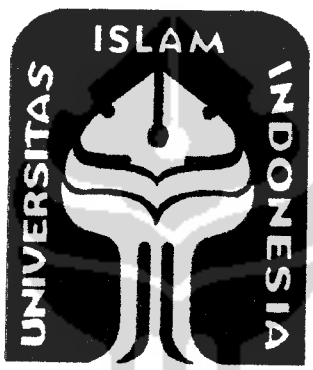
PERANCANGAN DAN PENGENDALIAN ROBOT LIMA KAKI MENGGUNAKAN MIKROKONTROLER AT89S51

TUGAS AKHIR

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



Disusun Oleh :

Nama : **Andri Permadi**

No. Mahasiswa : **00 524 123**

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2007

LEMBAR PENGESAHAN PEMBIMBING

PERANCANGAN DAN PENGENDALIAN ROBOT LIMA KAKI MENGGUNAKAN MIKROKONTROLER AT89S51

TUGAS AKHIR

Disusun oleh :

Nama : Andri Permadi

No. Mhs : 00524123

Yogyakarta, Juni 2007

Pembimbing I

(Wahyudi Budi Pramono, ST)

Pembimbing II

(Yusuf Aziz Amrulloh, ST)

LEMBAR PENGESAHAN PENGUJI

Perancangan dan Pengendalian Robot Lima Kaki Menggunakan Mikrokontroler AT89S51

TUGAS AKHIR

Disusun oleh :

Nama : Andri Permadi

No. Mhs : 00524123

Telah Dipertahankan di Depan Sidang Penguji sebagai
Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Juli 2007

Tim Penguji

Wahyudi Budi Pramono, ST
Ketua

Yusuf Aziz Amrullah, ST
Anggota I

Hendra Setiawan, ST.,MT
Anggota II

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

Tito Yuwono, ST.,M.Sc

ABSTRAK

Perkembangan teknologi saat ini sudah semakin pesat pada berbagai bidang, khususnya pada bidang robotika. Indonesia sendiri sudah memanfaatkan teknologi robotika meski masih jauh tertinggal jika dibandingkan negara-negara tetangga dan negara maju lainnya. Salah satunya dengan menyelenggarakan Kontes Robot Indonesia (KRI) sejak tahun 1990 oleh Depdiknas, demi perkembangan Sumber Daya Manusia Indonesia untuk dapat menguasai teknologi robot. Dalam perancangan dan pengendalian robot lima kaki terbagi menjadi dua, perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras terdiri dari mekanik dan elektronik, sedangkan perancangan perangkat lunak menggunakan mikrokontroler jenis AT89S51 sebagai pengendali utama pada robot robot lima kaki. Menggunakan motor servo sebagai penggerak kaki robot dan putarannya diatur dengan membangkitkan sinyal PWM pada mikrokontroler. Pergerakan kaki robot lima kaki mengikuti konsep urutan pergerakan yang meliputi gerakan kaki naik, gerakan kaki geser kedepan, gerakan kaki turun dan gerakan kaki geser kebelakang. Pada saat pergerakan kaki robot telah sesuai dengan konsep urutan pergerakan kaki, ternyata pergerakan sistem robot mengalami kesulitan. Salah satunya dengan memerlukan bantuan penyangga ketika robot bergerak agar robot tidak roboh. Untuk penelitian lebih lanjut sebaiknya melakukan pembuatan *prototype* miniature robot terlebih dahulu dan pembuatan mekanik yang tepat terutama kepresisian gir agar gerakan kaki robot dapat lebih baik lagi.

Kata kunci : robot, gir, motor, PWM dan mikrokontroler.

MOTTO

"Sesungguhnya shalatku, ibadahku, hidup dan matiku

hanya untuk Allah Tuhan semesta alam..."

(Al An'aam : 162)

Siapa saja yang menduga bahwa apabila seseorang mencurahkan tenaganya untuk mencapai tujuan, berarti dia tertolong. Barangsiapa yang menduga tanpa jerih payah ia akan mencapai tujuannya,berarti ia hanya berangan-angan. Maka perbaikilah pekerjaanmu niscaya doamu dikabulkan...

(HR Thabrani)

Orang yang pesimis melihat kesukaran dibalik kesempatan, sedangkan orang yang optimis melihat kesempatan disetiap kesukaran...

(Sir Wiston Churchill)

"Sebaik-baik manusia ialah orang yang banyak manfaatnya (kebaikannya) kepada manusia lainnya..."

(H.R. Qadla'ie dari Jabir)

"Rasa bahagia, rasa duka, rasa cinta dan kasih

selalu datang dari hati...

Dan Ingatlah Allah pada setiap rasa itu..."

LEMBAR PERSEMBAHAN

"*Untuk mereka tercintalah karya ini kupersembahkan...*"

Kepada Ayah dan Ibu yang telah membesarkankan aku
dan selalu berdoa dengan nasehat - nasehat yang
selalu mengiringi perjalananku...

Adikku Alice Setiawati yang selalu memberi dukungan
dan jadilah perawat yang ramah dan baik...

Segenap keluarga dan saudaraku di Klaten dan Solo,
terima kasih buat semua nasehat dan dukungannya...

Yang senantiasa menemaniku dan memberi semangat,

Yulina Fitrianingrum,

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Assalamulaikum Wr. Wb.

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, berkat ridlo-Nya saya dapat menyelesaikan dengan baik penyusunan tugas akhir (TA) dengan judul "**Perancangan dan Pengendalian Robot Lima Kaki Menggunakan Mikrokontroler AT89S51**" dan tidak lupa juga kita berikan shalawat serta salam pada junjungan kita Nabi besar Muhammad SAW beserta keluarga dan para pengikutnya sampai akhir zaman.

Adapun maksud dari penyusunan tugas akhir ini adalah untuk memenuhi kurikulum S-1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Disamping itu untuk menambah pengetahuan terhadap ilmu yang telah dipelajari di bangku perkuliahan untuk diterapkan ke masyarakat.

Dalam menyelesaikan penyusunan tugas akhir ini tidak terlepas dari berbagai pihak yang memberikan bantuan dan dukungan, maka penyusun mengucapkan terima kasih kepada:

1. Bapak Fathul Wahid, ST.M.Sc, selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST.M.Sc, selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

3. Bapak Wahyudi Budi Pramono, ST, selaku Dosen Pembimbing I.
4. Bapak Yusuf Azis Amrulloh, ST, selaku Dosen Pembimbing II.
5. Segenap dosen Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmunya.
6. Ayahanda dan Ibunda serta adikku Alice Setiawati, yang selalu mendoakan dan memberikan dukungan baik moril maupun spiritual dalam penyusunan tugas akhir.
7. Mas Heri, Mas Agung, Mas Tri dan seluruh asisten di laboratorium Jurusan Teknik Elektro UII. Terima kasih atas bantuan dan dukungannya.
8. Teman-teman Teknik Elektro angkatan 2000 Universitas Islam Indonesia.
9. Teman-teman kost Pandega Wreksa, terima kasih atas bantuan, doa dan dukungannya selama ini.
10. Semua pihak yang telah turut membantu penyusunan tugas akhir yang tidak dapat penyusun 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.

Wassalamulaikum Wr. Wb.

Yogyakarta, Juni 2007

Penyusun,

Andri Permadi

TAKARIR

AC (Alternating Current)

Adalah sistem arus listrik. Sistem AC adalah cara bekerjanya arus bolak-balik. Dimana arus yang berskala dengan harga rata-rata selama satu periode atau satu masa kerjanya dimana periodenya adalah nol.

Akumulator

Register yang digunakan untuk menyimpan semua proses aritmatika

ASCII (*American Standarts Committee for Information Interchange*)

Kode ini merupakan kode alphanmeric yang masing-masing kode berisi 7-bit karakter.

Assembler

Bahasa pemrograman mikrokontroler MCS-51

Assembly Listing

Hasil dari proses assembly dalam rupa campuran dari program objek, program sumber dan alamat-alamatnya

Atmel

Sebuah perusahaan pembuat *flash memory* berarsitektur MCS-51

Bit

Bit merupakan ukuran terkecil dalam data digital. Bit terdiri dari angka 0 dan 1

Byte

Byte adalah merupakan kumpulan beberapa bit (1 Byte = 8 bit)

Boucing

Logika saklar yang tidak dapat diperkirakan

Chip

Sebuah kepingan IC

CGRAM (*Character Random Access Memory*)

Memori untuk membentuk pola karakter pada LCD

CGROM (*Character Generator Read Only Memory*)

Memori untuk menyimpan pola karakter pada LCD

CPU (*Central Processing Unit*)

Unit pusat pemrosesan. Biasanya disebut sebagai otak komputer itu sendiri

Cycle

Kecepatan siklus mesin program

Data Pointer (DPTR)

Media yang digunakan untuk membuat alamat berukuran 16 bit untuk mengakses memori luar.

DDRAM (*Display Data Random Access Memory*)

Memori tempat karakter yang ditampilkan pada LCD

Delay

Waktu tunda

Digital

Data dalam bentuk angka 0 dan 1

Downloader

Perangkat yang digunakan untuk mengisi program dari komputer ke mikrokontroler

EEPROM (*Elecrtrical Erasable Programable Read Only Memory*)

Momori yang digunakan untuk menyimpan instruksi- instruksi MCS-51

Ground

Titik referensi tegangan biasanya untuk menentukan 0 V

Hardware

Perangkat keras pada suatu sistem atau alat yang berupa satu kesatuan komponen elektrik dan elektronik

Heksadesimal

Penulisan angka dalam format 16-an

IC (*Integrated Circuit*)

Sebuah alat yang didalamnya terdapat rangkaian elektronis terpadu dengan fungsi tertentu

Input

Masukan bagi alat atau sistem

Interface

Antamuka. Penghubung antara dua sistema atau alat. Media penghubung antara satu sub sistem dengan sub sistem lainnya

Interupsi

Sela atau pemberhentian sesuatu untuk sementara waktu

Keypad

Papan tombol terdiri dari beberapa tombol

LCD (*Liquid Crystal Display*)

Suatu penampil (Display) dari bahan cairan kristal yang pengoperasiannya menggunakan sistem dot matrix

Magnetic Card Reader

Sebuah alat yang digunakan untuk membaca kartu magnetik

MCS-51

Keluarga mikrokontroler Atmel AT89XX

MHz (*Mega Hertz*)

Jutaan gerak per detik

Mikrokontroler

Sebuah alat atau IC kecil yang dapat digunakan untuk mengendalikan sebuah sistem

Nibble

Nibble adalah merupakan kumpulan beberapa bit (1 Nibble = 4 bit)

Output

Keluaran dari alat atau sistem

PCB

Suatu papan yang berfungsi sebagai tempat terpasang dan tersambungnya berbagai komponen eletronik secara terpadu

Port

Sebuah jalur atau pintu yang dapat digunakan sebagai masukan atau keluaran

RAM (*Random Access Memory*)

Memori yang dapat diolah secara acak, biasanya digunakan sebagai penyimpan data untuk sementara waktu

Register

Sebuah kumpulan data digital dalam mikrokontroler, dapat digunakan untuk mengatur atau melihat keadaan mikrokontroler

ROM (*Read Only Memory*)

Memori yang hanya dapat dibaca, biasanya digunakan untuk menyimpan data program yang akan dijalankan pada mikrokontroler

RST (*Reset*)

Keadaan awal dari sistem

SFR (*Special Function Register*)

Merupakan register yang memiliki fungsi- fungsi khusus

SPI (*Serial Peripheral Interface*)

Antar muka untuk pemrograman serial

Timer/Counter

Aplikasi mikrokontroler untuk pewaktu dan penghitung yang dibedakan dari pemberian input *clock*.

DAFTAR ISI

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

BAB I

PENDAHULUAN

1.1. Latar Belakang

Saat ini secara sadar atau tidak, robot memang telah hadir di dalam kehidupan manusia dalam bentuk yang bermacam-macam. Terdapat bentuk desain robot yang sederhana untuk mengerjakan kegiatan yang mudah atau berulang-ulang. Di kalangan umum pengertian robot selalu dikaitkan dengan "makhluk hidup" berbentuk orang maupun binatang yang terbuat dari logam dan bertenaga listrik. Sementara itu dalam arti luas robot berarti alat yang dalam batas-batas tertentu dapat bekerja sendiri (otomatis) sesuai dengan perintah yang sudah diberikan oleh perancangnya. Dengan pengertian ini sangat erat hubungan antara robot dan otomatisasi sehingga dapat dipahami bahwa hampir setiap aktivitas kehidupan modern makin tergantung pada robot dan otomatisasi.

Indonesia sendiri sudah memanfaatkan teknologi robotika meski masih jauh tertinggal jika dibandingkan negara-negara tetangga dan negara maju lainnya. Salah satunya dengan menyelenggarakan Kontes Robot Indonesia (KRI) sejak tahun 1990 oleh Depdiknas, demi perkembangan Sumber Daya Manusia Indonesia untuk dapat menguasai teknologi robot.

Untuk menjawab tantangan tersebut, maka di buatlah salah satu sistem robot sederhana yaitu perancangan dan pengendalian robot lima kaki. Perancangan dan pengendalian robot lima kaki ini meliputi, perancangan perangkat keras pada mekanik serta rangkaian elektroniknya dan perangkat lunak pada pengendalian

program pergerakan sistem robot. Untuk pengedalian robot menggunakan mikrokontroler melalui program yang *didownload* sehingga sistem robot dapat bergerak sesuai dengan keinginan.

Robot lima kaki ini masih tergolong robot sederhana, yang dirancang hanya untuk bergerak maju dengan mengikuti konsep urutan pergerakan kaki robot. Konsep urutan pergerakan kaki robot lima kaki meliputi, kaki robot bergerak naik, bergerak geser kedepan, bergerak turun dan bergerak geser kebelakang. Jika semua konsep urutan pergerakan kaki pada robot lima kaki terpenuhi, maka diharapkan sistem robot dapat bekerja sesuai dengan yang diinginkan.

1.2. Rumusan Masalah

Dari uraian latar belakang dan dasar pemikiran di atas, maka dapat diambil suatu rumusan masalah "Bagaimana merancang dan mengendalikan robot lima kaki dengan menggunakan mikrokontroler.

1.3. Batasan Masalah

Agar penulisan lebih terarah, maka pembahasan penulisan ini dibatasi pada ruang lingkup pembahasan sebagai berikut :

1. Robot mempunyai 5 buah kaki.
2. Menggunakan motor servo sebagai penggerak kaki robot.
3. Menggunakan mikrokontroler AT89S51 sebagai pengendali pergerakan kaki-kaki robot.

4. Arah pergerakan robot maju, dengan pergerakan kaki meliputi gerakan kaki naik, kaki geser kedepan, gerak kaki turun dan kaki geser kebelakang.
5. Perangkat keras *(hardware)* menggunakan bahan akrilik.

1.4. Tujuan dan Manfaat Penelitian Tugas Akhir

Tujuan dan manfaat yang akan dicapai dalam penulisan tugas akhir ini adalah sebagai berikut :

1. Merancang dan membangun suatu sistem robot berbasis mikrokontroler yang dapat dikendalikan untuk bergerak sesuai dengan yang diinginkan.
2. Memperdalam pengetahuan dan pemahaman tentang konsep *software* dan *hardware* dalam suatu pemanfaatan sistem komputer sebagai pendukung dalam pembuatan sistem robot.

1.5. Sistematika Penulisan

Dalam sistematika penulisan tugas akhir ini diberikan uraian bab demi bab yang berurutan untuk mempermudah pembahasannya. Pokok-pokok permasalahan dalam penulisan ini dibagi menjadi lima bab :

BAB I Pendahuluan

Bab ini merupakan pengantar permasalahan yang dibahas seperti latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penulisan, manfaat penulisan dan sistematika penulisan.

BAB II Landasan Teori

Bab ini merupakan penjelasan secara terperinci mengenai teori-teori yang digunakan sebagai landasan untuk pemecahan masalah. Memberikan garis besar metode yang digunakan oleh peneliti sebagai kerangka pemecahan masalah.

BAB III Perancangan Sistem

Bab ini menjelaskan perancangan sistem, komponen yang digunakan serta desain perangkat keras dan perangkat lunaknya.

BAB IV Analisa dan Pembahasan

Bab ini membahas hasil pengujian sistem robot yang dibuat dan membandingkan hasil pengujian dengan dasar teori.

BAB V Kesimpulan dan Saran

Bab ini berisikan kesimpulan dan saran-saran yang diperoleh dari hasil perancangan, implementasi sistem, juga keterbatasan-keterbatasan yang ditemukan dengan asumsi yang dibuat selama melakukan tugas akhir.

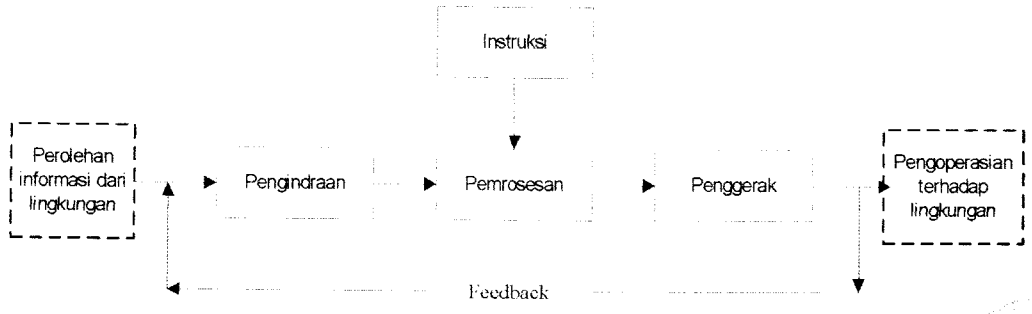
BAB II

LANDASAN TEORI

2.1. Robot dan Sistem Robotika

Robot menurut beberapa literatur adalah mesin yang dapat diprogram dan mempunyai sifat *anthropomopile* atau menyerupai manusia. *British Robot Association* menyatakan robot sebagai *"a reprogramable device designed to both manipulate and transport part, tools or specialised manufacturing implements through variabel programmed motions for the performance of specific manipulating task"* atau "sebuah robot merupakan manipulator berfungsi ganda yang dapat diprogramkan kembali serta dirancang untuk menggerakan bahan, onderdil, perkakas atau peralatan khusus melalui gerakan-gerakan terprogram untuk melaksanakan berbagai tugas khusus.

Sistem robot memiliki 3 bagian yang saling berhubungan, seperti pada Gambar 2.1. dijelaskan bahwa komponen pengolahan (pemrosesan) merupakan pusat dari kendali yang mendapatkan informasi dari luar dari sistem penginderaan dan setelah diolah akan diteruskan sebagai suatu perintah yang mengawali sejumlah tindakan pada bagian lainnya.



