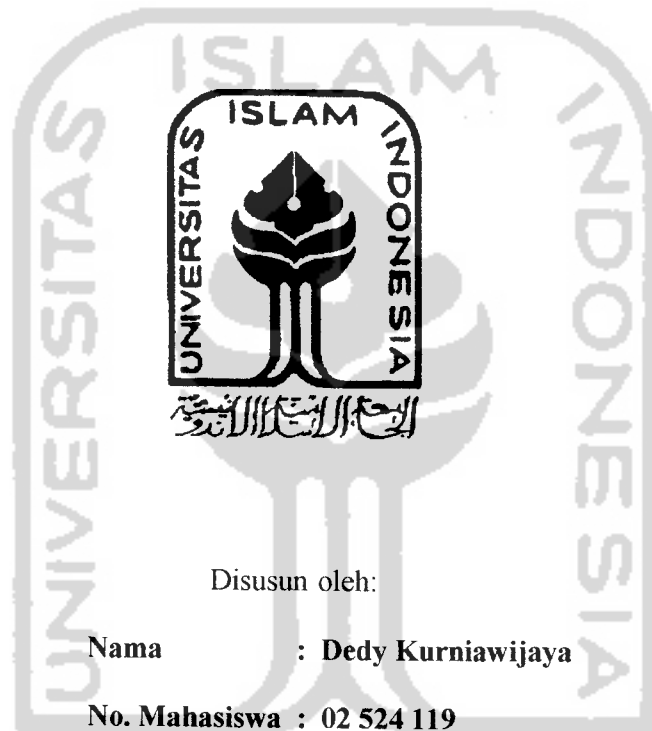


ROBOT PENGIKUT GARIS DENGAN KONTROL

BERBASIS ATURAN

TUGAS AKHIR

Diajukan Sebagai Syarat Untuk Memperoleh Gelar Sarjana Pada
Jurusan Teknik Elektro Fakultas Teknologi Industri
Universitas Islam Indonesia



Disusun oleh:

Nama : Dedy Kurniawijaya

No. Mahasiswa : 02 524 119

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2007


LEMBAR PENGESAHAN PEMBIMBING

ROBOT PENGIKUT GARIS DENGAN KONTROL

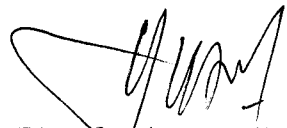
BERBASIS ATURAN



Pembimbing I


(Ir. Hj. Budi Astuti, MT)

Pembimbing II


(Yusuf Aziz Amrullah, ST)

LEMBAR PENGESAHAN PENGUJI
ROBOT PENGIKUT GARIS DENGAN KONTROL
BERBASIS ATURAN

Oleh :

Nama : Dedy Kurniawijaya

No. Mahasiswa : 02 524 119

Telah dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, Juni 2007

Tim Penguji :

Ir. Hj. Budi Astuti, MT

Ketua

Yusuf Aziz Amrullah, ST

Anggota I

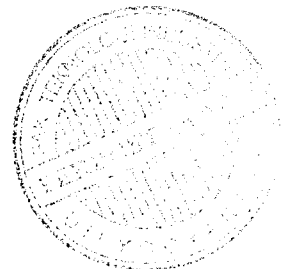
Dwi Ana Ratna Wati, ST

Anggota II

Mengetahui,
Ketua Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

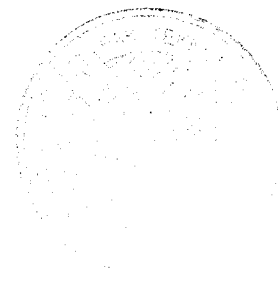


Dedy Kurniawijaya, ST, MSc)



ABSTRAK

Robot pengikut garis merupakan robot otonom yang banyak dirancang baik untuk tujuan penelitian, industri maupun kompetisi. Sesuai dengan namanya, tugas dari robot pengikut garis adalah mampu mengikuti garis pemandu yang diberikan dengan tingkat presisi tertentu. Hal ini akan sangat bermanfaat digunakan untuk melakukan pekerjaan yang tidak mungkin dilakukan manusia karena keterbatasan ruang gerak, situasi dan kondisi yang dapat membahayakan manusia. Penelitian robot pengikut garis dewasa ini umumnya berkonsentrasi pada algoritma perangkat lunak untuk mendapatkan tanggapan robot yang terbaik. Tujuan tugas akhir ini adalah merancang dan mengimplementasikan logika berbasis aturan-aturan yang diimplementasikan sebagai sistem pengontrol dalam suatu robot. Dalam perancangannya robot memiliki masukan dari sensor inframerah yang di kenali sebagai keadaan masukan sensor dan juga untuk mengurangi terjadinya osilasi sehingga didapat hasil yang baik, maka dari kedua masukan tersebut didapat aturan-aturan yang bisa diterapkan sebagai logika pengontrol. Keluaran robot berupa pergerakan motor servo yang dihasilkan dari nilai aturan tersebut sehingga terbentuk nilai PWM. Hasilnya adalah robot mampu mengikuti garis dengan jalur yang telah ditentukan dan dengan keadaan lebar jalur mulai dari selebar 1 cm sampai 9 cm tanpa mengubah kedudukan sensor serta mampu berbelok ke kanan dan kekiri dengan sudut belokkan mulai dari 10^0 sampai 90^0 serta memiliki kecepatan rata-rata sebesar 14,25 cm/s pada jalur lurus.



MOTTO

*Hai jemaah jin dan manusia, jika kamu sanggup menembus
(melintasi) penjuru langit dan bumi, maka lintasilah, kamu tidak
dapat menembusnya melainkan dengan kekuatan*

(QS. Ar Rahman : 33)

Sebuah keberhasilan terdapat 99% kegagalan

(Saichiro Honda)

Non Scholae sed vitae discimus

Kita belajar bukan untuk sekolah tapi untuk hidup

Gitu aja kok Repot...!!!

(Gus Dur)

HALAMAN PERSEMBAHAN

Dengan segala Rasa hormat, Kasih sayang, Kerendahan
hati dan Bakti Kupersembahkan karyaku ini untuk :

*Ayahanda Rjduan Daud dan Ibunda Husnah, beserta seluruh
keluarga besar, terima kasih atas segala Doa dan Kasih sayang
serta kepercayaan yang telah diberikan untuk didi selama ini.*



Banyak Cinta Buat Kalian

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

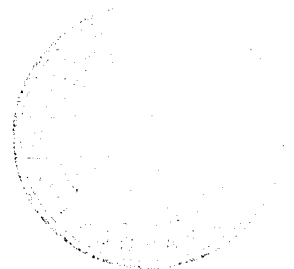
Assalamu 'alaikum Wr.Wb.

Alhamdulillah segala puji syukur kita panjatkan kehadiran Allah SWT karena dengan rahmat dan hidayah-Nya, tugas akhir ini dapat diselesaikan dengan baik. Shalawat serta salam semoga selalu tercurah kepada junjungan kita Nabi Muhammad SAW beserta keluarga dan para sahabat.

Tugas akhir dengan judul “**Robot Pengikut Garis Dengan Kontrol Berbasis Aturan**” disusun sebagai salah satu syarat dalam menyelesaikan program studi untuk memperoleh gelar Sarjana Teknik Elektro pada Fakultas Teknologi Industri, Universitas Islam Indonesia.

Ilmu, pengalaman, dorongan, semangat, motivasi dan doa banyak diberikan kepada penulis selama masa kuliah dan selama proses penyelesaian tugas akhir ini. Pada kesempatan ini, penulis ingin mengucapkan banyak terima kasih yang kepada :

1. Allah SWT dan junjungan kita Nabi Besar Muhammad SAW.
2. Bapak Tito Yuwono ST, MSc., selaku Ketua Jurusan Teknik Elektro Universitas Islam Indonesia
3. Ibu Ir. Hj. Budi Astuti, MT selaku Dosen Pembimbing I.
4. Bapak Yusuf Aziz Amrullah, ST selaku Dosen Pembimbing II.
5. Segenap dosen Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmunya.



6. Ayahanda dan Ibunda beserta kakak dan adik-adikku tercinta. Terima kasih atas kasih sayang yang diberikan, pengorbanan dan dorongan semangat yang selalu mengalir di setiap saat serta doa yang selalu mengiringi langkah ini. Semoga Allah SWT membalas semuanya dan memberikan tempat yang indah disisi-Nya kelak.
7. Mas Tri, Mas Agung, Mas Heri dan seluruh asisten di laboratorium Jurusan Teknik Elektro UII. Terima kasih atas bantuan dan dukungannya.
8. Teman-teman seperjuangan, teman-teman kos, terima kasih atas semua bantuan dan kerjasamanya.
9. Teman-teman KKN Unit R1 khusus angkatan 33, Terima kasih atas semua persahabatan ini, dan semoga kita tetap bersahabat selamanya.
10. Rekan-rekan angkatan 2001, 2002, 2003, 2004 Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia yang telah banyak membantu dalam penyusunan Tugas Akhir ini.
11. Semua pihak yang telah membantu dalam pengerjaan Tugas Akhir ini, yang tidak dapat penulis sebutkan satu persatu.

Dalam penyelesaian Tugas Akhir ini penulis menyadari bahwa masih terdapat kekurangan dan kesalahan baik dalam penulisan maupun bentuk lainnya. Oleh karena itu, kritik dan saran yang bersifat membangun akan penulis terima dengan harapan penulisan berikutnya menjadi lebih baik.

Akhir kata besar harapan penulis agar laporan ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi untuk masa yang akan datang.

Wassalamu'alaikum Wr.Wb

Yogyakarta, Juni 2007

Dedy Kuniawijaya



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
ABSTRAK	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan Penulisan	3
1.5. Sistematika Penulisan	3
BAB II DASAR TEORI	5
2.1. Pengertian Dasar Robot dan Sistem Robotika	5
2.2. Mikrokontroler AVR Atmega8	7
2.3. PWM (<i>Pulse Width Modulation</i>)	10
2.4. Motor Servo	11
2.5. Komparator	13

2.6. IRED (<i>infrared emitting diode</i>)	14
2.7. Fototransistor	15
2.8. Logika	16
2.8.1. Konjungsi	17
2.8.2. Disjungsi	17
2.8.3. Implikasi	18
2.8.4. Biimplikasi	19
BAB III PERANCANGAN SISTEM	20
3.1. Perancangan Sistem	20
3.2. Perancangan <i>Hardware</i> Robot	21
3.3. Bagian sensor garis	22
3.4. <i>Power</i> Suplai	23
3.5. Sistem Minimum Mikrokontroler	24
3.6. Perancangan Aturan Logika Pengontrol	26
3.7. Perancangan Perangkat Lunak	31
BAB IV ANALISIS DAN PEMBAHASAN	33
4.1. Pengujian Sensor Inframerah	33
4.2. Pengujian Terhadap Lebar Jalur	35
4.3. Pengujian Sistem Terhadap Sudut Belok	37
4.4. Pengujian Terhadap Variasi Bentuk Jalur	40
4.4.1. Pengujian bentuk jalur pertigaan dengan sudut 90^0	40

4.4.2. Pengujian bentuk jalur perempatan	41
4.5. Pengujian Kecepatan Saat Menempuh Jalur	41
4.5.1. Pengujian Kecepatan Saat Menempuh Jalur Lurus	42
4.5.2. Pengujian Kecepatan Saat Menempuh Jalur Tertentu	43
BAB V PENUTUP	45
5.1. Kesimpulan	45
5.2. Saran	46
DAFTAR PUSTAKA	
LAMPIRAN	



DAFTAR TABEL

Tabel 2.1. Nilai kebenaran konjungsi	17
Tabel 2.2. Nilai kebenaran disjungsi inklusif	18
Tabel 2.3. Nilai kebenaran disjungsi eksklusif	18
Tabel 3.1. Tabel aturan-aturan pada robot pengikut garis	29
Tabel 4.1. Hasil pengujian sensor inframerah pada output fototransistor	34
Tabel 4.2. Hasil pengujian sensor inframerah pada output IC LM324	34
Tabel 4.3. Hasil pengujian lebar jalur	36
Tabel 4.4. Hasil pengujian robot dengan berbagai variasi sudut belokan ke kanan	38
Tabel 4.5. Hasil pengujian robot dengan berbagai variasi sudut belokan ke kiri	39
Tabel 4.6. Hasil pengujian waktu tempuh jalur lurus	42
Tabel 4.7. Hasil pengujian waktu tempuh jalur tertentu	43

DAFTAR GAMBAR

Gambar 2.1. Konfigurasi Pin ATMEGA8	8
Gambar 2.2 Diagram blok mikrokontroler ATMEGA8	9
Gambar 2.3. Bentuk gelombang PWM	10
Gambar 2.4. Pengkabelan motor servo	12
Gambar 2.5 Bentuk dan bagian <i>gear</i> motor servo	12
Gambar 2.6 Rangkaian komparator	13
Gambar 2.7 Rangkaian infra merah	14
Gambar 2.8 Rangkaian fototransistor	15
Gambar 3.1. Blok diagram perancangan robot	20
Gambar 3.2. Robot tampak samping	21
Gambar 3.3. Susunan <i>optocoupler</i> tampak bawah	22
Gambar 3.4. Rangkaian sensor <i>optocoupler</i>	23
Gambar 3.5. Rangkaian <i>power</i> suplai	24
Gambar 3.6. Sistem minimum robot	25
Gambar 3.7. Rangkaian osilator	25
Gambar 3.8. Daerah masukan keadaan sensor	27
Gambar 3.9. Daerah masukan keadaan sensor sebelumnya	28
Gambar 3.10. Diagram alir program utama	32
Gambar 4.1. Lebar jalur yang digunakan dalam pengujian	35
Gambar 4.2. Bentuk jalur dalam berbagai variasi sudut belokan ke kanan	37

Gambar 4.3.	Bentuk jalur dalam berbagai variasi sudut belokan ke kiri	38
Gambar 4.4.	Bentuk jalur pertigaan dengan sudut 90°	40
Gambar 4.5.	Bentuk jalur perempatan	41
Gambar 4.6.	Jalur untuk pengujian kecepatan jalur lurus	42
Gambar 4.7.	Jalur untuk pengujian kecepatan jalur tertentu	43



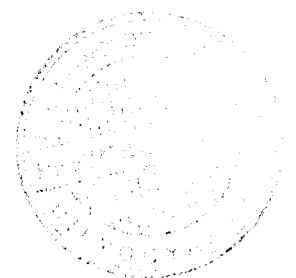
BAB I

PENDAHULUAN

1.1 Latar Belakang

Bidang Robotika merupakan bidang yang saat ini begitu banyak menyumbang dalam hal perkembangan teknologi, karena sangat diminatinya bidang robotika, saat ini ilmuwan dan pakar-pakar teknologi di dunia maupun di Indonesia telah mampu merencanakan berbagai macam bentuk robot. Sebagai contoh, robot yang di buat menyerupai tingkah laku manusia atau yang biasa disebut *humanoid* seperti Asimo yang dikembangkan oleh ilmuwan di Jepang, robot yang menyerupai hewan seperti *insectronic* dan *amphibionic* yang menyerupai serangga dan binatang amfibi, atau robot dengan kegunaan yang lebih sederhana seperti robot industri yang hanya terdiri dari lengan-lengan *manipulator* dan juga robot yang dibuat dengan tujuan khusus, seperti robot yang digunakan untuk situasi-situasi berbahaya, misalnya robot pendeteksi bahan peledak, robot untuk area dengan tingkat radiasi yang membahayakan manusia, ataupun robot yang dibuat untuk tujuan kompetisi robotika.

Robot pengikut garis merupakan robot otonom yang banyak dirancang baik untuk tujuan penelitian, industri maupun kompetisi. Sesuai dengan namanya, tugas dari robot pengikut garis adalah mampu mengikuti garis pemandu yang diberikan dengan tingkat presisi tertentu. Hal ini akan sangat bermanfaat digunakan untuk melakukan pekerjaan yang tidak mungkin dilakukan manusia



karena keterbatasan ruang gerak, situasi dan kondisi yang dapat membahayakan manusia.

Penelitian robot pengikut garis dewasa ini umumnya berkonsentrasi pada algoritma perangkat lunak untuk mendapatkan tanggapan robot yang terbaik. Seperti penerapan algoritma kecerdasan buatan yang saat ini banyak dilakukan agar robot pengikut garis mampu mengikuti garis dengan sangat presisi dan terbaik.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil suatu rumusan masalah sebagai berikut : “Bagaimana membuat sebuah robot yang mampu mengikuti garis (*line follower robot*) dengan menggunakan logika pengontrol serta memiliki kepresisian yang cukup akurat.”

1.3 Batasan Masalah

Dengan adanya batasan masalah, maka dapat lebih menyederhanakan dan mengarahkan penelitian dan pembuatan sistem agar tidak menyimpang dari apa yang diteliti dan dikembangkan.

Batasan-batasannya adalah sebagai berikut :

1. Penelitian difokuskan pada pemrograman suatu *mobile* robot yang mampu mengikuti suatu garis tertentu dengan sistem kontrol berbasis aturan.
2. Pembuatan dan perancangan *mobile* robot .

3. Menggunakan rangkaian elektronika dan mikrokontroler sebagai pengendali *input* dan *output*.
4. Pengujian sistem apakah bekerja sesuai dengan keinginan.

I. 4 Tujuan Penulisan

Adapun tujuan dari penelitian dan perancangan sistem ini adalah untuk mengimplementasikan suatu logika pengontrol dengan menggunakan aturan-aturan tertentu ke dalam sebuah robot agar dapat mengikuti garis yang disediakan, serta memperbaiki sistem yang selama ini dipakai pada robot pengikut garis, yaitu kontrol *on/off*.

1.5 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri dari 5 bab bagian isi laporan, dengan penjelasan bab sebagai berikut

BAB I : Pendahuluan

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II : Landasan Teori

Bab ini memuat teori-teori yang berhubungan dengan penelitian dan juga berisi dasar teori yang berhubungan dengan fungsi atau piranti yang akan digunakan.

BAB III : Perancangan Sistem

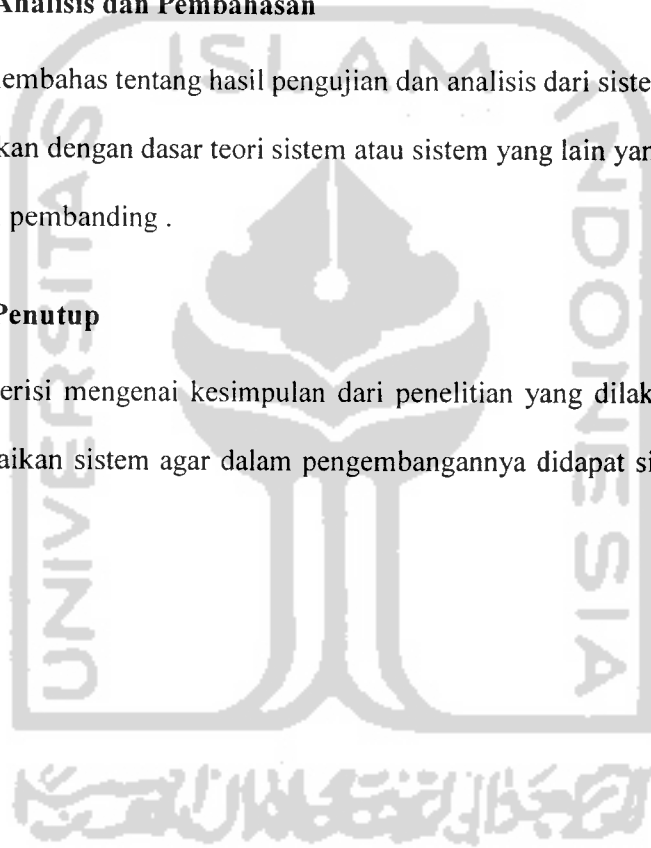
Bagian ini menjelaskan metode-metode perancangan yang digunakan, cara mensimulasikan rancangan dan pengujian sistem yang telah dibuat, pembagian fungsi kerja dalam diagram blok serta berisi lebih terperinci tentang apa yang telah disampaikan pada proposal Tugas Akhir ini.

BAB IV : Analisis dan Pembahasan

Bab ini membahas tentang hasil pengujian dan analisis dari sistem yang dibuat dibandingkan dengan dasar teori sistem atau sistem yang lain yang dapat dijadikan sebagai pembanding .

BAB V : Penutup

Bab ini berisi mengenai kesimpulan dari penelitian yang dilakukan serta saran untuk perbaikan sistem agar dalam pengembangannya didapat sistem yang lebih baik lagi.



BAB II

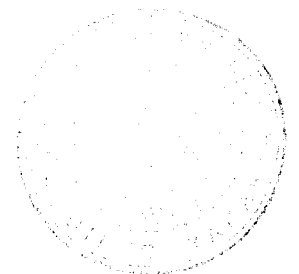
DASAR TEORI

2.1 Pengertian Dasar Robot dan Sistem Robotika

Dunia Robotika secara tidak kita sadari sebenarnya telah berkembang sejak sangat lama, dari zaman Yunani kuno seperti peralatan perang yang mampu melontarkan bom dengan jarak tertentu, hingga sekarang dimana robot sudah terkendali secara otomatis dan memiliki tingkat kecerdasan yang berbeda-beda seperti robot yang dapat berjalan bahkan berlari seperti manusia. Definisi robot secara umum adalah suatu alat multi fungsi otomatis yang dapat diprogram untuk melakukan pekerjaan tertentu. Jadi dalam perkembangannya robot juga dapat diartikan sebagai berikut :

1. Seperangkat peralatan/mesin/*device*.
2. Dapat diprogram.
3. Bekerja/bergerak secara manual dan otomatis.
4. Mampu melaksanakan tugas tertentu sesuai dengan apa yang diprogramkan.

