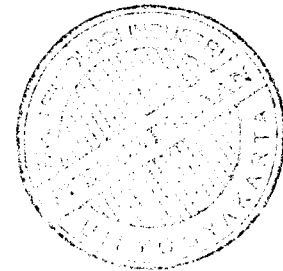
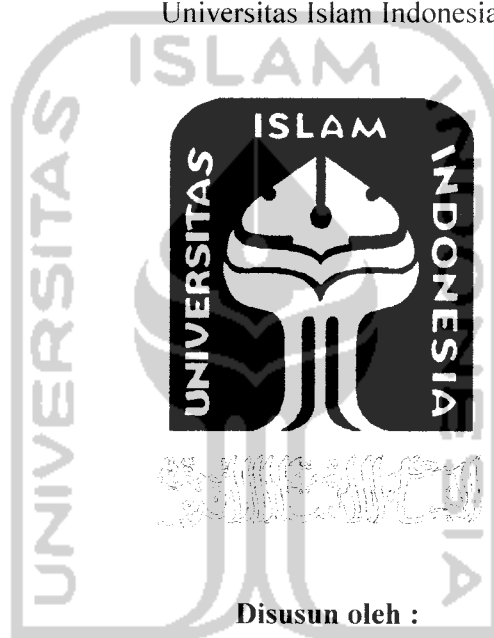


**SIMULASI PENGENDALI PINTU IRIGASI
DENGAN MENGGUNAKAN DTMF MOBILE PHONE
BERBASIS MIKROKONTROLER AT89S51**

TUGAS AKHIR

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



Disusun oleh :

Nama : Nur Budi Agung Nugroho

No Mahasiswa : 00 524 013

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

YOGYAKARTA

2007

LEMBAR PENGESAHAN PENGUJI

**SIMULASI PENGENDALI PINTU IRIGASI
DENGAN MENGGUNAKAN DTMF MOBILE PHONE
BERBASIS MIKROKONTROLER AT89S51**

TUGAS AKHIR

Oleh :

Nama : Nur Budi Agung Nugroho
No. Mahasiswa : 00 524 013

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

Yogyakarta, September 2007

Tim Penguji

Tito Yuwono, ST, M.Sc

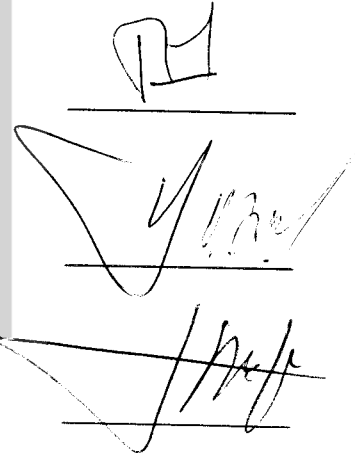
Ketua

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)

MOTTO

“...Sungguh, Allah tidak akan mengubah nasib suatu kaum sampai mereka sendiri mengubah dirinya”

(QS Ar Ra'd : 11)

“...Allah akan meninggikan orang-orang yang beriman diantara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat. Dan Allah mengetahui atas apa yang kamu kerjakan”

(QS. Al Mujadalah : 11)

“...Barang siapa melalui jalan mencari ilmu, maka Allah melewatkannya pada jalan menuju surga”

(Sabda Rasulullah SAW)

“...Sesungguhnya apabila kamu bersyukur, pasti Aku akan menambah nikmat-Ku, dan jika kamu mengingkari, sesungguhnya azab-Ku sangat pedih”

(Q.S Ibrahim: 7)

PERSEMBAHAN

*Dengan perasaan bahagia dan syukur kepada Allah SWT,
Kupersembahkan laporan Tugas Akhir ini kepada
Bapak dan Ibu tercinta,
Serta kakak dan adikku tersayang,
Yang telah memberikan dorongan semangat dan
segalanya...*



KATA PENGANTAR



Assalamu'alaikum Wr. Wb

Alhamdulillah, kami panjatkan puji syukur ke hadirat Allah SWT yang telah memberikan hidayah-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan judul **“SIMULASI PENGENDALI PINTU IRIGASI DENGAN MENGGUNAKAN DTMF MOBILE PHONE BERBASIS MIKROKONTROLER AT89S51”** yang merupakan salah satu syarat guna memperoleh gelar sarjana S-1 dikalangan Universitas Islam Indonesia Fakultas Teknologi Industri Jurusan Teknik Elektro.

Penulisan dan penyelesaian Tugas Akhir ini bisa selesai tidak lepas dari saran-saran, bimbingan, dukungan, serta bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya 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 Tito Yuwono, ST, M.Sc, selaku dosen pembimbing I, yang telah berkenan memberikan bimbingan selama berlangsungnya pembuatan Tugas Akhir ini.