Gambar 2.1 Bagian utama sistem robotik

Penggolongan robot dijadikan dua kelompok besar, yaitu *service robot* dan *mobile robot*. Sedangkan *mobile robot* dibagi dalam dua jenis, yaitu robot beroda *(wheeled robot)* dan robot dengan kaki *(legged walking robot)*. Meskipun definisi robot beragam, dunia robotika tak lepas dari sistem kendali *(control system)*. Robot dapat dikontrol oleh manusia, dirinya sendiri maupun oleh robot lainnya. Akan tetapi robot juga dapat bergerak sendiri secara terkendali dengan perintah yang dapat diproses oleh sistem kendali yang ada pada robot tersebut.

Aktifitas robot yang terkendali didalam ruangan berkaitan dengan 6 (enam) faktor yang perlu dipahami dalam membangun sebuah robot, yaitu :

1. Kinematika, bidang ilmu yang mempelajari geometri gerakan robot tanpa memperhatikan gaya yang terjadi pada robot.
2. Dinamika, merupakan bidang ilmu untuk mempelajari gaya yang membuat robot dapat bergerak.
3. *Control System*, mempelajari cara agar robot dapat bergerak secara tekendali.
4. Sensor, dikelompokan kedalam *vision* dan *non-vision*, dimana *non-vision* ini dapat dibagi lagi kedalam sub kelompok *contact* dan *non-contact*.
5. *Navigation*, bidang ilmu yang diperlukan terutama untuk *mobile robot*, karena robot harus mampu menentukan arah jalannya sehingga dapat tiba ditempat tujuan dengan posisi dan orientasi yang benar.
6. Mikrokontroler, otak dalam melakukan aktifitas kelima elemen faktor diatas. Keluaran yang didapat kemudian menjadi masukan yang akan diolah sesuai dengan program yang telah dibuat sebelumnya.

2.2. Mikrokontroler

Mikrokontroler adalah suatu alat atau komponen pengontrol atau pengendali yang berukuran kecil (mikro). Sebelum mikrokontroler ada, telah terlebih dahulu muncul apa yang disebut dengan mikroprosesor. Bila dibandingkan dengan mikroprosesor, mikrokontroler jauh lebih unggul. Alasannya adalah sebagai berikut :

1. Tersedia I/O

   I/O dalam mikrokontroler sudah tersedia, bahkan untuk AT89S51 ada 32 jalur I/O. Sementara pada mikroprosesor dibutuhkan IC tambahan untuk menangani I/O tersebut.

2. Memori internal

   Memori merupakan media untuk menyimpan program dan data sehingga mutlak harus ada. Mikroprosesor belum memiliki memori internal sehingga memerlukan IC memori eksternal.

3. Harga

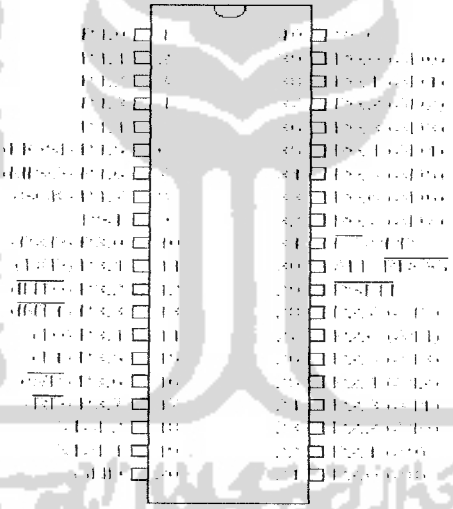
   Mikrokontroler harganya relatif murah sehingga terjangkau dibandingkan dengan mikroprosesor

Meski memiliki kelebihan dan kelemahan, mikroprosesor dan mikrokontroler mempunyai inti kerja yang sama, yaitu sebagai pengendali atau pengontrol utama suatu rangkaian.

2.3. Arsitektur Mikrokontroler AT89S51

Mikrokontroler yang digunakan dalam sistem kendali robot ini adalah mikrokontroler dari keluarga Atmel yaitu AT89S51 yang memiliki spesifikasi sebagai berikut:

1. Kompatibel dengan produk dan program assembler MCS-51.
2. 4 Kb memory program yang dapat ditulis hingga 1000 kali.
3. Kecepatan clock 0 – 33MHz.
4. 32 pin Input/Output (4 buah port parallel I/O) yang dapat diprogram.
5. 128 byte memory RAM internal.
6. 2 buah timer / counter 16 bit.
7. 6 Interrupt (2 timer, 2 counter, 1 serial, 1 reset).
8. Mempunyai 2 DPTR (Data Pointer).



Gambar 2.2. Konfigurasa pin IC AT89S51

Mikrokontroler AT89S51 memiliki pin berjumlah 40 dan masing – masing pin mempunyai kegunaan sebagai berikut :

1. 4 buah port sebagai I/O

   Terdiri dari port 0, port 1, port 2, dan port 3. Masing-masing port dengan lebar 8 pin atau 8 bit dan masing-masing port mempunyai fungsi yang berbeda, yaitu:

a. Port 0

Merupakan port I/O yang terdiri dari port 0.0 sampai port 0.7 dan berfungsi sebagai port alamat dan data.

b. Port 1

Merupakan salah satu port I/O dengan lebar 8 bit yang terdiri dari port 1.0 sampai port 1.7. Bersifat dua arah dengan resistor *pull-up* internal. Jika logika 1 atau tinggi dituliskan maka keluarannya akan berlogika 1 atau tinggi dan jika logika 0 atau rendah dituliskan maka keluarannya akan berlogika 0 atau rendah. Selain fungsi-fungsi tersebut port 1 juga memiliki fungsi alternatif seperti Tabel 2.1.

Tabel 2.1 Fungsi alternatif port 1

Pin Port	Fungsi alternatif
P1.5	MOSI (*Master data output, slave data input pin untuk SPI*)
P1.6	MISO (*Master data input, slave data output pin untuk SPI*)
P1.7	SCK (*Master clock input, slave data input pin untuk SPI*)

c. Port 2

Merupakan salah satu port I/O dengan lebar 8 bit yang terdiri dari port 2.0 sampai port 2.7. Berbeda dengan port 0, port ini tidak bersifat sebagai jalur data hanya sabagai pembawa alamat. Dengan demikian jelas bahwa untuk alamat AT89S51 menyediakan 16 bit dan untuk jalur data 8 bit.

d. Port 3

Merupakan port I/O dengan lebar 8 bit yang bersifat dua arah dan sudah mempunyai resistor *pull-up* secara internal. Jika logika 1 atau tinggi dituliskan maka keluarannya akan berlogika 1 atau tinggi dan jika logika 0 atau rendah dituliskan maka keluarannya akan berlogika 0 atau rendah. Port 3 juga memiliki fungsi khusus seperti pada Tabel 2.2.

Tabel 2.2 Fungsi alternatif port 3

Pin Port	Fungsi alternatif
P3.0	RXD (port masukan serial)
P3.1	TXD (port keluaran serial)
P3.2	INT0 (interupsi eksternal 0, aktif rendah)
P3.3	INT1 (interupsi eksternal 1, aktif rendah)
P3.4	T0 (masukan eksternal timer 0)
P3.5	T1 (masukan eksternal timer 1)
P3.6	WR (*signal* tulis untuk memori eksternal, aktif rendah)
P3.7	RD (*signal* Baca untuk memori eksternal, aktif rendah)

2. Jalur kontrol

AT89S51 mempunyai pin yang berfungsi secara khusus untuk mengontrol piranti lain, untuk melakukan sebuah eksekusi atau mengakses data. Jalur kontrol tersebut beranggotakan :

a. PSEN (*Program Store Enable*)

Merupakan pulsa pengaktif unuk membaca program memori luar.

b. ALE (*Address Latch Enable*)

Berfungsi untuk mengakses alamat, dengan memberikan signal ke IC latch agar menahan/menyimpan alamat dari port 0 yang akan menuju memori eksternal (alamat 0-7), dan selanjutnya memori eksternal akan mengeluarkan data yang melalui port 0.

c. EA (*External Access*)

Akan aktif jika menggunakan program memori internal maka EA dihubungkan dengan VCC.

d. RST (*Reset*)

Pin ini berfungsi sebagai input untuk melakukan reset terhadap mikro, artinya jika diaktifkan maka semua pin dan program akan terakses kembali seperti awal mulai bekerja.

3. Register pewaktu (*Timer Register*)

Sebanyak 2 buah yaitu *timer* 0 dan *timer* 1 yang masing-masing berkapasitas 16 bit. Register ini digunakan sebagai :

a. *Delay* atau jarak waktu

Sebagai contoh penggunaannya, mikrokontroler memberikan waktu kepada sebuah piranti I/O yang dikontrolnya untuk bekerja selama rentang waktu tertentu. Hal ini memerlukan *delay*.

b. *Counter* atau pencacah

Mikrokontroler mempunyai kemampuan untuk mencacah (menghitung) pulsa dari luar misalnya dari *signal generator*.

e. *Baud Rate* Serial Komunikasi

yaitu tekanan transfer dapat diubah-ubah sesuai dengan kebutuhan.

4. XTAL 1 dan XTAL 2

Merupakan pin input untuk kristal osilator.

5. GND

Pada kaki berfungsi sebagai pentanahan (*ground).*

6. RAM (*Random Acces Memory*)

Dengan kapasitas sebesar 128 bytes. RAM merupakan tempat menyimpan data sementara, dan akan hilang bila sumber tegangan dimatikan.

7. ROM (*Read Only Memory*)

Dengan kapasitas sebesar 4 Kb. ROM ini berisikan program-program yang akan dijalankan oleh Mikrokontroler. ROM hanya bisa dibaca dan tidak bisa ditulis pada saat eksekusi program. Untuk menghapus program di ROM ada berbagai macam cara yang disesuaikan dengan jenis ROM tersebut, yaitu:

a. Untuk EPROM (*Erasable Programmable* ROM)

Dapat dihapus dengan menggunakan sinar ultra violet selama kurang lebih 15 menit. Setelah itu EPROM dapat ditulis program menggunakan EPROM Programer.

b. Untuk EEPROM (*Electric Erasable Programmable* ROM)

Dapat dihapus dengan memberikan tegangan 5 *volt* selama beberapa saat pada pin tertentu, setelah itu dapat ditulis program kembali dengan menggunakan EEPROM Programer.

5. Kontrol Interupsi

   Interupsi yang dilayani oleh AT89S51 dapat berasal dari :

   a. Piranti diluar AT89S51. Untuk interupsi ini AT89S51 menyediakan dua buah kontrol yaitu INT0 dan INT1 (pin P3.3 dan P3.2).

   b. Timer Register baik Timer 0 dan Timer 1 .

   c. Port serial yaitu melalui register TI (*transmit interupt*) atau RI (*Receive Interupt*).

6. Port serial berfungsi untuk komunikasi serial dengan CPU lain. Sepasang Tx (*Transmiter*) dan Rx (*Receiver*) (pin P3.1 dan P3.0).

8. *Osilator On-Chip* AT89S51 sangat penting dalam menentukan siklus mesin dari AT89S51. Osilator ini dibangkitkan oleh kristal ataupun dari TTL (*Transitor Transistor Logic*) luar. Semakin besar frekuensi yang dipakai oleh *osilator on-chip* ini semakin cepat siklus mesin dari AT89S51 berarti semakin cepat pula kemampuan AT89S51 mengeksekusi suatu program.

9. VCC

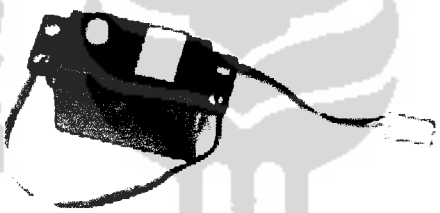
   Pada kaki ini berfungsi sebagai tempat sumber tegangan yang sebesar 5 Volt.

2.4. Motor Servo

Motor servo adalah motor khusus yang disertai dengan rangkaian tambahan atau umpan balik *(feedback)* posisi untuk menentukan posisi sumbu motor servo yang diinginkan. Jenis motor ini banyak digunakan para penggemar mainan *radio control*/RC seperti pesawat terbang model, perahu, mobil dan juga sering digunakan

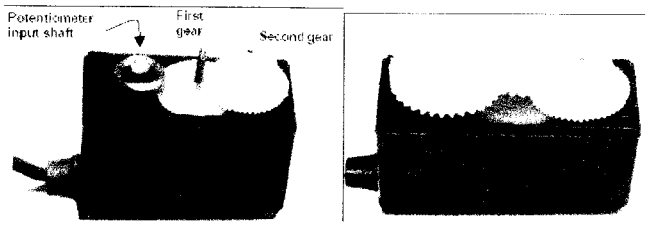
pada robot. Dengan menggunakan motor servo pengendalian posisi mainan tersebut lebih presisi sehingga memudahkan pengguna dalam memakainya.

Berbeda dengan motor dc dan motor stepper yang didesain menggunakan sistem open feedback (tanpa umpan balik), maka motor servo merupakan motor yang didisain bersistem closed feedback (dengan umpan balik). Hal yang unik dari motor servo adalah bahwa motor servo ini diatur/dikontrol menggunakan pulsa. Namun dengan menggunakan computer atau rangkaian mikrokontroler, akan dapat dengan mudah mengontrol motor servo.



Gambar 2.3. Motor servo

Motor servo jika dibuka dari badannya maka akan didapat part-part sebagai berikut. Sebuah motor yang merupakan jantung dari motor servo, sekumpulan gear-gear yang berguna untuk mengurangi kecepatan putar motor, sebuah potensiometer dan PCB yang berisi rangkaian control.



Gambar 2.4. Bagian-bagian motor servo

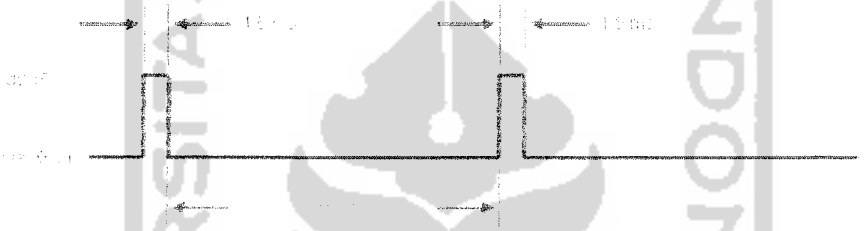
Sebagaimana dijelaskan didepan bahwa motor servo termasuk jenis close loop. Rangkaian *feedback* posisi tersebut berupa potensiometer yang dapat berputar sesuai dengan putaran gear utama motor servo. Rangkaian *feedback* akan terus membandingkan antara posisi motor servo yang diperoleh dari perubahan nilai hambatan pada potensiometer dengan pulsa masukan. Rangkaian control dan motor tersebut dicatu dengan tegangan dc sebesar 4,8 – 6 volt tergantung daya yang ada pada motor servo.

Motor servo terdapat dua jenis, yaitu :

2.4.1. Servo standar

Servo standar memiliki posisi tengah atau nol derajat yang merupakan titik tengah antara awal dan akhir jangkauan motor servo. Posisi titik tengah ini dapat dicapai dengan memberikan lebar pulsa tertentu pada s*tandard servo*. Pada motor s*tandard servo* pemberian sinyal kontrol menyebabkan sumbu keluaran berputar sebagian dalam *range* pergerakan maksimal sampai dengan 90° atau 180° dan mempunyai posisi yang cukup akurat dalam setiap pergerakannya. Servo didesain untuk menterjemahkan lebar pulsa positip (+3 sampai +5 volt) antara 0.9 sampai 2.1 milisecond(ms). Lebar pulsa ini merupakan pengontrol gerakan servo. Gerakan ini diatur oleh dua keadaan, yaitu pada tegangan sebesar 5 volt dengan lebar pulsa antara 0.9 sampai 2.1 ms dan pada keadaan ketika tanpa adanya tegangan dengan lebar pulsa yang dapat bervariasi antara 10 sampai 40 ms sesuai dengan rekomendasi. Pada lebar pulsa rendah ini tidak akan banyak mengubah atau mengurangi gerakan servo karena informasi arah posisi gerakan servo dibawa oleh lebar pulsa positif 5 volt itu sendiri. Lebar pulsa rendah hanya menentukan kehalusan gerakan putaran servo. Jika

lebar pulsa rendah terlalu besar, akan mengakibatkan gerakan putaran servo kurang halus. Lebar pulsa rendah antara 10 sampai 40 ms merupakan rekomendasi agar gerakan servo bisa halus. Pada s*tandard servo*, jika lebar pulsa 5 volt diberikan sebesar 1,5 ms, sumbu servo akan berputar menuju titik tengah (posisi netral). Jika pulsa yang diberikan lebih kecil dari 1,5 ms sumbu motor servo akan bergerak menuju arah 0 derajat. Jika pulsa yang diberikan lebih besar dari 1,5 ms, sumbu motor servo akan bergerak menuju arah 180 derajat.



Gambar 2.5. Pulsa motor servo

Motor ini juga mempunyai torsi yang besar tergantung dari jenis serinya.

2.4.2. *Continous Servo*

Motor servo dapat dimodifikasi agar dapat berputar secara kontinyu (360 derajat), yaitu dengan cara memanipulasi (memotong pembatas) *gear* yang terdapat pada motor servo dan rangkaian *feedback* yang berupa potensiometer.

Gambar 2.6. Gear *countinous servo* (

Continous servo merupakan servo yang sudah dimodifikasi, putaran akan bergerak kontinyu. Dengan memberikan lebar pulsa sebesar 1,3 ms akan menyebabkan motor

servo berputar searah jarum jam. Sedangkan jika lebar pulsa diberikan sepanjang 1,7 ms akan menyebabkan sumbu servo berputar berlawanan dengan arah jarum jam. Namun pulsa sepanjang 1,5 ms akan mengakibatkan sumbu servo tetap pada posisinya (tidak berputar). Hal ini sama dengan yang terjadi pada *standard servo*, yaitu dikarenakan servo berada pada posisi netral ketika lebar pulsa yang diterimanya 1,5 ms.