Pengertian lain dari robot menurut beberapa literatur adalah mesin yang dapat diprogram dan mempunyai sifat yang menyerupai manusia. *Robot Institute of America* menyatakan robot sebagai "*a reprogrammable multifunctional manipulator designed to move material, parts, tools or other specialized device through variable programmed motions for the performance of variety of tasks*" atau sebuah robot merupakan manipulator berfungsi ganda yang dapat



diprogramkan kembali serta dirancang untuk menggerakkan bahan, perkakas atau peralatan khusus melalui gerakan-gerakan sesuai program untuk melaksanakan berbagai tugas. Definisi ini mengharuskan sebuah robot mampu untuk di program ulang dan multifungsi yang menyebabkan robot dapat fleksibel. Keunggulan yang didapat dalam menggunakan sistem robot antara lain: kuat, tidak mudah lelah serta tahan terhadap lingkungan yang berbahaya.

Meskipun definisi robot Islam, dunia robotika tak lepas dari kendali (*control*). Meskipun dapat bergerak sendiri, pada akhirnya harus bergerak secara terkendali, baik itu dikontrol oleh manusia, dirinya sendiri maupun oleh robot lainnya.

Sebagaimana kita ketahui, robot berkembang berawal dari aplikasi-aplikasi dunia industri dalam struktur lingkungan yang lebih dikondisikan pada kawasan pabrik. Sehingga robot lebih banyak di desain dalam bentuk yang relatif khas sesuai dengan kebutuhan pabrik seperti manipulator, dan kebanyakan tidak bersifat *mobile*. Namun sekarang kehadiran robot juga banyak ditemui di lingkungan yang bersifat lebih fleksibel seperti misalnya rumah tangga, perkantoran, rumah sakit, daerah eksplorasi, bahkan pada pembangunan kawasan-kawasan berbahaya. Hal ini telah membuat suatu definisi ulang tentang fungsi robot dan telah menempatkan robot sebagai bagian dari kehidupan keseharian sehingga dikenal istilah *human-robot interaction*.

Struktur robot umumnya memiliki:

1. Manipulator yaitu badan atau *body*, pergelangan atau *wrist*, lengan atau *arm*.

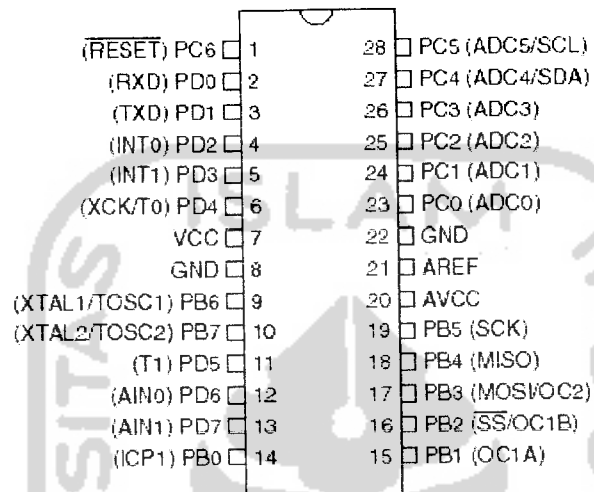
2. *End effector* yaitu penjepit, bor, jarum, pemanas, dan lain-lain.
3. Penggerak (*driver*) yaitu motor, piston, hidrolik, pneumatik
4. Sensor yaitu sensor posisi, kecepatan, torsi, dan lain-lain.
5. Pengendali (*controller*) yaitu komputer, mikroprosesor, mikrokontroler.

2.2 Mikrokontroler AVR ATmega 8

Dalam perancangan robot pengikut garis ini digunakan mikrokontroler keluarga AVR dari ATMEL. Digunakannya mikrokontroler generasi AVR (*Alf and Vegard's RICS processor*) karena mikrokontroler ini memiliki desain sistem dengan teknologi yang memiliki kapabilitas yang amat maju, tetapi dengan biaya ekonomis yang cukup minimal.

Mikrokontroler AVR memiliki arsitektur RICS (*Redyced Instruction Set Computing*) 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dieksekusi dalam satu siklus *clock*. Berbeda dengan mikrokontroler generasi sebelumnya, yaitu MCS51 yang hanya menggunakan arsitektur CISC (*Complex Instruction Set Computing*) yang membutuhkan 12 siklus *clock*. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga ATmega dan AT86RFxx. Dari semua keluarga AVR tersebut pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral* dan fungsinya, sedang dari segi aritektur dan instruksi yang digunakan semua keluarga AVR dapat dikatakan hampir sama.

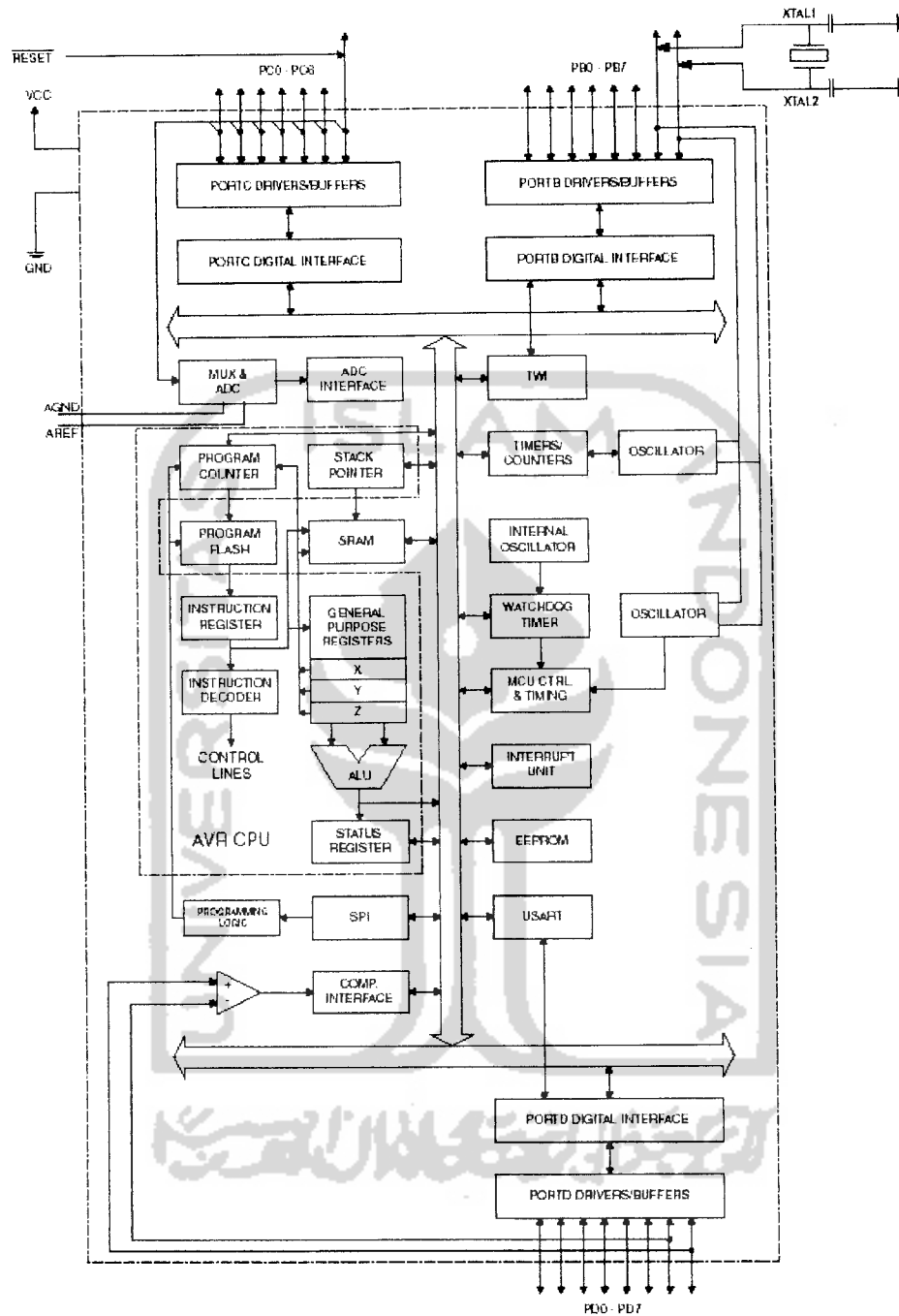
Dalam penggunaannya, keluarga AVR yang digunakan untuk penelitian ini adalah keluarga ATmega yaitu ATMEGA8. Untuk konfigurasi pin ATMEGA8 dapat dilihat pada gambar 2.1 berikut:



Gambar 2.1 Konfigurasi pin ATMEGA8

Konfigurasi pin ATMEGA8 diatas dapat dijelaskan sebagai berikut :

1. VCC adalah pin yang berfungsi sebagai pin masukan catu daya.
2. GND adalah pin *ground*.
3. *Port B* (PB0..PB7) merupakan pin I/O dan pin dengan fungsi khusus yaitu SPI dan fungsi XTAL serta TOSC.
4. *Port C* (PC0..PC6) merupakan pin I/O dan pin dengan fungsi khusus yaitu *internal ADC* dan *reset*.
5. *Port D* (PD0..PD7) merupakan pin I/O dan pin dengan fungsi khusus yaitu interupsi eksternal dan komunikasi serial.



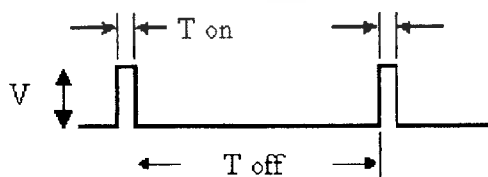
Gambar 2.2 Diagram blok mikrokontroler ATMEGA8

Dari Gambar 2.9 diatas dapat dilihat bahwa ATMEGA8 memiliki bagian sebagai berikut :

1. Saluran I/O sebanyak 23 buah, yaitu *port B*, *port C*, *port D*.
2. ADC 10 bit 6 saluran.
3. SRAM sebesar 1K *Byte*.
4. Memori *flash* sebesar 8K *byte*.
5. 512 *Byte* EEPROM.
6. *Port* antarmuka SPI.
7. *Port* USART untuk komunikasi serial.

2.3 PWM (*Pulse Width Modulation*)

Pulse Width Modulation (PWM) adalah suatu teknik manipulasi dalam pengendalian motor atau perangkat elektronik yang menggunakan arus cukup besar lainnya dengan menggunakan prinsip pemodulasian terhadap sebuah lebar pulsa, sehingga lebar pulsa hidup (*T on*), tidak sama lebarnya dengan pulsa mati (*T off*). Lebar pulsa PWM dinyatakan dalam *Duty Cycle*. Misal, *duty cycle* sebesar 10 % maka berarti bahwa lebar pulsa adalah 1/10 bagian dari satu perioda penuh.



Gambar 2.3 Bentuk gelombang PWM

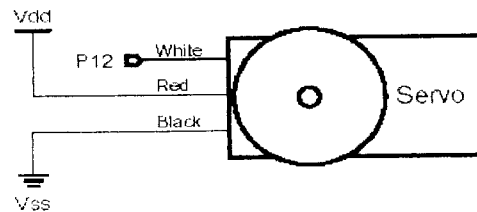
Rumus *duty cycle* :

$$\frac{T_{on}}{T_{on} + T_{off}} \times 100\%$$

2.4 Motor Servo

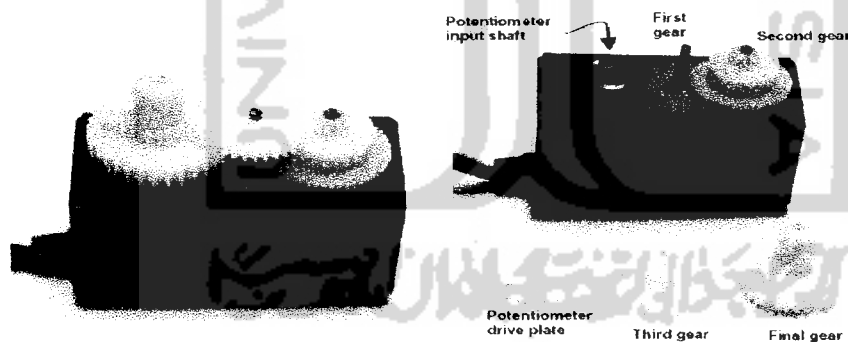
Dalam kamus *Oxford* istilah servo diartikan sebagai sebuah mekanisme yang dapat mengontrol mekanisme yang lebih besar (*a mechanism that controls a larger mechanism*). Motor servo menerima masukan berupa pulsa elektronik. Perputaran motor dipengaruhi dari sinyal pulsa yang diterima. Kecepatan dari motor tergantung dari lebar pulsa yang di kirimkan. Servo merupakan motor yang disertai dengan rangkaian tambahan atau *feedback* posisi untuk menentukan posisi dari sumbu motor servo yang diinginkan. Rangkain *feedback* posisi berupa potensiometer (resistor variabel) yang dapat berputar sesuai dengan putaran *gear* utama motor servo. Rangkaian *feedback* ini akan membandingkan nilai yang diberikan oleh perintah program terhadap nilai aktual potensiometer itu sendiri. Selisih atau *error* dari perbandingan kedua nilai tersebut akan dijadikan rujukan untuk menentukan arah posisinya, sehingga motor berputar untuk mengeliminasi selisih atau *error* menjadi nol, dengan demikian akan tercapailah posisi yang diinginkan.

Motor servo memiliki 3 jenis input yaitu merah untuk *power* (6 volt), hitam untuk *ground* dan putih untuk sinyal pengendalian servo yang dihubungkan dengan mikrokontroler



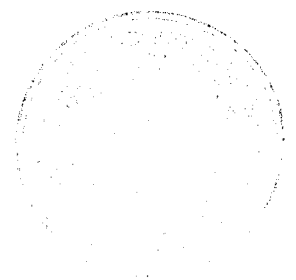
Gambar 2.4 Pengkabelan motor servo

Motor servo yang digunakan untuk robot ini adalah jenis kontinyu. Artinya motor dapat berputar 360 derajat tanpa harus memodifikasi *gear* yang biasa dilakukan pada motor servo standar. Motor servo akan bergerak dengan memberikan nilai lebar pulsa sebesar 1,3 ms maka motor akan berputar searah jarum jam, sedangkan jika lebar pulsa diberikan sepanjang 1,5 ms akan menyebabkan sumbu servo berputar berlawanan dengan arah jarum jam dan bila diberikan lebar pulsa sebesar 1,4 ms maka akan menyebabkan motor servo tetap pada posisinya (tidak berputar).



Gambar 2.5 Bentuk dan bagian *gear* motor servo

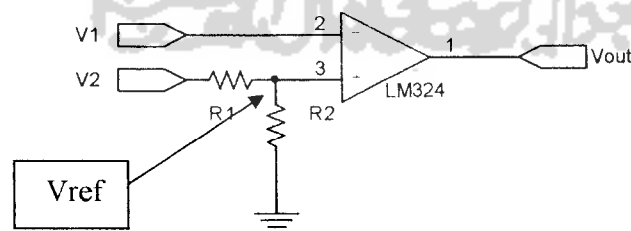
Namun pada kenyataannya, sangat sulit untuk bisa membuat pulsa sebesar 1,4 milidetik atau 1400 mikrodetik secara tepat. Biasanya akan kelebihan atau kekurangan beberapa mikrodetik. Untuk membuat servo diam atau berhenti dapat dilakukan dengan memutar trimpot (*trimmer potentiometer*) yang ada



dirangkaian *feedback*. Tetapi hal itu tentunya akan berpengaruh pada pergerakan ke searah maupun berlawanan arah dengan jarum jam. Pulsa untuk menggerakkan ke searah maupun berlawanan arah jarum jam akan menyesuaikan dengan putaran trimpot yang terdapat pada servo.

2.5 Komparator

Komparator merupakan suatu penguat operatif yang digunakan untuk membandingkan dua tegangan masukan yaitu V_1 dan V_2 sesuai dengan kebutuhannya. Prinsip kerja dari sebuah komparator adalah sebagai berikut: Jika tegangan masukan V_1 lebih kecil dari tegangan referensi maka keluaran dari komparator dalam keadaan *high* (V_{sat+}), sedangkan apabila tegangan masukan lebih besar dari tegangan referensi maka keluaran komparator akan bernilai *low* (V_{sat-}). Selama *input* mendapat masukan yang tegangannya lebih kecil dari tegangan referensi, maka keluaran dari komparator akan selalu dalam keadaan *high*. Tetapi apabila nilai tegangan masukan mengalami kenaikan sampai melebihi nilai tegangan referensi maka keluaran komparator akan berubah menjadi *low*.



Gambar 2.6 Rangkaian komparator

Dimana :

$$V_{ref} = \frac{R_2}{R_1 + R_2} \times V_2$$

2.6 IRED (*Infrared Emitting Diode*)

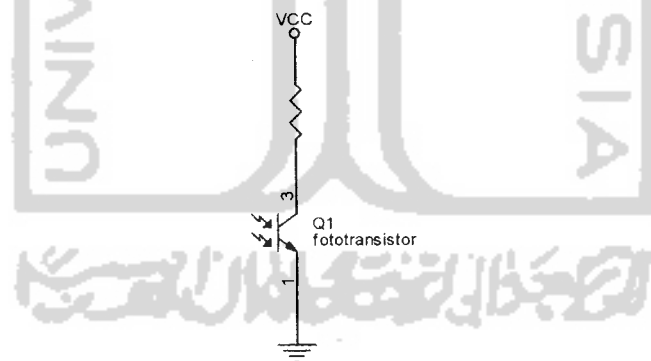
IRED adalah jenis dioda yang dapat memancarkan cahaya infra merah. Cahaya infra merah ini tidak dapat dilihat oleh mata. Penerapan dari infra merah ini biasanya sebagai perantara komunikasi yang berada pada ruangan yang relatif kecil dan tertutup. Pancaran cahaya infra merah tidak dapat menembus benda-benda padat, tetapi apabila cahaya infra merah mengenai suatu benda padat maka akan memantul. Penerapan infra merah paling banyak digunakan pada *remote control* alat-alat elektronik seperti televisi, *air conditioner*, *radio tape* dan lain sebagainya. Pada pemasangan IRED hampir sama dengan pemasangan LED biasa. Hanya dengan memerlukan tambahan resistor sebagai pembatas arus pada kaki anoda dan kaki katoda dihubungkan dengan *ground*.



Gambar 2.7 Rangkaian infra merah

2.7 Fototransistor

Fototransistor adalah sebuah jenis transistor yang cara kerjanya berhubungan dengan cahaya. Pada umumnya cahaya yang diperlukan untuk mengaktifkan fototransistor adalah cahaya infra merah. Fototransistor jika dilihat dari cara kerjanya hampir sama dengan sebuah saklar cahaya, apabila sebuah fototransistor terkena cahaya infra merah maka kaki kolektor-*emitter* akan tersambung dan berfungsi sebagai saklar yang terhubung singkat. Akan tetapi apabila fototransistor tidak terkena cahaya infra merah atau hanya terkena cahaya biasa maka kaki kolektor-*emitter* tidak terhubung. Dalam penerapannya fototransistor sangatlah mudah dalam merangkainya, hanya dengan menambahkan sebuah resistor pada kaki kolektor sebelum dihubungkan dengan V_{cc} , sedangkan kaki emitter dihubungkan dengan *ground*. Sedangkan outputnya diambil dari kaki kolektor.



Gambar 2.8 Rangkaian fototransistor

2.8 Logika

Logika sangat penting dalam aspek kehidupan manusia. Dengan menggunakan logika kita lebih efektif dalam mengenal dan menghindari kesalahan penalaran. Adapun aplikasi logika salah satunya di bidang teknologi informasi dimana di bidang ini kita mengenal perangkat komputer serta bahasa pemrogramannya. Perangkat komputer tersebut biasanya disebut *hardware* dan bahasa pemrograman biasa disebut *software*, dan baik *software* ataupun *hardware* banyak menggunakan prinsip-prinsip logika.

Dalam mengungkapkan sebuah logika maka diperlukan pernyataan, Pernyataan adalah suatu kalimat tertutup yang mempunyai satu nilai kebenaran yaitu benar atau salah. Akan tetapi tidak memiliki nilai kebenaran sekaligus keduanya, contoh : “hasil kali 3 dan 6 sama dengan 18” dan “semua bilangan prima adalah bilangan ganjil”. Pernyataan pertama bernilai benar, sedang pernyataan kedua bernilai salah. Sebuah pernyataan merupakan suatu kalimat, sedang suatu kalimat belum tentu pernyataan. Banyak pernyataan yang bisa langsung diterima kebenarannya, seperti misalnya pernyataan : “ada kehidupan di bulan atau tidak ada kehidupan di bulan” adalah benar, meskipun kita tidak tahu apakah ada orang yang pernah membuktikan atau tidak.

Dalam menerapkan suatu logika, terdapat beberapa operasi yang sering ditemui, seperti operasi negasi yaitu penyangkalan atau ingkaran. Ingkaran dari suatu pernyataan merupakan pernyataan baru dengan nilai kebenaran berlawanan dengan nilai pernyataan sebelumnya. Ingkaran dari suatu pernyataan diperoleh

dengan cara memberi tambahan kata “tidak benar” di awal kalimat, atau dapat pula memberi tambahan kata “tidak” atau “bukan” pada pernyataan tersebut.

Operasi lainnya adalah operasi biner. Operasi biner adalah operasi antara dua pernyataan, operasi biner antara dua pernyataan tunggal akan menghasilkan suatu pernyataan majemuk. Ada empat macam operasi biner, yaitu konjungsi, disjungsi, implikasi dan biimplikasi.

