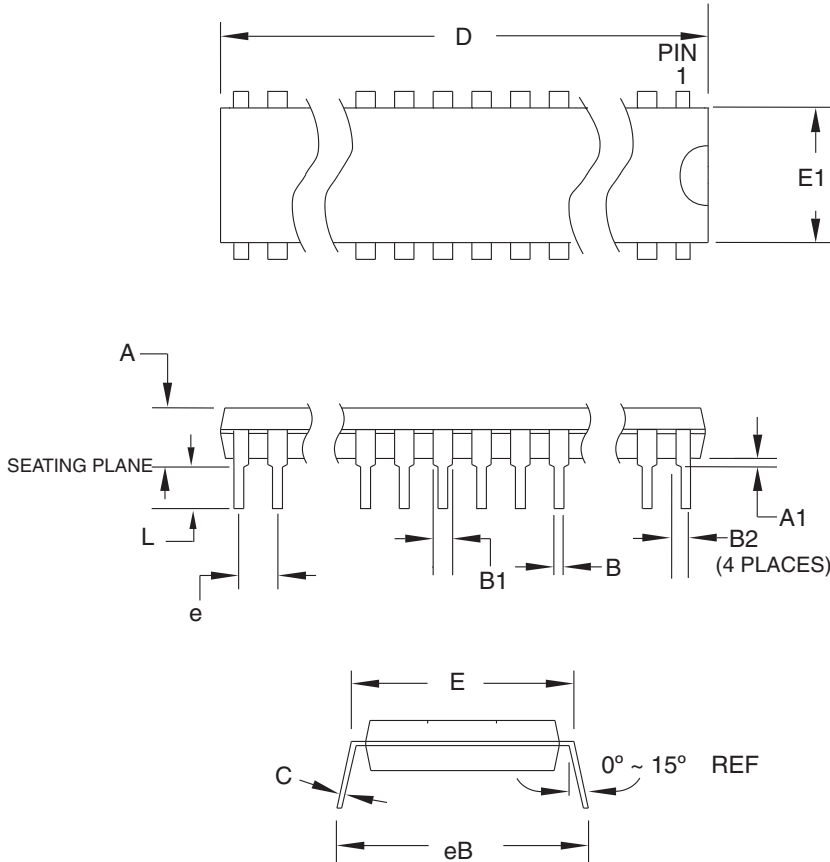
E





# ATmega48A/PA/88A/PA/168A/PA/328/P

10.5 28P3



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.5724	
A1	0.508	–	–	
D	34.544	–	34.798	Note 1
E	7.620	–	8.255	
E1	7.112	–	7.493	Note 1
B	0.381	–	0.533	
B1	1.143	–	1.397	
B2	0.762	–	1.143	
L	3.175	–	3.429	
C	0.203	–	0.356	
eB	–	–	10.160	
e	2.540 TYP			

Note: 1. Dimensions D and E1 do not include mold Flash or Protrusion.  
Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01



2325 Orchard Parkway  
San Jose, CA 95131

**TITLE**

**28P3**, 28-lead (0.300"/7.62 mm Wide) Plastic Dual  
Inline Package (PDIP)

**DRAWING NO.**

28P3

**REV.**

B



## 11. Errata

### 11.1 Errata ATmega48A

The revision letter in this section refers to the revision of the ATmega48A device.

#### 11.1.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

### 11.2 Errata ATmega48PA

The revision letter in this section refers to the revision of the ATmega48PA device.

#### 11.2.1 Rev. D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

### 11.3 Errata ATmega88A

The revision letter in this section refers to the revision of the ATmega88A device.

#### 11.3.1 Rev. F

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

## 11.4 Errata ATmega88PA

The revision letter in this section refers to the revision of the ATmega88PA device.

### 11.4.1 Rev. F

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

## 11.5 Errata ATmega168A

The revision letter in this section refers to the revision of the ATmega168A device.

### 11.5.1 Rev. E

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

## 11.6 Errata ATmega168PA

The revision letter in this section refers to the revision of the ATmega168PA device.

### 11.6.1 Rev E

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.



## 11.7 Errata ATmega328

The revision letter in this section refers to the revision of the ATmega328 device.

### 11.7.1 Rev D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

### 11.7.2 Rev C

Not sampled.

### 11.7.3 Rev B

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

**Problem Fix/ Workaround**

None.

### 11.7.4 Rev A

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

**Problem Fix/ Workaround**

None.

## 11.8 Errata ATmega328P

The revision letter in this section refers to the revision of the ATmega328P device.

### 11.8.1 Rev D

- **Analog MUX can be turned off when setting ACME bit**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

### 11.8.2 Rev C

Not sampled.

### 11.8.3 Rev B

- **Analog MUX can be turned off when setting ACME bit**
- **Unstable 32kHz Oscillator**

1. **Analog MUX can be turned off when setting ACME bit**

If the ACME (Analog Comparator Multiplexer Enabled) bit in ADCSRB is set while MUX3 in ADMUX is '1' (ADMUX[3:0]=1xxx), all MUX'es are turned off until the ACME bit is cleared.

**Problem Fix/Workaround**

Clear the MUX3 bit before setting the ACME bit.

2. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

**Problem Fix/ Workaround**

None.