2.4.3. Sistem Konektor

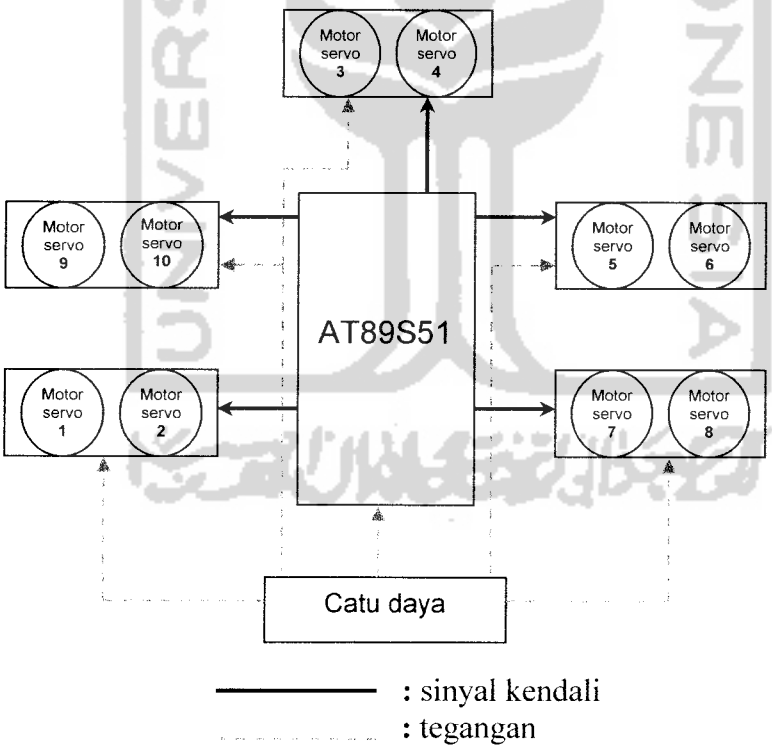
Motor servo memiliki konektor dengan 3 titik hubung, masing-masing adalah untuk tegangan berwarna merah (4,8 – 6 VDC), Ground berwarna hitam dan untuk sinyal/data berwarna putih.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem

Perancangan secara umum dari sebuah robot berjalan pada tugas akhir ini adalah sebuah sistem minimum mikrokontroler AT89S51 yang menjadi kendali utama dengan di beri masukan atau *input* pada sistem robot dan keluaran atau *output* yang berupa gerakan motor servo yang berfungsi sebagai penggerak utama pada robot. Blok diagram sistem dapat dilihat pada Gambar 3.1.



Gambar 3.1. Blok diagram robot secara umum

Dari tiap blok gambar dapat dilihat bahwa masing-masing komponen saling berhubungan antara satu dengan yang lain. Fungsi dari masing-masing blok tersebut adalah :

a. Mikrokontroler ; merupakan pengolah data-data. Dari data yang didapat dan diolah tersebut maka akan dihasilkan suatu perintah yang akan mengendalikan kaki robot. Dengan kata lain mikrokontroler merupakan otak dari suatu sistem robot tersebut.
b. Motor Servo ; merupakan motor penggerak kaki pada robot. Data yang didapat dari mikrokontroler akan diterjemahkan sebagai gerakan mekanis oleh motor servo.
c. Catu daya ; tegangan yang digunakan untuk mengoperasikan sistem pada robot.dengan tegangan 12 Volt.
d. Mekanik ; bagian perangkat keras (hardware) pada perancangan robot. Bahan yang digunakan adalah akrilik.

3.2 Perancangan Perangkat Keras (*Hardware*) Robot

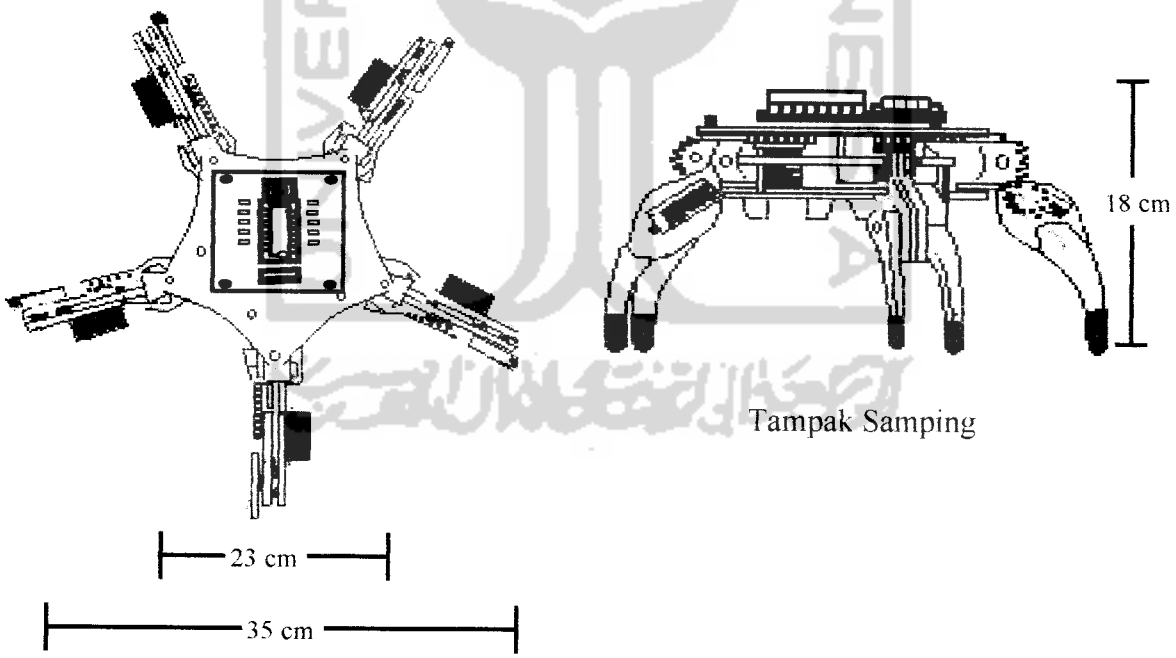
Robot dibuat dengan bentuk persegi lima dengan lima kaki sebagai pergerakannya. Dasar badan robot berupa lembaran akrilik dengan tebal 3 mm sebagai tempat untuk meletakkan komponen elektronik, motor servo dan komponen pendukung lainnya. Badan robot memiliki ukuran panjang 35 cm, lebar 18 cm dan tinggi 23 cm. Bahan-bahan yang digunakan dalam perancangan robot ini adalah sebagai berikut :

a. Komponen mekanik pembentuk robot

1. Akrilik dengan ketebalan 3 mm.
2. Sepuluh buah motor servo Hitec HS-311 standart.
3. Gir.
4. Mur dan baut.

b. Komponen elektronik robot

1. Mikrokontroler Atmel AT89S51.
2. Resistor,transistor, kapasitor.
3. PCB (*Printed Board Circuit*).
4. Kabel, pin deret dan *housing*.



Gambar 3.2 *layout* robot dalam beberapa posisi

(www.rybots.com)

3.2.1 Perancangan Komponen Mekanik Pembentuk Robot

Untuk perancangan ini dibagi menjadi dua bagian, yaitu :

1. Perancangan kaki (*leg design)*

   Meliputi : pangkal (*hip)*, paha (*thigh)*, kaki bawah (*bottom leg).*

2. Perancangan body (*body design)*

   Meliputi : atas (*top)*, tengah (*middle)*, bawah (*bottom).*

Berikut ini penjelasan pada masing-masing perancangan adalah sebagai berikut ;

1. Perancangan kaki (*leg design)*

   Seperti yang telah dijelaskan diatas, perancangan kaki/*leg* structure meliputi 3 bagian rakitan/*assembly*, yaitu : pangkal, paha dan kaki bawah. Berikut ini penjelasannya :

   a. Rakitan pangkal/*hip assembly*

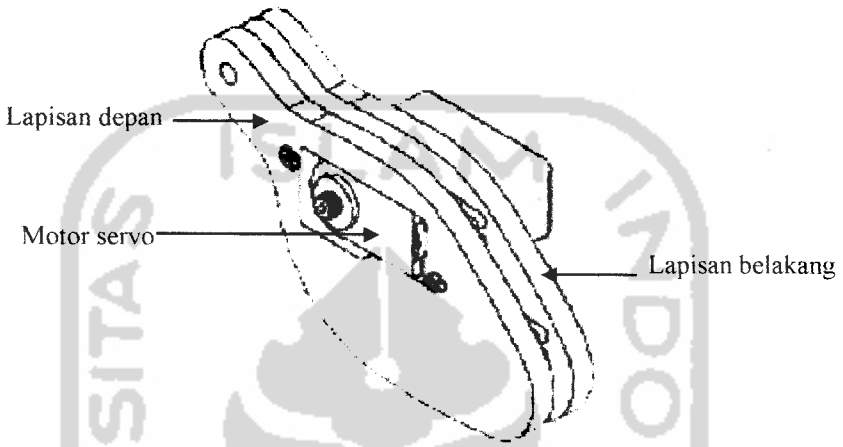
      Terdiri dari beberapa bagian seperti; gir pangkal atas, pangkal bawah, pangkal samping dan gear pangkal samping. Seperti pada gambar di bawah ini :

Gambar 3.3. *Hip assembly*/rakitan pangkal

(www.rybots.com)

b. Rakitan paha/*thigh assembly*

Terdiri dari 2 lapisan paha yaitu ; lapisan depan dan belakang. Terdapat 1 buah motor servo yang fungsinya untuk pergerakan mengangkat dan menurunkan kaki robot. Seperti gambar dibawah ini :



Gambar 3.4. *Thigh assembly*/rakitan paha

(www.rybots.com)

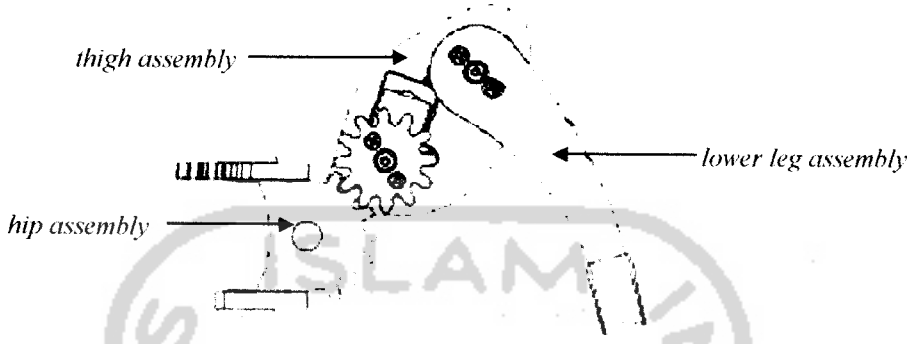
c. Rakitan kaki bawah/*lower leg assembly*

Digunakan untuk tumpuan robot dengan ujung kaki yang dilapisi dengan karet biar tidak licin dan robot bisa berjalan. Sepertti gambar dibawah ini :

Gambar 3.5. *Lower leg assembly*/rakitan kaki bawah

(www.rybots.com)

Kemudian semua bagian/rakitan pangkal, paha dan kaki di rakit menjadi satu maka akan diperoleh rakitan seperti gambar dibawah ini ;



Gambar 3.6. Rakitan dari semua perancangan kaki

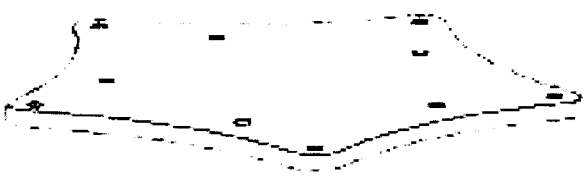
(www.rybots.com)

2. Perancangan badan (*body design)*

Seperti yang telah di jelaskan di atas, perancangan body meliputi 3 bagian, yaitu : atas, tengah dan bawah. Berikut ini penjelasan masing-masing bagian tersebut.

a. Bagian Atas/*top part*

Bagian atas robot dan sebagai pelindung atas atau penutup. Biasanya dipasang setelah robot telah selesai dirakit. Seperti gambar di bawah ini.

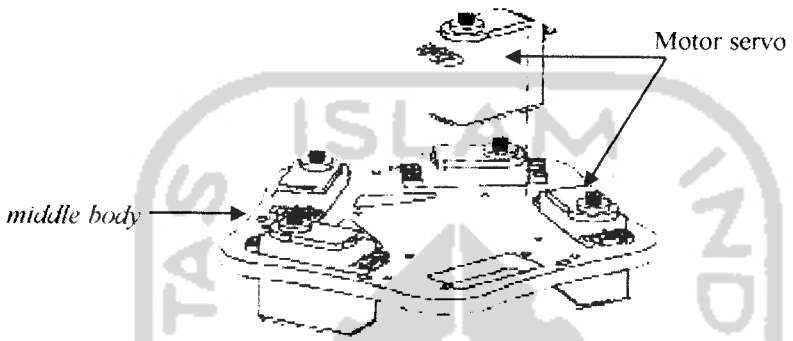


Gambar 3.7. *Top part*/bagian atas

(www.rybots.com)

b. Bagian Tengah/*middle part*

Bagian tengah terdapat 5 buah motor servo yang berfungsi untuk berputar atau menggeser kaki ke kanan atau kiri sebagai pergerakan motor servonya.



Gambar 3.8. Middle part/bagian tengah

(www.rybots.com)

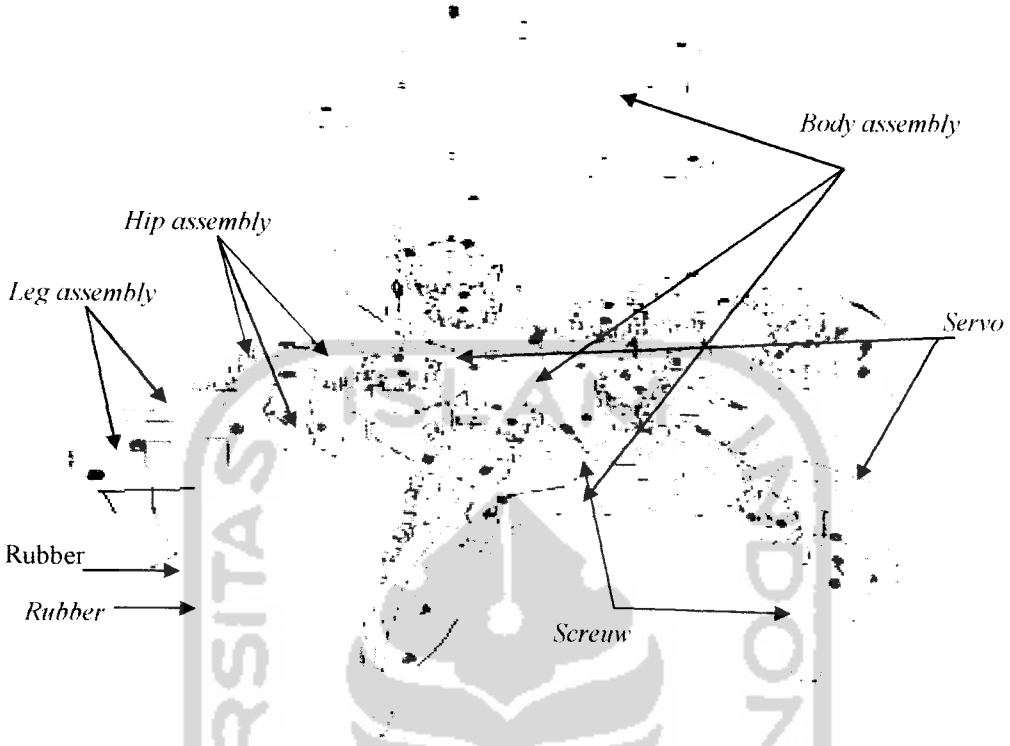
c. Bagian Bawah/*bottom part*

Berfungsi sebagai tempat *power supply*/baterai untuk daya pada robot.

Gambar 3.9. *Bottom part*/bagian bawah

(www.rybots.com)

Bila semua rancangan telah selesai dibuat, kemudian di rakit menjadi satu yaitu sebagai mekanik dari perangkat keras sehingga akan terbentuk sebuah mekanik robot sesuai dengan desain yang dirancang. Berikut di bawah ini gambar semua perancangan mekanik robot yang telah di rakit menjadi satu.



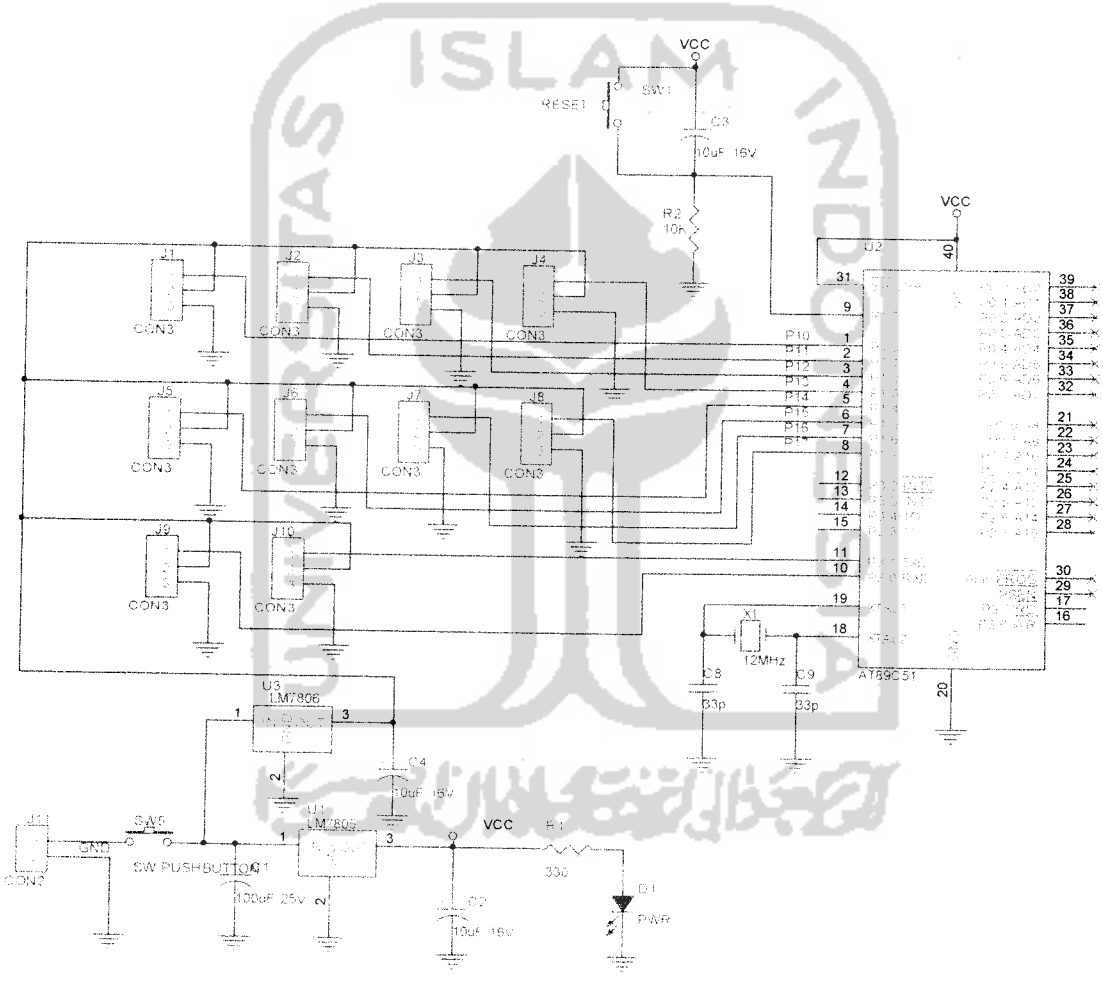
Gambar 3.10. Rakitan total seluruh perancangan mekanik robot

(www.rybots.com)

3.2.2 Sistem Minimum AT89S51

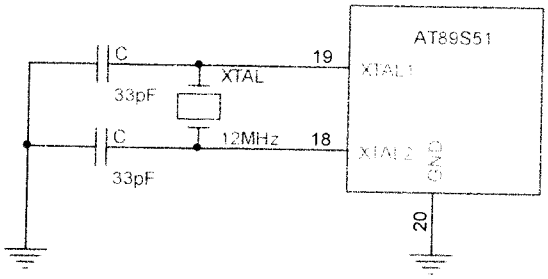
Pada pengolahan data dirobot ini sistem mikrokontroler yang digunakan adalah sistem minimum AT89S51 yang terdiri dari rangkaian osilator, *power on reset* dan catu daya. Sistem minimum ini memerlukan tegangan sebesar 5 volt, sedangan tengangan untuk motor servo sebesar 6 volt. Sedangkan keluaran mikrokontroler yang digunakan motor servo sebagai input adalah *port* 1.0 sampai dengan *port* 1.7 ditambah *port* 3.0 dan *port* 3.1. Dengan pembagian *port* 1.0 dan *port* 1.1 sebagai keluaran untuk menggerakkan kaki pertama, *port* 1.2 dan *port* 1.3 sebagai keluaran untuk menggerakkan kaki kedua, *port* 1.4 dan *port* 1.5 sebagai

keluaran untuk menggerakkan kaki ketiga, *port* 1.6 dan *port* 1.7 sebagai keluaran untuk menggerakkan kaki keempat dan *port* 3.0 dan *port* 3.1 sebagai keluaran untuk menggerakkan kaki kelima. Berikut gambar rangkaian sistem minimum AT89S51 pada robot.