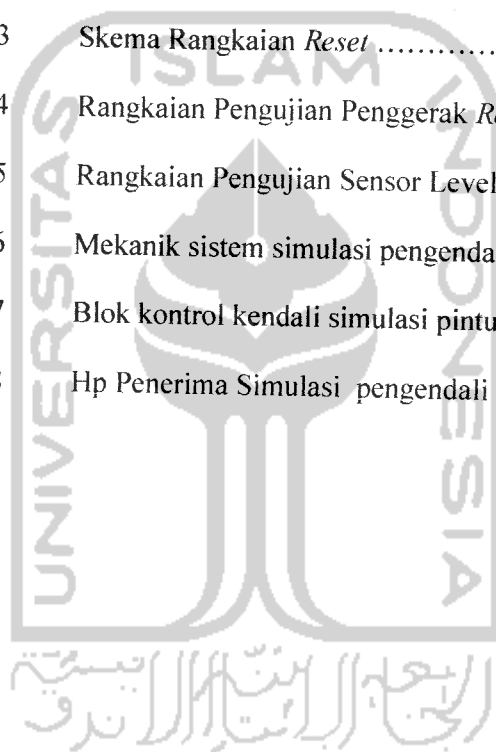
DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI	iii
HALAMAN MOTTO	iv
HALAMAN PERSEMBAHAN	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Identifikasi Masalah	2
1.3 Batasan Masalah	3
1.4 Rumusan Masalah	4
1.5 Tujuan Penelitian	4
1.6 Manfaat Penelitian	5
1.7 Sistematika Penulisan	5
BAB II LANDASAN TEORI	7
2.1 Mikrokontroler AT89S51	7
2.1.1 Perangkat Keras (<i>Hardware</i>)	7

2.1.2 Perangkat Lunak (<i>Software</i>)	13
2.2 Sinyal DTMF	16
2.3 MT 8870 (<i>DTMF Receiver</i>)	16
2.4 Penguat Sinyal (LM 386)	20
2.5 Transistor sebagai saklar	21
2.6 <i>Relay</i>	26
2.7 Motor DC	28
2.8 <i>Micro Switch</i>	30
2.9 <i>Software</i>	30
BAB III PERANCANGAN SISTEM	32
3.1 Prinsip Kerja	32
3.2 Perancangan Perangkat Keras	34
3.2.1 Penerima DTMF (<i>DTMF Receiver</i>)	35
3.2.2 Sistem Minimum Mikrokontroler AT89S51	37
3.2.3 Rangkaian Sensor Pembatas dengan <i>Micro Switch</i>	38
3.2.4 Rangkaian Penggerak Motor	39
3.2.5 Rangkaian Sensor Level Air	42
3.2.6 Rangkaian Pengendali <i>Hand Phone</i> Penerima	43
3.2.7 Rangkaian <i>Power Supply</i>	44
3.3 Perancangan Perangkat Lunak	46
3.3.1 Program Utama	48
3.3.2 <i>Subroutin</i> Program CEK_HP	49

BAB IV ANALISA DAN PEMBAHASAN	50
4.1 Pengujian Rangkaian Penerima DTMF	50
4.2 Pengujian Sistem Minimum Mikrokontroler	51
4.3 Pengujian Rangkaian Sensor Pembatas	54
4.4 Pengujian Rangkaian Penggerak <i>Relay</i>	55
4.5 Pengujian Rangkaian Sensor Level Air	56
4.6 Gambar Sistem Simulasi Pengendali Pintu Irigasi	57
4.7 Perhitungan Arus dan Daya Simulasi Pintu Irigasi	58
4.7.1 Timing buka tutup pintu irigasi	58
4.7.2 Perhitungan arus dan daya pada output sistem	59
4.7.3 Perhitungan arus dan daya pada output power supply (5V,6V,24V)	59
BAB V PENUTUP	61
5.1 Kesimpulan	61
5.2 Saran	62
DAFTAR PUSTAKA	
LAMPIRAN	

Gambar 3.7	Rangkaian Sensor Level Air	42
Gambar 3.8	Rangkaian Pengendali <i>Hand Phone</i> Penerima	44
Gambar 3.9	Rangkaian Catu Daya (<i>Power Supply</i>)	45
Gambar 3.10	<i>Flowchart</i> Program Utama	48
Gambar 3.11	<i>Flowchart Subroutin</i> Program CEK_HP	49
Gambar 4.1	Rangkaian Uji Unit Penerima DTMF	50
Gambar 4.2	Rangkaian Pengujian Sistem Minimum	52
Gambar 4.3	Skema Rangkaian <i>Reset</i>	53
Gambar 4.4	Rangkaian Pengujian Penggerak <i>Relay</i>	55
Gambar 4.5	Rangkaian Pengujian Sensor Level Air	57
Gambar 4.6	Mekanik sistem simulasi pengendali pintu irigasi	57
Gambar 4.7	Blok kontrol kendali simulasi pintu irigasi	58
Gambar 4.8	Hp Penerima Simulasi pengendali pintu irigasi	58



DAFTAR TABEL

Tabel 2.1	Fungsi-Fungsi Khusus Kaki-Kaki <i>Port</i> 1 AT89S51	9
Tabel 2.2	Fungsi Khusus <i>Port</i> 3	9
Tabel 2.3	Nilai Frekuensi DTMF	16
Tabel 2.4	Penjelasan tiap pin pada IC MT 8870	18
Tabel 2.5	Fungsi Frekuensi DTMF	20
Tabel 3.1	Logika Input Rangkaian Penggerak Motor	42
Tabel 4.1	Data Hasil Pengujian Rangkaian Penerima DTMF	51
Tabel 4.2	Hasil Pengujian Rangkaian Tombol	54
Tabel 4.3	Hasil Pengujian Rangkaian Penggerak <i>Relay</i>	56
Tabel 4.4	Timing buka tutup pintu irigasi	58