2.8.1. Konjungsi

Konjungsi merupakan operasi biner pada dua pernyataan yang menggunakan kata “dan” dinotasikan dengan lambang “ \wedge ”. Jadi konjungsi antara pernyataan p dan q ditulis $p \wedge q$. Nilai kebenaran dari operasi konjungsi dinyatakan seperti pada tabel berikut :

Tabel 2.1 Nilai kebenaran konjungsi

p	q	$p \wedge q$
Benar	Benar	Benar
Benar	Salah	Salah
Salah	Benar	Salah
Salah	Salah	Salah

2.8.2. Disjungsi

Disjungsi adalah operasi biner pada dua pernyataan yang menggunakan kata penghubung “atau”. Ada dua jenis disjungsi, yaitu disjungsi inklusif yang menggunakan lambang “ \vee ” dan disjungsi eksklusif yang menggunakan lambang

“ \vee ”. Untuk tabel kebenaran disjungsi inklusif atau disjungsi eksklusif dapat dilihat pada tabel berikut :

Tabel 2.2 Nilai kebenaran disjungsi inklusif

p	q	$p \vee q$
Benar	Benar	Benar
Benar	Salah	Benar
Salah	Benar	Benar
Salah	Salah	Salah

Tabel 2.3 Nilai kebenaran disjungsi eksklusif

p	q	$p \wedge q$
Benar	Benar	Salah
Benar	Salah	Benar
Salah	Benar	Benar
Salah	Salah	Salah

2.8.3. Implikasi

Pernyataan yang berbentuk “jika p maka q” dinamakan pernyataan implikasi atau pernyataan kondisional. Operasi implikasi menggunakan lambang “ \Rightarrow ” sehingga pernyataan “jika p maka q” dinotasikan oleh “ $p \Rightarrow q$ ”. Pada pernyataan implikasi, p disebut anteseden atau hipotesis atau alasan atau sebab. Adapun q disebut konsekuen atau konklusi atau kesimpulan atau akibat.

2.8.4. Biimplikasi

Biimplikasi merupakan operasi dua pernyataan dengan menggunakan lambang " \Leftrightarrow " yang berarti "jika dan hanya jika". Pernyataan "p jika hanya jika q" dinotasikan dengan " $p \Leftrightarrow q$ ". Biimplikasi " $p \Leftrightarrow q$ " mempunyai arti yaitu "jika p maka q dan jika q maka p". Pernyataan biimplikasi " $p \Leftrightarrow q$ " bernilai benar jika p dan q memiliki nilai yang sama, sedangkan jika nilai kebenaran p dan q berbeda, maka nilai kebenaran $p \Leftrightarrow q$ bernilai salah.

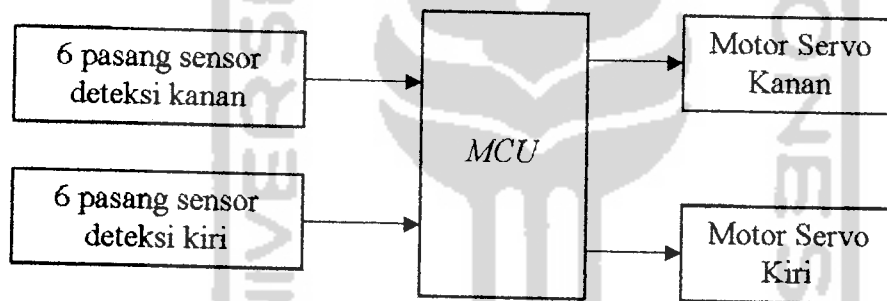


BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem

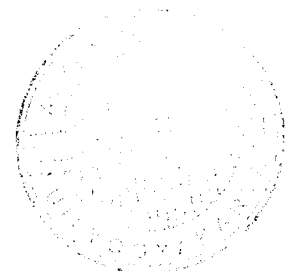
Secara umum sistem robot pada penelitian ini terdiri dari sistem mikrokontroler ATMEGA8 sebagai kendali utama. Masukan atau *input* berupa sensor infra merah dan keluaran atau *output* berupa penggerak motor servo. Blok diagram sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok diagram perancangan robot

Pada diagram blok diatas, maka dapat dijelaskan sebagai berikut

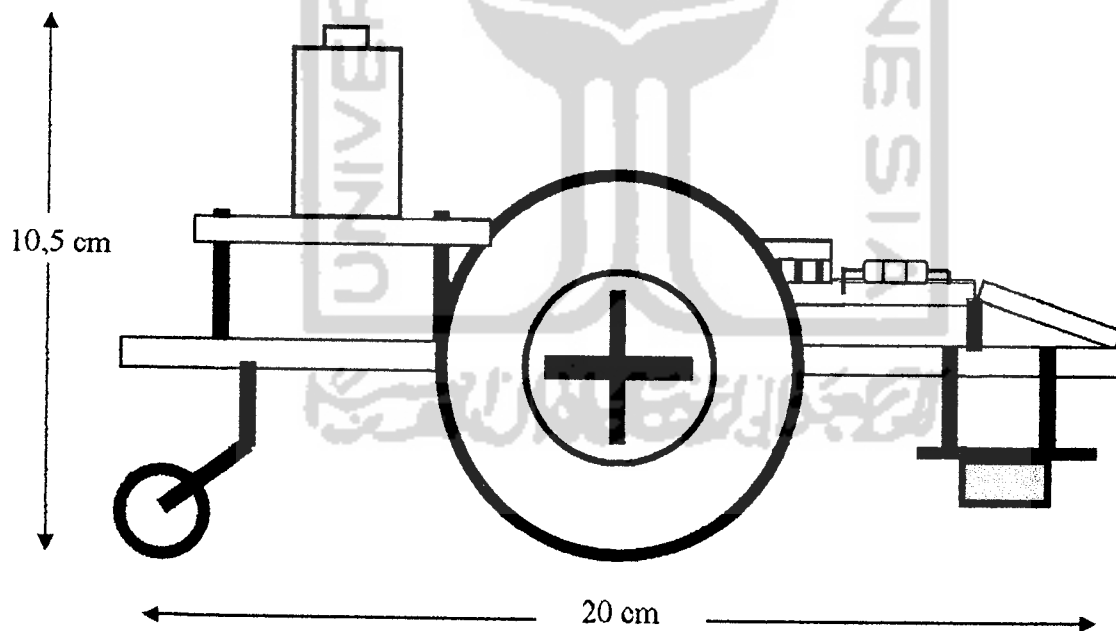
1. *Sensor*, digunakan untuk mengenali garis. Sensor yang digunakan adalah infra merah dan fototransistor.
2. *Microcontroller Unit (MCU)*, merupakan komponen pengolah data masukan dari sensor, penerapan aturan-aturan atau logika yang digunakan dan juga untuk mengirimkan data ke motor servo.



3. Motor, sebuah alat yang mengubah besaran listrik ke dalam wujud mekanis berupa rotasi. Dalam penelitian digunakan motor servo yang dikendalikan oleh PWM.

3.2 Perancangan *Hardware* Robot

Robot mobil dirancang sekecil mungkin, agar memudahkan robot berjalan ke segala arah, serta memperkecil *tracking error*. Ukuran robot ini adalah panjang 20 cm lebar 16 cm dan mempunyai tinggi 10,5 mm. Gambar perancangan robot dapat dilihat pada Gambar 3.2 dibawah ini :



Gambar 3.2 Robot tampak samping

Bahan – bahan yang digunakan dalam perancangan robot ini adalah sebagai berikut:

1. Acrylic 2 mm
2. 2 buah motor servo *parallax*
3. 3 buah roda, 2 buah roda penggerak robot terbuat dari kepingan CD diameter 80 mm.
4. Komponen elektronik
 - a. 12 pasang sensor *optocoupler*
 - b. IC, resistor, dioda, kapasitor
 - c. Mikrokontroler Atmel ATMEGA8.
 - d. PCB, kabel dan baterai.

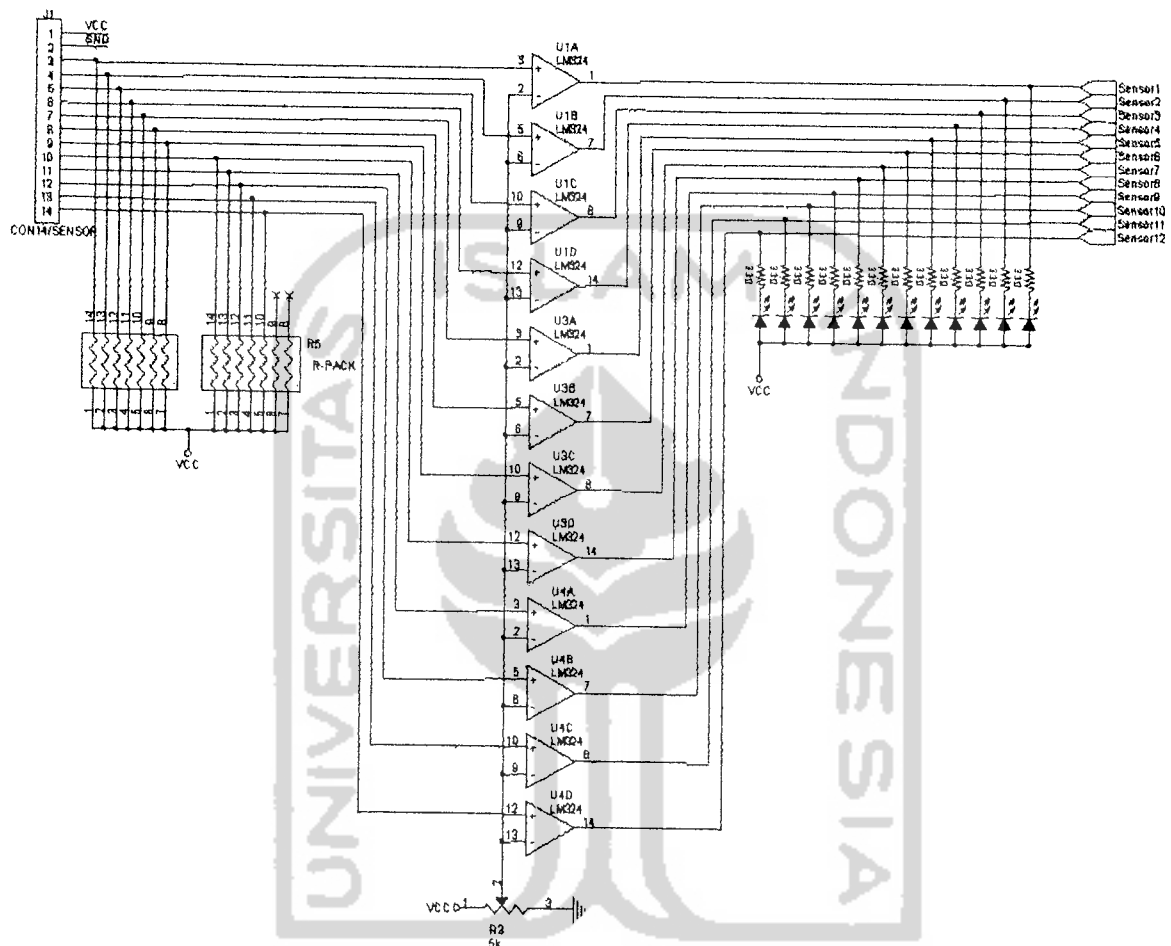
3.3. Bagian Sensor Garis

Sensor yang digunakan dalam perancangan robot ini adalah sensor *optocoupler* jenis pantul yang terdiri dari LED Infra merah sebagai pemancar dan *phototransistor* sebagai penerima. Dalam mendeteksi garis digunakan 12 pasang *optocoupler*, dengan susunan sebagai berikut :



Gambar 3.3 Susunan sensor *optocoupler* tampak bawah

Kemudian agar sensor dapat membedakan antara garis hitam dan putih, maka ditambah rangkaian komparator, seperti terlihat pada Gambar 3.4 berikut :

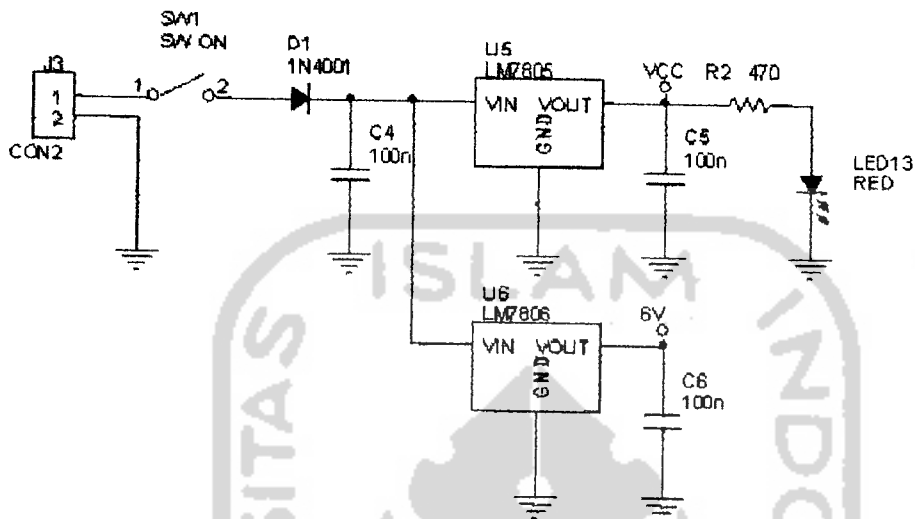


Gambar 3.4 Rangkaian sensor *optocoupler*

3.4 Power Suplai

Robot mendapatkan sumber tegangan dari baterai. Baterai yang digunakan adalah jenis baterai yang dapat diisi ulang atau *charge*. Tegangan dari baterai didistribusikan ke 3 bagian, yaitu rangkaian sensor *optocoupler* sebesar 5 volt, sistem minimum mikrokontroler 5 volt dan motor servo sebesar 6 volt.

8	PC5(ADC5)
7	PC4(ADC4)
6	PC3(ADC4)
5	PC2(ADC2)
4	PC1(ADC1)
3	PC0(ADCC)
2	PD2 (INT0)
1	PD3 (INT1)
0	PD4 (T0)
1	PD5 (T1)
2	PD6 (AINC)
3	PD7 (AIN')
9	PB6 (OSC)
0	PB7 (OSC)

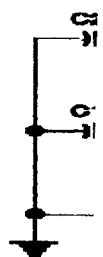


Gambar 3.5 Rangkaian power suplai

3.5 Sistem Minimum Mikrokontroler

Pengendalian robot dilakukan melalui sistem minimum mikrokontroler ATMEGA8 yang terdiri dari rangkaian osilator, *reset* dan catu daya seperti terlihat pada Gambar 3.6.

Port ATMEGA8 yang digunakan sebagai masukan sensor *optocoupler* adalah *port C*, yaitu *port C.0* sampai *port C.5* dan *port D*, yaitu *port D.2* sampai *D.7* dimana semuanya berjumlah 12 buah *port* untuk 12 pasang sensor yang digunakan, sedangkan dua buah keluaran digunakan yang digunakan untuk motor servo, yaitu *port B.1* untuk motor servo 1 dan *port B.2* untuk motor servo 2.



Gambar

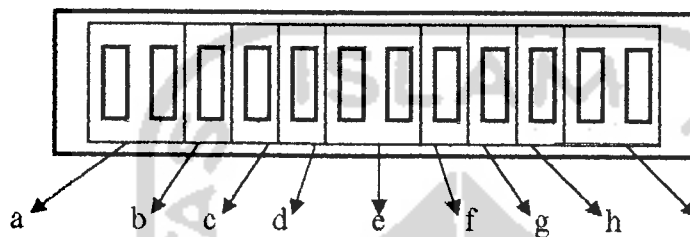
Sedangkan rangkaian *reset* berfungsi untuk menjaga agar pin RST mikrokontroler selalu berlogika rendah saat mikrokontroler mengeksekusi program. Untuk dapat mengeksekusi program dari awal program (alamat 00H) maka mikrokontroler akan direset secara otomatis saat catu daya pertama kali dihidupkan dimana untuk reset otomatis ini dilakukan oleh C dan R (*Power Off Reset*).

Sistem minimum mikrokontroler ATMEGA8 digunakan sebagai pengolah dan sekaligus sebagai unit penyimpan program. PWM (*Pulse Width Modulation*) akan dibangkitkan melalui program dengan menggunakan mikrokontroler. PWM ini yang akan mengatur apakah robot mobil berjalan maju, belok, mundur atau berhenti.

3.6 Perancangan Aturan Logika Pengontrol

Untuk mengendalikan robot digunakan susunan aturan-aturan yang dirancang agar dapat mengikuti garis dengan baik, ide dasar pembuatan logika aturan tersebut yaitu mengenali nilai masukkan sensor fototransistor apakah sensor mendeteksi nilai kecil, sedang, besar atau sangat besar. Setiap nilai tersebut menganalogikan seberapa jauh sensor dari nilai nol, dimana nol diartikan juga sebagai jalur lurus. Untuk mengenali area kecil, sedang, besar atau sangat besar maka data terluar dari sensor fototransistor yang mendeteksi garis hitam dijadikan sebagai keanggotaan masukan. Sehingga apabila sensor mendeteksi garis hitam di dekat daerah nol, maka hal tersebut dikenali sebagai area kecil sedangkan apabila sensor paling jauh dari daerah nol maka hal tersebut dikenali sebagai area sangat

besar. Nilai yang dideteksi oleh sensor juga menyatakan bentuk jalur yang dilewati robot, apakah robot menemukan belokkan kecil, sedang, besar atau sangat besar. Masukan ini di kenali sebagai himpunan masukan sensor.



Gambar 3.8 Daerah masukan kadaan sensor

Keterangan :

a = Masukan sensor daerah -sangat besar (-SBS).

b = Masukan sensor daerah -besar (-BS).

c = Masukan sensor daerah -sedang (-SD).

d = Masukan sensor daerah -kecil (-KC).

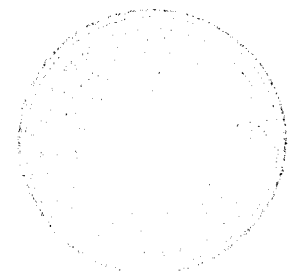
e = Masukan sensor daerah lurus (NOL).

f = Masukan sensor daerah + kecil (+KC).

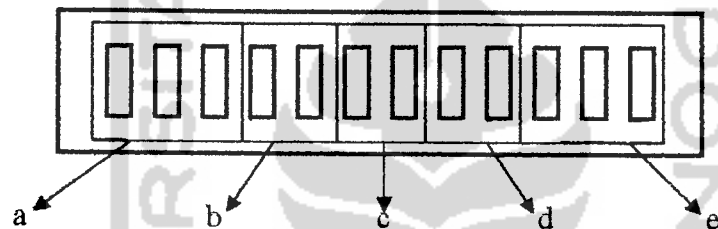
g = Masukan sensor daerah + sedang (+SD).

h = Masukan sensor daerah + besar (+BS).

i = Masukan sensor daerah + sangat besar (+SBS).



Dalam perancangan pada robot juga diterapkan satu buah kendali masukkan lagi selain masukan sensor, yaitu keadaan sensor sebelumnya (KSS). Ini dimaksudkan untuk mengurangi terjadinya osilasi (*tracking error*). Berbeda dengan masukan sensor yang memiliki 8 daerah masukan, yaitu 4 area sensor kiri, 4 area kanan dan kondisi area nol, maka pada masukan keadaan sensor sebelumnya (KSS) hanya memiliki 5 kondisi area, yaitu 2 area kiri, 2 kondisi area kanan dan kondisi area nol.



Gambar 3.9 Daerah masukan keadaan sensor sebelumnya

Keterangan :

a = Masukan keadaan sensor sebelumnya -besar (- BS).

b = Masukan keadaan sensor sebelumnya -kecil (- KC).

c = Masukan keadaan sensor sebelumnya lurus (NOL).

d = Masukan keadaan sensor sebelumnya + kecil (+ KC).

e = Masukan keadaan sensor sebelumnya + besar (+ BS).

Saat kedua kendali masukan didapatkan maka dibuatlah suatu logika aturan pengontrol dengan membandingkan keadaan sensor yang didapat dengan

keadaan daerah sensor sebelumnya, adapun kondisi aturan-aturan tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Tabel aturan-aturan pada robot pengikut garis

		Masukan sensor								
		-SBS	-BS	-SD	-KC	NOL	+KC	+SD	+BS	+SBS
KSS	-BS	-SBS	-BS	-BS	-SD	-BS	+BS	+SBS	+SBS	+SBS
	-KC	-BS	-SD	-SD	-KC	-KC	+SD	+BS	+SBS	+SBS
	NOL	-SBS	-BS	-SD	-KC	NOL	+KC	+SD	+BS	+SBS
	+KC	-SBS	-SBS	-BS	-SD	+KC	+KC	+SD	+SD	+BS
	+BS	-SBS	-SBS	-SBS	-BS	+BS	+SD	+BS	+BS	+SBS

Keterangan :

- SBS = Kondisi keluaran - sangat besar dengan nilai -64
- BS = Kondisi keluaran - besar dengan nilai -48
- SD = Kondisi keluaran - sedang dengan nilai -32
- KC = Kondisi keluaran - kecil dengan nilai -16
- NOL = Kondisi keluaran - lurus dengan nilai 0
- + KC = Kondisi keluaran kecil dengan nilai 16
- + SD = Kondisi keluaran sedang dengan nilai 32
- + BS = Kondisi keluaran besar dengan nilai 48
- + SBS = Kondisi keluaran sangat besar dengan nilai 64

Setiap aturan keluaran diatas diperoleh dengan cara meng-implikasikan keadaan sensor dengan keadaan sensor sebelumnya, contoh : apabila keadaan sensor bernilai besar (+BS) dan kondisi sensor sebelumnya bernilai kecil (+KC),

maka implikasinya sebagai berikut : “JIKA keadaan sensor bernilai besar (+BS) DAN kondisi sensor sebelumnya bernilai kecil (+KC) MAKA kondisi keluaran bernilai sedang (+SD)”.

Dari aturan-aturan diatas, dapat diketahui bahwa terdapat 45 aturan yang mungkin terpenuhi selama robot mengikuti garis tertentu, dimana dari aturan tersebut diketahui juga bahwa dalam penyusunan logika pengontrol, robot hanya memiliki 9 kondisi keluaran, dan masing-masing kondisi tersebut memiliki nilai yang telah ditentukan sebelumnya. Nilai-nilai yang diberikan nantinya akan berkaitan dengan penulisan data PWM ke motor servo kanan dan servo kiri.