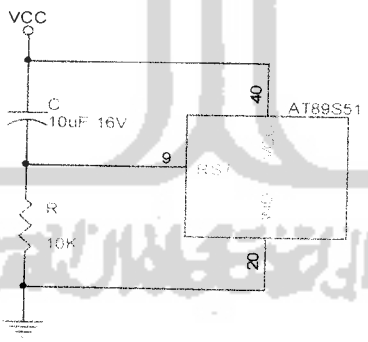
Gambar 3.11 Sistem minimum AT89S51 pada robot

Pada rangkaian osilator di sistem minimum AT89S51 ini menggunakan kristal 12 MHz dan dua kapasitor 33 pF. Fungsi rangkaian osilator ini adalah agar dapat membangkitkan *clock*. Dapat dilihat pada gambar 3.12 di bawah ini :



Gambar 3.12 Rangkaian osilator

Rangkaian *power on reset* pada gambar 3.13 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 C1 dan R1 (*Power On Reset*), (*Najmurrokhman,2005*).



Gambar 3.13 Rangkaian *power on reset*

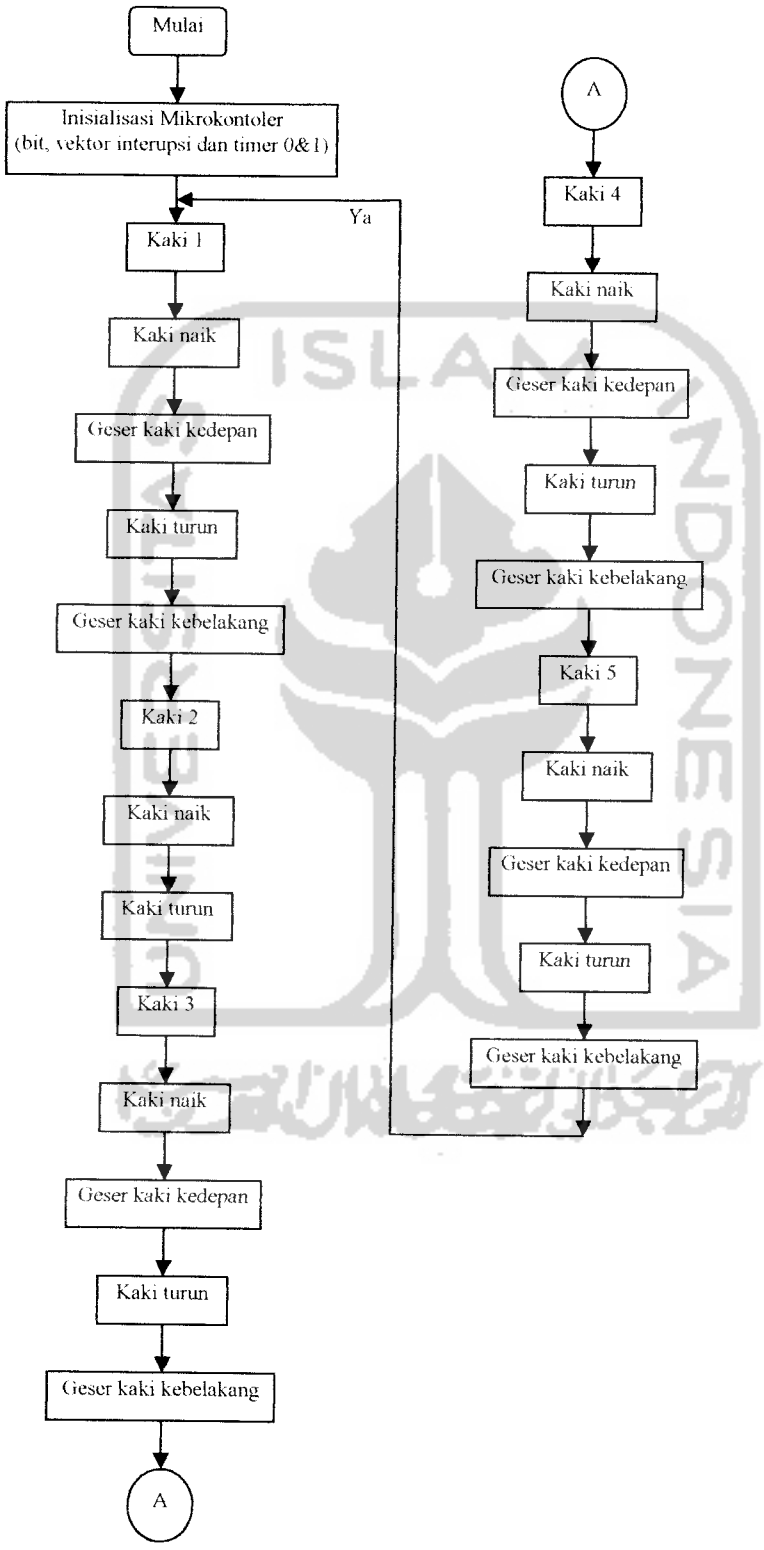
Selain sebagai pengolah data, sistem minimum mikrokontroler AT89S51 juga digunakan sebagai unit penyimpan program. Didalam sistem minimum inilah semua data diproses dan kemudian mengeluarkan instruksi untuk melakukan suatu tindakan.

3.3 Perancangan Perangkat Lunak (*Software*) Robot

Pada perancangan perangkat lunak terlebih dahulu dibuat algoritmanya, yang berisi proses pembuatan program yaitu sebagai berikut :

1. Program selalu diawali dengan melakukan inisialisasi untuk memudahkan proses selanjutnya.
2. Program yang dibuat menggunakan *Pulse Width Modulation* (PWM), yang akan menggerakan motor servo dengan memberikan lebar pulsa.
3. Motor servo akan menggerakan masing – masing kaki dengan perintah pergerakan kaki naik, geser depan, turun dan geser belakang.
4. Apabila semua proses diatas dilaksanakan, maka progam dapat menggerakan kaki robot sesuai perintah dan diharapkan sistem robot bisa bekerja sesuai dengan keinginan

3.3.1. Diagram Alir Program



Gambar 3.14 Diagram alir program

3.3.2. Implementasi Diagram Alir

3.3.2.1. Inisialisasi Milrokontroler

Program inisialisasi mikrokontroler terdiri dari inisialisasi bit, inisialisasi vektor interupsi dan inisialisasi timer 0 dan 1. Berikut ini listing program inisialisasi bit, inisialisasi vektor interupsi dan inisialisasi timer 0 dan 1:

```
;-------------------------------------------------------------------
;###########inisialisasi bit##########################
;-------------------------------------------------------------------
        flag1       bit     00h
        flag2       bit     01h
        a1          bit     18h
        b1          bit     19
        a2          bit     1ah
        b2          bit     1bh
        a3          bit     1ch
        b3          bit     1dh
        a4          bit     1eh
        b4          bit     1fh
        a5          bit     20h
        b5          bit     21h
        kaki1a      bit     p1.0
        kaki1b      bit     p1.1
        kaki2a      bit     p1.2
        kaki2b      bit     p1.3
        kaki3a      bit     p1.4
        kaki3b      bit     p1.5
        kaki4a      bit     p1.6
        kaki4b      bit     p1.7
        kaki5a      bit     p3.0
        kaki5b      bit     p3.1

;-------------------------------------------------------------------
;###############inisialisasi vektor interupsi##############
;-------------------------------------------------------------------
        org         00h
        sjmp        mulai
        org         0bh
        acall       waktu0
        reti
        org         1bh
        acall       waktu1
        reti
```

```
-------------------------------------------------------------------------
;################inisialisasi timer 0 & 1##################
-------------------------------------------------------------------------
mulai:  mov   r0, #00h
star:   mov   @r0, #00h
        inc   r0
        cjne  r0, #70h,star
        MOV   SP, #60H
        MOV   IE, #10001010B
        MOV   TMOD, #12H
        MOV   TL0, #-0fbH          ;faH,0e6h
        MOV   TH0, #-0fbH          ;faH,0e6h
        MOV   TL1, #-050H          ;050H
        MOV   TH1, #-0c3H          ;0c3H
```

3.3.2.2 Program utama

Dalam program utama berisi program untuk mengatur putaran motor servo yang akan menggerakan kaki robot. Pengaturannya berupa putar kanan dan kiri, sehingga bila di aplikasikan pada kaki robot berupa gerakan naik, turun, geser depan dan geser belakang. Programnya ditampilkan sebagai berikut:

```
-----------------------------------------------------
;############ Program Utama ##############
-----------------------------------------------------
aaa:
;kaki 1
        mov   23h, #00000001b      ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
        acall delay
        mov   23h, #00000010b      ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
        acall delay
        mov   23h, #00000001b      ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull
        acall delay
```

```
        mov   23h, #00000010b     ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull
        acall delay

;kaki2
        mov   23h, #00000100b     ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
        acall delay;
        mov   23h, #00001000b     ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
        mov   23h, #00000100b     ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull
        acall delay
        mov   23h, #00001000b     ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull

;kaki3
        mov   23h, #00010000b     ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
        acall delay
        mov   23h, #00100000b     ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull
        acall delay
        mov   23h, #00010000b     ;satu a
        mov   24h, #00000000b
        acall kiri
        acall ull
        acall delay
        mov   23h, #00100000b     ;satu a
        mov   24h, #00000000b
        acall kanan
        acall ull
```

```
        acall   delay

;kaki4
        mov     23h, #01000000b     ;satu a
        mov     24h, #00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h, #10000000b     ;satu a
        mov     24h, #00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h, #01000000b     ;satu a
        mov     24h, #00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h, #10000000b     ;satu a
        mov     24h, #00000000b
        acall   kiri
        acall   ull
        acall   delay

;kaki5
        mov     23h, #00000000b     ;satu a
        mov     24h, #00000001b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h, #00000000b     ;satu a
        mov     24h, #00000010b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h, #00000000b     ;satu a
        mov     24h, #00000001b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h, #00000000b     ;satu a
        mov     24h, #00000010b
        acall   kanan
        acall   ull
        acall   delay
```

```
        ljmp    aaa
```

3.3.2.3 Pengaturan Pulsa

Pengaturan pulsa di atur melalui keluaran PWM motor servo pada mikrokontroler. Port yang digunakan adalah P1.0 – P1.7, port 3.0 dan port 3.1 sebagai output.

```
;################ subrutin pulsa kanan ############
kiri:   mov     21h, #07h
        setb    tr0
        setb    tr1
        ret
;################ subrutin pulsa kiri ##############
kanan:  mov     21h, #05h
        setb    tr0
        setb    tr1
        ret
;################subrutin pulsa####################
ull:    jnb     flag1, time
        clr     flag1
        jnb     a1, satu
        cpl     kaki1a
        sjmp    ull
satu:   jnb     b1, dua
        cpl     kaki1b
        sjmp    ull
dua:    jnb     a2, duaa
        cpl     kaki2a
        sjmp    ull
duaa:   jnb     b2, tiga
        cpl     kaki2b
        sjmp    ull
tiga:   jnb     a3, tigaa
        cpl     kaki3a
        sjmp    ull
```

```
tigaa:    jnb   b3, empat
          cpl   kaki3b
          sjmp  ull
empat:    jnb   a4, empata
          cpl   kaki4a
          sjmp  ull
empata:   jnb   b4, lima
          cpl   kaki4b
          sjmp  ull
lima:     jnb   a5, limaa
          cpl   kaki5a
          sjmp  ull
limaa:    jnb   b5, ull
          cpl   kaki5b
          sjmp  ull
time:     jnb   flag2, ull
          clr   flag2
          clr   tr0
          clr   tr1
          mov   p1, #00h
          mov   p3, #00h
          ret
```

3.3.2.4. Subrutin pulsa timer

Program pulsa timer di pakai untuk mengatur keluaran pulsa timer, maka dalam aplikasi ini timer yang digunakan adalah pulsa timer 0 dan pulsa timer 1. Berikut listing program dalam subrutin pulsa timer 0 dan subrutin pulsa timer 1 :

```
-------------------------------------------------------
;####### subrutin pulsa timer 0##############
-------------------------------------------------------
    waktu0:   inc   22h
              mov   a,21h
              cjne  a,22h,belum
              mov   22h,#00h
              setb  flag1
    belum:    ret
```

```
;-------------------------------------------------------
;####### subrutin pulsa timer 1##############
;-------------------------------------------------------
        waktu1:     MOV  TL1,#-050H     ;50H
                    MOV  TH1,#-0c3H     ;c3H
                    inc  r7
                    cjne r7,#0ah,belom
                    mov  r7,#00h
                    setb flag2
        belom  :    ret
```

3.3.2.5. Program Tunda

Program tunda dibuat tidak dengan menggunakan timer karena tidak harus membutuhkan presisi timer yang benar-benar teliti/akurat. Listing programnya sebagai berikut :

```
;-------------------------------------------------------
;####### pengaturan tunda ##################
;-------------------------------------------------------
        delay:      mov  50h, #00h
        delay1:     mov  51h, #07fh
        delay2:     mov  52h, #01h      ;lama tunda 05h,
        delay3:
                    nop
                    djnz 52h, delay3
                    djnz 51h, delay2
                    djnz 50h, delay1
                    ret
        end.
```

BAB IV

ANALISA DAN PEMBAHASAN

4.1. Pengujian Sistem

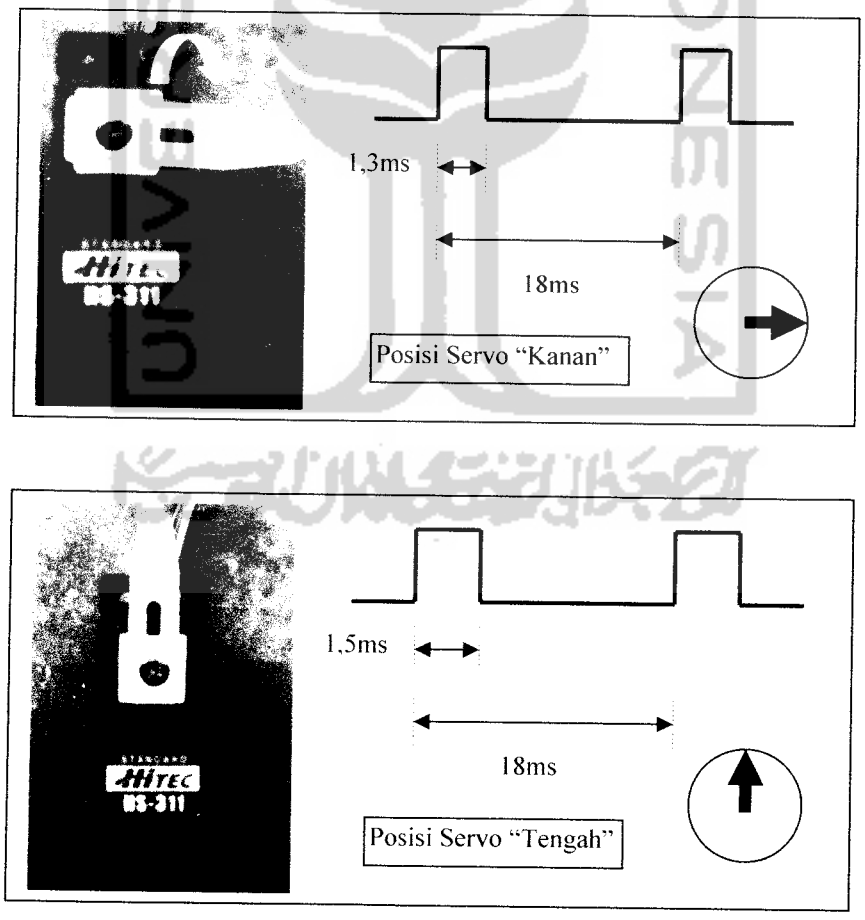
Pada bab ini akan dijelaskan mengenai pengujian terhadap keseluruhan sistem. Pengujian alat dan analisis hasil pengujian berfungsi untuk membandingkan antara perancangan dengan hasil pengujian untuk mengetahui unjuk kerja dari alat. Pengujian dilakukan terhadap keluaran atau masukan rangkaian yang membentuk sistem robot. Adapun analisa dan pembahasan di bagi menjadi dua, yaitu analisa pengamatan PWM dari motor servo dan konsep dari pergerakan robot.