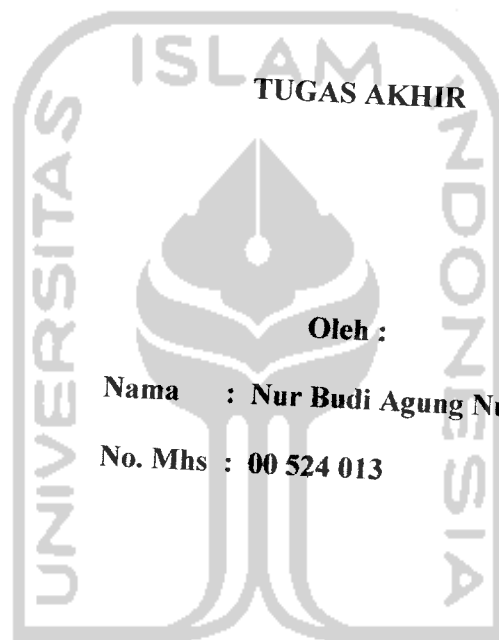


ABSTRAK

Tugas akhir ini bertujuan untuk mengembangkan pengendalian pintu irigasi di daerah pedesaan yang umumnya masih dikendalikan secara manual menjadi pengendalian secara otomatis dengan menggunakan sinyal DTMF *mobile phone* sebagai *input* ke sistem dan aplikasi *password*, sehingga sistem pengendalian pintu irigasi menjadi lebih efisien dan aman. Sistem pengendali pintu irigasi ini terdiri dari (1) Sistem minimum mikrokontroler AT89S51, (2) Rangkaian penerima sinyal DTMF, (3) Rangkaian pengendali motor, (4) Rangkaian sensor level air, (5) Rangkaian sensor pembatas dengan menggunakan *micro switch*, (6) Rangkaian pengendali *hand phone* penerima. Rangkaian penerima DTMF berfungsi untuk menerima sinyal *input* dari *mobile phone*, dimana sinyal ini dibuat suatu kode (*password*) yang dapat menentukan proses pengendalian. Pengendali motor berfungsi untuk memutar motor ke kanan atau ke kiri yang digunakan untuk membuka dan menutup pintu irigasi. Sensor level air berfungsi untuk mendeteksi ketinggian air yang digunakan untuk mengetahui kondisi air penuh atau kosong. Sensor pembatas dengan menggunakan *micro switch* berfungsi untuk membatasi gerak motor atas dan bawah/untuk menghentikan putaran motor apabila telah menyentuh saklar yang dipasang di atas dan di bawah mekanik sistem. Rangkaian pengendali *hand phone* penerima berfungsi untuk melakukan panggilan (*call*) ke *hand phone* pengendali apabila sensor level air mendeteksi kondisi air penuh. Sistem ini dirancang sebagai peringatan untuk pengendali. Mikrokontroler merupakan komponen utama pengendali secara keseluruhan dengan perangkat lunak (*software*) yang dimasukkan kedalamnya. Setelah melakukan uji kerja dari keseluruhan sistem, sistem kendali pintu irigasi dengan menggunakan DTMF dapat berfungsi untuk membuka dan menutup pintu irigasi sesuai dengan aplikasi *password* yang telah ditentukan (telah di *setting*). Sistem mampu memberikan data ketinggian air dengan menggunakan sensor level air yang berfungsi untuk mendeteksi kondisi air penuh atau kosong. Sistem mampu melakukan pengendalian jarak jauh menggunakan jaringan telekomunikasi. Dengan demikian simulasi pengendali pintu irigasi dengan menggunakan DTMF ini sudah dapat bekerja dengan baik sesuai dengan yang diharapkan.

LEMBAR PENGESAHAN PEMBIMBING

**SIMULASI PENGENDALI PINTU IRIGASI
DENGAN MENGGUNAKAN DTMF MOBILE PHONE
BERBASIS MIKROKONTROLER AT89S51**



Oleh :

Nama : Nur Budi Agung Nugroho

No. Mhs : 00 524 013

Yogyakarta, 05 September 2007

Pembimbing I

Tito Yuwono, ST, M.Sc

Pembimbing II

Yusuf Aziz Amrullah, ST

DAFTAR GAMBAR

Gambar 2.1	Susunan Kaki Mikrokontroler AT89S51	8
Gambar 2.2	Blok Diagram AT89S51	12
Gambar 2.3	Pin MT 8870	17
Gambar 2.4	IC LM 386	21
Gambar 2.5	Aplikasi LM 386 Untuk Gain 200 Kali	21
Gambar 2.6	(a) Transistor NPN (b) Transistor PNP	22
Gambar 2.7	Karakteristik Beban DC Transistor	23
Gambar 2.8	(a) Transistor Dalam Keadaan Konduktif	24
	(b) Ekuivalen Saklar Tertutup	24
Gambar 2.9	(a) Transistor Dalam Keadaan Non Konduktif	24
	(b) Ekuivalen Saklar Terbuka	24
Gambar 2.10	(a) Simbol <i>Relay</i> SPDT (b) <i>Relay</i> SPDT	27
Gambar 2.11	Dasar Pengaturan Arah Putaran Motor	29
Gambar 2.12	Pengaturan Arah Motor Dengan Menggunakan Saklar	29
Gambar 2.13	Sensor <i>Micro Switch</i>	30
Gambar 3.1	Blok Diagram Simulasi Pengendali Pintu Irigasi	33
Gambar 3.2	Blok Diagram Perangkat Keras (<i>Hardware</i>)	34
Gambar 3.3	Rangkaian Penerima DTMF	36
Gambar 3.4	Rangkaian Sistem Minimum Mikrokontroler AT89S1	38
Gambar 3.5	Rangkaian <i>Micro Switch</i>	39
Gambar 3.6	Rangkaian Penggerak Motor	41