Untuk langkah selanjutnya yaitu memberikan nilai tersebut ke motor servo. Motor servo dijalankan dengan cara diberikan data PWM berupa lebar pulsa, apabila robot ingin berjalan maju maka servo kanan harus berputar berlawanan dengan arah jarum jam ($1401 \mu\text{s}$ sampai $2000 \mu\text{s}$) dan motor servo kiri berputar searah jarum jam ($800 \mu\text{s}$ sampai $1399 \mu\text{s}$). Saat diketahui keadaan masukan sensor dan keadaan sensor sebelumnya, maka didapatlah kondisi keluaran dengan nilai tertentu, kemudian program dibuat untuk menuliskan data PWM ke servo kanan sebesar $((80\mu\text{s} - \text{nilai kondisi keluaran}) + 1400\mu\text{s})$ dan memberikan servo kiri PWM sebesar $(- (80\mu\text{s} + \text{nilai kondisi keluaran}) + 1400\mu\text{s})$. Sedangkan nilai $80\mu\text{s}$ adalah nilai tetapan yang membuat servo tetap berputar pelan, jadi tidak ada servo yang berhenti berputar.

Sebagai contoh, saat robot berjalan diketahui bahwa keadaan sensor ada pada kondisi –sangat besar(-SBS), dengan keadaan sensor sebelumnya ada pada kondisi –besar(-BS), maka aturan yang dipakai dapat dilihat pada tabel kolom ke

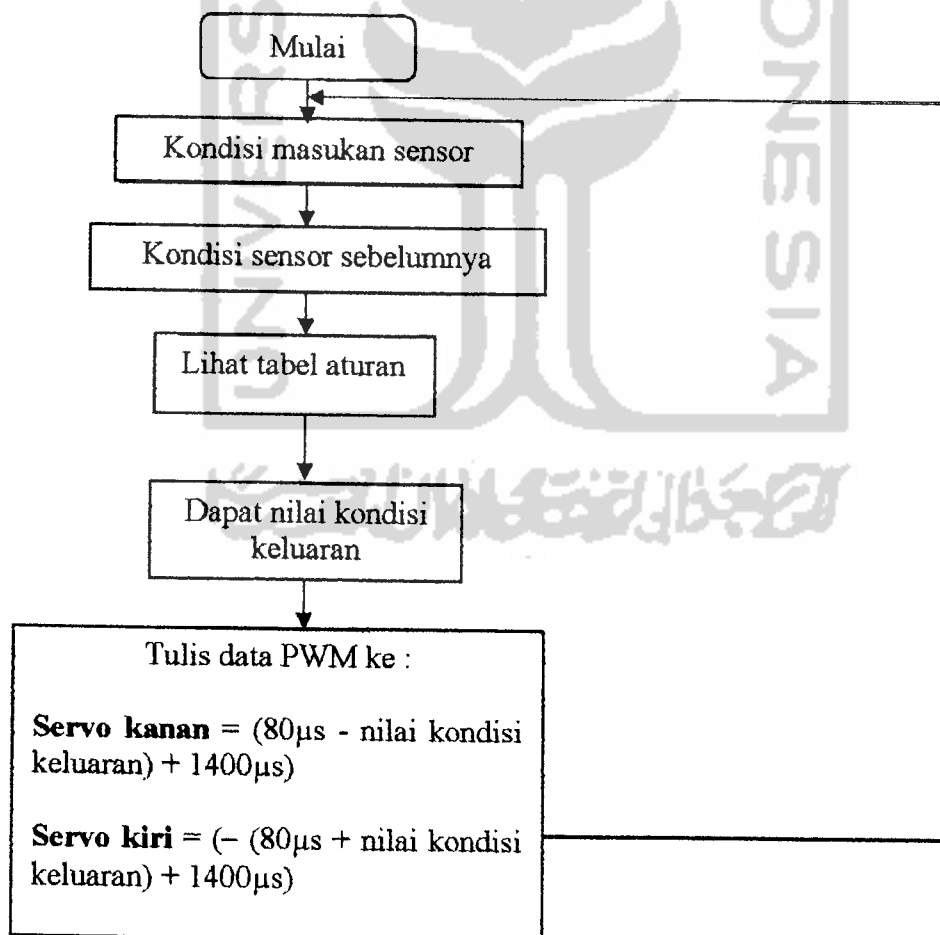
1 baris ke 3, dengan implikasi yaitu : “jika sensor bernilai – sangat besar dan keadaan sensor sebelumnya bernilai – besar maka kondisi keluaran bernilai - sangat besar (-64)”. Kondisi keluaran –sangat besar bernilai -64, kemudian memberikan nilai tersebut ke PWM kanan sebesar $(80 \mu s - (-64)) + 1400 \mu s = 1544 \mu s$ dan PWM kiri sebesar $(- (80 \mu s + (-64))) + 1400 \mu s = 1384 \mu s$. Maka motor kanan akan berputar kencang sedang motor kiri berputar pelan dan menyebabkan robot berbelok ke kiri.

3.7 Perancangan Perangkat Lunak

Robot yang dibuat pada penelitian tugas akhir ini pada dasarnya hampir sama dengan robot pengikut garis biasa, yang berbeda adalah penerapan aturan-aturan yang diterapkan agar robot dapat berjalan mengikuti garis pemandu yang disediakan dengan baik dan dengan sesempurna mungkin serta memperkecil *tracking error*. Adapun dasar pemikiran dalam merancang perangkat lunak atau program robot ini adalah, pada saat tombol *start* ditekan robot akan langsung membaca keadaan sensor, kemudian akan didapat kondisi masukan sensor. Selain dari kondisi masukan sensor, robot juga akan membaca nilai keadaan masukan sensor sebelumnya. Setelah didapat kedua nilai tersebut maka program akan melihat aturan-aturan yang telah ditentukan berkaitan dengan kedua kondisi masukan tersebut, setelah didapat aturan dengan cara melihat tabel yang disediakan, selanjutnya program akan memberikan data ke motor servo kanan dan motor servo kiri sesuai nilai yang didapat dari tabel aturan, apakah robot akan mengarah ke kanan, ke kiri atau lurus. Setelah selesai program akan kembali ke

keadaan semula untuk mulai mengambil nilai kondisi masukan sensor yang baru dan juga kondisi masukan sensor sebelumnya sehingga didapat nilai aturan yang baru serta data PWM untuk motor yang baru. Sebagai acuan dalam membuat program maka terlebih dahulu dibuat susunan algoritma robot pengikut garis seperti yang terlihat pada Gambar 3.11

Dalam penyusunan dan pembuatan program, digunakan bahasa C sebagai bahasa yang akan diprogramkan pada mikrokontroller ATMEGA8, digunakannya bahasa C karena lebih mendekati bahasa tingkat tinggi serta lebih mudah dipahami dan dipelajari dibandingkan bahasa *assembly*.



Gambar 3.10 Diagram alir program utama

BAB IV

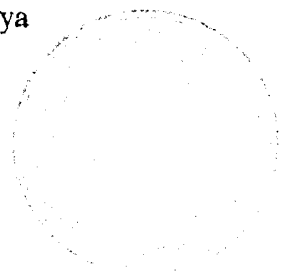
ANALISIS DAN PEMBAHASAN

Bab ini akan dijelaskan mengenai hasil pengujian terhadap keseluruhan sistem. Materi pengujian meliputi pengujian sensor infra merah dan pengujian sistem terhadap berbagai macam variasi lebar jalur dan sudut belokan. Pengujian sensor infra merah adalah pengujian dengan mengukur tegangan keluaran infra merah setelah melewati rangkaian komparator. Selanjutnya pengujian sistem dengan berbagai macam variasi jalur meliputi jalur dengan lebar 1 cm sampai 9 cm. Kemudian pengujian sistem terhadap belokan dengan variasi sudut mulai dari 10° sampai dengan 90° .

4.1 Pengujian Sensor Inframerah

Pada sensor inframerah ini pengujian dilakukan pada penggunaannya sebagai sensor garis. Sensor inframerah ini menggunakan IRED (*Infrared Emitting Diode*) yang berfungsi sebagai pemancar cahaya inframerah dan fototransistor sebagai penerima cahaya inframerah. Untuk mendapatkan keluaran 1 dan 0, maka keluaran pada fototransistor dibandingkan dengan tegangan referensi yaitu sebesar 3,2 Volt dengan menggunakan IC LM324 yang berfungsi sebagai komparator.

Warna jalur yang digunakan adalah hitam dengan dasar berwarna putih. Apabila sensor inframerah berada diatas garis hitam maka cahaya inframerah tidak akan memantul, sehingga fototransistor tidak aktif, begitu juga sebaliknya



apabila sensor inframerah berada pada dasar berwarna putih maka cahaya inframerah akan memantul, sehingga fototransistor akan aktif. Tegangan yang terukur pada kaki *output* IC LM324 ketika berada diatas garis hitam berkisar pada 0,9 V sedangkan pada dasar berwarna putih adalah 4,5 V.

Untuk pengujian sensor infra merah ini dilakukan dengan mengubah-ubah jarak pantul dari sensor terhadap jalur.

Tabel 4.1 Hasil pengujian sensor inframerah pada *output* fototransistor

Pengujian	Jarak pantul sensor (cm)	Tegangan <i>output</i> pada fototransistor (volt)		
		Putih	Hitam	Hijau Tua
1	0,1	4,08	4,2	4,2
2	0,2	3,9	4,1	3,9
3	0,3	3,1	4,0	3,8
4	0,5	2,5	4,0	3,6
5	0,7	2,1	4,0	3,6
6	1	1,7	3,9	2,8
7	1,5	3,3	3,2	2,7

Tabel 4.2 Hasil pengujian sensor inframerah pada *output* IC LM324

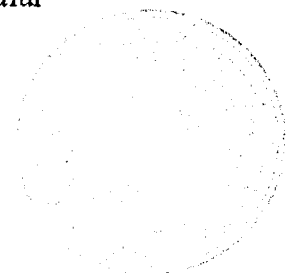
Pengujian	Jarak pantul sensor (cm)	Tegangan <i>output</i> pada IC LM324 (volt)		
		Putih	Hitam	Hijau Tua
1	0,1	0,9	0,9	0,9
2	0,2	0,9	0,9	0,9
3	0,3	4,5	0,9	0,9
4	0,5	4,5	0,9	0,9
5	0,7	4,5	0,9	0,9
6	1	4,5	0,9	4,5
7	1,5	0,9	4,5	4,5

Pada pengujian, jalur yang dibuat berupa garis lurus dengan menggunakan isolatip berwarna hitam. Pengujian dilakukan dengan cara menjalankan robot pada jalur yang telah dibuat. Adapun hasil dari pengujian lebar jalur dapat dilihat pada tabel berikut :

Tabel 4.3 Hasil pengujian lebar jalur

No.	Lebar Jalur	Hasil	Keterangan
1.	1 cm	Berhasil	Berhasil dengan baik
2.	2 cm	Berhasil	Berhasil dengan baik
3.	3 cm	Berhasil	Berhasil dengan baik
4.	4 cm	Berhasil	Berhasil dengan baik
5.	5 cm	Berhasil	Berhasil dengan baik
6.	6 cm	Berhasil	Berhasil dengan baik
7.	7 cm	Berhasil	Berhasil dengan baik
8.	8 cm	Berhasil	Berhasil dengan <i>tracking error</i>
9.	9 cm	Berhasil	Berhasil dengan <i>tracking error</i>

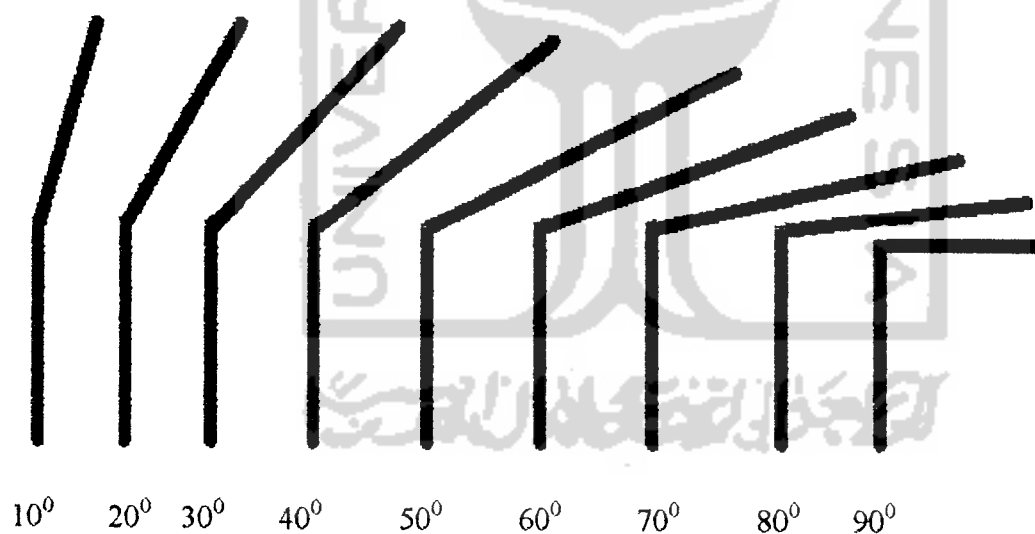
Saat pengujian dilakukan robot tidak mengalami banyak kesulitan dalam mengikuti jalur, karena aturan-aturan yang dibuat mampu membuat robot mengikuti semua garis dengan berbagai jenis ukuran lebar dan juga memperkecil terjadinya kesalahan yang dapat menyebabkan robot mengikuti jalur dengan tidak sempurna. Dari Tabel 4.3 terlihat bahwa semua jalur dengan lebar 1 cm sampai dengan 9 cm berhasil diikuti oleh robot, walaupun robot ternyata mengikuti jalur



dengan lebar 8 cm dan 9 cm dengan *tracking error* tetapi mampu diikuti dengan cukup baik.

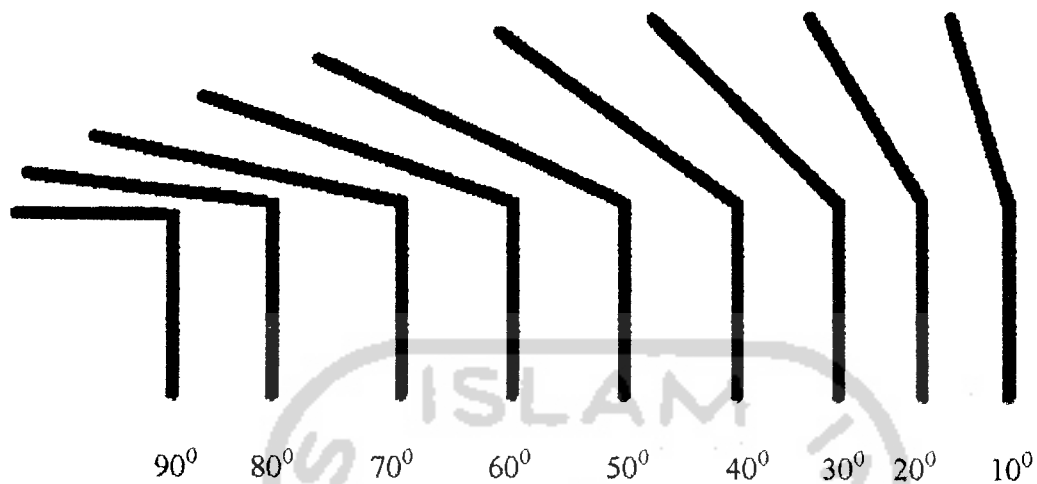
4.3 Pengujian Sistem Terhadap Sudut Belok

Untuk pengujian terhadap sudut belok ini robot akan dijalankan pada jalur lurus dengan panjang 35 cm kemudian berbelok, sudut belokan pada pengujian ini bernilai 10° , 20° , 30° , 40° , 50° , 60° , 70° , 80° dan 90° . Jalur yang digunakan pada pengujian ini menggunakan isolatip berwarna hitam dengan latar berwarna putih dengan lebar jalur sebesar 3 cm.



Gambar 4.2 Bentuk jalur dalam berbagai variasi sudut belokan ke kanan

Dari semua variasi bentuk sudut belokkan mampu diikuti oleh robot dengan baik, seperti terlihat pada data Tabel 4.4 dan Tabel 4.5



Gambar 4.3 Bentuk jalur dalam berbagai variasi sudut belokan ke kiri

Tabel 4.4 Hasil pengujian robot dengan berbagai variasi sudut belokan ke kanan

No.	Besar Sudut	Hasil
1.	10°	Berhasil
2.	20°	Berhasil
3.	30°	Berhasil
4.	40°	Berhasil
5.	50°	Berhasil
6.	60°	Berhasil
7.	70°	Berhasil
8.	80°	Berhasil
9.	90°	Berhasil

Pengujian yang dilakukan dalam berbagai variasi sudut belokan ke kanan, hasilnya robot dapat mengikuti semua sudut jalur belokan. Akan tetapi setelah dilakukan beberapa kali percobaan, pada sudut 10° sampai dengan sudut 80° robot

mampu berbelok dengan baik, sedangkan untuk sudut 90° setelah dilakukan beberapa kali percobaan ada beberapa kali *error* dimana robot terlewat jalur yang ditentukan, tetapi hal tersebut jarang terjadi karena kecenderungan berhasil lebih besar.

Tabel 4.5 Hasil pengujian robot dengan berbagai variasi sudut belokan ke kiri

No.	Besar Sudut	Hasil
1.	10°	Berhasil
2.	20°	Berhasil
3.	30°	Berhasil
4.	40°	Berhasil
5.	50°	Berhasil
6.	60°	Berhasil
7.	70°	Berhasil
8.	80°	Berhasil
9.	90°	Berhasil

Sama halnya pada pengujian yang dilakukan dalam berbagai variasi sudut belokan ke kanan, pada pengujian belokan ke kiri ini robot dapat mengikuti semua sudut jalur belokan. Dan sama halnya dengan belokkan kanan, yaitu pada belokkan kiri sudut 10° sampai dengan sudut 80° robot mampu berbelok dengan baik untuk beberapa percobaan, sedangkan untuk sudut 90° setelah dilakukan beberapa kali percobaan ada beberapa kali *error* dimana robot terlewat jalur yang

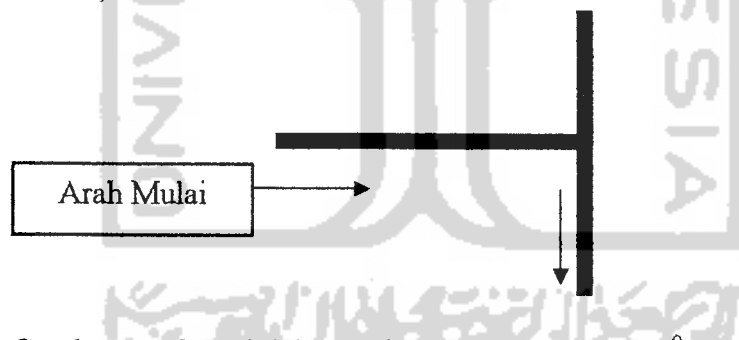
ditentukan dan sekali lagi hal tersebut jarang terjadi karena kecenderungan berhasil tetap lebih besar.

4.4 Pengujian Terhadap Variasi Bentuk Jalur

Pada pengujian terhadap bentuk jalur ini yang akan dipakai adalah jalur dengan bentuk pertigaan dengan sudut 90° dan juga jalur dengan bentuk perempatan, pengujian ini dilakukan untuk mengetahui arah jalan robot jika menemui jalur dengan pola yang bervariasi.

4.4.1 Pengujian bentuk pertigaan dengan sudut 90°

Untuk pengujian ini jalur yang dipakai menggunakan isolatip berwarna hitam dengan lebar 3,5 cm.

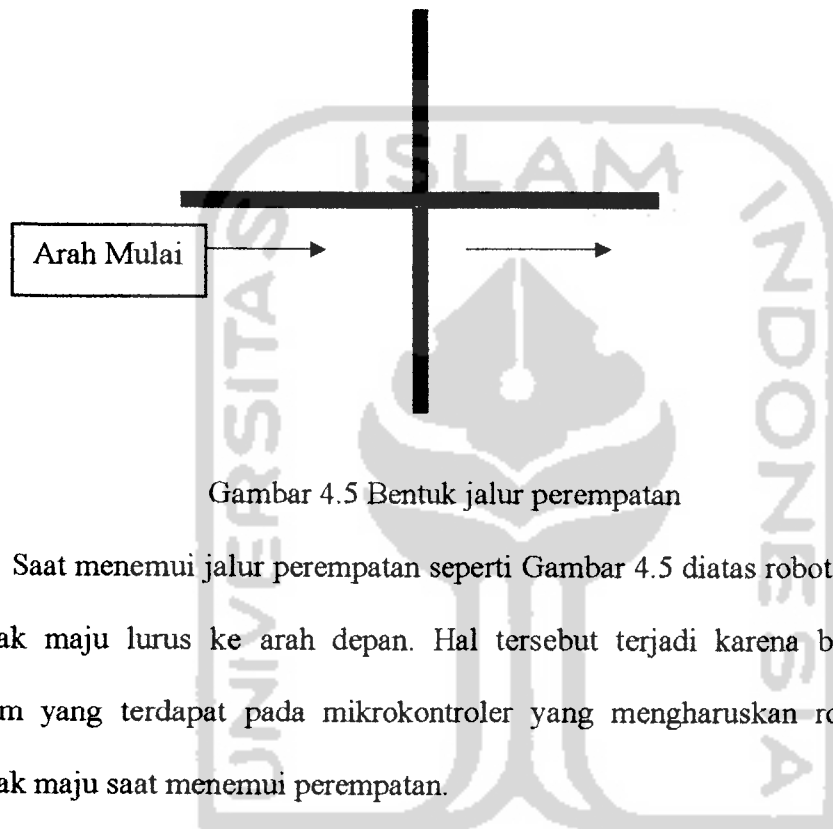


Gambar 4.4 Bentuk jalur pertigaan dengan sudut 90°

Robot saat menemui jalur pertigaan seperti Gambar 4.4 diatas robot akan berbelok ke arah kanan, sebenarnya robot tidak diprogram khusus untuk berbelok ke arah kanan, saat mulai robot akan mengikuti jalur lurus dan saat menemukan pertigaan robot sempat melewati jalur dan saat robot mendapatkan jalur berwarna putih robot akan bergerak mundur perlahan, saat mundur itu pula robot bergerak cenderung ke arah kanan, maka robot akan berbelok ke arah kanan.