### 11.8.4 Rev A

- **Unstable 32kHz Oscillator**

1. **Unstable 32kHz Oscillator**

The 32kHz oscillator does not work as system clock. The 32kHz oscillator used as asynchronous timer is inaccurate.

**Problem Fix/ Workaround**

None.

## 12. Datasheet Revision History

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

### 12.1 Rev. 8271D – 05/11

1. Added Atmel QTouch Sensing Capability Feature
2. Updated "Register Description" on page 94 with PINxn as R/W.
3. Added a footnote to the PINxn, page 94.
4. Updated
5. Updated "Ordering Information", "ATmega328" on page 546. Added "ATmega328-MMH" and "ATmega328-MMHR".
6. Updated "Ordering Information", "ATmega328P" on page 547. Added "ATmega328P-MMH" and "ATmega328P-MMHR".
7. Added "Ordering Information" for ATmega48PA/88PA/168PA/328P @ 105°C
8. Updated "Errata ATmega328" on page 555 and "Errata ATmega328P" on page 556
98. Updated the datasheet according to the Atmel new brand style guide.

### 12.2 Rev. 8271C – 08/10

1. Added 32UFBGA Pinout, Table 1-1 on page 2.
2. Updated the "SRAM Data Memory", Figure 8-3 on page 19.
3. Updated "Ordering Information" on page 540 with CCU and CCUR code related to "32CC1" Package drawing.
4. "32CC1" Package drawing added on "Packaging Information" on page 548.

### 12.3 Rev. 8271B – 04/10

1. Updated Table 9-8 with correct value for timer oscillator at xtal2/tos2
2. Corrected use of SBIS instructions in assembly code examples.
3. Corrected BOD and BODSE bits to R/W in Section 10.11.2 on page 46, Section 12.5 on page 70 and Section 14.4 on page 94
4. Figures for bandgap characterization added, Figure 30-34 on page 350, Figure 30-81 on page 375, Figure 30-128 on page 400, Figure 30-175 on page 425, Figure 30-222 on page 450, Figure 30-269 on page 475, Figure 30-316 on page 500 and Figure 30-363 on page 525.
5. Updated "Packaging Information" on page 548 by replacing 28M1 with a correct corresponding package.



## 12.4 Rev. 8271A – 12/09

1. New datasheet 8271 with merged information for ATmega48PA, ATmega88PA, ATmega168PA and ATmega48A, ATmega88A and ATmega168A. Also included information on ATmega328 and ATmega328P
2. Changes done:
  - New devices added: ATmega48A/ATmega88A/ATmega168A and ATmega328
  - Updated Feature Description
  - Updated [Table 2-1 on page 6](#)
  - Added note for BOD Disable on [page 41](#).
  - Added note on BOD and BODSE in "MCUCR – MCU Control Register" on [page 94](#) and "Register Description" on [page 295](#)
  - Added limitation informatin for the application "Boot Loader Support – Read-While-Write Self-Programming" on [page 280](#)
  - Added limitation information for "Program And Data Memory Lock Bits" on [page 297](#)
  - Added specified DC characteristics
  - Added typical characteristics
  - Removed exception information in "Address Match Unit" on [page 224](#).



# UNIVERSITAS BRAWIJAYA

## **Atmel Corporation**

2325 Orchard Parkway  
San Jose, CA 95131  
USA

**Tel:** (+1)(408) 441-0311  
**Fax:** (+1)(408) 487-2600  
[www.atmel.com](http://www.atmel.com)

## **Atmel Asia Limited**

Unit 1-5 & 16, 19/F  
BEA Tower, Millennium City 5  
418 Kwun Tong Road  
Kwun Tong, Kowloon  
HONG KONG  
**Tel:** (+852) 2245-6100  
**Fax:** (+852) 2722-1369

## **Atmel Munich GmbH**

Business Campus  
Parkring 4  
D-85748 Garching b. Munich  
GERMANY  
**Tel:** (+49) 89-31970-0  
**Fax:** (+49) 89-3194621

## **Atmel Japan**

9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
JAPAN  
**Tel:** (+81)(3) 3523-3551  
**Fax:** (+81)(3) 3523-7581

© 2011 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, AVR® and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

**Disclaimer:** The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.



# IRF640 IRF640FP

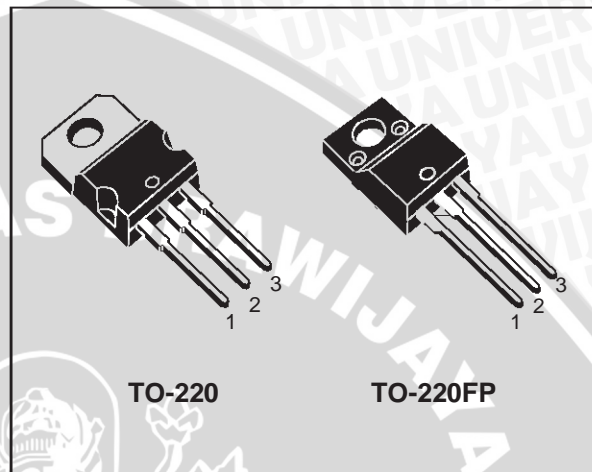
## N - CHANNEL 200V - 0.150Ω - 18A TO-220/TO-220FP MESH OVERLAY™ MOSFET

TYPE	V <sub>DSS</sub>	R <sub>DS(on)</sub>	I <sub>D</sub>
IRF640	200 V	< 0.18 Ω	18 A
IRF640FP	200 V	< 0.18 Ω	18 A

- TYPICAL R<sub>DS(on)</sub> = 0.150 Ω
- EXTREMELY HIGH dv/dt CAPABILITY
- VERY LOW INTRINSIC CAPACITANCES
- GATE CHARGE MINIMIZED

### DESCRIPTION

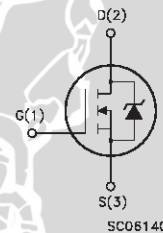
This power MOSFET is designed using the company's consolidated strip layout-based MESH OVERLAY™ process. This technology matches and improves the performances compared with standard parts from various sources.