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pangan merupakan salah satu kebutuhan pokok dari manusia. Untuk mendapatkan hasil pertanian yang memuaskan, maka tidak lepas dari sistem irigasi yang baik. Umumnya pada musim penghujan, para petani tidak begitu memperlakukan sistem irigasi, karena air dapat datang kapan saja. Saat musim kemarau datang, petani biasanya menggunakan air irigasi baik dari sungai, waduk ataupun rawa. Untuk mendapatkan sistem irigasi yang baik dan adil maka diperlukan manajemen dari pengelola irigasi. Jika tidak dimenejemen dengan baik, biasanya menimbulkan kesenjangan dikalangan para petani. Untuk mengatasi hal ini, maka diperlukan pengaturan debit air yang akan didistribusikan ke areal persawahan petani. Salah satu cara pengaturan debit air ini adalah dengan pengaturan pintu air.

Permasalahannya adalah keamanan dan efisiensi dalam pengendalian pintu air tersebut. Keamanan diperlukan agar tidak setiap orang bisa membuka atau menutup pintu air yang ada. Efisiensi diperlukan agar pengendalian dapat dilakukan dengan mudah dan dapat dikendalikan dari mana saja. Untuk mengatasi hal ini, maka pengendalian jarak jauh merupakan alternatif yang bisa dipakai. Dengan pengendali jarak jauh, maka pengendali dapat membuka/menutup pintu irigasi dimanapun dia berada, tidak harus datang untuk membuka/menutup pintu tersebut. Untuk keamanan agar tidak sembarang orang dapat mengendalikan

pintu, maka pengendali pintu air diaplikasikan dengan penggunaan *password*. Jadi untuk membuka atau menutup pintu air yang ada, maka pengendali harus memasukkan *password*. Jika *password* benar maka pintu dapat dikendalikan sesuai keinginan, tapi jika *password* salah, maka pintu tidak akan bergerak. Dengan sistem ini, sistem pendistribusian air dapat diawasi dan terorganisir dengan baik.

Mengingat perkembangan teknologi komunikasi makin berkembang saat ini, dan setiap daerah sebagian besar sudah terjamah oleh jaringan komunikasi, maka telepon ataupun *mobile phone* merupakan perangkat yang tepat untuk mengendalikan sistem pengendalian pintu irigasi ini. Cara yang mudah adalah dengan memanfaatkan nada tombol yang ada di perangkat *mobile phone* atau telepon. Nada tombol ini merupakan gabungan dua frekuensi yang berbeda yang disebut DTMF (*Dual Tone Multiple Frequency*). Dengan DTMF tidak diperlukan perangkat lain untuk mengirim maupun menerima sinyal ini. Untuk menerima sinyal dilakukan dengan memanfaatkan jalur *handsfree* yang sudah tersedia pada *mobile phone*.

1.2. Identifikasi Masalah

Permasalahan yang ada pada pembuatan simulasi pengendali pintu irigasi dengan menggunakan DTMF (*Dual Tone Multiple Frequency*) ini dapat diidentifikasi sebagai berikut :

1. Banyaknya pintu irigasi di daerah pertanian, terutama di pedesaan yang masih dikendalikan secara manual, sehingga pengendalian menjadi tidak efisien

karena harus dilakukan bolak-balik ke tempat pintu irigasi untuk membuka dan menutupnya.

2. Kurangnya manajemen pendistribusian air irigasi di daerah pertanian karena sistem masih dikendalikan secara manual sehingga tidak efisien dalam pengaturan pendistribusian air.
3. Kurangnya keamanan pintu irigasi, sehingga sering terjadi penyalahgunaan pendistribusian air irigasi karena sistem masih dikendalikan secara manual sehingga untuk membuka maupun menutupnya dapat dilakukan siapapun yang mengakibatkan keamanan menjadi kurang terjamin.
4. Kurangnya tanggung jawab dari orang yang berwenang menangani pintu irigasi, hal ini bisa disebabkan karena sistem yang masih dikendalikan secara manual sehingga kadang-kadang pengendali merasa capek dan lelah untuk bolak-balik membuka maupun menutup pintu irigasi.

1.3. Batasan Masalah

Sistem rancangan alat ini terdiri dari dua perangkat utama, blok pengendali dan blok kontrol kendali pintu irigasi. Untuk menghindari adanya pembahasan diluar materi dan mengingat keterbatasan waktu, maka penyusun dalam mendesain simulasi pengendali pintu irigasi dengan menggunakan DTMF membatasi permasalahan pada :

1. Pengendali

Blok pengendali ini memanfaatkan nada tombol pada *handphone* (*keypad tones*) yang disebut DTMF (*Dual Tone Multiple Frequency*).

2. Kontrol Kendali Pintu Irigasi

Blok kontrol kendali pintu irigasi dibatasi pada unit penerima data, pengkode / penerjemah data, pengolah data, sensor, dan mekanik sistem pintu irigasi.

Dalam proses pengolahan data digunakan mikrokontroler AT89S51 sebagai kendali utama untuk sistem yang dirancang.