4.4.2 Pengujian bentuk jalur perempatan

Pada pengujian ini jalur yang dipakai juga menggunakan isolatip berwarna hitam berlatar warna putih dengan lebar jalur 3,5 cm.



Gambar 4.5 Bentuk jalur perempatan

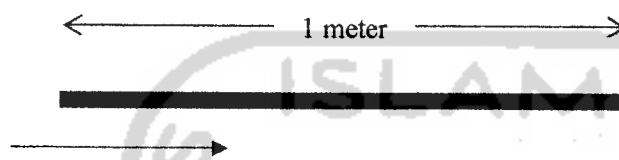
Saat menemui jalur perempatan seperti Gambar 4.5 diatas robot akan terus bergerak maju lurus ke arah depan. Hal tersebut terjadi karena berdasarkan program yang terdapat pada mikrokontroler yang mengharuskan robot untuk bergerak maju saat menemui perempatan.

4.5 Pengujian Kecepatan Saat Menempuh Jalur

Pengujian ini dilakukan dengan cara membuat robot mengikuti jalur tertentu sambil dihitung waktu tempuhnya, pengujian dilakukan sebanyak lima kali lalu diambil rata-rata waktu tempuh robot sehingga didapat kecepatan rata-rata robot. Pengujian dilakukan dengan cara membuat robot mengikuti garis lurus dan garis yang sudah dibuat dengan bentuk melingkar dengan panjang tertentu.

4.5.1 Pengujian Kecepatan Saat Menempuh Jalur Lurus

Pengujian ini adalah membuat robot mengikuti sebuah garis lurus sepanjang 1 meter sambil dihitung waktu tempuhnya. Jalur yang diikuti terbuat dari isolatip berwarna hitam dengan lebar jalur sebesar 3,5 cm.



Gambar 4.6 Jalur untuk pengujian kecepatan jalur lurus

Hasil dari pengujian kecepatan ini dapat dilihat pada Tabel 4.6 berikut :

Tabel 4.6 Hasil pengujian waktu tempuh jalur lurus

Pengujian ke-	Waktu Tempuh (detik)
1	6,84
2	6,92
3	7,16
4	7,18
5	7,02

Dari hasil diatas maka dapat dicari waktu tempuh rata-rata, yaitu :

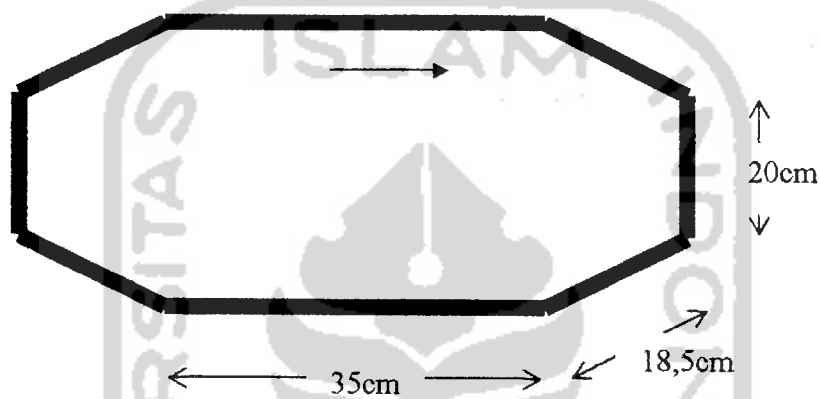
$$s \text{ rata-rata} = \frac{6,84 + 6,92 + 7,16 + 7,18 + 7,02}{5} = 7,02 \text{ detik}$$

Setelah didapat waktu tempuh rata-rata, selanjutnya mencari kecepatan rata-rata saat robot menempuh jalur lurus, yaitu :

$$V_{\text{rata-rata}} = \frac{1 \text{ meter}}{7,02 \text{ s}} = 0,1425 \text{ m/s} = 14,25 \text{ cm/s}$$

4.5.2 Pengujian Kecepatan Saat Menempuh Jalur Tertentu

Dengan cara yang sama dengan pengujian kecepatan jalur lurus, yaitu membuat robot mengikuti jalur tertentu kemudian dihitung waktu tempuhnya. Adapun jalur yang dimaksud dapat dilihat seperti pada Gambar 4.7, dan hasil dari pengujian waktu tempuh dapat dilihat pada Tabel 4.7.



Gambar 4.7 Jalur untuk pengujian kecepatan jalur tertentu

Tabel 4.7 Hasil pengujian waktu tempuh jalur tertentu

Pengujian ke-	Waktu Tempuh (detik)
1	12,47
2	12,29
3	12,98
4	12,81
5	13,08

Dari hasil diatas maka dapat dicari waktu tempuh rata-rata, yaitu :

$$s \text{ rata-rata} = \frac{12,47 + 12,29 + 12,98 + 12,81 + 13,08}{5} = 12,73 \text{ detik}$$

Setelah didapat waktu tempuh rata-rata, selanjutnya mencari kecepatan rata-rata saat robot menempuh jalur lurus, yaitu :

$$\text{Vrata-rata} = \frac{1,84\text{meter}}{12,73\text{s}} = 0,1445 \text{ m/s} = 14,45 \text{ cm/s}$$



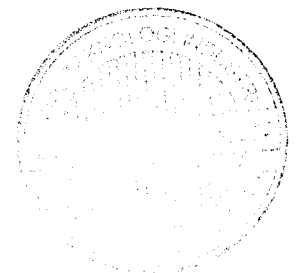
BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pada perancangan sistem dan hasil analisa yang didapat maka dalam pembuatan Robot pengikut garis memanfaatkan logika sederhana dengan kontrol berbasis aturan ini dapat disimpulkan beberapa hal, yaitu :

1. Sensor inframerah dalam perancangannya dapat difungsikan sebagai pendeteksi garis pada jalur.
2. Dengan menggunakan rangkaian komparator maka, keluaran sensor dapat dilogika-kan sebagaimana logika digital yaitu 0 dan 1.
3. Dalam perancangan sistem untuk robot pengikut garis ini memiliki masukan berupa keadaan sensor dari sensor fototransistor untuk mendeteksi area garis hitam dan keadaan sensor sebelumnya (KSS) yang merupakan keadaan sensor yang didapat pada siklus kerja sebelum keadaan sekarang.
4. Dengan memanfaatkan logika-logika yang cukup sederhana berbasis aturan-aturan tertentu maka dapat membuat robot mampu mengikuti garis dengan jalur yang telah ditentukan dan dengan keadaan lebar jalur mulai dari selebar 1 cm sampai 9 cm tanpa mengubah kedudukan sensor.
5. Dari hasil pengujian robot mampu berbelok ke kanan dan kekiri dengan sudut belokkan mulai dari 10^0 sampai 90^0 serta memiliki kecepatan rata-rata sebesar 14,25 cm/s pada jalur lurus.



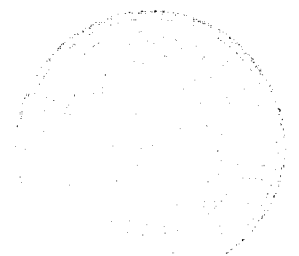
5.2 Saran

Untuk mengembangkan sistem dimasa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

1. Untuk pengembangan dari robot ini diharapkan agar dapat memiliki kecepatan yang lebih dari yang ada saat ini, karena dalam perancangan ini menggunakan motor servo, ada baiknya agar dapat juga dikembangkan untuk penggunaannya dengan motor DC dan Steper.
2. Robot ini dapat dikembangkan lagi dengan membuat robot pengikut garis dengan warna jalur tidak hanya hitam atau putih, tetapi juga dapat mengikuti jalur dengan warna berbeda.
3. Diharapkan untuk robot pengikut garis selanjutnya sudah dapat menerapkan logika-logika yang lebih kompleks lagi, atau penerapan sensor yang memiliki masukan lebih rumit seperti dengan mengganti sensor infra merah menjadi kamera.

DAFTAR PUSTAKA

1. Budioko, Totok., 2005. *Belajar dengan mudah Pemrograman Bahasa C dengan SDCC [Small Device C Compiler] Pada Mikrokontroller AT 89X051 / AT 89C51 / 52 Teori simulasi dan aplikasi*, Yogyakarta : Gava Media.
2. Kanginan, Marthen, Ir.,M.Sc., 2001. *Matematika untuk SMU kelas III*, Bandung: Grafindo Media Pratama.
3. Malik, Ibnu, Moh., 2006. *Pengantar Membuat ROBOT*, Yogyakarta : Gava Media.
4. Pitowarno, Endra., 2006. *Robotika Desain, Kontrol, dan Kecerdasan Buatan*, Yogyakarta : Penerbit Andi.
5. Suprpto., 2003. *Logika Informatika (Dasar-dasar Logika untuk Pemrograman Komputer & Perancangan Komputer)*, Yogyakarta : Gava Media.
6. Wardhana, Lingga., 2006. *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware dan Aplikasi*, Yogyakarta : Penerbit Andi.
7. www.atmel.com/
8. www.elm-chan.org/
9. www.en.wikipedia.org/
10. www.seattlerobotics.org/



LAMPIRAN - LAMPIRAN



Lampiran :

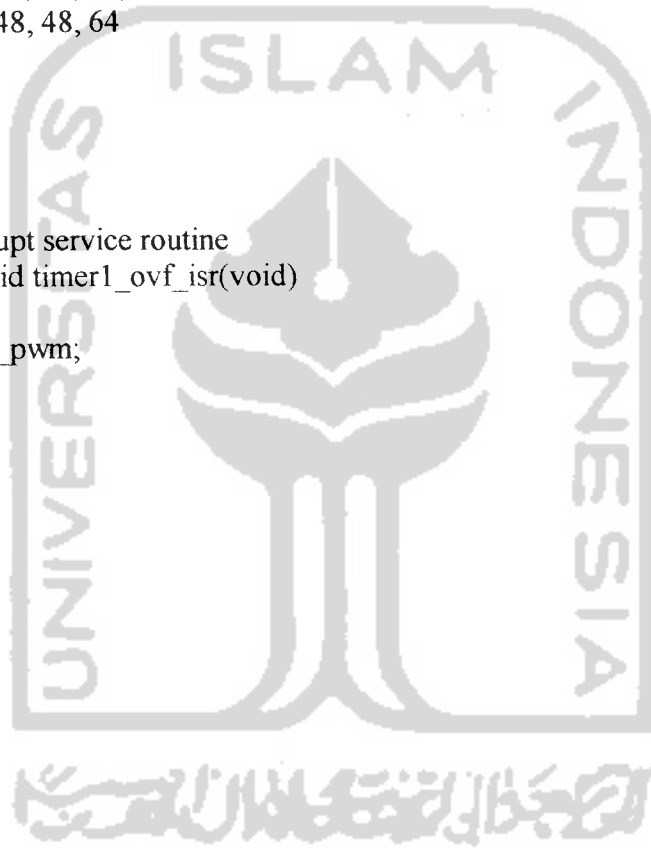
```
#include <mega8.h>
#include <delay.h>

#define ka 0
#define ki 1

flash int tabel_aturan[45]={
-64, -48, -48, -32, -48, 48, 64, 64, 64,
-48, -32, -32, -16, -16, 32, 48, 64, 64,
-64, -48, -32, -16, 0, 16, 32, 48, 64,
-64, -64, -48, -32, 16, 16, 32, 32, 48,
-64, -64, -64, -48, 48, 32, 48, 48, 64
};

int sensor_ka,sensor_ki;
unsigned char th[2],tl[2];

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    static unsigned char idx_pwm;
    if(idx_pwm==0)
    {
        TCCR1B=0x02;
        TCNT1H=0xd8;
        TCNT1L=0xf0;
        PORTB.1=0;
        PORTB.2=0;
    }
    else if(idx_pwm==1)
    {
        TCCR1B=0x01;
        TCNT1H=th[0];
        TCNT1L=tl[0];
        PORTB.1=1;
        PORTB.2=0;
    }
    else if(idx_pwm==2)
    {
        TCNT1H=th[1];
        TCNT1L=tl[1];
        PORTB.1=0;
        PORTB.2=1;
    }
}
```



```

    idx_pwm++;if(idx_pwm==3)idx_pwm=0;
}

void set_pwm(char n,int val)
{
    unsigned int tim_val;

    if(!n)val=-val;
    tim_val=65536-((val+1400)*8);
    th[n]=(tim_val>>8) & 0xff;
    tl[n]=tim_val & 0xff;
}

int get_error()
{
    sensor_ka=(~PIND>>2) & 0x3f;
    sensor_ki=~PINC & 0x3f;
    return (sensor_ka-sensor_ki); //error
}

void main(void)
{
    int error,error1,d_error;
    int pwm_ka,pwm_ki,v0,nilai_out,x,y;

    DDRB=0x07;
    DDRC=0x00;
    DDRD=0x00;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 8000,000 kHz
    // Mode: Normal top=FFFFh
    // OC1A output: Discon.
    // OC1B output: Discon.
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    TCCR1A=0x00;
    TCCR1B=0x01;
    TCNT1H=0x00;
    TCNT1L=0x00;

    // Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK=0x04;

```

```

// Global enable interrupts
#asm("sei")

error1=0;

pwm_ka=0;
pwm_ki=0;
v0=80;
set_pwm(ka,pwm_ka); //Motor kanan = 0 maju negatif
set_pwm(ki,pwm_ki); //Motor kiri = 1 maju positif

while (1)
{
PORTB.0=1;

error=get_error();
d_error=error-error1;
error1=error;
y=(d_error+64)/32;

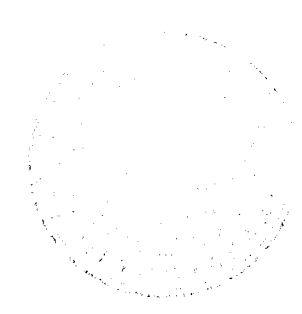
if(error<-48)x=0;
else if(error<-32)x=1;
else if(error<-16)x=2;
else if(error<0)x=3;
else if(error==0)x=4;
else if(error<16)x=5;
else if(error<32)x=6;
else if(error<48)x=7;
else if(error<64)x=8;

if(d_error<-32)y=0;
else if(d_error<0)y=1;
else if(d_error==0)y=2;
else if(d_error<32)y=3;
else if(d_error<64)y=4;

nilai_out=tabel_aturan[x+y*9];

set_pwm(ka,v0+nilai_out);
set_pwm(ki,v0-nilai_out);
}

```

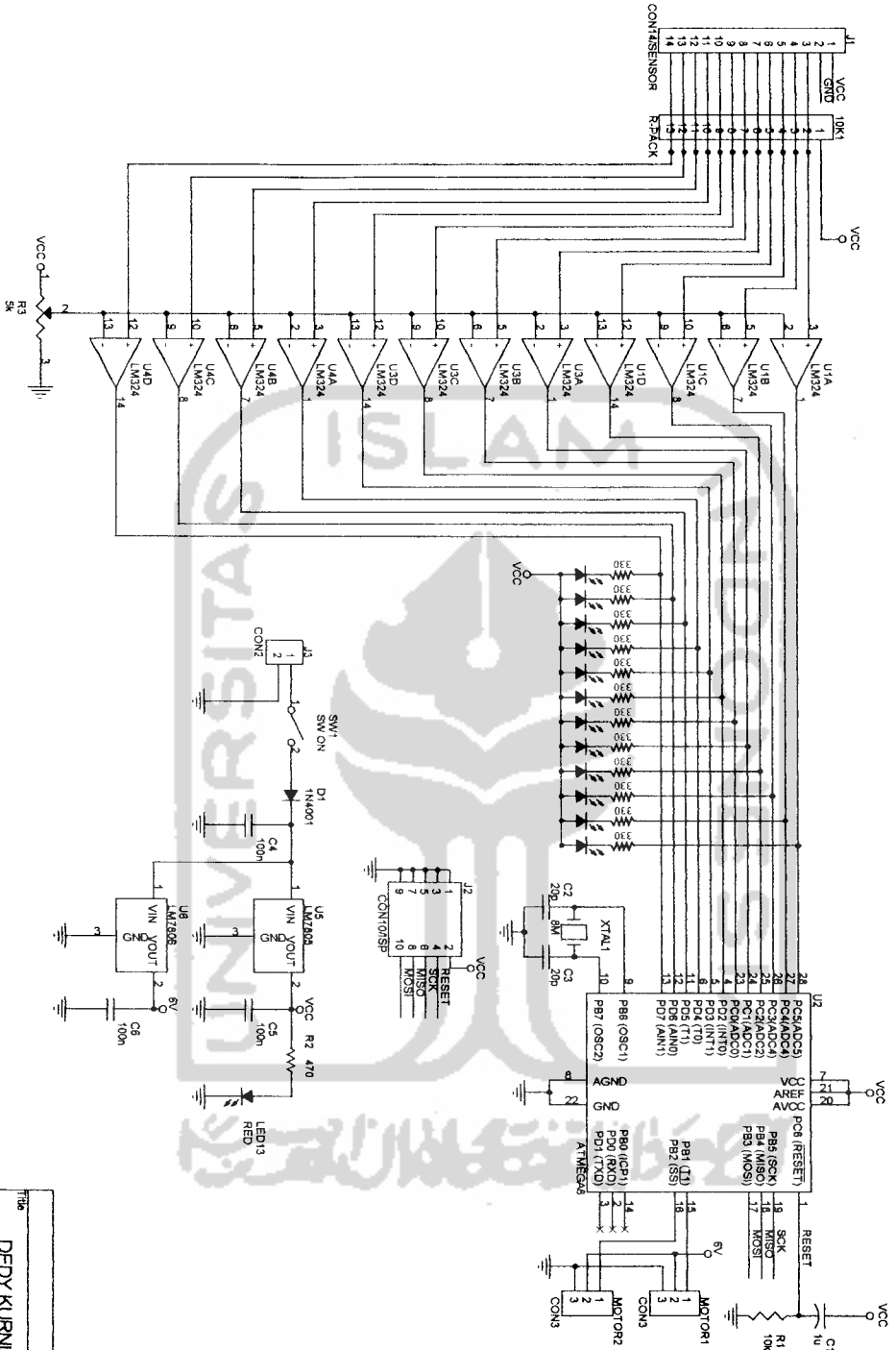



```
if(sensor_ka==0 && sensor_ki==0)
{
  set_pwm(ka,-30);
  set_pwm(ki,-30);
}

PORTB.0=0;
delay_ms(500);
};
}
```



LINE FOLLOWER ROBOT



Title	DEDY KURNIAMALAYA
Size	8
Document Number	02 524 119
Date	Sheet 1 of 1
Rev	

atures

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



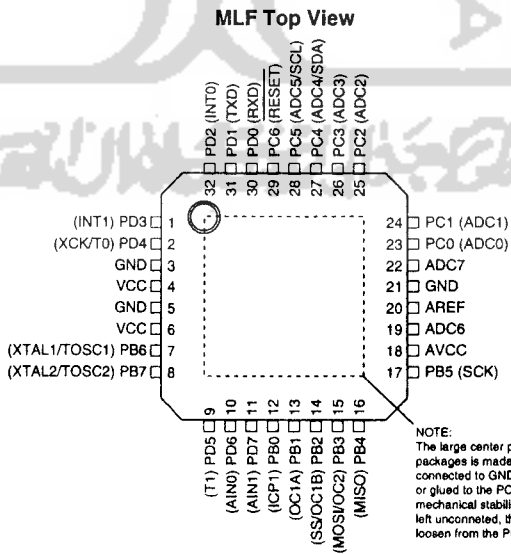
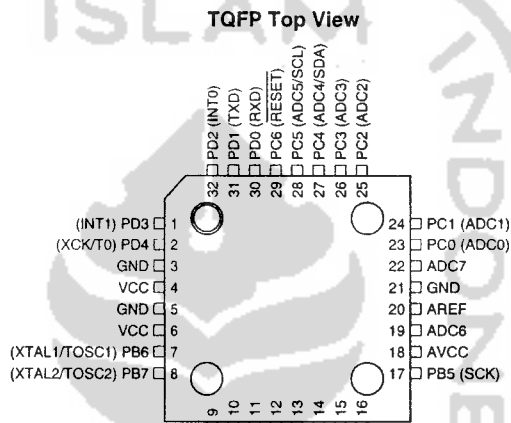
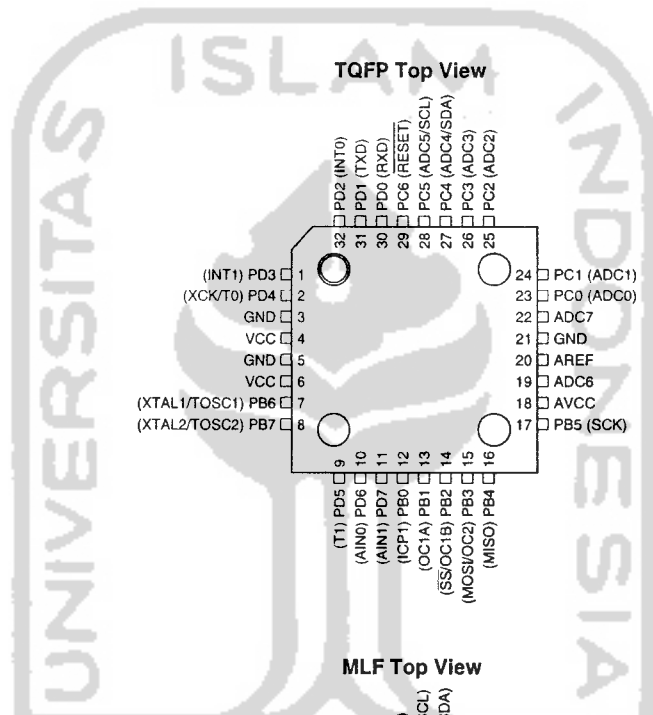
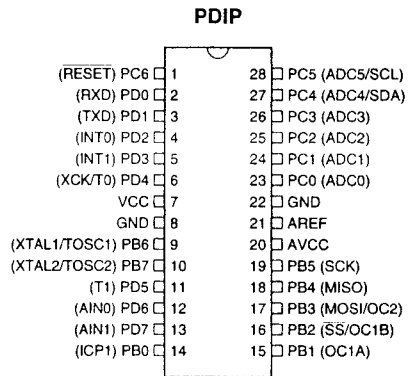
8-bit **AVR[®]**
with 8K Bytes
In-System
Programmable
Flash