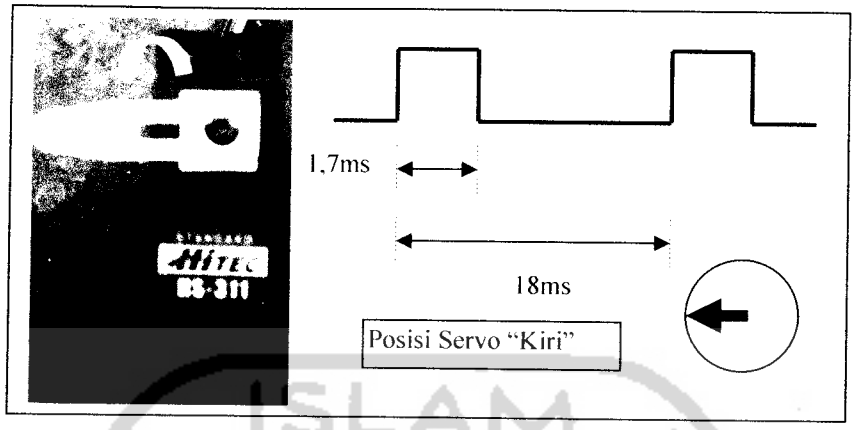
4.2. Pengujian Motor Servo

4.2.1. Motor Servo

Pada dasarnya 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.

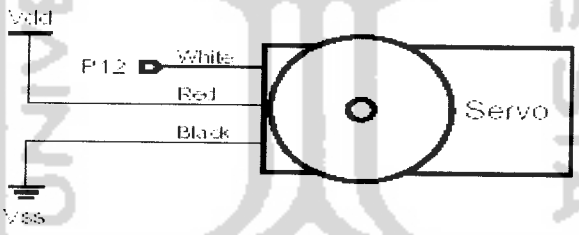
Motor servo yang digunakan untuk robot ini adalah jenis *continous*. Artinya motor dapat berputar 360 derajat tanpa harus memodifikasi *gear* yang biasa dilakukan pada motor servo standar. Motor servo akan bergerak dengan memberikan nilai lebar pulsa sebesar 1,3 ms maka motor akan berputar searah jarum jam (kanan), sedangkan jika lebar pulsa diberikan sepanjang 1,7 ms akan menyebabkan servo berputar berlawanan dengan arah jarum jam (kiri) dan bila diberikan lebar pulsa sebesar 1,5 ms maka akan menyebabkan motor servo tetap pada posisinya (tidak berputar). Berikut dibawah ini hubungan lebar pulsa dengan posisi "*horn*" servo ditunjukan pada gambar 4.1.





Gambar 4.1. Hubungan lebar pulsa dengan posisi "*horn*" servo

Motor servo memiliki 3 jenis input yaitu merah untuk *power* (5 volt), hitam untuk *ground* dan putih untuk sinyal pengendalian servo yang dihubungkan dengan mikrokotroler.

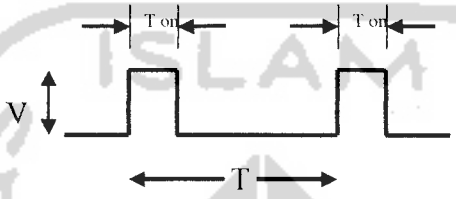


Gambar 4.2. pengkabelan motor servo

Motor servo bersifat proposional, kecepatan dan torsi yang dihasilkan proposional dengan tegangan yang diberikan pada pin powernya. Semakin besar tegangan yang diberikan, maka semakin cepat dan semakin besar juga torsi yang dihasilkan, begitu juga sebaliknya.

4.2.2. Pengamatan Sinyal Keluaran Dari Motor Servo

Pulsa Width Modulation (PWM) adalah pemodulasian terhadap sebuah pulsa, sehingga pulsa hidup (T *on*), merupakan perbandingan dari lebar pulsa (T). Modulasi yang dilakukan pada PWM adalah modulasi *duty cycle*, yaitu perbandingan antara T *on* dan T.

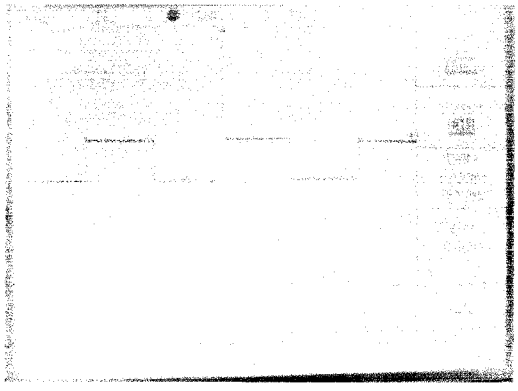


Gambar 4.3. Bentuk gelombang PWM

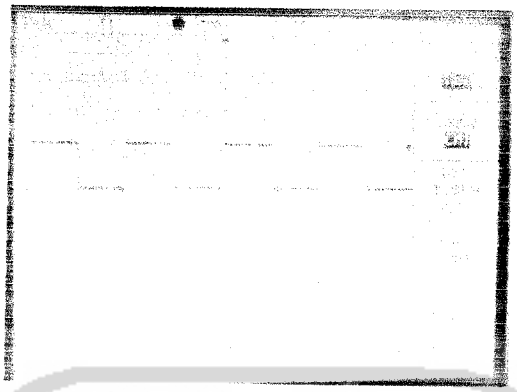
Dari hasil pengamatan didapat analisa Sinyal keluaran dari motor servo pada program pergerakan kaki robot adalah sebagai berikut pada gambar dibawah ini :

a. Analisa Sinyal pada program gerak kaki naik dan turun

Pada analisa sinyal keluaran pada program gerak kaki naik merupakan persamaan pergerakan dari motor putar kiri. Sedangkan analisa sinyal keluaran pada program gerak kaki turun merupakan persamaan pergerakan dari motor putar kanan. Berikut gambar sinyal keluaran yang di hasilkan :



Gambar 4.4. Sinyal keluaran pada gerak kaki naik (putar kiri)



Gambar 4.5. Sinyal keluaran pada gerak kaki turun (putar kanan)

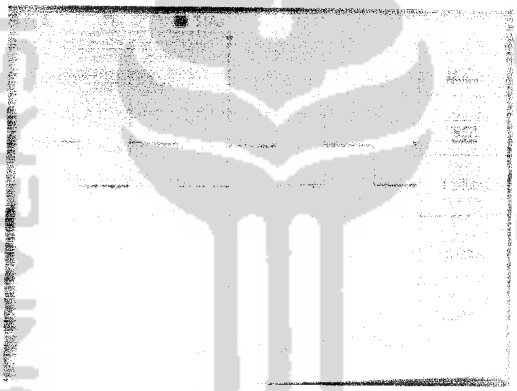
Dari hasil pengujian dan pengamatan, gelombang yang dihasilkan adalah gelombang kotak. Maka analisa dan pembahasan yang didapat antara teori dengan pengamatan adalah sebagai berikut :

1. Putaran motor pada saat keadaan motor putar kiri (gerak kaki naik) adalah 1,720 ms. Artinya untuk membuat motor berputar kekiri maka diberikan pulsa dengan lebar 1,720ms. Sehingga di dapat analisa dan pembahasan dalam menentukan besar lebar pulsa untuk menentukan posisi putar kiri servo adalah sesuai dan terbukti dengan teori yaitu lebar pulsa pada servo continous untuk posisi motor putar kiri sebesar 1,7ms. Dengan pertimbangan lebar pulsa putar kiri servo antara 1,5ms sampai 2ms.
2. Putaran motor pada saat keadaan motor putar kanan (gerak kaki turun) adalah 1,240 ms. Artinya untuk membuat motor berputar kekanan maka diberikan pulsa dengan lebar 1,240ms. Sehingga di dapat analisa dan pembahasan dalam menentukan besar lebar pulsa untuk menentukan posisi putar kanan servo adalah sesuai dan terbukti dengan teori yaitu

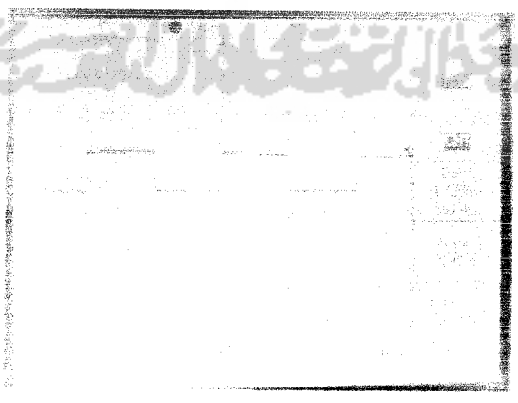
lebar pulsa pada servo continous untuk posisi motor putar kanan sebesar ≤1,3ms. Dengan pertimbangan lebar pulsa putar kanan servo antara 1ms sampai 1,5ms.

b. Analisa Sinyal pada program gerak kaki geser depan dan geser ke belakang

Pada analisa sinyal keluaran pada program gerak kaki geser depan merupakan persamaan pergerakan dari motor putar kanan. Sedangkan analisa sinyal keluaran pada program gerak kaki geser belakang merupakan persamaan pergerakan dari motor putar kiri. Berikut gambar sinyal keluaran yang di hasilkan :



Gambar 4.6. Sinyal keluaran pada gerak kaki geser depan



Gambar 4.7. Sinyal keluaran pada gerak kaki belakang

Dari hasil pengujian dan pengamatan, gelombang yang dihasilkan adalah gelombang kotak. Maka analisa dan pembahasan yang didapat antara teori dengan pengamatan adalah sebagai berikut :

1. Putaran motor pada saat keadaan motor putar kanan (gerak kaki geser depan) adalah 1,240 ms. Artinya untuk membuat motor berputar kekanan maka diberikan pulsa dengan lebar 1,240ms. Sehingga di dapat analisa dan pembahasan dalam menentukan besar lebar pulsa untuk menentukan posisi putar kanan servo adalah sesuai dan terbukti dengan teori yaitu lebar pulsa pada servo continous untuk posisi motor putar kanan sebesar ≤1,3ms. Dengan pertimbangan lebar pulsa putar kanan servo antara 1ms sampai 1,5ms.
2. Putaran motor pada saat keadaan motor putar kiri (gerak kaki geser belakang) adalah 1,720 ms. Artinya untuk membuat motor berputar kekiri maka diberikan pulsa dengan lebar 1,720ms. Sehingga di dapat analisa dan pembahasan dalam menentukan besar lebar pulsa untuk menentukan posisi putar kiri servo adalah sesuai dan terbukti dengan teori yaitu lebar pulsa pada servo continous untuk posisi motor putar kiri sebesar 1,7ms. Dengan pertimbangan lebar pulsa putar kiri servo antara 1,5ms sampai 2ms.

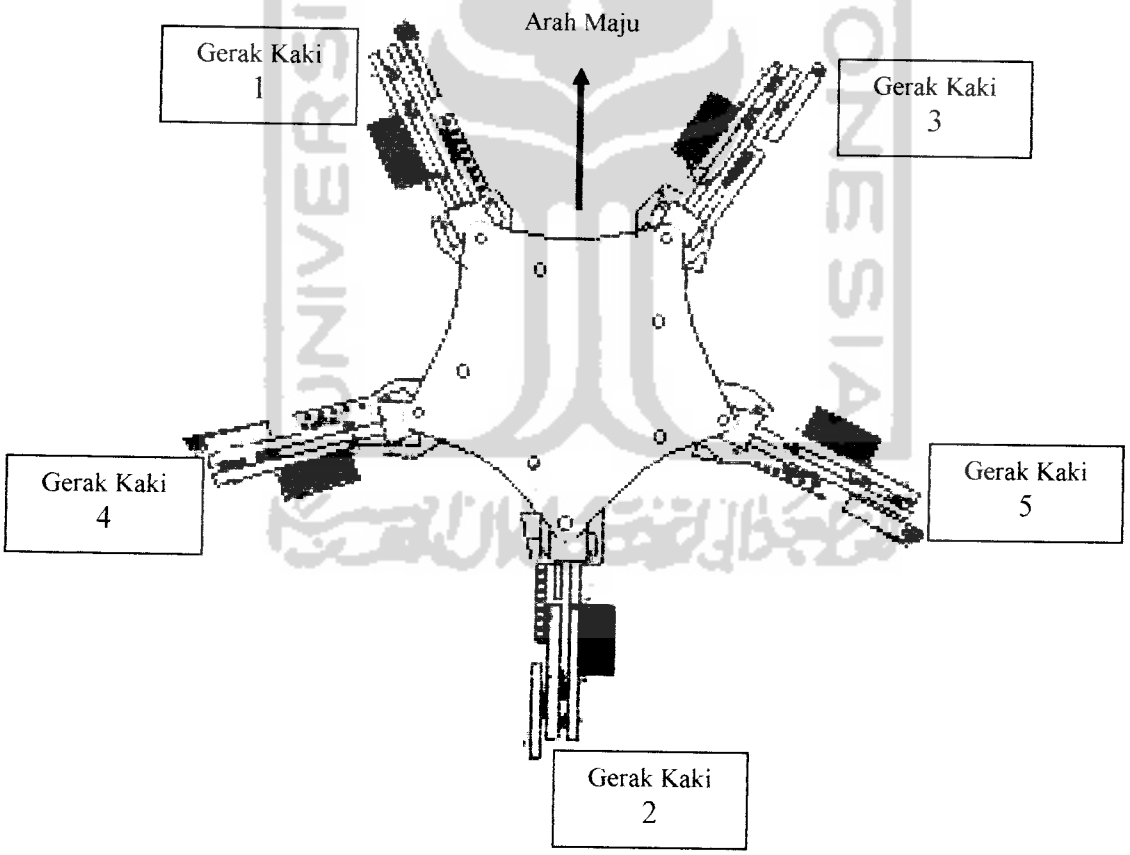
Maka dalam hal ini dengan melihat hasil data yang didapat, antara pengamatan dan teori tentang besar lebar pulsa pada pengaturan posisi motor servo putar kanan dan lebar pulsa posisi motor servo putar kiri adalah sesuai dan terbukti. Hal ini ditunjukan pada tabel 4.1 dibawah ini.

Tabel 4.1. Perbandingan hasil antara pengamatan dan teori

No	Posisi motor servo	Hasil pengamatan lebar pulsa (ms)	Hasil teori lebar pulsa (ms)
1	Putar kanan	1,240	1,3
2	Putar kiri	1,720	1,7

4.3. Pengujian Gerak Kaki Robot

Pengujian gerak pada kaki robot mempunyai konsep urutan gerak kaki seperti pada gambar 4.5 di bawah ini :



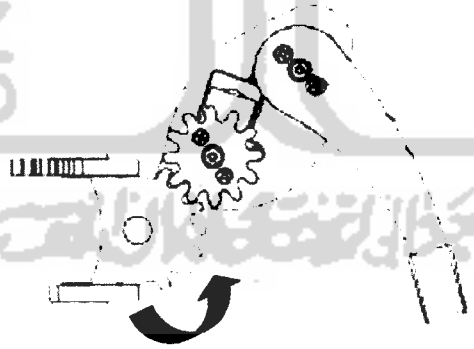
Gambar 4.8. Konsep urutan pergerakan kaki robot

(www.rybots.com)

Konsep pergerakan kaki robot meliputi 4 gerakan untuk kesatu, ketiga, keempat dan kelima, sedangkan kaki kedua mempunyai 2 gerakan dengan tujuan sebagai pendorong. Adapun gerakan – gerakan yang di berikan pada kaki robot kesatu, ketiga, keempat dan ketiga adalah gerakan kaki naik, gerakan kaki geser depan, gerakan kaki turun dan gerakan kaki geser belakang, sedangkan kaki kedua dengan gerakan kaki naik dan kaki turun. Berikut analisa dan pembahasan untuk masing – masing gerakan kaki pada tiap kaki robot.

4.3.1. Gerakan Kaki Naik

Agar kaki robot bergerak naik, maka motor servo berputar kekiri dengan memberikan pulsa pada motor tersebut. Kaki robot bergerak naik dan akan sesuai dengan perintah program yang telah ditentukan. Dibawah ini ini gambar kaki robot bergerak naik.



Gambar 4.9. Gerakan kaki naik (www.rybots.com)

4.3.2. Gerakan Kaki Geser Depan

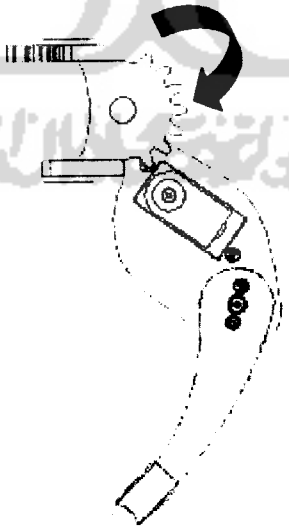
Agar kaki robot bergeser ke depan, maka motor servo berputar kekanan dengan memberikan pulsa pada motor tersebut. Kaki robot bergerak geser kedepan

dan akan sesuai dengan perintah program yang telah ditentukan. Dibawah ini gambar kaki robot bergerak geser kedepan.

Gambar 4.10. Gerakan kaki geser ke depan (www.rybots.com)

4.3.3. Gerakan Kaki Turun

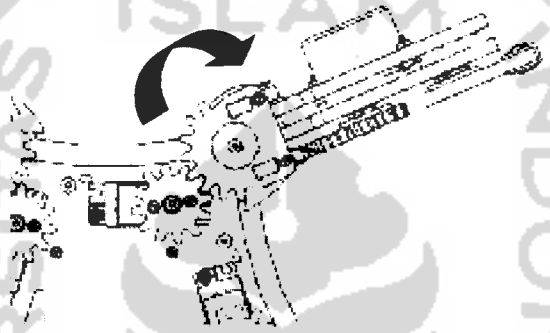
Agar kaki robot bergerakan turun, maka motor servo berputar kekanan dengan memberikan pulsa pada motor tersebut. Kaki robot bergerak turun dan akan sesuai dengan perintah program yang telah ditentukan. Dibawah ini ini gambar kaki robot bergerak turun.



Gambar 4.11. Gerakan kaki turun (www.rybots.com)

4.3.4. Gerakan Kaki Geser Belakang

Agar kaki robot bergeser ke depan, maka motor servo berputar kekiri dengan memberikan pulsa pada motor tersebut. Kaki robot bergerak geser ke belakang dan akan sesuai dengan perintah program yang telah ditentukan. Dibawah ini gambar kaki robot bergerak geser ke belakang.