1.4. Rumusan Masalah

Dari uraian latar belakang masalah yang telah dijabarkan diatas, maka dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana menggabungkan tiap blok rangkaian dengan rangkaian sistem minimum mikrokontroler agar tercipta sistem pengendali pintu irigasi yang dapat berfungsi dengan baik?
2. Bagaimana kinerja dari simulasi pengendali pintu irigasi dengan menggunakan DTMF?

1.5. Tujuan Penelitian

Simulasi pengendali pintu irigasi menggunakan DTMF berbasis mikrokontroler AT89S51 mempunyai tujuan sebagai berikut :

1. Mendesain sebuah simulasi pengendali pintu irigasi dengan menggunakan DTMF sebagai input masukkan sistem.
2. Mengetahui prinsip kerja dari sistem simulasi pengendali pintu irigasi menggunakan DTMF berbasis mikrokontroler AT89S51.

1.6. Manfaat Penelitian

Manfaat yang dapat diambil dari dikembangkannya alat ini adalah :

1. Bagi Mahasiswa :

Hasil dari penelitian (tugas akhir) ini diharapkan dapat lebih mendorong mahasiswa untuk lebih mengembangkan dan menghasilkan karya teknologi yang lebih inovatif.

2. Bagi Masyarakat :

- 1). Membantu masyarakat khususnya pada lingkungan pertanian dalam pengendalian irigasi pertanian.
- 2). Membantu masyarakat lingkungan pertanian untuk mendapatkan sistem pengendalian irigasi yang aman, mudah, dan efisien.

3. Bagi Universitas Islam Indonesia :

Hasil penelitian (tugas akhir) ini merupakan wujud pengabdian pada masyarakat kaitanya dengan tridharma perguruan tinggi dan penerapan ilmu pengetahuan dan pengembangan teknologi yang bermanfaat bagi masyarakat.

1.7. 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, identifikasi masalah, batasan masalah, perumusan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II : LANDASAN TEORI

Merupakan penjelasan secara terperinci mengenai teori-teori yang digunakan sebagai landasan untuk pemecahan masalah. Memberikan garis besar metode yang digunakan oleh penyusun sebagai kerangka pemecahan masalah.

BAB III : PERANCANGAN SISTEM

Bagian ini menjelaskan metode-metode perancangan yang digunakan, cara mengimplementasikan rancangan yang telah dibuat serta batasan dan hambatan yang ditemui selama proses perancangan dan implementasi sistem.

BAB IV : ANALISA DAN PEMBAHASAN

Bab ini membahas hasil uji dan pengamatan dari sistem yang dibuat dibandingkan dengan dasar teori sistem.

BAB V : PENUTUP

Bab ini merupakan bab terakhir yang berisi kesimpulan dan saran-saran. Kesimpulan diambil dari proses perancangan, implementasi, dan analisis kinerja sistem elektronis. Saran-saran yang dikemukakan berdasar pada keterbatasan-keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan tugas akhir.

BAB II

LANDASAN TEORI

2.1. Mikrokontroler AT89S51

Perusahaan ATMEL mendapatkan lisensi dari Intel untuk mengembangkan mikrokontroler MCS-51. Salah satu tipe yang diperkenalkan oleh ATMEL adalah AT89S51. Mikrokontroler AT89S51 ini adalah mikrokontroler yang sesuai dengan bahasa pemrograman MCS-51. Mikrokontroler AT89S51 menggunakan *Flash Programmable Erasable Read Only Memory* (Flash PEROM) yang mempunyai banyak keunggulan dalam hal kepraktisannya sehingga penghapusan data dapat dilakukan secara cepat dan serentak (tidak *byte* demi *byte* seperti pada EPROM) dengan menggunakan energi listrik.

2.1.1. Perangkat keras (*Hardware*)

a. Deskripsi Umum

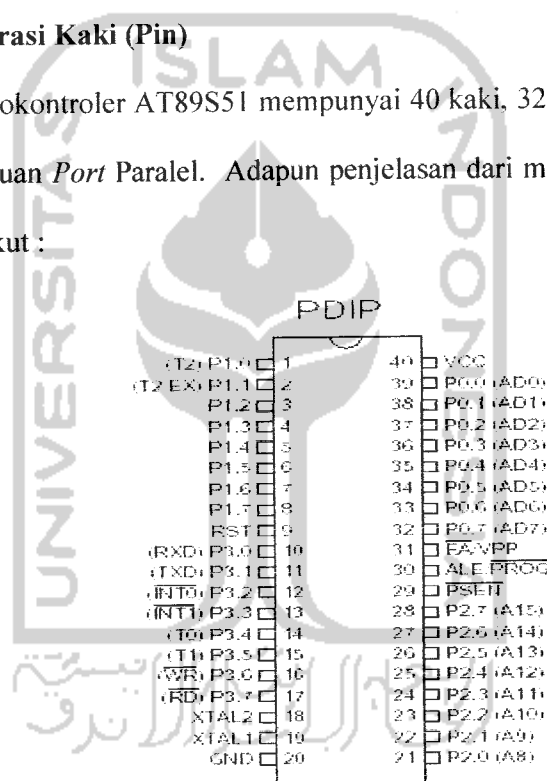
Mikrokontroler AT89S51 merupakan salah satu mikrokontroler buatan *Atmel Corporation* yang termasuk dalam keluarga MCS-51. AT89S51 memiliki keistimewaan sebagai berikut :

- 1) Kompatibel dengan produk MCS-51.
- 2) Mempunyai 4K byte *In System Programmable* (ISP), Flash Memory yang dapat diprogram ulang 1000 kali siklus.
- 3) Mempunyai tegangan kerja 4-5 volt.
- 4) Beroperasi secara penuh pada frekuensi 0 sampai 33 MHz.
- 5) Memiliki tiga tingkat penguncian program memori.