### APPLICATIONS

- HIGH CURRENT SWITCHING
- UNINTERRUPTIBLE POWER SUPPLY (UPS)
- DC/DC CONVERTERS FOR TELECOM, INDUSTRIAL, AND LIGHTING EQUIPMENT.

### INTERNAL SCHEMATIC DIAGRAM



### ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value		Unit
		IRF640	IRF640FP	
V <sub>DS</sub>	Drain-source Voltage (V <sub>GS</sub> = 0)	200		V
V <sub>DGR</sub>	Drain- gate Voltage (R <sub>GS</sub> = 20 kΩ)	200		V
V <sub>GS</sub>	Gate-source Voltage	± 20		V
I <sub>D</sub>	Drain Current (continuous) at T <sub>c</sub> = 25 °C	18	18(**)	A
I <sub>D</sub>	Drain Current (continuous) at T <sub>c</sub> = 100 °C	11	11(**)	A
I <sub>DM</sub> (●)	Drain Current (pulsed)	72	72	A
P <sub>tot</sub>	Total Dissipation at T <sub>c</sub> = 25 °C	125	40	W
	Derating Factor	1.0	0.32	W/°C
dv/dt(1)	Peak Diode Recovery voltage slope	5	5	V/ns
V <sub>ISO</sub>	Insulation Withstand Voltage (DC)	—	2000	V
T <sub>stg</sub>	Storage Temperature	-65 to 150		°C
T <sub>j</sub>	Max. Operating Junction Temperature	150		°C

(●) Pulse width limited by safe operating area

(1) I<sub>SD</sub> ≤ 18A, di/dt ≤ 300 A/μs, V<sub>DD</sub> ≤ V<sub>(BR)DSS</sub>, T<sub>j</sub> ≤ T<sub>JMAX</sub>

First Digit of the Datecode Being Z or K Identifies Silicon Characterized in this Datasheet

(\*\*) Limited only by Maximum Temperature Allowed





## IRF640/FP

### THERMAL DATA

		TO-220	TO-220FP		
$R_{thj-case}$	Thermal Resistance Junction-case	Max	1.0	3.12	$^{\circ}\text{C}/\text{W}$
$R_{thj-amb}$	Thermal Resistance Junction-ambient	Max	62.5		$^{\circ}\text{C}/\text{W}$
$R_{thc-sink}$	Thermal Resistance Case-sink	Typ	0.5		$^{\circ}\text{C}/\text{W}$
$T_l$	Maximum Lead Temperature For Soldering Purpose		300		$^{\circ}\text{C}$

### AVALANCHE CHARACTERISTICS

Symbol	Parameter	Max Value	Unit
$I_{AR}$	Avalanche Current, Repetitive or Not-Repetitive (pulse width limited by $T_j$ max)	18	A
$E_{AS}$	Single Pulse Avalanche Energy (starting $T_j = 25^{\circ}\text{C}$ , $I_D = I_{AR}$ , $V_{DD} = 50\text{ V}$ )	280	mJ

### ELECTRICAL CHARACTERISTICS ( $T_{case} = 25^{\circ}\text{C}$ unless otherwise specified)

#### OFF

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{(BR)DSS}$	Drain-source Breakdown Voltage	$I_D = 250\ \mu\text{A}$ , $V_{GS} = 0$	200			V
$I_{DSS}$	Zero Gate Voltage Drain Current ( $V_{GS} = 0$ )	$V_{DS} = \text{Max Rating}$ $V_{DS} = \text{Max Rating}$ , $T_c = 125^{\circ}\text{C}$			1 10	$\mu\text{A}$ $\mu\text{A}$
$I_{GSS}$	Gate-body Leakage Current ( $V_{DS} = 0$ )	$V_{GS} = \pm 20\text{ V}$			$\pm 100$	nA

#### ON (\*)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$V_{GS(th)}$	Gate Threshold Voltage	$V_{DS} = V_{GS}$ , $I_D = 250\ \mu\text{A}$	2	3	4	V
$R_{DS(on)}$	Static Drain-source On Resistance	$V_{GS} = 10\text{ V}$ , $I_D = 9\text{ A}$		0.15	0.18	$\Omega$
$I_{D(on)}$	On State Drain Current	$V_{DS} > I_{D(on)} \times R_{DS(on)max}$ $V_{GS} = 10\text{ V}$	18			A

#### DYNAMIC

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$g_{fs}$ (*)	Forward Transconductance	$V_{DS} > I_{D(on)} \times R_{DS(on)max}$ , $I_D = 9\text{ A}$	7	11		S
$C_{iss}$	Input Capacitance	$V_{DS} = 25\text{ V}$ , $f = 1\text{ MHz}$ , $V_{GS} = 0$		1200	1560	pF
$C_{oss}$	Output Capacitance			200	260	pF
$C_{rss}$	Reverse Transfer Capacitance			60	80	pF