ATmega8
ATmega8L

2486QS-AVR-10/06



Configurations

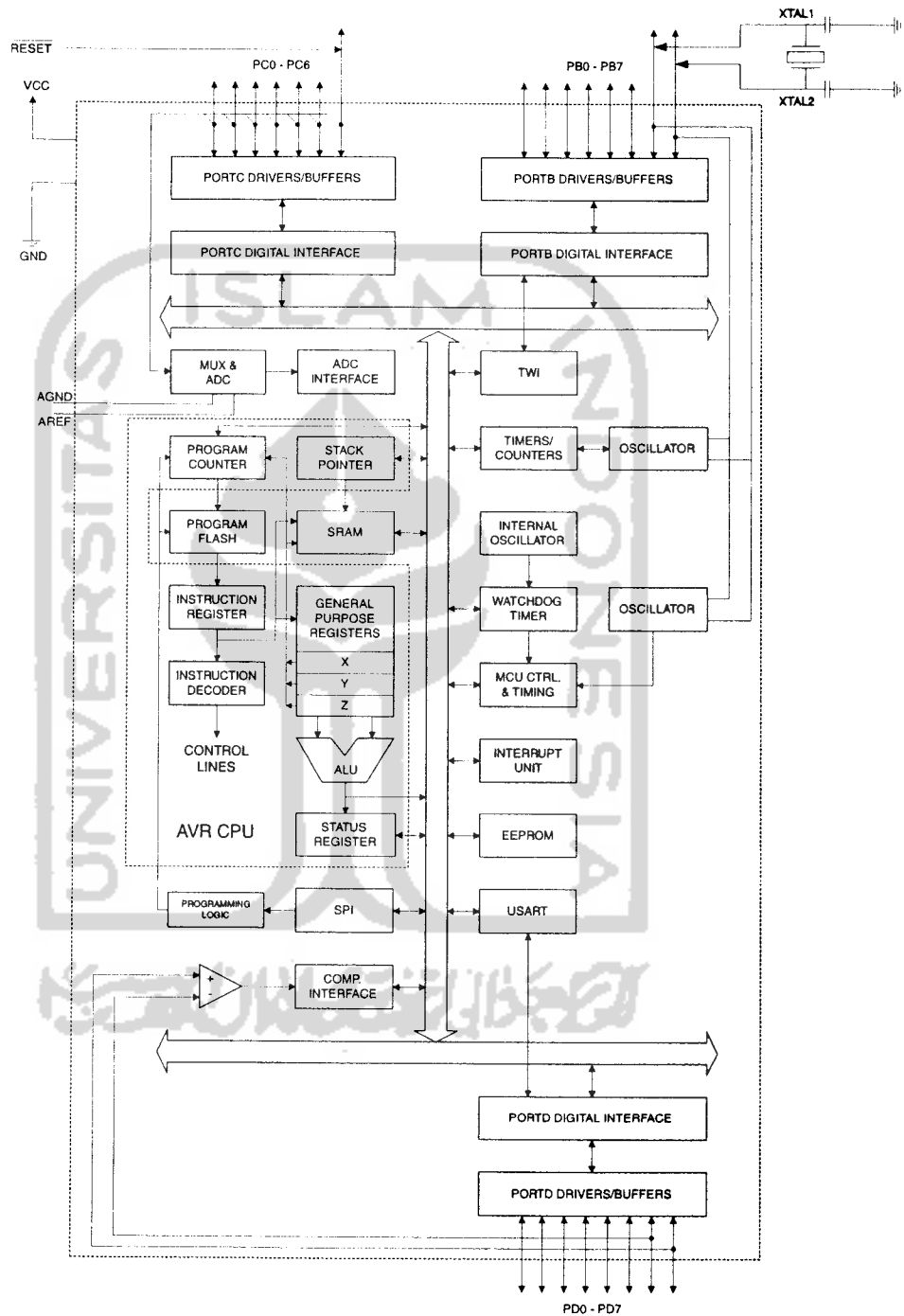


Overview

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

Block Diagram

Figure 1. Block Diagram





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

The ATmega8 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes of EEPROM, 1K byte of SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, a 6-channel ADC (eight channels in TQFP and QFN/MLF packages) with 10-bit accuracy, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The Flash Program memory can be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash Section will continue to run while the Application Flash Section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega8 AVR is supported with a full suite of program and system development tools, including C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

Disclaimer

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

Descriptions

Digital supply voltage.

Ground.

B (PB7..PB0) **L1/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 58 and "System Clock and Clock Options" on page 25.

t C (PC5..PC0)

Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

6/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 15 on page 38. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated on page 61.

rt D (PD7..PD0)

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

Port D also serves the functions of various special features of the ATmega8 as listed on page 63.

SET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.

AV_{CC} is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that Port C (5..4) use digital supply voltage, V_{CC} .

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

**7..6 (TQFP and QFN/MLF
page 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.



Sources

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





Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
F (0x5F)	SREG	I	T	H	S	V	N	Z	C	11
E (0x5E)	SPH	–	–	–	–	–	SP10	SP9	SP8	13
D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	13
C (0x5C)	Reserved									
B (0x5B)	GICR	INT1	INT0	–	–	–	–	IVSEL	IVCE	49, 67
A (0x5A)	GIFR	INTF1	INTF0	–	–	–	–	–	–	68
39 (0x59)	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0	72, 102, 122
38 (0x58)	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0	73, 103, 122
37 (0x57)	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN	213
36 (0x56)	TWCR	TWNT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWME	171
35 (0x55)	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00	33, 66
34 (0x54)	MCUCSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	41
33 (0x53)	TCCR0	–	–	–	–	–	CS02	CS01	CS00	72
32 (0x52)	TCNT0	Timer/Counter0 (8 Bits)								72
31 (0x51)	OSCCAL	Oscillator Calibration Register								31
30 (0x50)	SFIOR	–	–	–	–	ACME	PUD	PSR2	PSR10	58, 75, 123, 193
2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10	97
2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	100
2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High byte								101
2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low byte								101
2B (0x4B)	OCR1AH	Timer/Counter1 – Output Compare Register A High byte								101
2A (0x4A)	OCR1AL	Timer/Counter1 – Output Compare Register A Low byte								101
x29 (0x49)	OCR1BH	Timer/Counter1 – Output Compare Register B High byte								101
x28 (0x48)	OCR1BL	Timer/Counter1 – Output Compare Register B Low byte								101
x27 (0x47)	ICR1H	Timer/Counter1 – Input Capture Register High byte								102
x26 (0x46)	ICR1L	Timer/Counter1 – Input Capture Register Low byte								102
x25 (0x45)	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20	117
x24 (0x44)	TCNT2	Timer/Counter2 (8 Bits)								119
x23 (0x43)	OCR2	Timer/Counter2 Output Compare Register								119
x22 (0x42)	ASSR	–	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB	119
x21 (0x41)	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0	43
0 ⁽¹⁾ (0x40 ⁽¹⁾)	UBRRH	URSEL	–	–	–	–	UBRR[11:8]			158
	UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	156
x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	20
x1E (0x3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	20
x1D (0x3D)	EEDR	EEPROM Data Register								20
x1C (0x3C)	EECR	–	–	–	–	EERIE	EEMWE	EERE	–	20
x1B (0x3B)	Reserved									
x1A (0x3A)	Reserved									
x19 (0x39)	Reserved									
x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	65
x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	65
x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	65
x15 (0x35)	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	65
x14 (0x34)	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	65
x13 (0x33)	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	65
x12 (0x32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	65
x11 (0x31)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	65
x10 (0x30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	65
x0F (0x2F)	SPDR	SPI Data Register								131
x0E (0x2E)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	131
x0D (0x2D)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	129
x0C (0x2C)	UDR	USART I/O Data Register								153
x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	154
x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	155
x09 (0x29)	UBRRL	USART Baud Rate Register Low byte								158
x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	194
x07 (0x27)	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	205
x06 (0x26)	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	207
x05 (0x25)	ADCH	ADC Data Register High byte								208
x04 (0x24)	ADCL	ADC Data Register Low byte								208
x03 (0x23)	TWDR	Two-wire Serial Interface Data Register								173
x02 (0x22)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	174

Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x21	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	173
0x20	TWBR	Two-wire Serial Interface Bit Rate Register								171

- 1. Refer to the USART description for details on how to access UBRRH and UCSRC.
- 2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
- 3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.





Instruction Set Summary

Monomials	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
	Rdl, K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
	Rdl, K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
	Rd, K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
	Rd, K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (0xFF - K)$	Z,N,V	1
	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \lll 1$	Z,C	2
BRANCH INSTRUCTIONS					
AP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
P		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
ALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
LL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
T		Subroutine Return	$PC \leftarrow STACK$	None	4
TI		Interrupt Return	$PC \leftarrow STACK$	I	4
SE	Rd, Rr	Compare, Skip if Equal	if $(Rd = Rr) PC \leftarrow PC + 2$ or 3	None	1/2/3
	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
C	Rd, Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
I	Rd, K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
RC	Rr, b	Skip if Bit in Register Cleared	if $(Rr(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
RS	Rr, b	Skip if Bit in Register is Set	if $(Rr(b)=1) PC \leftarrow PC + 2$ or 3	None	1/2/3
IC	P, b	Skip if Bit in I/O Register Cleared	if $(P(b)=0) PC \leftarrow PC + 2$ or 3	None	1/2/3
IS	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
BS	s, k	Branch if Status Flag Set	if $(SREG(s) = 1) PC \leftarrow PC + k + 1$	None	1/2
BC	s, k	Branch if Status Flag Cleared	if $(SREG(s) = 0) PC \leftarrow PC + k + 1$	None	1/2
EQ	k	Branch if Equal	if $(Z = 1) PC \leftarrow PC + k + 1$	None	1/2
NE	k	Branch if Not Equal	if $(Z = 0) PC \leftarrow PC + k + 1$	None	1/2
CS	k	Branch if Carry Set	if $(C = 1) PC \leftarrow PC + k + 1$	None	1/2
CC	k	Branch if Carry Cleared	if $(C = 0) PC \leftarrow PC + k + 1$	None	1/2
SH	k	Branch if Same or Higher	if $(C = 0) PC \leftarrow PC + k + 1$	None	1/2
LO	k	Branch if Lower	if $(C = 1) PC \leftarrow PC + k + 1$	None	1/2
MI	k	Branch if Minus	if $(N = 1) PC \leftarrow PC + k + 1$	None	1/2
PL	k	Branch if Plus	if $(N = 0) PC \leftarrow PC + k + 1$	None	1/2
GE	k	Branch if Greater or Equal, Signed	if $(N \oplus V = 0) PC \leftarrow PC + k + 1$	None	1/2
LT	k	Branch if Less Than Zero, Signed	if $(N \oplus V = 1) PC \leftarrow PC + k + 1$	None	1/2
HS	k	Branch if Half Carry Flag Set	if $(H = 1) PC \leftarrow PC + k + 1$	None	1/2
HC	k	Branch if Half Carry Flag Cleared	if $(H = 0) PC \leftarrow PC + k + 1$	None	1/2
TS	k	Branch if T Flag Set	if $(T = 1) PC \leftarrow PC + k + 1$	None	1/2
TC	k	Branch if T Flag Cleared	if $(T = 0) PC \leftarrow PC + k + 1$	None	1/2
VS	k	Branch if Overflow Flag is Set	if $(V = 1) PC \leftarrow PC + k + 1$	None	1/2
VC	k	Branch if Overflow Flag is Cleared	if $(V = 0) PC \leftarrow PC + k + 1$	None	1/2
Monomials	Operands	Description	Operation	Flags	#Clocks

Instruction Set Summary (Continued)

	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
TRANSFER INSTRUCTIONS					
	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
V	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
	Rd, K	Load Immediate	Rd ← K	None	1
	Rd, X	Load Indirect	Rd ← (X)	None	2
	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
	Rd, -X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
	Rd, Y	Load Indirect	Rd ← (Y)	None	2
	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
	Rd, -Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
	Rd, Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
	Rd, Z	Load Indirect	Rd ← (Z)	None	2
	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z + 1	None	2
	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
	X, Rr	Store Indirect	(X) ← Rr	None	2
	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
	-X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
	Y, Rr	Store Indirect	(Y) ← Rr	None	2
	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
	-Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
	Y+q, Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
	Z, Rr	Store Indirect	(Z) ← Rr	None	2
	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
	Z+q, Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
		Load Program Memory	R0 ← (Z)	None	3
	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z + 1	None	3
		Store Program Memory	(Z) ← R1:R0	None	-
	Rd, P	In Port	Rd ← P	None	1
	P, Rr	Out Port	P ← Rr	None	1
	Rr	Push Register on Stack	STACK ← Rr	None	2
	Rd	Pop Register from Stack	Rd ← STACK	None	2
AND BIT-TEST INSTRUCTIONS					
	P, b	Set Bit in I/O Register	I/O(P, b) ← 1	None	2
	P, b	Clear Bit in I/O Register	I/O(P, b) ← 0	None	2
	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z, C, N, V	1
	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z, C, N, V	1
	Rd	Rotate Left Through Carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z, C, N, V	1
	Rd	Rotate Right Through Carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z, C, N, V	1
	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z, C, N, V	1
AP	Rd	Swap Nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
IT	s	Flag Set	SREG(s) ← 1	SREG(s)	1
R	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
		Set Carry	C ← 1	C	1
		Clear Carry	C ← 0	C	1
		Set Negative Flag	N ← 1	N	1
		Clear Negative Flag	N ← 0	N	1
		Set Zero Flag	Z ← 1	Z	1
		Clear Zero Flag	Z ← 0	Z	1
		Global Interrupt Enable	I ← 1	I	1
		Global Interrupt Disable	I ← 0	I	1
		Set Signed Test Flag	S ← 1	S	1
		Clear Signed Test Flag	S ← 0	S	1
		Set Twos Complement Overflow	V ← 1	V	1
		Clear Twos Complement Overflow	V ← 0	V	1
		Set T in SREG	T ← 1	T	1
Inemomics	Operands	Description	Operation	Flags	#Clocks



Instruction Set Summary (Continued)

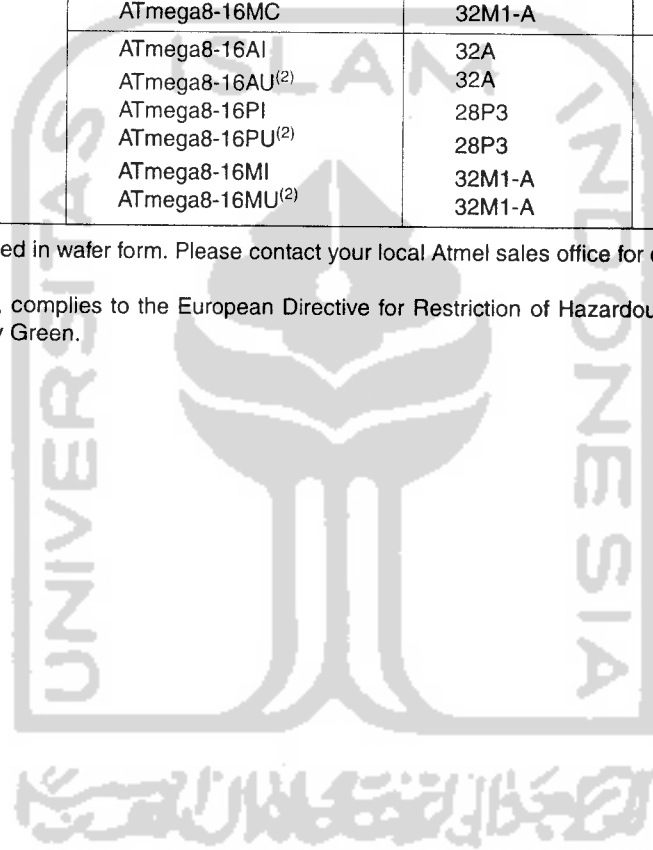
		Clear T in SREG	T ← 0	T	1
		Set Half Carry Flag in SREG	H ← 1	H	1
		Clear Half Carry Flag in SREG	H ← 0	H	1
CONTROL INSTRUCTIONS					
		No Operation		None	1
EP		Sleep	(see specific descr. for Sleep function)	None	1
		Watchdog Reset	(see specific descr. for WDR/timer)	None	1



Ordering Information

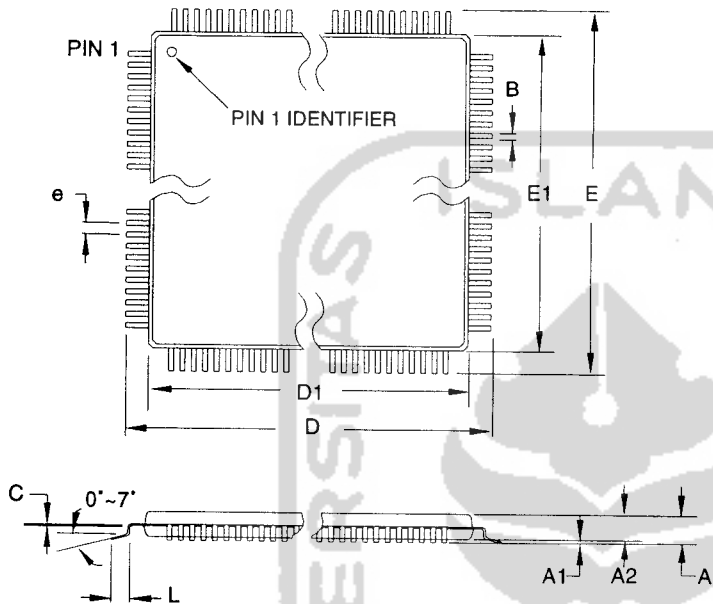
Speed (MHz)	Power Supply	Ordering Code	Package ⁽¹⁾	Operation Range
8	2.7 - 5.5	ATmega8L-8AC	32A	Commercial (0°C to 70°C)
		ATmega8L-8PC	28P3	
		ATmega8L-8MC	32M1-A	
		ATmega8L-8AI	32A	Industrial (-40°C to 85°C)
		ATmega8L-8AU ⁽²⁾	32A	
		ATmega8L-8PI	28P3	
ATmega8L-8PU ⁽²⁾	28P3			
16	4.5 - 5.5	ATmega8-16AC	32A	Commercial (0°C to 70°C)
		ATmega8-16PC	28P3	
		ATmega8-16MC	32M1-A	
		ATmega8-16AI	32A	Industrial (-40°C to 85°C)
		ATmega8-16AU ⁽²⁾	32A	
		ATmega8-16PI	28P3	
ATmega8-16PU ⁽²⁾	28P3			
		ATmega8-16MI	32M1-A	
		ATmega8-16MU ⁽²⁾	32M1-A	

- Notes:
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 alternative, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.



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

Packaging Information



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.

10/5/2001



2325 Orchard Parkway
San Jose, CA 95131

TITLE

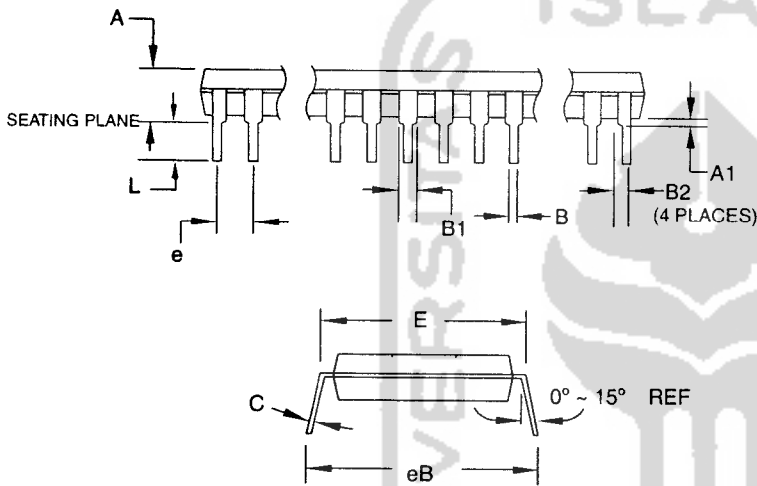
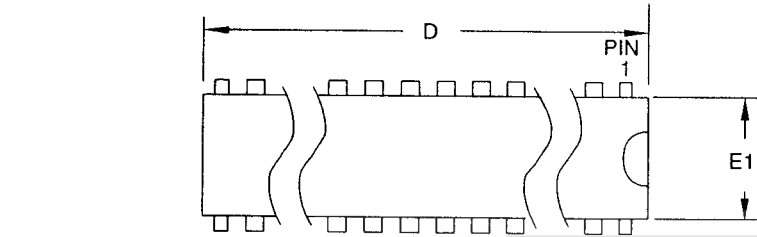
**32A, 32-lead, 7 x 7 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)**

DRAWING NO.

32A

REV.