Gambar 4.12. Gerakan kaki geser ke belakang (www.rybots.com)

4.3.5. Proses Pergerakan kaki robot secara keseluruhan

Pada sistem perancangan dan pengendalian robot lima kaki, aksi keseluruhan robot ini adalah bagaimana kaki robot tersebut dapat bergerak dan memungkinkan robot dapat berjalan. Pertama mengikuti konsep urutan pergerakan kaki yaitu dimulai dari kaki satu kemudian di ikuti kaki dua, kaki tiga, kaki empat dan kaki lima. Maka dari konsep urutan pergerakan di dapat pergerakan kaki robot secara keseluruhan sebagai berikut :

1. Kaki kesatu dengan pergerakan kaki bergerak naik kemudian kaki bergeser ke depan, kaki bergerak turun dan kaki kembali bergeser ke belakang.

2. Kaki kedua dengan pergerakan kaki bergerak naik kemudian kembali bergerak turun yang seolah – olah sebagai pendorong.
3. Kaki ketiga dengan pergerakan kaki bergerak naik, kemudian kaki bergeser kedepan, kaki bergerak turun dan kaki kembali bergeser ke belakang.
4. Kaki keempat dengan pergerakan kaki bergerak naik, kemudian kaki bergeser kedepan, kaki bergerak turun dan kaki kembali bergeser ke belakang.
5. Kaki kelima dengan pergerakan kaki bergerak naik, kemudian kaki bergeser kedepan, kaki bergerak turun dan kaki kembali bergeser ke belakang.

Dengan mengikuti urutan konsep pergerakan kaki, maka di dapat kaki robot bergerak sesuai dengan konsep urutan pergerakan kaki yaitu, kaki kesatu, kaki ketiga, kaki keempat dan kaki kelima bergerak dengan gerakan kaki bergerak naik, bergeser ke depan, bergerak turun dan bergeser kembali ke belakang. Sedangkan untuk kaki kedua dengan pergerakan kaki naik kemudian kaki bergerak turun.

Setelah masing-masing kaki robot dapat bergerak sesuai dengan konsep urutan pergerakan kaki, ternyata robot masih belum bisa berjalan dengan baik

4.3.6. Analisa Pergerakan Kaki Robot Tehadap Keseluruhan Robot

Analisa yang didapat dari pengamatan antara pergerakan kaki robot terhadap keseluruhan robot adalah sebagai berikut :

1. Pergerakan robot terlihat begitu berat pada saat kaki robot bergerak dengan tujuan untuk membuat robot dapat berjalan.

2. Pada saat kaki robot bergerak bergantian, di dapat kaki robot kurang mampu menahan berat badan robot sehingga dapat roboh.
3. Karena kaki robot pada saat bergerak bergantian kurang mampu menahan berat badan, sehingga di perlukan adanya tamabahan penyangga agar robot tidak roboh.
4. Beberapa kemungkinan yang membuat kaki robot kurang mampu menahan berat badan robot pada saat bergerak, yaitu adanya bahan yang berat/tebal, pembuatan mekanik salah satunya kurangnya kepresisian gir.

Dengan adanya pengamatan dan analisa ini, maka dalam pembuatan robot yang perlu di perhatikan salah satunya adalah pembuatan mekanik yang tepat dan perlunya melakukan penelitian, salah satunya pembuatan *prototype* miniatur robot yang akan di rancang dan di buat.

4.4. Power Suplai

Sistem robot ini memerlukan suplai tegangan dari luar berupa tegangan DC 6 dan 5V. Semua rangkaian elektronis memerlukan tegangan kerja sebesar 5V, hanya motor servo memerlukan tegangan 6V. Untuk mendapatkan tegangan 6V dan 5V ini, maka digunakan LM7806 dan LM7805 sebagai regulator tegangan. Sedangkan sumber tegangan berasal dari power suplai dengan tegangan *output* sebesar 12V.

Penggunaan sumber tegangan dari baterai akan dapat menjadikan sistem robot ini lebih *efisien*, akan tetapi dikarenakan penggunaan baterai dalam jumlah yang banyak akan memberikan tambahan berat pada robot, maka penggunaan baterai digantikan dengan sumber tegangan dari *power suplai* yang dihubungkan dengan

kabel. Hasil pengukuran tegangan *output* pada power suplai dan regulator dapat dilihat pada Tabel 4.2.

Tabel 4.2. Hasil pengukuran tegangan catu daya

Komponen	Keadaan	Pengukuran	Tegangan terukur (DC *volt*)
Power Suplai	Aktif	Kaki Positif	12
LM7806	Aktif	Kaki 3 *output*	5,98
LM7805	Aktif	Kaki 3 *output*	4,95

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pada perancangan sistem dan hasil analisa yang didapat maka dalam perancangan dan pengendalian robot lima kaki ini, dapat diambil beberapa kesimpulan, yaitu :

1. Kaki robot dapat bergerak sesuai dengan program yang diinginkan, yaitu kaki gerak naik, geser depan, gerak turun dan geser depan.
2. Pergerakan robot kurang sempurna di karenakan pembuatan mekanik yang kurang akurat, terutama gir yang kurang presisi.
3. Tingkat kepresisian pada motor servo dapat terganggu karena beberapa faktor seperti berat beban dan jalan yang tidak rata, kaki robot tidak mampu menahan kaki yang lain pada saat bergerak sehingga perlu tambahan tumpuan, mur dan baut pada mekanik yang kurang sesuai sehingga membuat kaki robot mengalami kesulitan pada saat bergerak.

5.2 Saran

Sebagai langkah lebih lanjut dalam penyempurnaan perancangan dan pengendalian robot lima kaki ini, maka beberapa saran berikut dapat digunakan dalam pengembangan robot ini, yaitu :

1. Perlunya melakukan penelitian, salah satunya pembuatan *prototype* miniatur robot yang akan di rancang dan di buat.
2. Pemilihan motor servo yang sesuai dengan robot yang akan di buat.
3. Pembuatan mekanik lebih akurat terutama gir sebaiknya lebih presisi agar pergerakan robot lebih lancar lagi.
4. Mekanik robot sebaiknya dibuat dari bahan yang sangat ringan agar tidak membebani kerja motor.
5. Robot ini masih sangat sederhana dan diperlukan penelitian lebih lanjut untuk mengembangkan robot dengan sensor infra red, sensor api dan sensor ultrasonik.
6. Menggunakan *solar cell* sebagai sumber tenaga.

DAFTAR PUSTAKA

[1] Abdallah C.T, Robot Control '*Dinamics, motion planning and analysis*', IEEE Press, 1993

[2] *Datasheet AT89S51, available at* www.atmel.com

[3] Eko Putra, Agfianto. *Belajar Mikrokontroler AT89C51 52 55 Teori dan Aplikasi*. Gava Media, Yogyakarta, 2002

[4] Jackson Jesse, *Ultrasonics and Robotics*, *available at* www. seattlerobotics.org/servo/

[5] Motor Servo HS-311, *available at* www.hitecred.com

[6] *Redefining Robotic 'Assembly Guide', available at* www.rybots.com

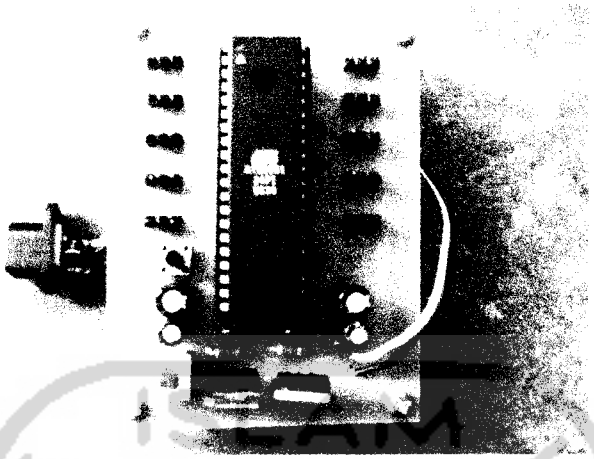
[7] *Robot InformationCentral, available at* www.robotics.com/robots.html

[8] Robotika dan otomatisasi, *available at* robotictechnologi.net

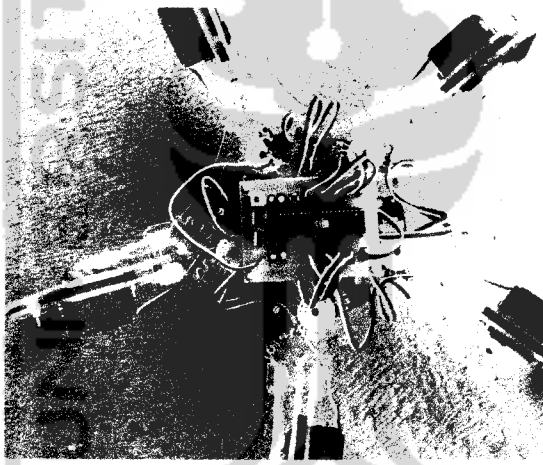
[9] *Speed Controllers, available at* www.homepages.which.net/Motors.html

[10] Tim Lab. Mikroprosesor, Pemrograman Mikrokontroler AT89S51 dengan C/C++ dan Assembler, ANDI,Yogyakarta.

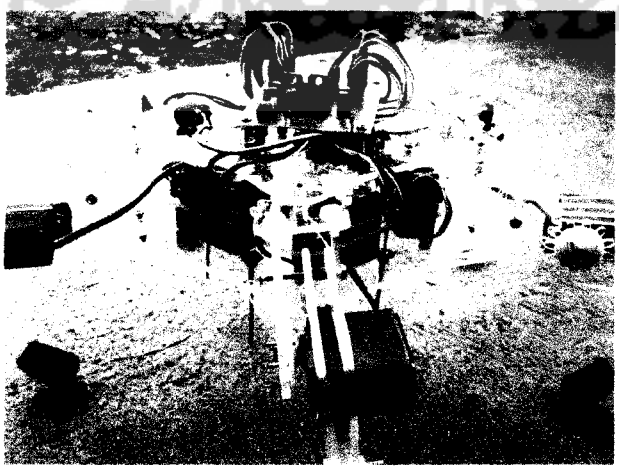
LAMPIRAN



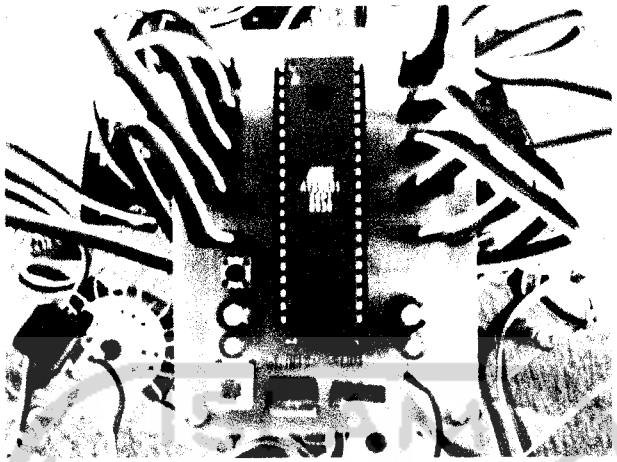
Gambar : Sistem minimum mikrokontroler AT89S51
Pada sistem robot



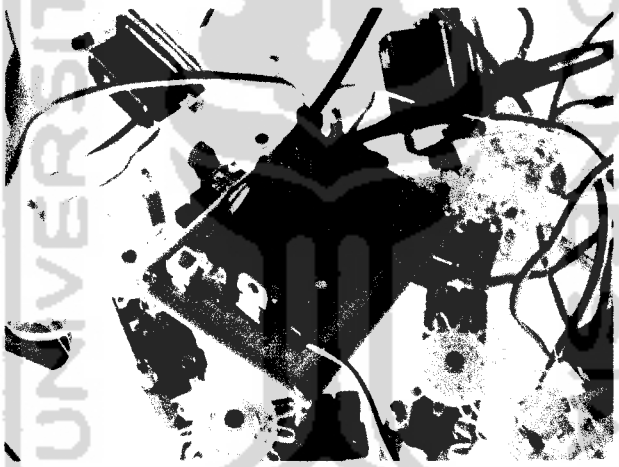
Gambar : Robot tampak dari atas



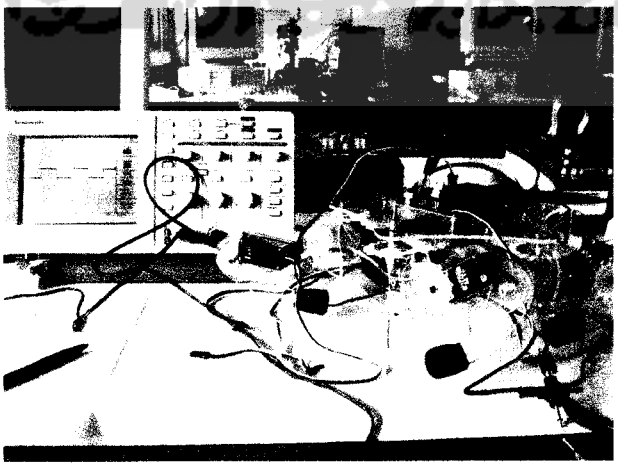
Gambar : Robot tampak dari samping



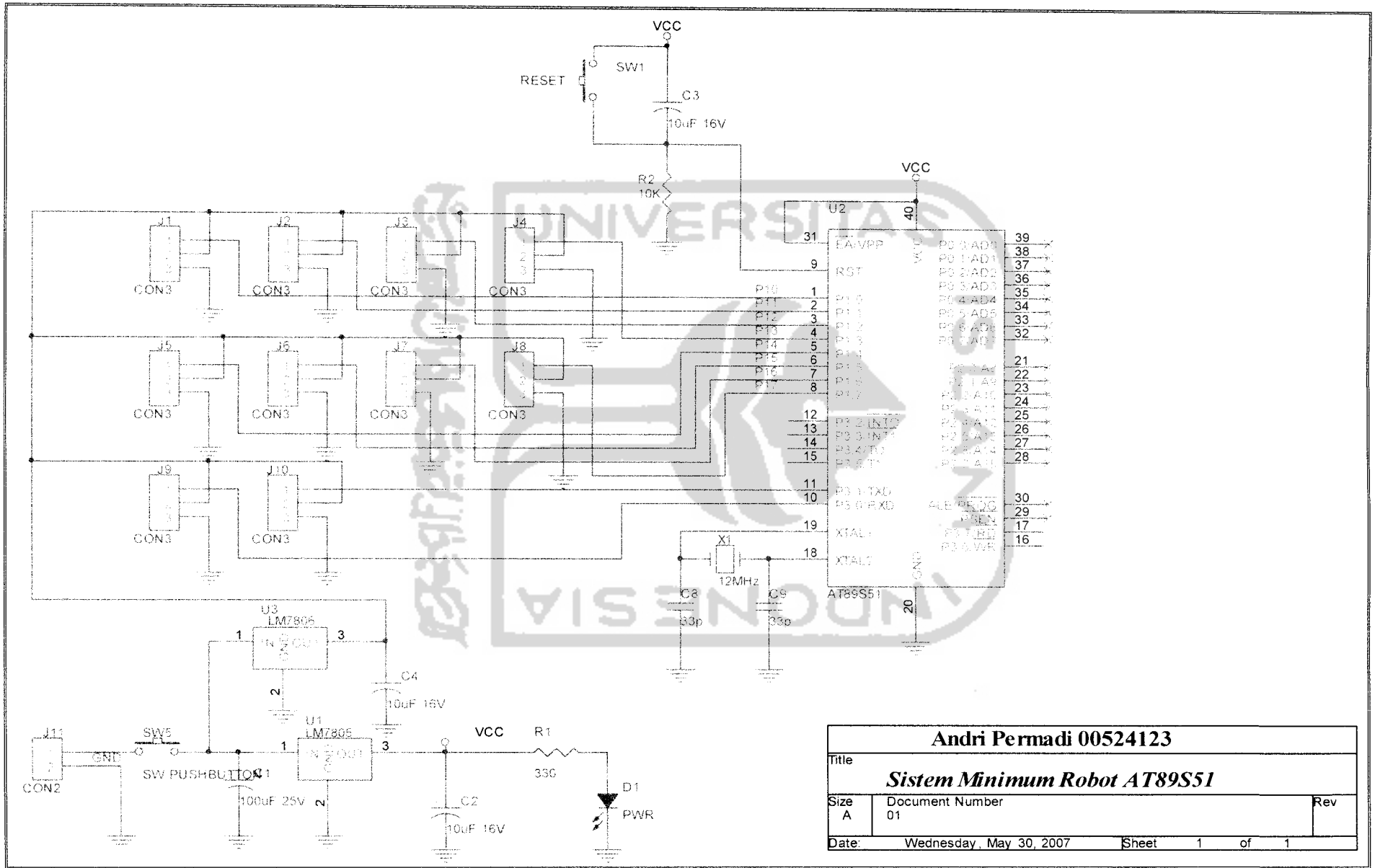
Gambar : Motor servo terhubung dengan sistem minimum robot



Gambar : Analisa PWM dengan menghubungkan output data dengan ground



Gambar : Pengamatan Sinyal PWM dengan Osiloskop Textronik



Andri Permadi 00524123		
Title	***Sistem Minimum Robot AT89S51***	
Size A	Document Number 01	Rev
Date: Wednesday, May 30, 2007	Sheet 1 of 1	