**ELECTRICAL CHARACTERISTICS** (continued)

**SWITCHING ON**

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$t_{d(on)}$	Turn-on Time	$V_{DD} = 100\text{ V}$ $I_D = 9\text{ A}$		13	17	ns
$t_r$	Rise Time	$R_G = 4.7\ \Omega$ $V_{GS} = 10\text{ V}$ (see test circuit, figure 3)		27	35	ns
$Q_g$	Total Gate Charge	$V_{DD} = 160\text{ V}$ $I_D = 18\text{ A}$ $V_{GS} = 10\text{ V}$		55	72	nC
$Q_{gs}$	Gate-Source Charge			10		nC
$Q_{gd}$	Gate-Drain Charge			21		nC

**SWITCHING OFF**

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$t_r(V_{off})$	Off-voltage Rise Time	$V_{DD} = 160\text{ V}$ $I_D = 18\text{ A}$		21	27	ns
$t_f$	Fall Time	$R_G = 4.7\ \Omega$ $V_{GS} = 10\text{ V}$ (see test circuit, figure 5)		25	32	ns
$t_c$	Cross-over Time			50	65	ns

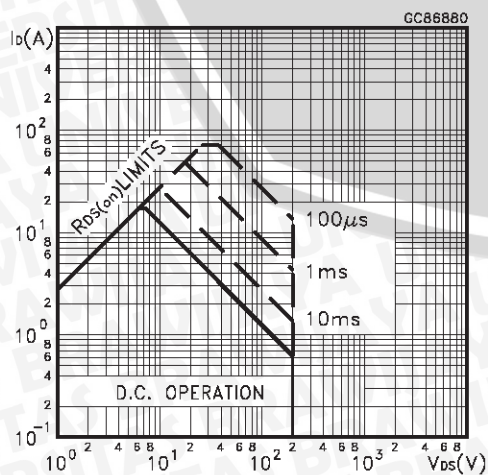
**SOURCE DRAIN DIODE**

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
$I_{SD}$	Source-drain Current				18	A
$I_{SDM}(\bullet)$	Source-drain Current (pulsed)				72	A
$V_{SD} (*)$	Forward On Voltage	$I_{SD} = 18\text{ A}$ $V_{GS} = 0$			1.5	V
$t_{rr}$	Reverse Recovery Time	$I_{SD} = 18\text{ A}$ $di/dt = 100\text{ A}/\mu\text{s}$ $V_{DD} = 50\text{ V}$ $T_j = 150\text{ }^\circ\text{C}$ (see test circuit, figure 5)		240		ns
$Q_{rr}$	Reverse Recovery Charge			1.8		$\mu\text{C}$
$I_{RRM}$	Reverse Recovery Current			15		A

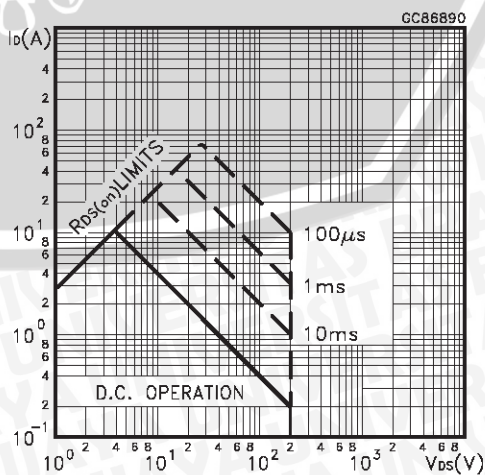
(\*) Pulsed: Pulse duration = 300  $\mu\text{s}$ , duty cycle 1.5 %