B



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)

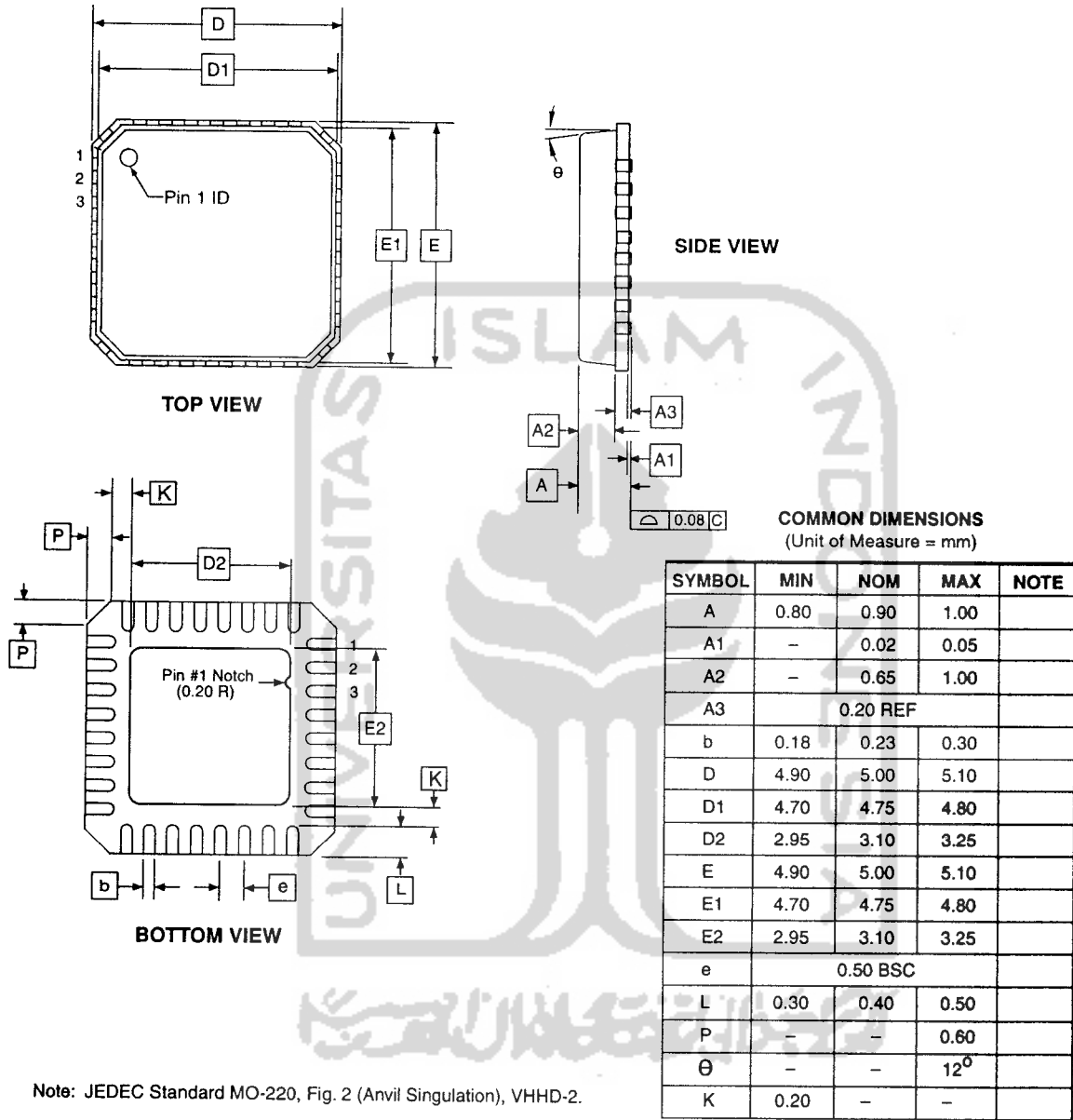
DRAWING NO.

28P3

REV.

B





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.	REV.
		32M1-A	E

atas

The revision letter in this section refers to the revision of the ATmega8 device.

nega8
/. D to I

- **First Analog Comparator conversion may be delayed**
- **Interrupts may be lost when writing the timer registers in the asynchronous timer**
- **Signature may be Erased in Serial Programming Mode**
- **CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2**

1. First Analog Comparator conversion may be delayed

If the device is powered by a slow rising V_{CC} , the first Analog Comparator conversion will take longer than expected on some devices.

Problem Fix/Workaround

When the device has been powered or reset, disable then enable the Analog Comparator before the first conversion.

2. Interrupts may be lost when writing the timer registers in the asynchronous timer

If one of the timer registers which is synchronized to the asynchronous timer2 clock is written in the cycle before a overflow interrupt occurs, the interrupt may be lost.

Problem Fix/Workaround

Always check that the Timer2 Timer/Counter register, TCNT2, does not have the value 0xFF before writing the Timer2 Control Register, TCCR2, or Output Compare Register, OCR2

3. Signature may be Erased in Serial Programming Mode

If the signature bytes are read before a chip erase command is completed, the signature may be erased causing the device ID and calibration bytes to disappear. This is critical, especially, if the part is running on internal RC oscillator.

Problem Fix/Workaround:

Ensure that the chip erase command has exceeded before applying the next command.

4. CKOPT Does not Enable Internal Capacitors on XTALn/TOSCn Pins when 32 KHz Oscillator is Used to Clock the Asynchronous Timer/Counter2

When the internal RC Oscillator is used as the main clock source, it is possible to run the Timer/Counter2 asynchronously by connecting a 32 KHz Oscillator between XTAL1/TOSC1 and XTAL2/TOSC2. But when the internal RC Oscillator is selected as the main clock source, the CKOPT Fuse does not control the internal capacitors on XTAL1/TOSC1 and XTAL2/TOSC2. As long as there are no capacitors connected to XTAL1/TOSC1 and XTAL2/TOSC2, safe operation of the Oscillator is not guaranteed.

Problem fix/Workaround

Use external capacitors in the range of 20 - 36 pF on XTAL1/TOSC1 and XTAL2/TOSC2. This will be fixed in ATmega8 Rev. G where the CKOPT Fuse will control internal capacitors also when internal RC Oscillator is selected as main clock source. For ATmega8 Rev. G, CKOPT = 0 (programmed) will enable the internal capacitors on XTAL1 and XTAL2. Customers who want compatibility between Rev. G and older revisions, must ensure that CKOPT is unprogrammed (CKOPT = 1).



atasheet Revision story

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.

anges from Rev.
36P- 02/06 to Rev.
36Q- 10/06

1. Updated “Timer/Counter Oscillator” on page 32.
2. Updated “Fast PWM Mode” on page 89.
3. Updated code example in “USART Initialization” on page 138.
4. Updated Table 37 on page 98, Table 39 on page 99, Table 42 on page 117, Table 44 on page 118, and Table 98 on page 240.
5. Updated “Erratas” on page 17.

anges from Rev.
86O-10/04 to Rev.
86P- 02/06

1. Added “Resources” on page 7.
2. Updated “External Clock” on page 32.
3. Updated “Serial Peripheral Interface – SPI” on page 124.
4. Updated Code Example in “USART Initialization” on page 138.
5. Updated Note in “Bit Rate Generator Unit” on page 170.
6. Updated Table 98 on page 240.
7. Updated Note in Table 103 on page 248.
8. Updated “Erratas” on page 17.

anges from Rev.
86N-09/04 to Rev.
86O-10/04

1. Removed to instances of “analog ground”. Replaced by “ground”.
2. Updated Table 7 on page 29, Table 15 on page 38, and Table 100 on page 244.
3. Updated “Calibrated Internal RC Oscillator” on page 30 with the 1 MHz default value.
4. Table 89 on page 225 and Table 90 on page 225 moved to new section “Page Size” on page 225.
5. Updated descripton for bit 4 in “Store Program Memory Control Register – SPMCR” on page 213.
6. Updated “Ordering Information” on page 13.

anges from Rev.
86M-12/03 to Rev.
86N-09/04

1. Added note to MLF package in “Pin Configurations” on page 2.
2. Updated “Internal Voltage Reference Characteristics” on page 42.
3. Updated “DC Characteristics” on page 242.

ATmega8(L)

4. ADC4 and ADC5 support 10-bit accuracy. Document updated to reflect this. Updated features in “Analog-to-Digital Converter” on page 196. Updated “ADC Characteristics” on page 248.

5. Removed reference to “External RC Oscillator application note” from “External RC Oscillator” on page 29.

anges from Rev.
86L-10/03 to Rev.
86M-12/03

1. Updated “Calibrated Internal RC Oscillator” on page 30.

anges from Rev.
86K-08/03 to Rev.
86L-10/03

1. Removed “Preliminary” and TBDs from the datasheet.
2. Renamed ICP to ICP1 in the datasheet.
3. Removed instructions CALL and JMP from the datasheet.
4. Updated t_{RST} in Table 15 on page 38, V_{BG} in Table 16 on page 42, Table 100 on page 244 and Table 102 on page 246.
5. Replaced text “XTAL1 and XTAL2 should be left unconnected (NC)” after Table 9 in “Calibrated Internal RC Oscillator” on page 30. Added text regarding XTAL1/XTAL2 and CKOPT Fuse in “Timer/Counter Oscillator” on page 32.
6. Updated Watchdog Timer code examples in “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 45.
7. Removed bit 4, ADHSM, from “Special Function IO Register – SFIOR” on page 58.
8. Added note 2 to Figure 103 on page 215.
9. Updated item 4 in the “Serial Programming Algorithm” on page 238.
10. Added t_{WD_FUSE} to Table 97 on page 239 and updated Read Calibration Byte, Byte 3, in Table 98 on page 240.
11. Updated Absolute Maximum Ratings* and DC Characteristics in “Electrical Characteristics” on page 242.

anges from Rev.
86J-02/03 to Rev.
86K-08/03

1. Updated V_{BOT} values in Table 15 on page 38.
2. Updated “ADC Characteristics” on page 248.
3. Updated “ATmega8 Typical Characteristics” on page 249.
4. Updated “Erratas” on page 17.

anges from Rev.
86I-12/02 to Rev.
86J-02/03

1. Improved the description of “Asynchronous Timer Clock – clk_{ASY} ” on page 26.
2. Removed reference to the “Multipurpose Oscillator” application note and the “32 kHz Crystal Oscillator” application note, which do not exist.



3. Corrected OCn waveforms in Figure 38 on page 90.
4. Various minor Timer 1 corrections.
5. Various minor TWI corrections.
6. Added note under "Filling the Temporary Buffer (Page Loading)" on page 216 about writing to the EEPROM during an SPM Page load.
7. Removed ADHSM completely.
8. Added section "EEPROM Write during Power-down Sleep Mode" on page 23.
9. Removed XTAL1 and XTAL2 description on page 5 because they were already described as part of "Port B (PB7..PB0) XTAL1/XTAL2/TOSC1/TOSC2" on page 5.
10. Improved the table under "SPI Timing Characteristics" on page 246 and removed the table under "SPI Serial Programming Characteristics" on page 241.
11. Corrected PC6 in "Alternate Functions of Port C" on page 61.
12. Corrected PB6 and PB7 in "Alternate Functions of Port B" on page 58.
13. Corrected 230.4 Mbps to 230.4 kbps under "Examples of Baud Rate Setting" on page 159.
14. Added information about PWM symmetry for Timer 2 in "Phase Correct PWM Mode" on page 113.
15. Added thick lines around accessible registers in Figure 76 on page 169.
16. Changed "will be ignored" to "must be written to zero" for unused Z-pointer bits under "Performing a Page Write" on page 216.
17. Added note for RSTDISBL Fuse in Table 87 on page 223.
18. Updated drawings in "Packaging Information" on page 14.

Changes from Rev.
86H-09/02 to Rev.
86I-12/02

1. Added errata for Rev D, E, and F on page 17.

Changes from Rev.
86G-09/02 to Rev.
86H-09/02

1. Changed the Endurance on the Flash to 10,000 Write/Erase Cycles.

Changes from Rev.
86F-07/02 to Rev.
86G-09/02

- 1 Updated Table 103, "ADC Characteristics," on page 248.

ATmega8(L)

anges from Rev.
6E-06/02 to Rev.
6F-07/02

- 1 Changes in “Digital Input Enable and Sleep Modes” on page 55.
- 2 Addition of OCS2 in “MOSI/OC2 – Port B, Bit 3” on page 59.
- 3 The following tables has been updated:
Table 51, “CPOL and CPHA Functionality,” on page 132, Table 59, “UCPOL Bit Settings,” on page 158, Table 72, “Analog Comparator Multiplexed Input⁽¹⁾,” on page 195, Table 73, “ADC Conversion Time,” on page 200, Table 75, “Input Channel Selections,” on page 206, and Table 84, “Explanation of Different Variables used in Figure 103 and the Mapping to the Z-pointer,” on page 221.
- 5 Changes in “Reading the Calibration Byte” on page 234.
- 6 Corrected Errors in Cross References.

anges from Rev.
86D-03/02 to Rev.
86E-06/02

- 1 Updated Some Preliminary Test Limits and Characterization Data
The following tables have been updated:
Table 15, “Reset Characteristics,” on page 38, Table 16, “Internal Voltage Reference Characteristics,” on page 42, DC Characteristics on page 242, Table , “ADC Characteristics,” on page 248.
- 2 Changes in External Clock Frequency
Added the description at the end of “External Clock” on page 32.
Added period changing data in Table 99, “External Clock Drive,” on page 244.
- 3 Updated TWI Chapter
More details regarding use of the TWI bit rate prescaler and a Table 65, “TWI Bit Rate Prescaler,” on page 173.

anges from Rev.
86C-03/02 to Rev.
86D-03/02

- 1 Updated Typical Start-up Times.
The following tables has been updated:
Table 5, “Start-up Times for the Crystal Oscillator Clock Selection,” on page 28, Table 6, “Start-up Times for the Low-frequency Crystal Oscillator Clock Selection,” on page 28, Table 8, “Start-up Times for the External RC Oscillator Clock Selection,” on page 29, and Table 12, “Start-up Times for the External Clock Selection,” on page 32.
- 2 Added “ATmega8 Typical Characteristics” on page 249.

anges from Rev.
86B-12/01 to Rev.
86C-03/02

- 1 Updated TWI Chapter.
More details regarding use of the TWI Power-down operation and using the TWI as Master with low TWBRR values are added into the datasheet.
Added the note at the end of the “Bit Rate Generator Unit” on page 170.
Added the description at the end of “Address Match Unit” on page 170.
- 2 Updated Description of OSCCAL Calibration Byte.
In the datasheet, it was not explained how to take advantage of the calibration bytes for 2, 4, and 8 MHz Oscillator selections. This is now added in the following sections:

Improved description of “Oscillator Calibration Register – OSCCAL” on page 31 and “Calibration Byte” on page 225.

3 Added Some Preliminary Test Limits and Characterization Data.

Removed some of the TBD’s in the following tables and pages:

Table 3 on page 26, Table 15 on page 38, Table 16 on page 42, Table 17 on page 44, “ $T_A = -40^{\circ}\text{C}$ to 85°C , $V_{CC} = 2.7\text{V}$ to 5.5V (unless otherwise noted)” on page 242, Table 99 on page 244, and Table 102 on page 246.

4 Updated Programming Figures.

Figure 104 on page 226 and Figure 112 on page 237 are updated to also reflect that AV_{CC} must be connected during Programming mode.

5 Added a Description on how to Enter Parallel Programming Mode if RESET Pin is Disabled or if External Oscillators are Selected.

Added a note in section “Enter Programming Mode” on page 228.





Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

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

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests

www.atmel.com/literature

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 ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, 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 OF 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. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2006 Atmel Corporation. All rights reserved. ATMEL®, logo and combinations thereof, Everywhere You Are®, AVR®, AVR Studio®, and others are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

LM2902, LM324/LM324A, LM224/ LM224A

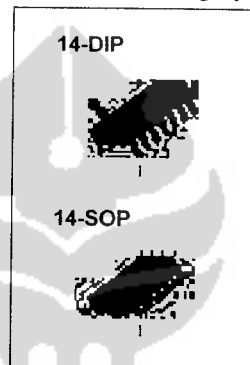
Quad Operational Amplifier

Features

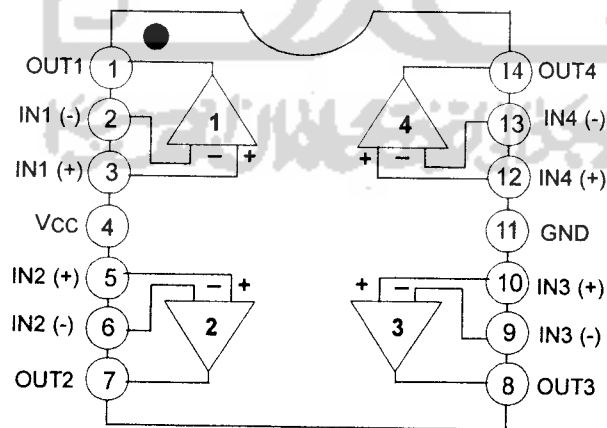
- Internally Frequency Compensated for Unity Gain
- Large DC Voltage Gain: 100dB
- Wide Power Supply Range:
LM224/LM224A, LM324/LM324A : 3V~32V (or $\pm 1.5 \sim 16V$)
LM2902: 3V~26V (or $\pm 1.5V \sim 13V$)
- Input Common Mode Voltage Range Includes Ground
- Large Output Voltage Swing: 0V to $V_{CC} - 1.5V$
- Power Drain Suitable for Battery Operation

Description

The LM324/LM324A, LM2902, LM224/LM224A consist of four independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide voltage range. operation from split power supplies is also possible so long as the difference between the two supplies is 3 volts to 32 volts. Application areas include transducer amplifier, DC gain blocks and all the conventional OP Amp circuits which now can be easily implemented in single power supply systems.

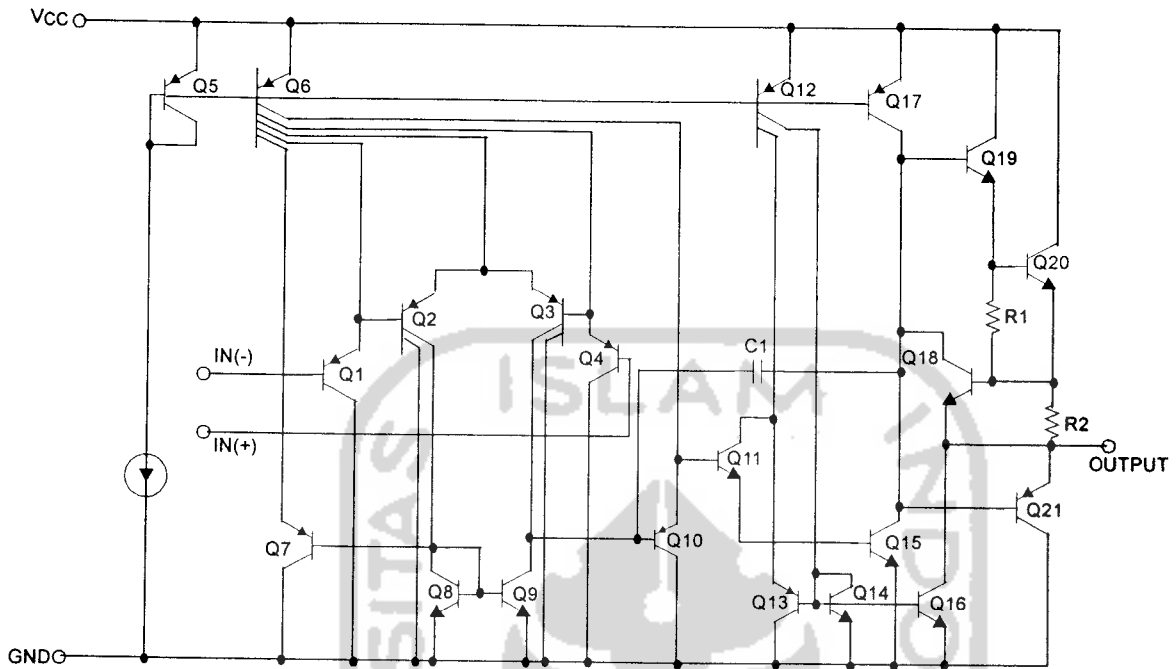


Internal Block Diagram



Schematic Diagram

(One Section Only)



Absolute Maximum Ratings

Parameter	Symbol	LM224/LM224A	LM324/LM324A	LM2902	Unit
Power Supply Voltage	VCC	±16 or 32	±16 or 32	±13 or 26	V
Differential Input Voltage	VI(DIFF)	32	32	26	V
Input Voltage	VI	-0.3 to +32	-0.3 to +32	-0.3 to +26	V
Output Short Circuit to GND VCC≤15V, TA=25°C(one Amp)	-	Continuous	Continuous	Continuous	-
Power Dissipation, TA=25°C	PD	1310	1310	1310	mW
14-DIP		640	640	640	
14-SOP					
Operating Temperature Range	TOPR	-25 ~ +85	0 ~ +70	-40 ~ +85	°C
Storage Temperature Range	TSTG	-65 ~ +150	-65 ~ +150	-65 ~ +150	°C

Thermal Data

Parameter	Symbol	Value	Unit
Thermal Resistance Junction-Ambient Max.			
14-DIP	Rθja	95	°C/W
14-SOP		195	

Electrical Characteristics

($V_{CC} = 5.0V$, $V_{EE} = GND$, $T_A = 25^\circ C$, unless otherwise specified)

Parameter	Symbol	Conditions	LM224			LM324			LM2902			Unit	
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.		
Input Offset Voltage	V_{IO}	$V_{CM} = 0V$ to $V_{CC} - 1.5V$ $V_{O(P)} = 1.4V$, $R_S = 0\Omega$ (Note1)	-	1.5	5.0	-	1.5	7.0	-	1.5	7.0	mV	
Input Offset Current	I_{IO}	$V_{CM} = 0V$	-	2.0	30	-	3.0	50	-	3.0	50	nA	
Input Bias Current	I_{BIAS}	$V_{CM} = 0V$	-	40	150	-	40	250	-	40	250	nA	
Input Common-Mode Voltage Range	$V_{I(R)}$	Note1	0	-	$V_{CC} - 1.5$	0	$V_{CC} - 1.5$	-	0	-	$V_{CC} - 1.5$	V	
Supply Current	I_{CC}	$R_L = \infty$, $V_{CC} = 30V$ (LM2902, $V_{CC} = 26V$)	-	1.0	3	-	1.0	3	-	1.0	3	mA	
		$R_L = \infty$, $V_{CC} = 5V$	-	0.7	1.2	-	0.7	1.2	-	0.7	1.2	mA	
Large Signal Voltage Gain	G_V	$V_{CC} = 15V$, $R_L = 2k\Omega$ $V_{O(P)} = 1V$ to $11V$	50	100	-	25	100	-	25	100	-	V/ mV	
Output Voltage Swing	$V_{O(H)}$	Note1	$R_L = 2k\Omega$	26	-	-	26	-	-	22	-	-	V
			$R_L = 10k\Omega$	27	28	-	27	28	-	23	24	-	V
	$V_{O(L)}$	$V_{CC} = 5V$, $R_L = 10k\Omega$	-	5	20	-	5	20	-	5	100	mV	
Common-Mode Rejection Ratio	CMRR	-	70	85	-	65	75	-	50	75	-	dB	
Power Supply Rejection Ratio	PSRR	-	65	100	-	65	100	-	50	100	-	dB	
Channel Separation	CS	$f = 1kHz$ to $20kHz$ (Note2)	-	120	-	-	120	-	-	120	-	dB	
Short Circuit to GND	I_{SC}	$V_{CC} = 15V$	-	40	60	-	40	60	-	40	60	mA	
Output Current	I_{SOURCE}	$V_{I(+)} = 1V$, $V_{I(-)} = 0V$ $V_{CC} = 15V$ $V_{O(P)} = 2V$	20	40	-	20	40	-	20	40	-	mA	
		$V_{I(+)} = 0V$, $V_{I(-)} = 1V$ $V_{CC} = 15V$ $V_{O(P)} = 2V$	10	13	-	10	13	-	10	13	-	mA	
	I_{SINK}	$V_{I(+)} = 0V$, $V_{I(-)} = 1V$ $V_{CC} = 5V$, $V_{O(R)} = 200mV$	12	45	-	12	45	-	-	-	-	μA	
Differential Input Voltage	$V_{I(DIFF)}$	-	-	V_{CC}	-	-	V_{CC}	-	-	V_{CC}	V		

Note :

- $V_{CC} = 30V$ for LM224 and LM324, $V_{CC} = 26V$ for LM2902
- This parameter, although guaranteed, is not 100% tested in production.