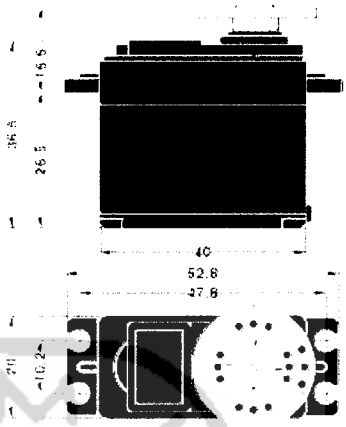
```
;-------------------------------------------------------------------
;########Listing Program Pengaturan Pergerakan Kaki Pada Robot#######
;-------------------------------------------------------------------
              org     00h
              sjmp    mulai
;-------------------------------------------------------
;############inisialisasi mikroprosesor#########
;-----------------------------------------------
              flag1   bit     00h
              flag2   bit     01h
              a1      bit     18h
              b1      bit     19h
              a2      bit     1ah
              b2      bit     1bh
              a3      bit     1ch
              b3      bit     1dh
              a4      bit     1eh
              b4      bit     1fh
              a5      bit     20h
              b5      bit     21h
              kaki1a  bit     p1.0
              kaki1b  bit     p1.1
              kaki2a  bit     p1.2
              kaki2b  bit     p1.3
              kaki3a  bit     p1.4
              kaki3b  bit     p1.5
              kaki4a  bit     p1.6
              kaki4b  bit     p1.7
              kaki5a  bit     p3.0
              kaki5b  bit     p3.1
              org     00h
              sjmp    mulai
              org     0bh
              acall   waktu0
              reti
              org     1bh
              acall   waktu1
              reti
      mulai:  mov     r0,#00h
      star:   mov     @r0,#00h
              inc     r0
              cjne    r0,#70h,star
              MOV     SP,#60H
              MOV     IE,#10001010B
              MOV     TMOD,#12H
              MOV     TL0,#-0fbH      ;faH,0e6h
              MOV     TH0,#-0fbH      ;faH,0e6h
              MOV     TL1,#-050H      ;050H
              MOV     TH1,#-0c3H      ;0c3H
;-----------------------------------------------
;################Program Utama#################
;-----------------------------------------------
      aaa:
      ;kaki 1
              mov     23h,#00000001b      ;satu a
              mov     24h,#00000000b
              acall   kanan
              acall   ull
```

```
        acall   delay
        mov     23h,#00000010b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h,#00000001b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00000010b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
;kaki2
        mov     23h,#00000100b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay;
        mov     23h,#00001000b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        mov     23h,#00000100b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00001000b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
;kaki3
        mov     23h,#00010000b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h,#00100000b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00010000b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00100000b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay
```

```
;kaki4
        mov     23h,#01000000b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h,#10000000b      ;satu a
        mov     24h,#00000000b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h,#01000000b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#10000000b      ;satu a
        mov     24h,#00000000b
        acall   kiri
        acall   ull
        acall   delay
;kaki5
        mov     23h,#00000000b      ;satu a
        mov     24h,#00000001b
        acall   kanan
        acall   ull
        acall   delay
        mov     23h,#00000000b      ;satu a
        mov     24h,#00000010b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00000000b      ;satu a
        mov     24h,#00000001b
        acall   kiri
        acall   ull
        acall   delay
        mov     23h,#00000000b      ;satu a
        mov     24h,#00000010b
        acall   kanan
        acall   ull
        acall   delay
        ljmp    aaa
;-----------------------------------------
;############subrutin pulsa kanan##########
;-----------------------------------------
    kiri:   mov     21h,#07h
            setb    tr0
            setb    tr1
            ret
;-----------------------------------------
;############subrutin pulsa kiri###########
;-----------------------------------------
    kanan:  mov     21h,#05h
            setb    tr0
```

```
              setb    tr1
              ret
;-----------------------------------------
;########subrutin pulsa kiri################
;-----------------------------------------
      STOP:   mov     21h,#06h
              setb    tr0
              setb    tr1
              ret
;-----------------------------------------
;##############subrutin pulsa###############
;-----------------------------------------
      ull:    jnb     flag1,time
              clr     flag1
              jnb     a1,satu
              cpl     kaki1a
              sjmp    ull
      satu:   jnb     b1,dua
              cpl     kaki1b
              sjmp    ull
      dua:    jnb     a2,duaa
              cpl     kaki2a
              sjmp    ull
      duaa:   jnb     b2,tiga
              cpl     kaki2b
              sjmp    ull
      tiga:   jnb     a3,tigaa
              cpl     kaki3a
              sjmp    ull
      tigaa:  jnb     b3,empat
              cpl     kaki3b
              sjmp    ull
      empat:  jnb     a4,empata
              cpl     kaki4a
              sjmp    ull
      empata: jnb     b4,lima
              cpl     kaki4b
              sjmp    ull
      lima:   jnb     a5,limaa
              cpl     kaki5a
              sjmp    ull
      limaa:  jnb     b5,ull
              cpl     kaki5b
              sjmp    ull
      time:   jnb     flag2,ull
              clr     flag2
              clr     tr0
              clr     tr1
              mov     p1,#00h
              mov     p3,#00h
              ret
;-----------------------------------------
;##########subrutin pulsa timer 0############
;-----------------------------------------
     waktu0:  inc     22h
              mov     a,21h
```

```
                cjne    a,22h,belum
                mov     22h,#00h
                setb    flag1
        belum:  ret
;---------------------------------------
;######## subrutin timer 1 ################
;---------------------------------------
        waktu1: MOV     TL1,#-050H      ;50H
                MOV     TH1,#-0c3H      ;c3H
                inc     r7
                cjne    r7,#0ah,blom
                mov     r7,#00h
                setb    flag2
        blom:   ret
;---------------------------------------
;######pengaturan tunda###################
;---------------------------------------
        delay:  mov     50h,#00h
        delay1: mov     51h,#07fh
        delay2: mov     52h,#01h        ; lama tunda 05h
        delay3:
                nop
                djnz    52h,delay3
                djnz    51h,delay2
                djnz    50h,delay1
                ret
        end.
```



The HS-311 sport servo provides the novice modeler all the performance and reliability you would expect to find in a more expensive servo. Combined with precise resin gears and SMT circuitry, the HS-311 represents a remarkable value in today's R/C market.

Operating Speed: 4.8/6.0v: 0.19 / 0.15 sec.
Output Torque: 4.8/6.0v: 42 / 49 oz. 3.0 / 3.7 kg.
Size: 1.51 oz. 43 g.
Weight: 1.6 x 0.8 x 1.4" 40 x 20 x 37mm

DETAILED SPECIFICATIONS

Control System: +Pulse Width Control 1500usec Neutral
Required Pulse: 3-5 Volt Peak to Peak Square Wave
Operating Voltage: 4.8-6.0 Volts
Operating Temperature Range: -20 to +60 Degree C
Operating Speed (4.8V): 0.19sec/60 degrees at no load
Operating Speed (6.0V): 0.15sec/60 degrees at no load
Stall Torque (4.8V): 42 oz/in (3.0 kg/cm)
Stall Torque (6.0V): 48.60 oz/in (4.5 kg/cm)
Current Drain (4.8V): 7.4mA/idle and 160mA no load operating
Current Drain (6.0V): 7.7mA/idle and 180mA no load operating
Dead Band Width: 5usec
Operating Angle: 40 Deg. one side pulse traveling 400usec
Direction: Clockwise/Pulse Traveling 1500 to 1900usec
Motor Type: Cored Metal Brush
Potentiometer Drive: 4 Slider/Direct Drive
Bearing Type: Top/Resin Bushing
Gear Type: Nylon
360 Modifiable: Yes
Connector Wire Length: 11.81" (300mm)
Dimensions: 1.57" x 0.79"x 1.44" (40 x 20 x 36.5mm)
Weight: 1.52oz (43g)
www.hitecrcd.com

Features

- Compatible with MCS-51® Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
  - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

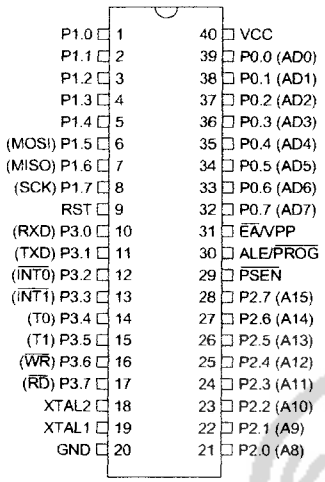
8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

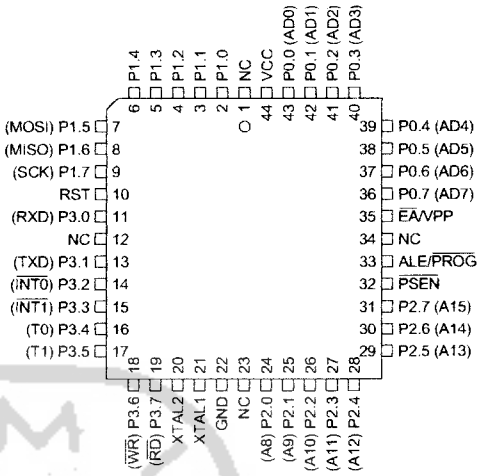
Rev. 2487A–10/01

Pin Configurations

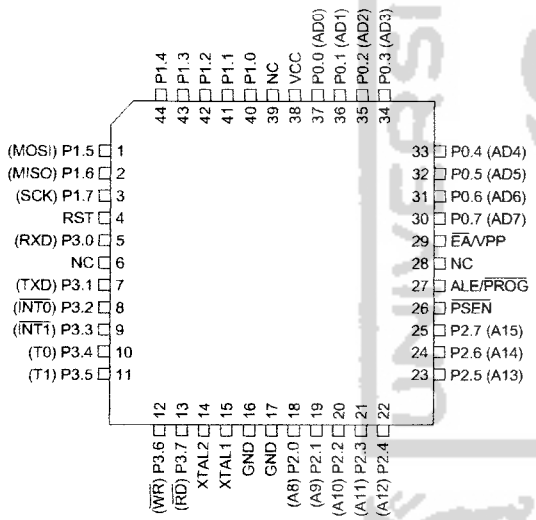
PDIP



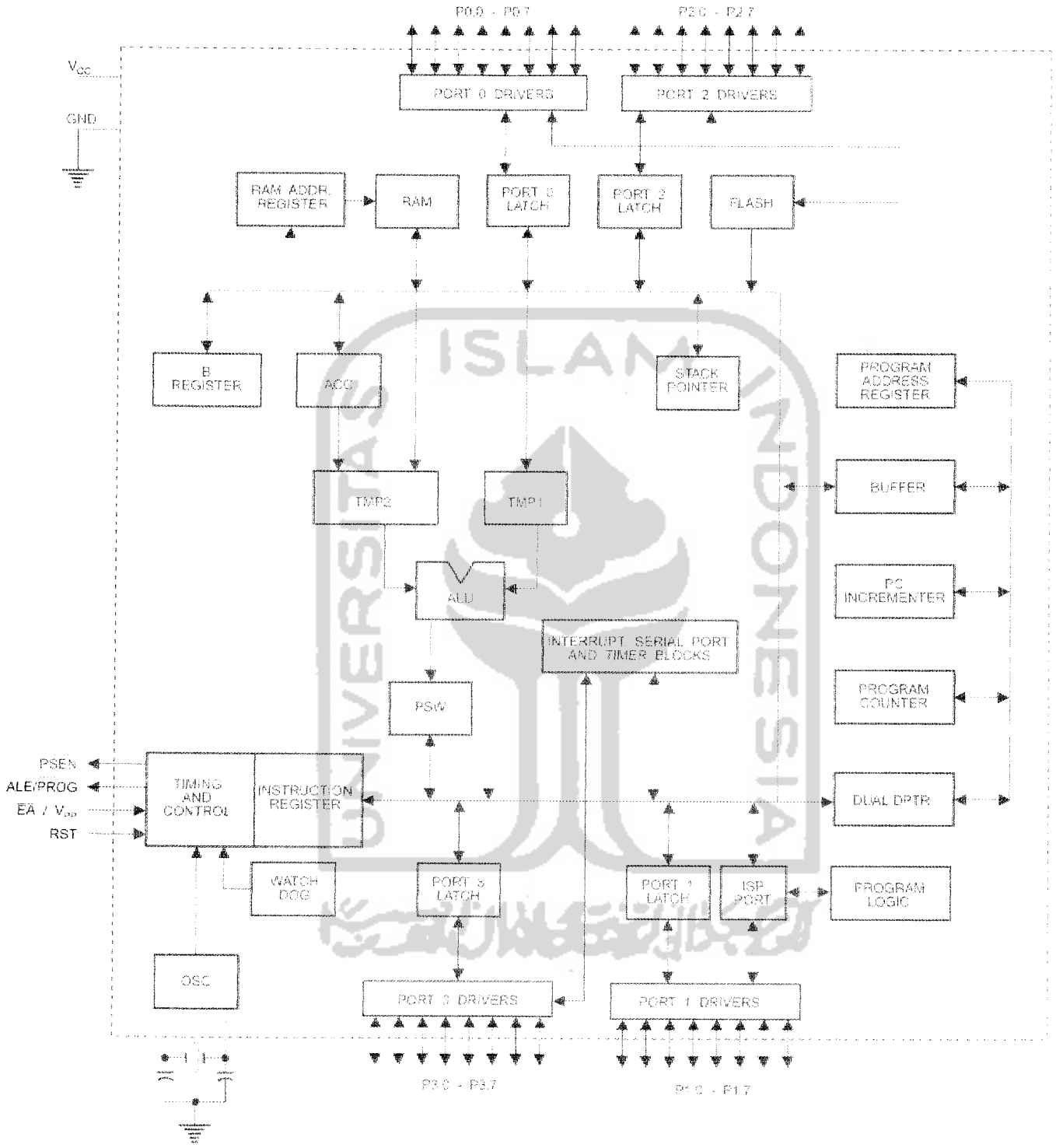
PLCC



TQFP



Block Diagram



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port Pin	Alternate Functions
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pull-ups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pull-ups.

Port 3 receives some control signals for Flash programming and verification.

Port 3 also serves the functions of various special features of the AT89S51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/$\overline{PROG}$

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable ($\overline{PSEN}$) is the read strobe to external program memory.

When the AT89S51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

Table 1. AT89S51 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the five interrupt sources in the IP register.

Table 2. AUXR: Auxiliary Register

AUXR Address = 8EH Reset Value = XXX00XX0B

Not Bit Addressable

Bit	–	–	–	WDIDLE	DISRTO	–	–	DISALE
	7	6	5	4	3	2	1	0

– Reserved for future expansion

DISALE Disable/Enable ALE

DISALE Operating Mode

0 ALE is emitted at a constant rate of 1/6 the oscillator frequency

1 ALE is active only during a MOVX or MOVC instruction

DISRTO Disable/Enable Reset out

DISRTO

0 Reset pin is driven High after WDT times out

1 Reset pin is input only

WDIDLE Disable/Enable WDT in IDLE mode

WDIDLE

0 WDT continues to count in IDLE mode

1 WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag: The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and rest under software control and is not affected by reset.

Table 3. AUXR1: Auxiliary Register 1

AUXR1

Address = A2H

Reset Value = XXXXXXX0B

Not Bit Addressable

	–	–	–	–	–	–	–	DPS
Bit	7	6	5	4	3	2	1	0

– Reserved for future expansion

DPS Data Pointer Register Select

DPS	
0	Selects DPTR Registers DP0L, DP0H
1	Selects DPTR Registers DP1L, DP1H

Memory Organization

MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory

If the $\overline{EA}$ pin is connected to GND, all program fetches are directed to external memory.

On the AT89S51, if $\overline{EA}$ is connected to V_{CC}, program fetches to addresses 0000H through FFFH are directed to internal memory and fetches to addresses 1000H through FFFFH are directed to external memory.

Data Memory

The AT89S51 implements 128 bytes of on-chip RAM. The 128 bytes are accessible via direct and indirect addressing modes. Stack operations are examples of indirect addressing, so the 128 bytes of data RAM are available as stack space.

Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is 98xTOSC, where TOSC=1/FOSC. To make the best use of the WDT, it

should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt, which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S51 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S51 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the ATMEL Web site (http://www.atmel.com). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 4 shows that bit position IE.6 is unimplemented. In the AT89S51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle

Table 4. Interrupt Enable (IE) Register

(MSB) (LSB)

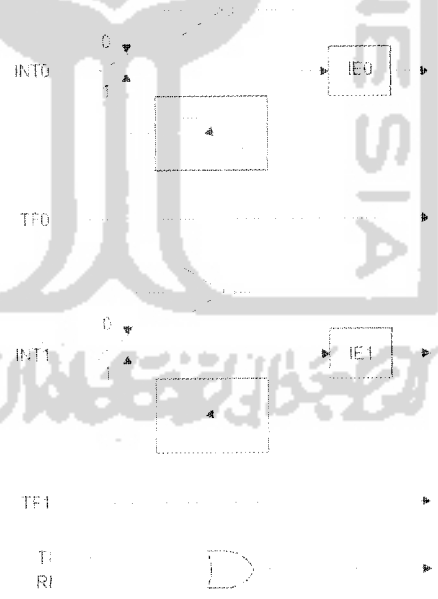
EA	–	–	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved
–	IE.5	Reserved
ES	IE.4	Serial Port interrupt enable bit
ET1	IE.3	Timer 1 interrupt enable bit
EX1	IE.2	External interrupt 1 enable bit
ET0	IE.1	Timer 0 interrupt enable bit
EX0	IE.0	External interrupt 0 enable bit
User software should never write 1s to reserved bits, because they may be used in future AT89 products.		

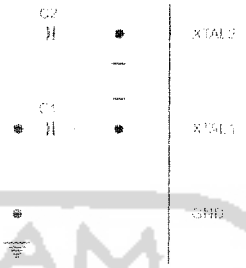
Figure 1. Interrupt Sources



Oscillator Characteristics

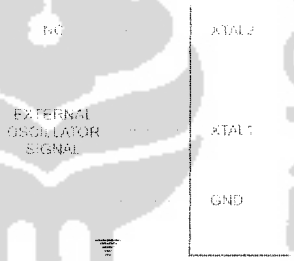
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals = 40 pF ± 10 pF for Ceramic Resonators

Figure 3. External Clock Drive Configuration



Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special function registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by activation of an enabled external interrupt into $\overline{INT0}$ or $\overline{INT1}$. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Table 5. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89S51 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 6. Lock Bit Protection Modes

	Program Lock Bits			
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled
3	P	P	U	Same as mode 2, but verify is also disabled
4	P	P	P	Same as mode 3, but external execution is also disabled

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash – Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S51, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 µs. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89S51 features $\overline{Data}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{\text{Busy}}$: The progress of byte programming can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. P3.0 is pulled low after ALE goes high during programming to indicate $\overline{\text{BUSY}}$. P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(100H) = 51H indicates 89S51
(200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/$\overline{\text{PROG}}$ low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC}. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
   Apply power between VCC and GND pins.
   Set RST pin to "H".
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

Set XTAL1 to "L" (if a crystal is not used).

Set RST to "L".

Turn V_{CC} power off.

Data Polling: The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 8 on page 18.

Programming Interface – Parallel Mode

Every code byte in the Flash array can be programmed by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 7. Flash Programming Modes

Mode	V_{CC}	RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.3-0 Address	P1.7-0 Address
Write Code Data	5V	H	L	(2)	12V	L	H	H	H	H	D_{IN}	A11-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A11-8	A7-0
Write Lock Bit 1	5V	H	L	(3)	12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L	(3)	12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L	(3)	12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L	(1)	12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	51H	0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	0010	00H

Notes:
1. Each PROG pulse is 200 ns - 500 ns for Chip Erase.
2. Each PROG pulse is 200 ns - 500 ns for Write Code Data.
3. Each PROG pulse is 200 ns - 500 ns for Write Lock Bits.
4. RDY/BSY signal is output on P3.0 during programming.
5. X = don't care.