- 6) Memiliki 128 x 8 bit RAM internal.
- 7) Memiliki 32 jalur I/O yang dapat diprogram.
- 8) Memiliki dua buah *timer/counter* 16 bit.
- 9) Memiliki enam buah sumber interupsi.
- 10) Memiliki kanal serial yang dapat diprogram.
- 11) Memiliki mode *Low Power Idle* dan *Power Down*.

b. Konfigurasi Kaki (Pin)

Mikrokontroler AT89S51 mempunyai 40 kaki, 32 diantaranya adalah kaki untuk keperluan *Port Paralel*. Adapun penjelasan dari masing-masing pin adalah sebagai berikut :



Gambar 2.1. Susunan kaki mikrokontroler AT89S51

1). Pin 1 sampai 8 (*Port 1*)

Port 1 merupakan *port I/O* 8 bit dua arah (*bidirectional*) dengan *pull-up* internal. Keluaran *port 0* dapat menangani delapan input TTL. *Port 1* juga

mempunyai fungsi khusus menerima alamat bagian rendah (*low byte*) selama pemrograman dan verifikasi *flash*. Dengan spesifikasi sebagai berikut :

Tabel 2.1. Fungsi-fungsi khusus kaki-kaki *Port 1* AT89S51

Kaki <i>Port</i>	Fungsi Alternatif
P1.5	MOSI (digunakan untuk <i>In-System Programming</i>)
P1.6	MISO (digunakan untuk <i>In-System Programming</i>)
P1.7	SCK (digunakan untuk <i>In-System Programming</i>)

2). Pin 9 (RST)

Pin RST merupakan masukan reset (aktif *high*). CPU memberi logika tinggi pada pin reset selama dua siklus mesin pewaktu osilator aktif akan menyebabkan reset peralatan.

3). Pin 10 sampai 17 (*port 3*)

Port 3 merupakan *port* I/O 8 bit dua arah (*bidirectional*) dengan *pull-up* internal. Keluaran *port 0* dapat menangani 4 input TTL. *Port 3* juga memiliki fungsi khusus, yaitu:

Tabel 2.2. Fungsi khusus *Port 3*

Pin <i>Port</i>	Fungsi
P3.0	RXD (<i>port</i> masukan serial)
P3.1	TXD (<i>port</i> keluaran serial)
P3.2	INT0 (interupsi 0 eksternal)
P3.3	INT1 (interupsi 1 eksternal)
P3.4	T0 (<i>timer 0</i> eksternal)
P3.5	T1 (<i>timer 1</i> eksternal)
P3.6	WR (<i>write strobe</i> memori data eksternal)
P3.7	RD (<i>read strobe</i> memori data eksternal)

4). Pin 18 (XTAL 2)

Pin XTAL2 digunakan sebagai keluaran ke rangkaian osilator internal.

Pin ini dipakai bila menggunakan osilator kristal.

5). Pin 19 (XTAL 1)

Pin XTAL1 digunakan sebagai pin masukan ke rangkaian osilator internal. Sebuah osilator kristal atau sumber osilator luar dapat digunakan.

6). Pin 20 (GND)

Dihubungkan dengan *ground supply*.

7). Pin 21 sampai 28 (*port 2*)

Port 2 merupakan *port I/O* 8-bit yang bersifat bidireksional yang dilengkapi dengan *pull-up internal*. *Port* ini juga dapat dikonfigurasi sebagai jalur (*bus*) alamat/data byte tinggi (*high byte*) selama pengaksesan data dan program eksternal. Masing-masing kaki diberi nama P2.0, P2.1, P2.2, P2.3, P2.4, P2.5, P2.6, P2.7.

8). Pin 29 (PSEN)

PSEN (*Program Store Enable*) merupakan sinyal pengontrol yang mengizinkan untuk mengeksekusi memori program eksternal. PSEN diaktifkan dua kali masing-masing siklus mesin

9). Pin 30 (ALE/PROG)

Pulsa ALE (*Address Latch Enable*) untuk menahan alamat bit rendah selama mengeksekusi memori program eksternal.

10). Pin 31 (EA)

Bila pin ini diberi logika tinggi maka mikrokontroler akan melaksanakan instruksi dari memori program internal. Untuk mengeksekusi memori program eksternal EA (*External Access Enable*) harus diberi logika rendah atau dihubungkan ke ground.

11). Pin 32 sampai 39 (*port 0*)

Port 0 merupakan *port I/O* 8 bit dua arah (*bidirectional*) open drain. Keluaran *port 0* dapat menangani 4 input TTL. *Port 0* juga dapat dikonfigurasi sebagai bus alamat/data bagian rendah (*low byte*) selama proses pengaksesan memori data dan program eksternal. Jika digunakan dalam mode ini *Port 0* memiliki *pull-up* internal.