( $\bullet$ ) Pulse width limited by safe operating area

Safe Operating Area for TO-220



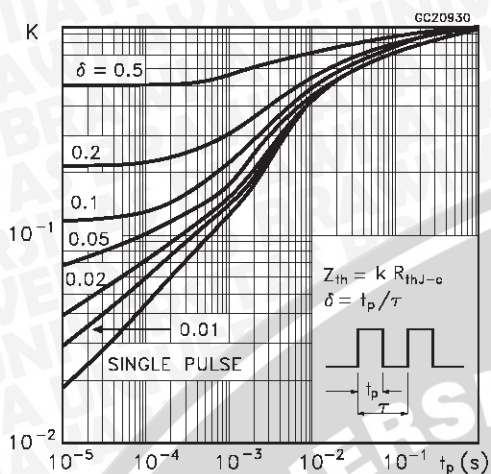
Safe Operating Area for TO-220FP



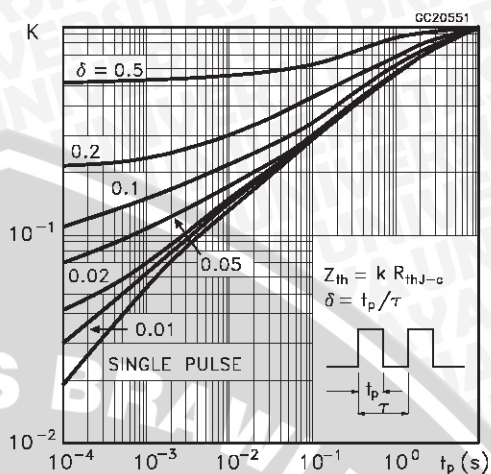


IRF640/FP

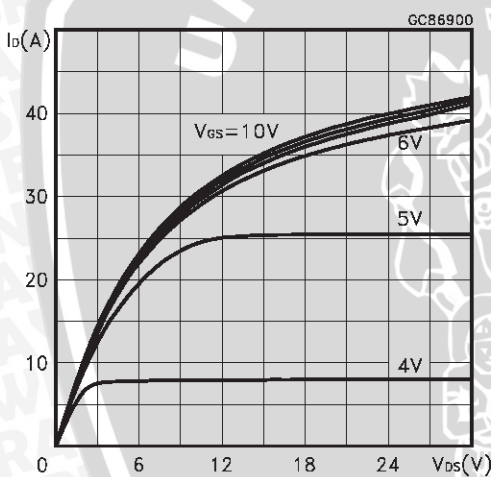
Thermal Impedance for TO-220



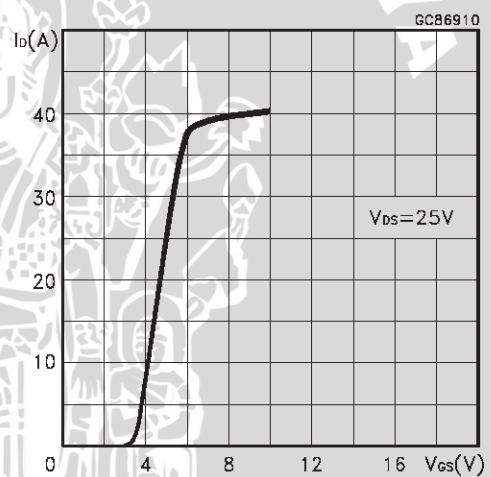
Thermal Impedance for TO-220FP



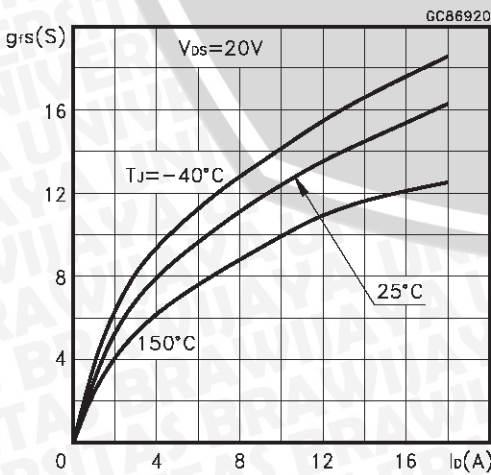
Output Characteristics