Figure 4. Programming the Flash Memory (Parallel Mode)

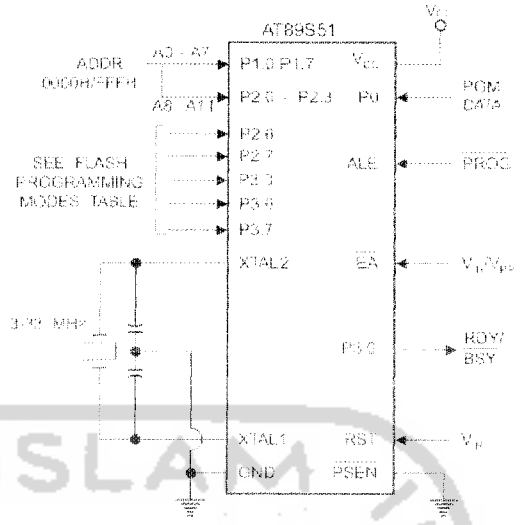
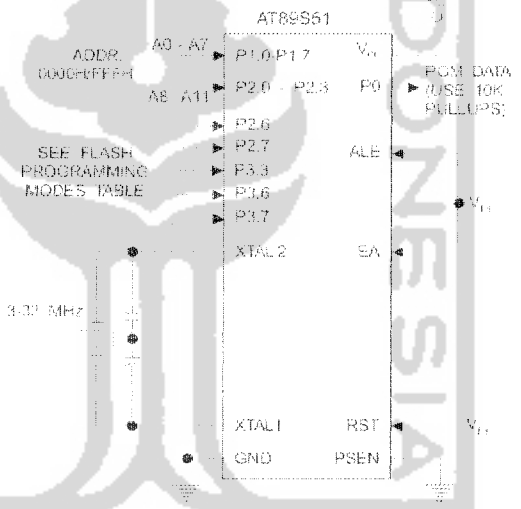


Figure 5. Verifying the Flash Memory (Parallel Mode)



Flash Programming and Verification Characteristics (Parallel Mode)

T_A = 20°C to 30°C, V_{CC} = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
$1/t_{CLCL}$	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	0.2	1	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		50	µs

Figure 6. Flash Programming and Verification Waveforms – Parallel Mode

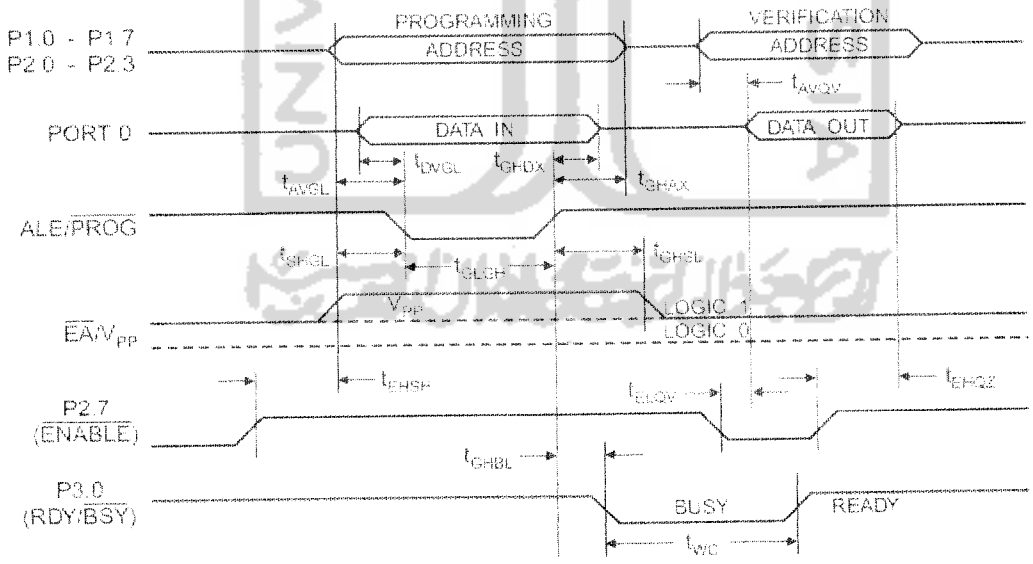
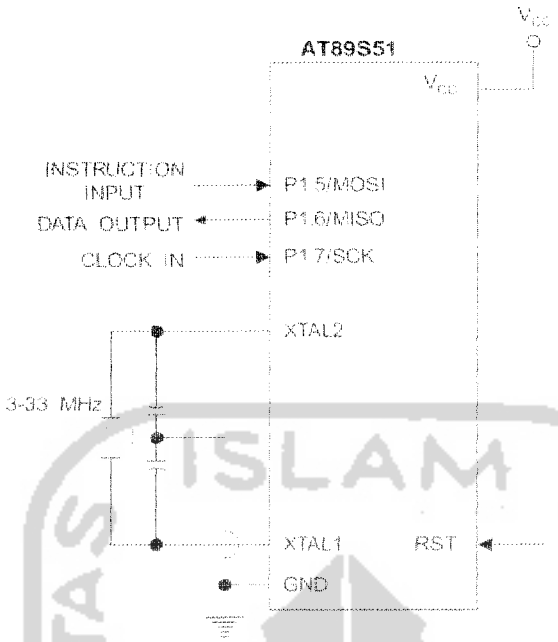


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms

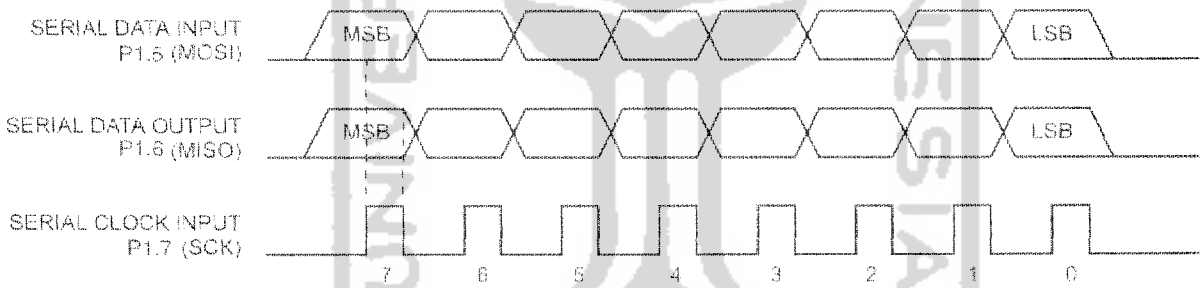


Table 8. Serial Programming Instruction Set

Instruction	Instruction Format				
	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**	**Operation**
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits[2]	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes[1]	0010 1000	xxx A5 A4 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Notes: 1. The signature bytes are not readable in Lock Bit Modes 3 and 4.

2. B1 = 0, B2 = 0 → Mode 1, no lock protection
   B1 = 0, B2 = 1 → Mode 2, lock bit 1 activated
   B1 = 1, B2 = 0 → Mode 3, lock bit 2 activated
   B1 = 1, B1 = 1 → Mode 4, lock bit 3 activated

   } Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

After Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at XTAL1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

Serial Programming Characteristics

Figure 9. Serial Programming Timing

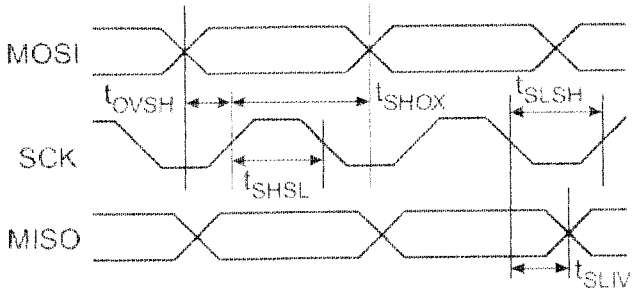


Table 9. Serial Programming Characteristics, T_A = -40° C to 85° C, V_{CC} = 4.0 - 5.5V (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	8 t_{CLCL}			ns
t_{SLSH}	SCK Pulse Width Low	8 t_{CLCL}			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	2 t_{CLCL}			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	µs

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 4.0V to 5.5V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ±10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode[2]	V_{CC} = 5.5V		50	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port:
   Port 0: 26 mA Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

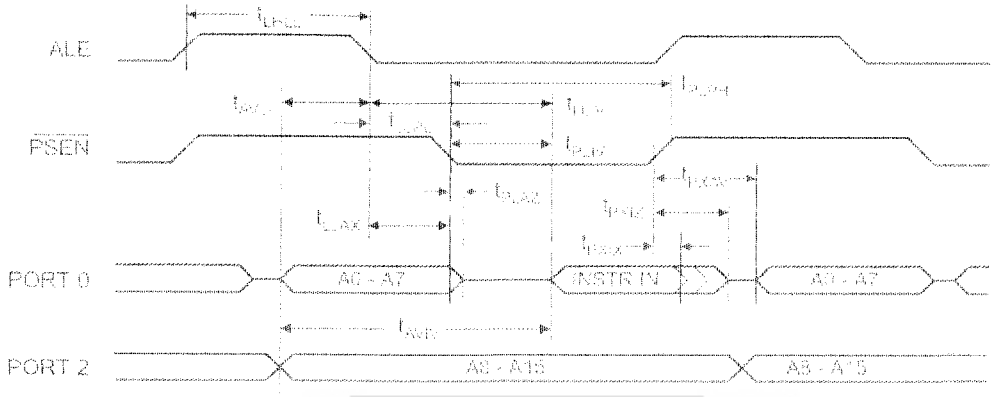
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

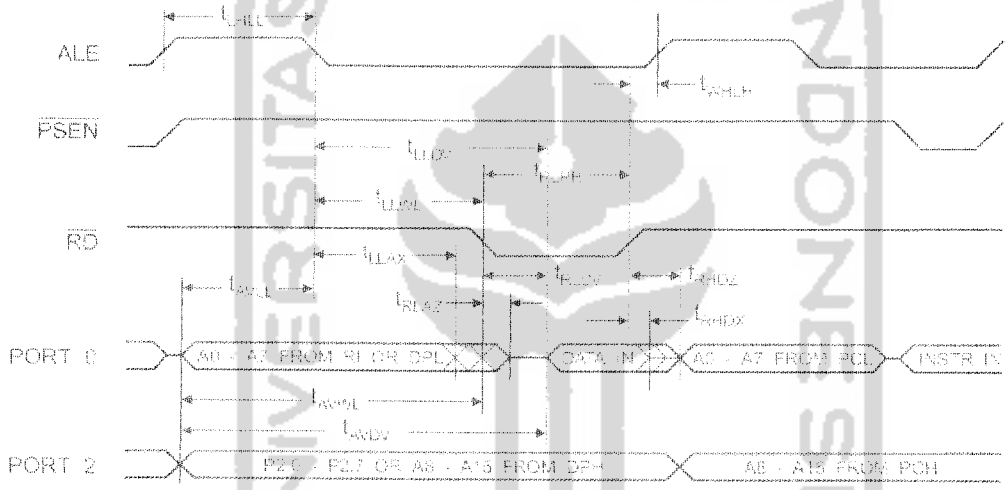
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-45		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-130		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

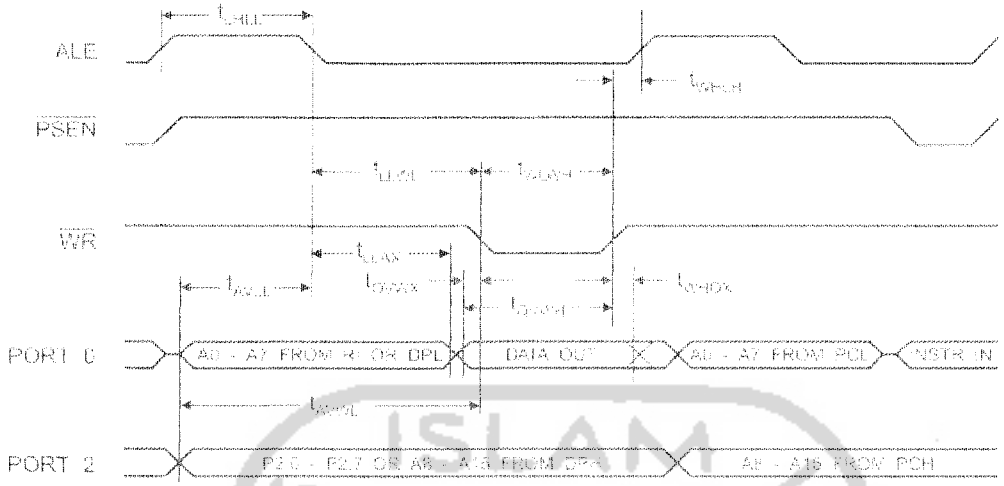
External Program Memory Read Cycle



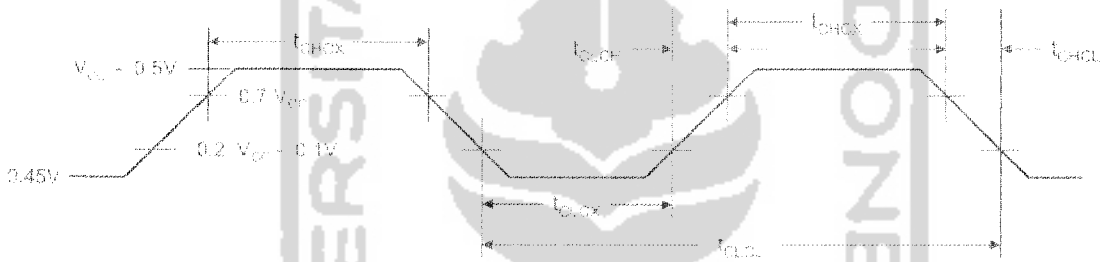
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

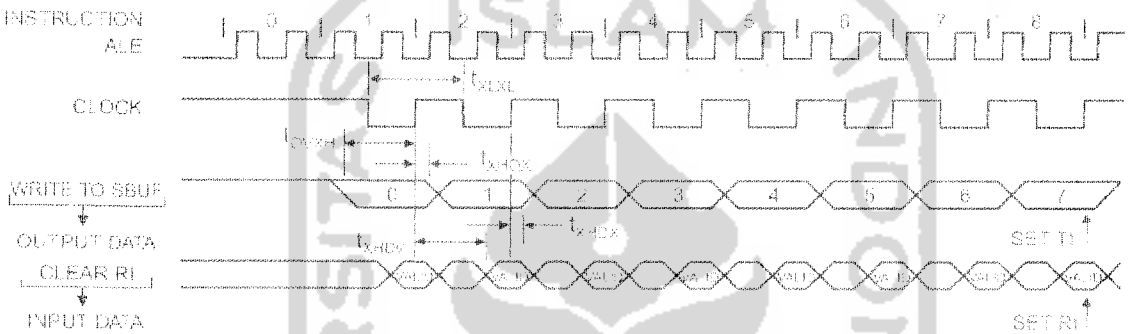
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
t_{CLCL}	Clock Period	30		ns
t_{CHCX}	High Time	12		ns
t_{CLCX}	Low Time	12		ns
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

Serial Port Timing: Shift Register Mode Test Conditions

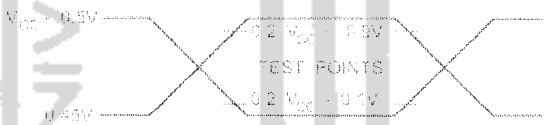
The values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-80		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms



AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]

Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

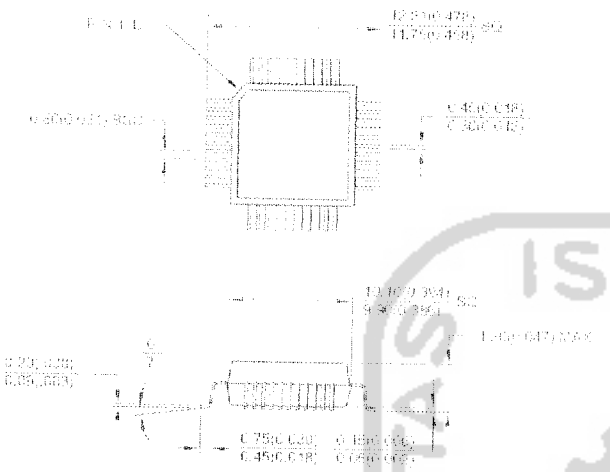
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AC AT89S51-24JC AT89S51-24PC	44A 44J 40P6	Commercial (0° C to 70° C)
		AT89S51-24AI AT89S51-24JI AT89S51-24PI	44A 44J 40P6	Industrial (-40° C to 85° C)
33	4.5V to 5.5V	AT89S51-33AC AT89S51-33JC AT89S51-33PC	44A 44J 40P6	Commercial (0° C to 70° C)

= Preliminary Availability

	Package Type
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

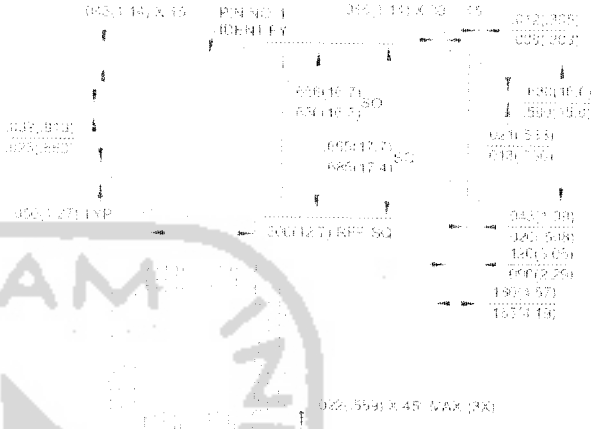
Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Millimeters and (Inches)*

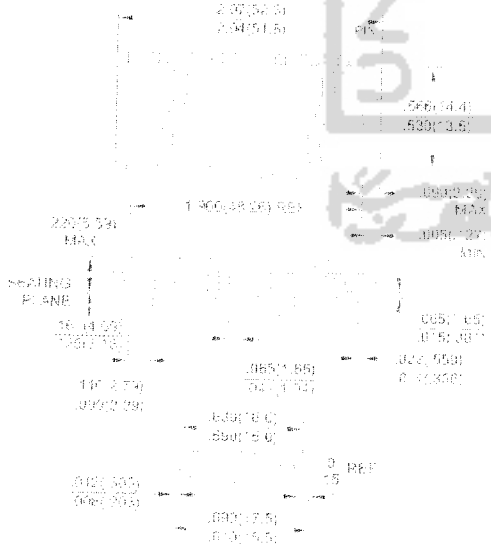


*Controlling dimension: millimeters

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)



40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AC



Atmel Headquarters

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

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

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

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

Atmel Product Operations

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

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

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

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

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

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

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

2487A–10/01/xM