Electrical Characteristics (Continued)

(VCC = 5.0V, VEE = GND, unless otherwise specified)

The following specifications apply over the range of $-25^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for the LM224; and the $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for the LM324; and the $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for the LM2902

Parameter	Symbol	Conditions	LM224			LM324			LM2902			Unit	
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.		
Input Offset Voltage	V_{IO}	$V_{ICM} = 0\text{V to } V_{CC} - 1.5\text{V}$ $V_{O(P)} = 1.4\text{V}$, $R_S = 0\Omega$ (Note1)	-	-	7.0	-	-	9.0	-	-	10.0	mV	
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	$R_S = 0\Omega$ (Note2)	-	7.0	-	-	7.0	-	-	7.0	-	$\mu\text{V}/^{\circ}\text{C}$	
Input Offset Current	I_{IO}	$V_{CM} = 0\text{V}$	-	-	100	-	-	150	-	-	200	nA	
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	$R_S = 0\Omega$ (Note2)	-	10	-	-	10	-	-	10	-	$\text{pA}/^{\circ}\text{C}$	
Input Bias Current	I_{BIAS}	$V_{CM} = 0\text{V}$	-	-	300	-	-	500	-	-	500	nA	
Input Common-Mode Voltage Range	$V_{I(R)}$	Note1	0	-	$V_{CC} - 2.0$	0	-	$V_{CC} - 2.0$	0	-	$V_{CC} - 2.0$	V	
Large Signal Voltage Gain	G_V	$V_{CC} = 15\text{V}$, $R_L = 2.0\text{k}\Omega$ $V_{O(P)} = 1\text{V to } 11\text{V}$	25	-	-	15	-	-	15	-	-	V/mV	
Output Voltage Swing	$V_{O(H)}$	Note1	$R_L = 2\text{k}\Omega$	26	-	-	26	-	-	22	-	-	V
			$R_L = 10\text{k}\Omega$	27	28	-	27	28	-	23	24	-	V
	$V_{O(L)}$	$V_{CC} = 5\text{V}$, $R_L = 10\text{k}\Omega$	-	5	20	-	5	20	-	5	100	mV	
Output Current	ISOURCE	$V_{I(+)} = 1\text{V}$, $V_{I(-)} = 0\text{V}$ $V_{CC} = 15\text{V}$, $V_{O(P)} = 2\text{V}$	10	20	-	10	20	-	10	20	-	mA	
	ISINK	$V_{I(+)} = 0\text{V}$, $V_{I(-)} = 1\text{V}$ $V_{CC} = 15\text{V}$, $V_{O(P)} = 2\text{V}$	10	13	-	5	8	-	5	8	-	mA	
Differential Input Voltage	$V_{I(DIFF)}$	-	-	-	V_{CC}	-	-	V_{CC}	-	-	V_{CC}	V	

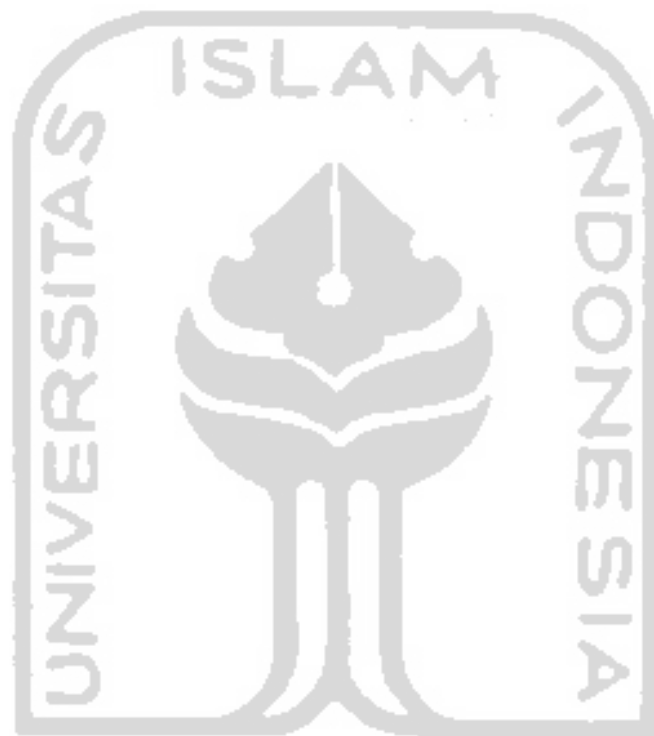
Note:

1. $V_{CC} = 30\text{V}$ for LM224 and LM324, $V_{CC} = 26\text{V}$ for LM2902
2. These parameters, although guaranteed, are not 100% tested in production.

Ordering Information

Product Number	Package	Operating Temperature
LM324N	14-DIP	0 ~ +70°C
LM324AN		
LM324M	14-SOP	
LM324AM		
LM2902N	14-DIP	-40 ~ +85°C
LM2902M	14-SOP	
LM224N	14-DIP	-25 ~ +85°C
LM224AN		
LM224M	14-SOP	
LM224AM		





DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Electrical Characteristics (Continued)(V_{CC} = 5.0V, V_{EE} = GND, T_A = 25°C, unless otherwise specified)

Parameter	Symbol	Conditions	LM224A			LM324A			Unit	
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Input Offset Voltage	V _{IO}	V _{CM} = 0V to V _{CC} -1.5V V _{O(P)} = 1.4V, R _S = 0Ω (Note1)	-	1.0	3.0	-	1.5	3.0	mV	
Input Offset Current	I _{IO}	V _{CM} = 0V	-	2	15	-	3.0	30	nA	
Input Bias Current	I _{BIAS}	V _{CM} = 0V	-	40	80	-	40	100	nA	
Input Common-Mode Voltage Range	V _{I(R)}	V _{CC} = 30V	0	-	V _{CC} -1.5	0	-	V _{CC} -1.5	V	
Supply Current	I _{CC}	V _{CC} = 30V, R _L = ∞	-	1.5	3	-	1.5	3	mA	
		V _{CC} = 5V, R _L = ∞	-	0.7	1.2	-	0.7	1.2	mA	
Large Signal Voltage Gain	G _V	V _{CC} = 15V, R _L = 2kΩ V _{O(P)} = 1V to 11V	50	100	-	25	100	-	V/mV	
Output Voltage Swing	V _{O(H)}	Note1	R _L = 2kΩ	26	-	-	26	-	-	V
			R _L = 10kΩ	27	28	-	27	28	-	V
	V _{O(L)}	V _{CC} = 5V, R _L = 10kΩ	-	5	20	-	5	20	mV	
Common-Mode Rejection Ratio	CMRR	-	70	85	-	65	85	-	dB	
Power Supply Rejection Ratio	PSRR	-	65	100	-	65	100	-	dB	
Channel Separation	CS	f = 1kHz to 20kHz (Note2)	-	120	-	-	120	-	dB	
Short Circuit to GND	I _{SC}	V _{CC} = 15V	-	40	60	-	40	60	mA	
Output Current	I _{SOURCE}	V _{I(+)} = 1V, V _{I(-)} = 0V V _{CC} = 15V, V _{O(P)} = 2V	20	40	-	20	40	-	mA	
		V _{I(+)} = 0V, V _{I(-)} = 1V V _{CC} = 15V, V _{O(P)} = 2V	10	20	-	10	20	-	mA	
	I _{SINK}	V _{I(+)} = 0V, V _{I(-)} = 1V V _{CC} = 5V V _{O(P)} = 200mV	12	50	-	12	50	-	μA	
Differential Input Voltage	V _{I(DIFF)}	-	-	-	V _{CC}	-	-	V _{CC}	V	

Note:

- V_{CC} = 30V for LM224A, LM324A
- This parameter, although guaranteed, is not 100% tested in production.

Electrical Characteristics (Continued)

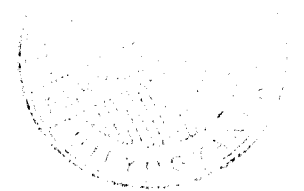
(VCC = 5.0V, VEE = GND, unless otherwise specified)

The following specifications apply over the range of $-25^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for the LM224A; and the $0^{\circ}\text{C} \leq T_A \leq +70^{\circ}\text{C}$ for the LM324A

Parameter	Symbol	Conditions	LM224A			LM324A			Unit	
			Min.	Typ.	Max.	Min.	Typ.	Max.		
Input Offset Voltage	V _{IO}	V _{CM} = 0V to V _{CC} -1.5V V _{O(P)} = 1.4V, R _S = 0Ω (Note1)	-	-	4.0	-	-	5.0	mV	
Input Offset Voltage Drift	ΔV _{IO} /ΔT	R _S = 0Ω (Note2)	-	7.0	20	-	7.0	30	μV/°C	
Input Offset Current	I _{IO}	V _{CM} = 0V	-	-	30	-	-	75	nA	
Input Offset Current Drift	ΔI _{IO} /ΔT	R _S = 0Ω (Note2)	-	10	200	-	10	300	pA/°C	
Input Bias Current	I _{BIAS}	-	-	40	100	-	40	200	nA	
Input Common-Mode Voltage Range	V _{I(R)}	Note1	0	-	V _{CC} -2.0	0	-	V _{CC} -2.0	V	
Large Signal Voltage Gain	G _V	V _{CC} = 15V, R _L = 2.0kΩ	25	-	-	15	-	-	V/mV	
Output Voltage Swing	V _{O(H)}	Note1	R _L = 2kΩ	26	-	-	26	-	-	V
		R _L = 10kΩ	27	28	-	27	28	-	-	V
	V _{O(L)}	V _{CC} = 5V, R _L = 10kΩ	-	5	20	-	5	20	mV	
Output Current	I _{SOURCE}	V _{I(+)} = 1V, V _{I(-)} = 0V V _{CC} = 15V, V _{O(P)} = 2V	10	20	-	10	20	-	mA	
	I _{SINK}	V _{I(+)} = 0V, V _{I(-)} = 1V V _{CC} = 15V, V _{O(P)} = 2V	5	8	-	5	8	-	mA	
Differential Input Voltage	V _{I(DIFF)}	-	-	-	V _{CC}	-	-	V _{CC}	V	

Note:

- V_{CC} = 30V for LM224A and LM324A.
- These parameters, although guaranteed, are not 100% tested in production.



Typical Performance Characteristics

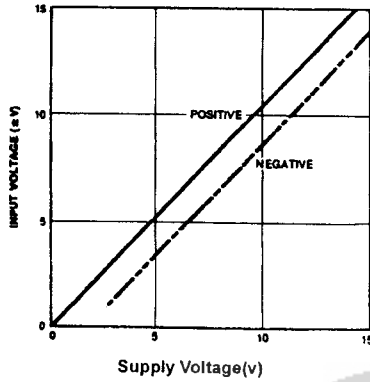


Figure 1. Input Voltage Range vs Supply Voltage

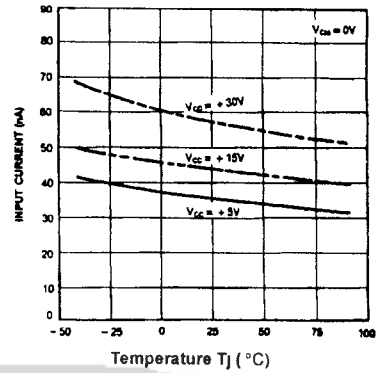


Figure 2. Input Current vs Temperature

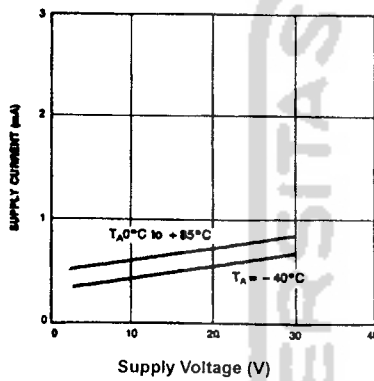


Figure 3. Supply Current vs Supply Voltage

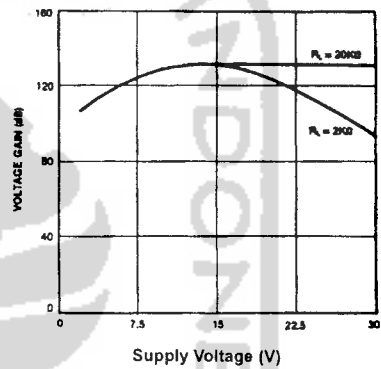


Figure 4. Voltage Gain vs Supply Voltage

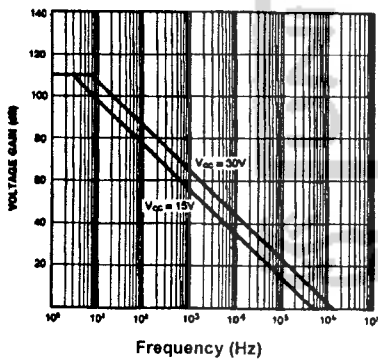


Figure 5. Open Loop Frequency Response

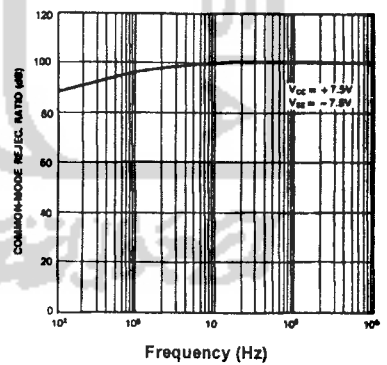


Figure 6. Common mode Rejection Ratio

Typical Performance Characteristics (Continued)

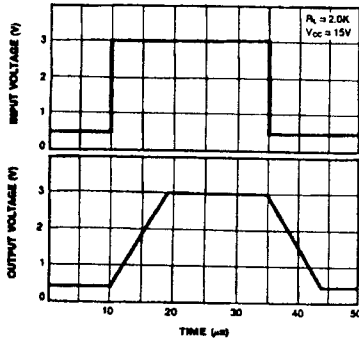


Figure 7. Voltage Follower Pulse Response

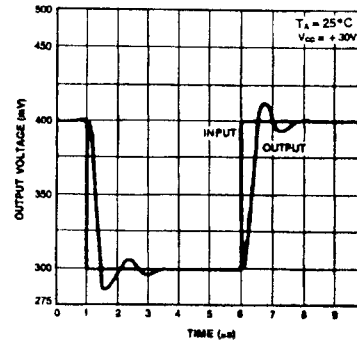


Figure 8. Voltage Follower Pulse Response (Small Signal)

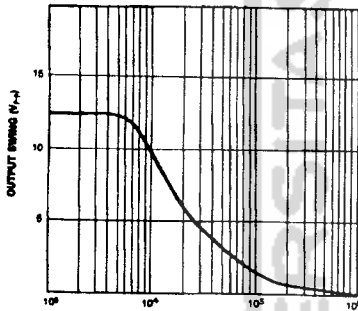


Figure 8. Large Signal Frequency Response

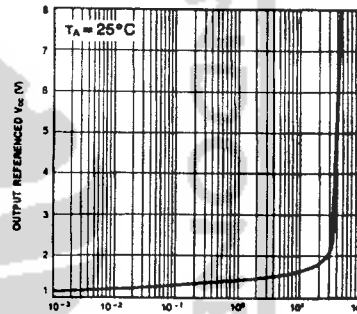


Figure 9. Output Characteristics vs Current Sourcing

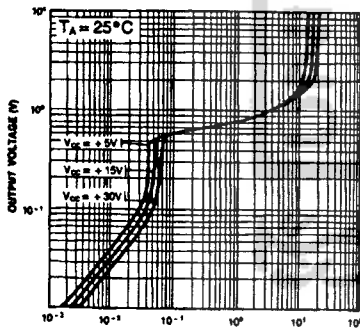


Figure 10. Output Characteristics vs Current Sinking

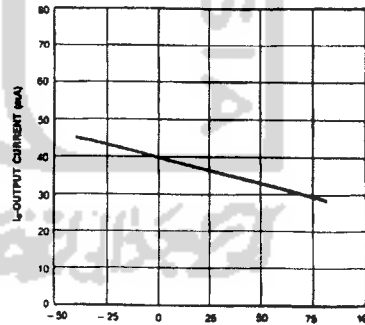


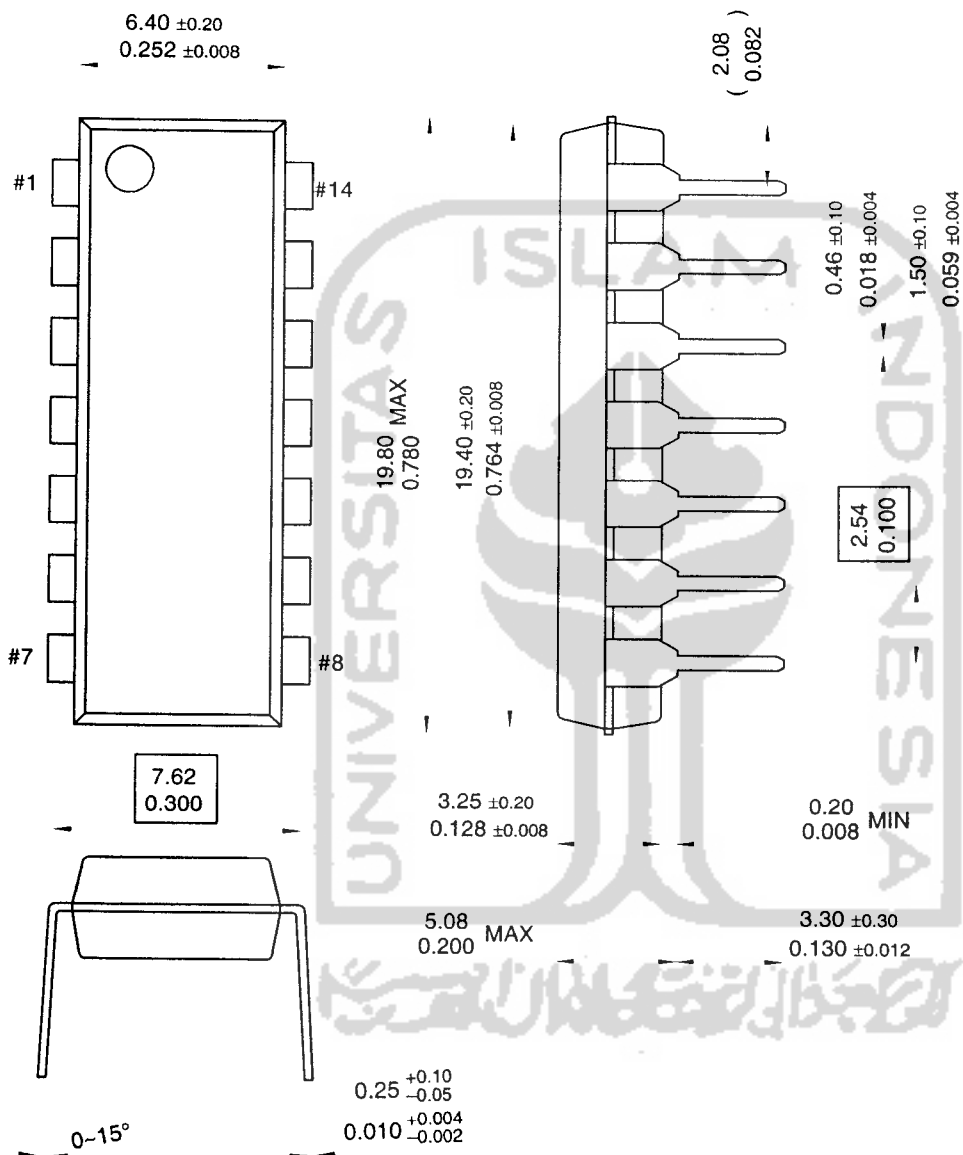
Figure 11. Current Limiting vs Temperature

Mechanical Dimensions

Package

Dimensions in millimeters

14-DIP



Mechanical Dimensions (Continued)

Package

Dimensions in millimeters

14-SOP