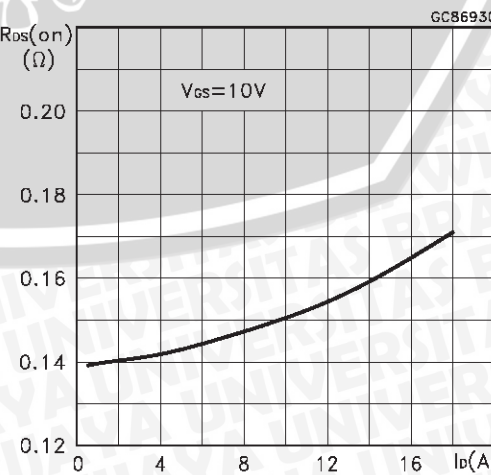
Transfer Characteristics



Transconductance

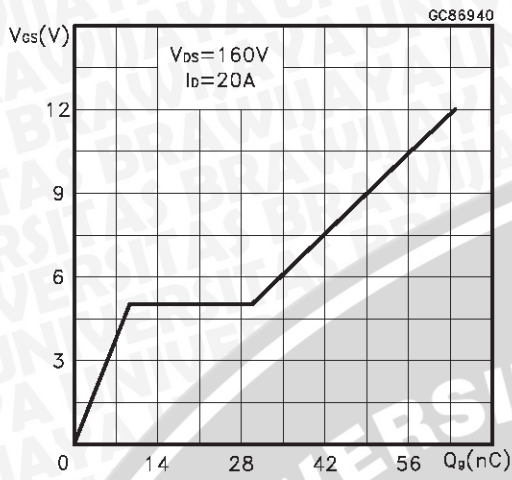


Static Drain-source On Resistance

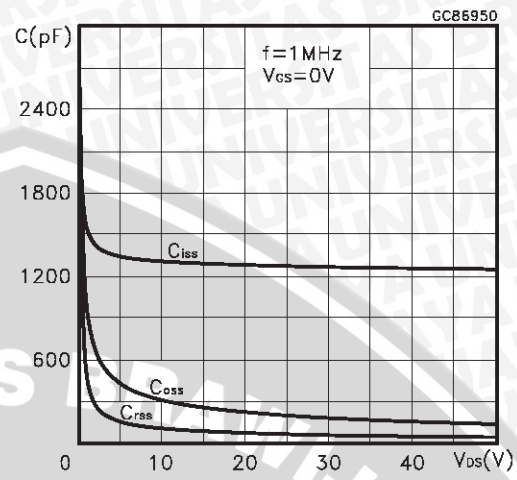




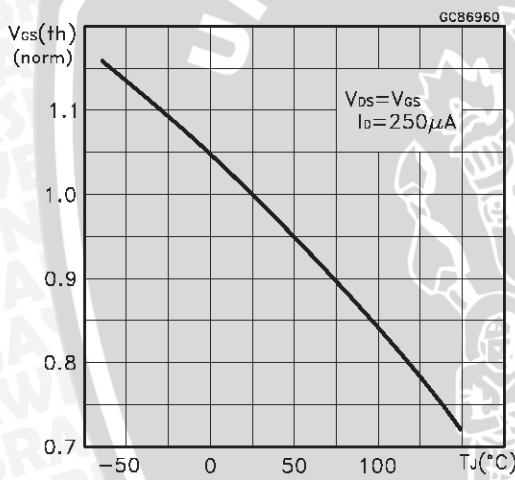
Gate Charge vs Gate-source Voltage



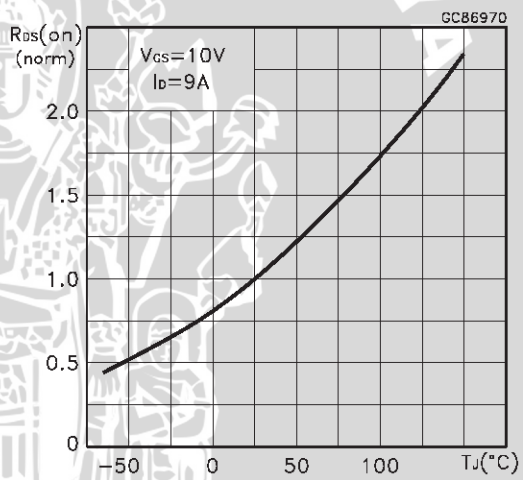
Capacitance Variations



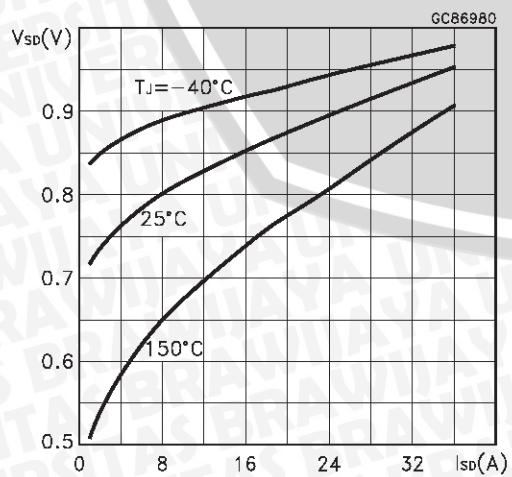
Normalized Gate Threshold Voltage vs Temperature