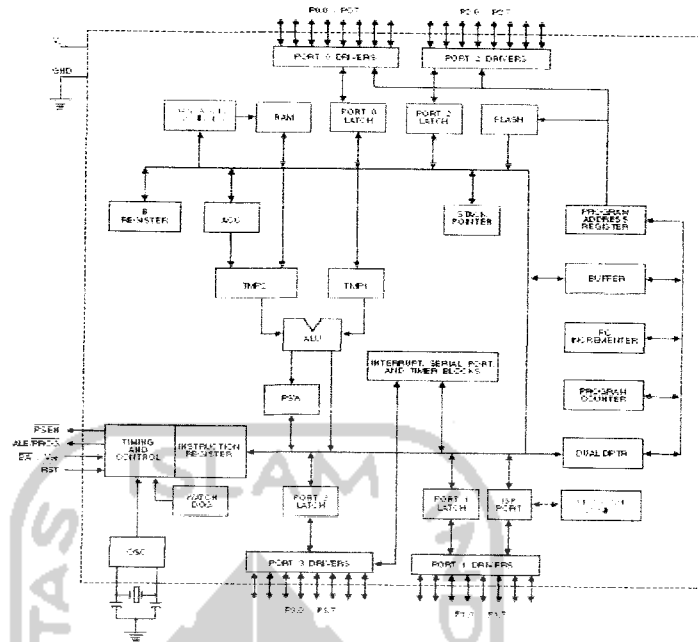
12). Pin 40 (VCC)

Dihubungkan ke VCC 5 volt sebagai sumber tegangan bagi mikrokontroler.

c. Blok Diagram

Pada blok diagram AT89S51 terdapat 4 *port* untuk *input/output* data serta tersedia pula Akumulator, Register, RAM, *Stack Pointer*, *Arithmetic Logic Unit* (ALU), Pengunci (*latch*), dan rangkaian osilasi yang membuat AT89S51 dapat beroperasi hanya dengan satu keping IC.

Mikrokontroler AT89S51 mempunyai blok diagram sebagai berikut :



Gambar 2.2. Blok diagram AT89S51

d. Organisasi Memori

Semua mikrokontroler keluarga MCS-51 memiliki pembagian ruang alamat untuk memori program dan memori data. Pemisahan memori program dengan memori data tersebut membolehkan memori data untuk diakses oleh alamat 8-bit.

Mikrokontroler AT89S51 memiliki memori program yang terpisah dengan data. Kapasitas memori program internal sebanyak 4 Kbyte yaitu dari alamat 0000H – 0FFFH. Namun memori program AT89S51 ini dapat ditingkatkan sampai 64 Kbyte dengan menggunakan memori program eksternal. Pembatasan alamat sampai 64 Kbyte ini disebabkan karena mikrokontroler AT89S51 hanya memiliki 16 jalur alamat ($2^{16} = 65536$ byte), keterangan 1 byte = $2^4 = 16$.

Mikrokontroler AT89S51 juga memiliki memori data internal yang disebut sebagai RAM internal. Ruang memori data dibagi menjadi tiga blok, yaitu bagian rendah 128-byte (*lower 128-byte*), bagian tinggi 128-byte (*upper 128-byte*) dan SFR (*Special Function Register*).

2.1.2. Perangkat Lunak (*Software*)

Mikrokontroler AT89S51 memiliki 110 macam instruksi yang dikelompokkan dalam 5 bagian yaitu instruksi transfer data, instruksi aritmatika, instruksi logika, manipulasi variabel Boolean, dan instruksi percabangan.

a. Instruksi Transfer Data

Instruksi ini memindahkan antara register dengan register, memori-memori, antarmuka register, dan antarmuka memori. Instruksi transfer data meliputi :

- | | |
|--------------------------|--------------------------|
| - Mov Rn, #data | - Mov DPTR, #data 16 bit |
| - Mov Rn, A | - Movc A, @A + DPTR |
| - Mov Rn, alamat data | - Movc A, @A + PC |
| - Mov A, #data | - Movx @DPTR, A |
| - Mov A, @Ri | - Movc A, @DPTR |
| - Mov A, Rn | - Movx A, @Ri |
| - Mov alamat data, A | - Movc @Ri, A |
| - Mov alamat data, Rn | - PUSH alamat data |
| - Mov alamat data, @Ri | - POP alamat data |
| - Mov alamat data, #data | - XCH A, Rn |
| - Mov alamat 1, alamat 2 | - XCH A, alamat data |
| - Mov @Ri, A | - XCH A, @Ri |
| - Mov @Ri, alamat data | - XCHD A, @Ri |
| - Mov @Ri, #data | |

b. Instruksi Aritmatika

Instruksi ini melakukan operasi aritmatika yang meliputi penjumlahan, pengurangan, penjumlahan satu, pengurangan satu, perkalian, dan pembagian.