Normalized On Resistance vs Temperature



Source-drain Diode Forward Characteristics



IRF640/FP

Fig. 1: Unclamped Inductive Load Test Circuit

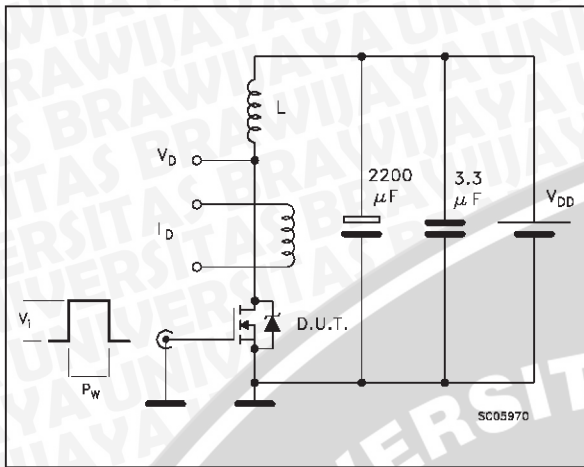


Fig. 2: Unclamped Inductive Waveform

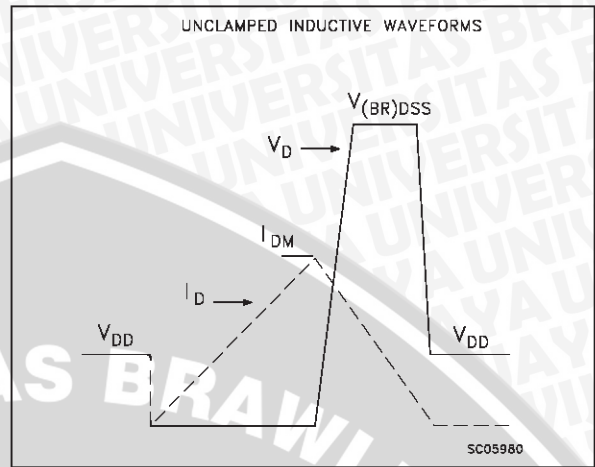


Fig. 3: Switching Times Test Circuits For Resistive Load

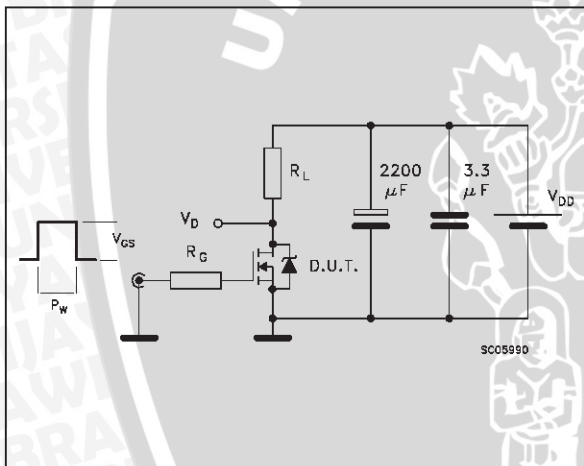


Fig. 4: Gate Charge test Circuit

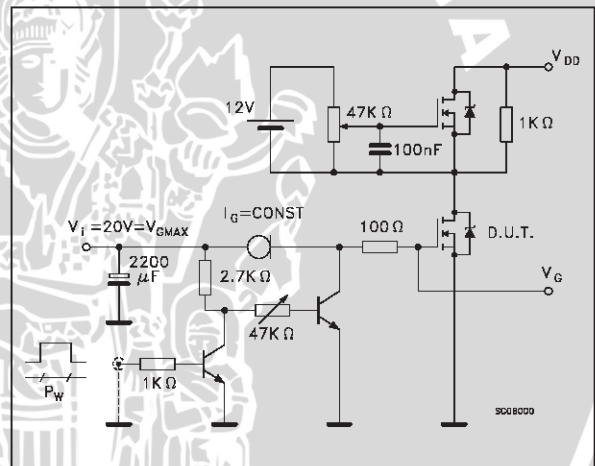
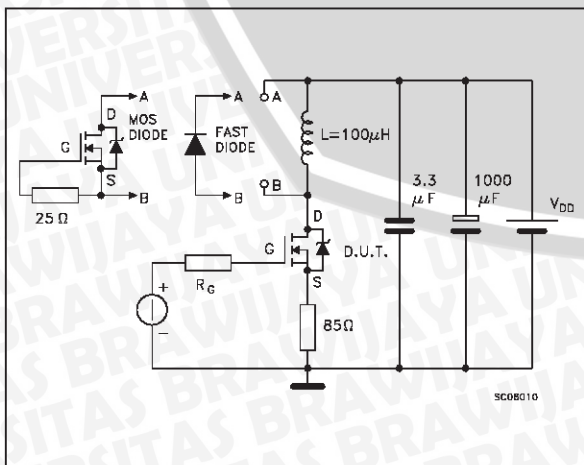
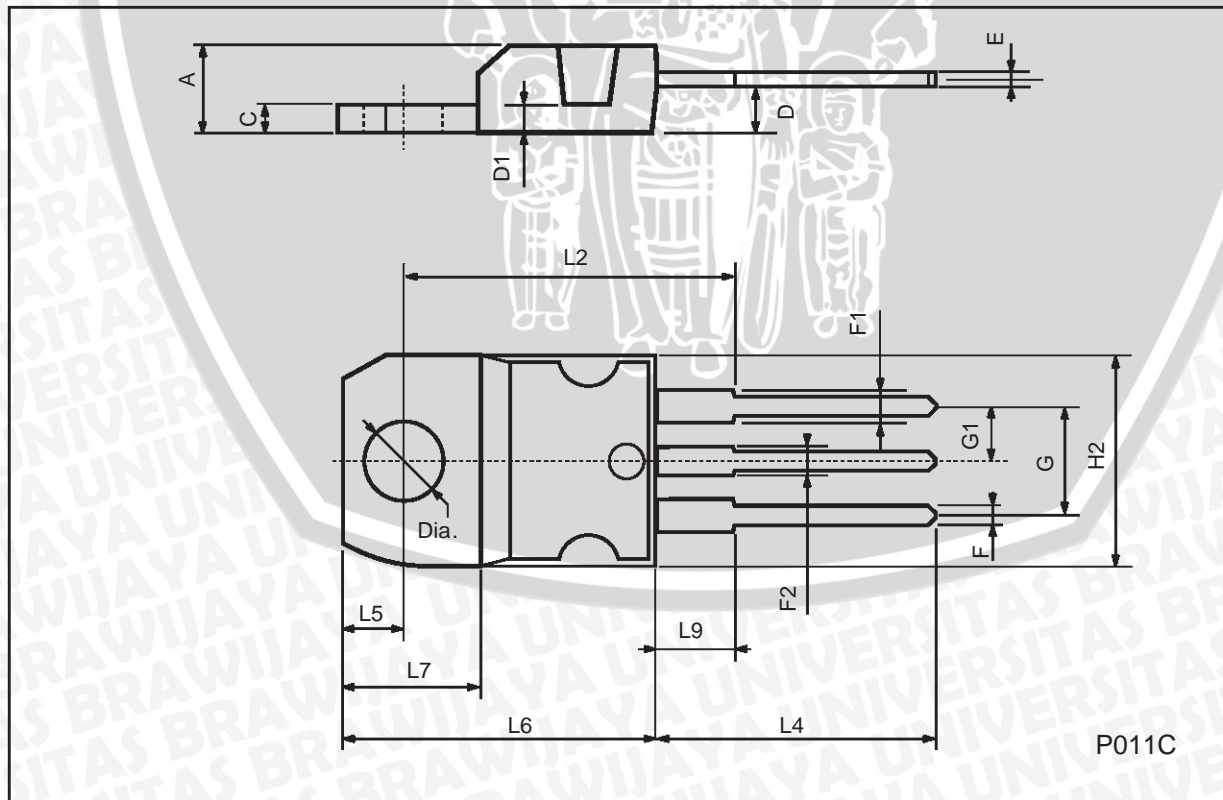


Fig. 5: Test Circuit For Inductive Load Switching And Diode Recovery Times



**TO-220 MECHANICAL DATA**

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	4.40		4.60	0.173		0.181
C	1.23		1.32	0.048		0.051
D	2.40		2.72	0.094		0.107
D1		1.27			0.050	
E	0.49		0.70	0.019		0.027
F	0.61		0.88	0.024		0.034
F1	1.14		1.70	0.044		0.067
F2	1.14		1.70	0.044		0.067
G	4.95		5.15	0.194		0.203
G1	2.4		2.7	0.094		0.106
H2	10.0		10.40	0.393		0.409
L2		16.4			0.645	
L4	13.0		14.0	0.511		0.551
L5	2.65		2.95	0.104		0.116
L6	15.25		15.75	0.600		0.620
L7	6.2		6.6	0.244		0.260
L9	3.5		3.93	0.137		0.154
DIA.	3.75		3.85	0.147		0.151



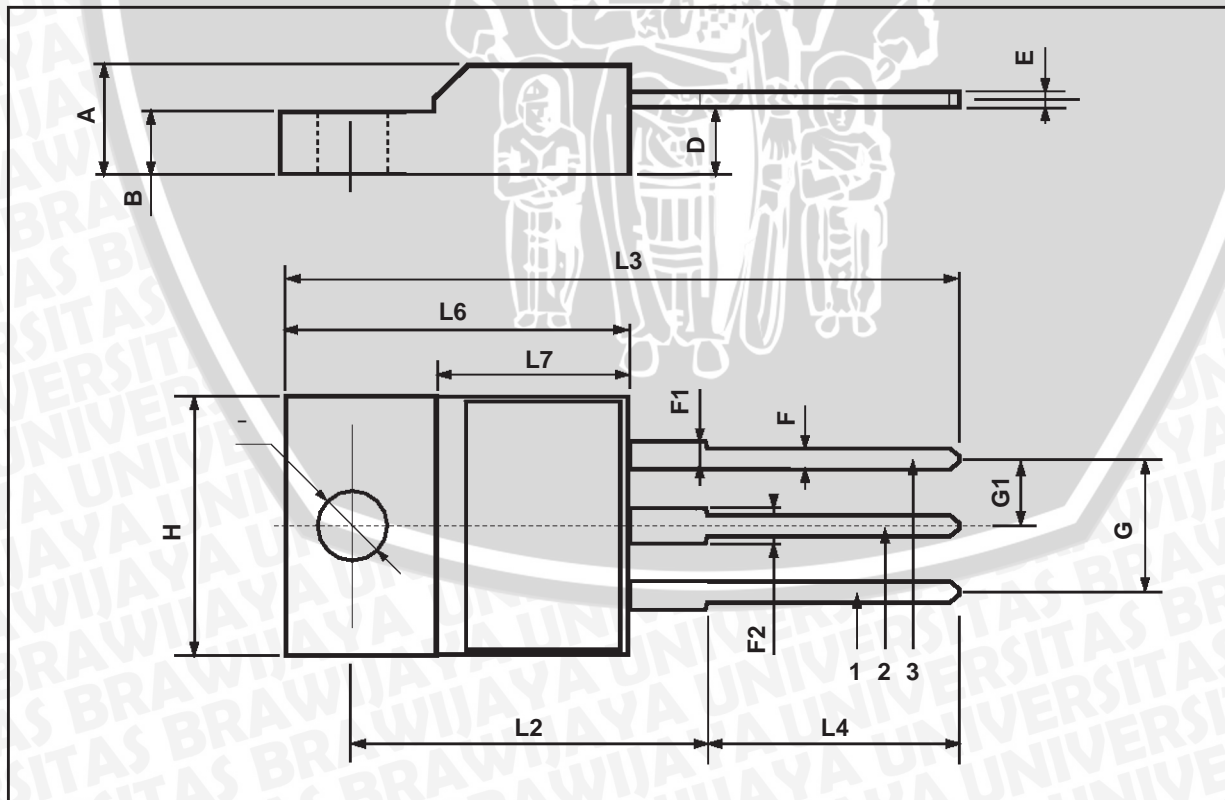




IRF640/FP

TO-220FP MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A	4.4		4.6	0.173		0.181
B	2.5		2.7	0.098		0.106
D	2.5		2.75	0.098		0.108
E	0.45		0.7	0.017		0.027
F	0.75		1	0.030		0.039
F1	1.15		1.7	0.045		0.067
F2	1.15		1.7	0.045		0.067
G	4.95		5.2	0.195		0.204
G1	2.4		2.7	0.094		0.106
H	10		10.4	0.393		0.409
L2		16			0.630	
L3	28.6		30.6	1.126		1.204
L4	9.8		10.6	0.385		0.417
L6	15.9		16.4	0.626		0.645
L7	9		9.3	0.354		0.366
∅	3		3.2	0.118		0.126





Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a trademark of STMicroelectronics

© 1999 STMicroelectronics – Printed in Italy – All Rights Reserved  
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.



<http://www.st.com>



This datasheet has been download from:

[www.datasheetcatalog.com](http://www.datasheetcatalog.com)

Datasheets for electronics components.