- | | |
|-----------------------|-----------------------|
| - ADD A, #data | - INC A |
| - ADD A, @Ri | - INC @Ri |
| - ADD A, Rn | - INC DPTR |
| - ADD A, alamat data | - INC Rn |
| - ADDC A, #data | - INC alamat data |
| - ADDC A, @Ri | - MUL AB |
| - ADDC A, Rn | - DIV AB |
| - ADDC A, alamat data | - DA A |
| - DEC A | - SUBB A, #data |
| - DEC @Ri | - SUBB A, Rn |
| - DEC Rn | - SUBB A, alamat data |
| - DEC alamat data | - SUBB A, @Ri |

c. Instruksi Logika

Instruksi ini melakukan operasi logika seperti AND, OR, XOR, perbandingan, pergeseran, serta komplemen. Yang termasuk dalam kelompok ini adalah :

- | | |
|--------------------------|--------------------------|
| - ANL A, #data | - XRL A, #data |
| - ANL A, #Rn | - XRL A, #Rn |
| - ANL A, @Ri | - XRL A, @Ri |
| - ANL A, Rn | - XRL A, Rn |
| - ANL A, alamat data | - XRL alamat data, A |
| - ANL alamat data, A | - XRL alamat data, #data |
| - ORL A, #data | - CLR A |
| - ORL A, #Rn | - CPL A |
| - ORL A, @Ri | - RL A |
| - ORL alamat data, A | - RR A |
| - ORL alamat data, #data | - RRC A |
| - ORL A, Rn | - SWAP A |

d. Manipulasi Variabel Boolean

Instruksi ini melakukan manipulasi variabel Boolean. Yang termasuk dalam instruksi ini adalah :

- | | |
|---------------|--------------------|
| - CLR C | - ORL A, /bit |
| - CLR bit | - MOV C, bit |
| - SETB C | - MOV bit, C |
| - SETB bit | - JC relative |
| - CPL C | - JNC relative |
| - CPL bit | - JB bit, relative |
| - ANL A, bit | - JNB bit, relatif |
| - ANL A, /bit | - JBC bit, relatif |
| - ORL A, bit | |

e. Percabangan

Instruksi ini mengakibatkan suatu program melompat ke suatu alamat tertentu. Percabangan dibedakan menjadi 2 yaitu percabangan tanpa syarat dan percabangan bersyarat.

1. Percabangan tanpa syarat

- | | |
|-----------------------|----------------------|
| - ACALL alamat 11 bit | - LJMP alamat 16 bit |
| - LCALL alamat 16 bit | - SJMP relative |
| - RET | - JMP @A+DPTR |
| - RETI | - JZ relative |
| - AJMP alamat 11 bit | - JNZ relatif |

2. Percabangan bersyarat

- | | |
|-------------------------------|------------------------------|
| - CJNE A,alamat data,relative | - DJNZ Rn,relative |
| - CJNE A, #data, relative | - DJNZ alamat data, relative |
| - CJNE Rn, #data, relative | - NOP |
| - CJNE @Ri, #data, relative | |

2.2. Sinyal DTMF

Salah satu hasil teknologi yang saat ini banyak dipakai orang adalah telepon. Hasil teknologi telepon telah merambah hampir pada seluruh lapisan masyarakat. Oleh karena hal tersebut, maka dalam perkembangannya, telepon dipakai sebagai sarana pengendali suatu sistem.

Salah satu sinyal telepon yang bisa digunakan untuk mengendalikan suatu sistem adalah sinyal DTMF. Sinyal DTMF (*Dual Tone Multiple Frequency*), yang terjadi saat pengguna telepon memasukkan nomor telepon tujuan. Sinyal ini berupa gabungan dua buah frekuensi dengan kombinasi sesuai tabel berikut:

Tabel 2.3. Nilai frekuensi DTMF

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

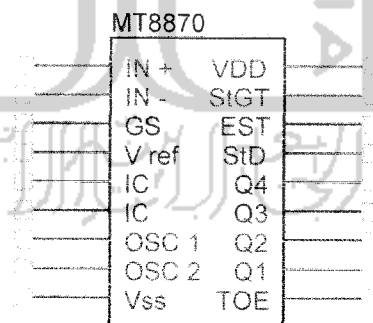
Tiap tombol pada pesawat telepon merupakan gabungan dari dua buah frekuensi yang berbeda, seperti ditunjukkan pada Tabel 2.3. Sebagai contoh, nada tombol '1' merupakan hasil penjumlahan dari frekuensi 697 Hz dan 1209 Hz.

2.3. MT 8870 (DTMF Receiver)

Pesawat telepon elektronik bekerja secara elektronik yang dilengkapi dengan rangkaian *dialer* berupa pulsa-pulsa dekodik atau *tone* DTMF. DTMF dihasilkan dari konfigurasi yang berhubungan secara matrik. Tombol- tombol ini merupakan kombinasi 8 frekuensi. Dari kombinasi delapan frekuensi ini dapat dipakai untuk mengkodekan 16 tanda untuk tombol pesawat telepon, tapi biasanya tombol 'A', 'B', 'C', dan 'D' tidak dipakai. Pada pesawat telepon, 8 frekuensi

yang ada dibangkitkan oleh 8 rangkaian osilator. Tiap frekuensi dibangkitkan oleh satu rangkaian osilator. Selain itu terdapat rangkaian pencampur frekuensi untuk mengirimkan 2 frekuensi terpilih apabila tone DTMF ingin dikirimkan.

Cara menerima sinyal DTMF ini ada berbagai cara, melalui jalur kabel data, jalur *head phone / hands free*, dan melalui *speaker hand phone*. IC MT 8870 merupakan penerima DTMF, menerima sinyal dari saluran telepon kalau ternyata sinyal yang diterima merupakan kombinasi nada yang sesuai dengan ketentuan DTMF. IC ini merubah data analog yang berupa nada *tone* menjadi data digital 4 bit. Penerima DTMF ini dibentuk dari 8 buah *filter* dan *decoder*. Dari rangkaian *filter* dapat dipisahkan group frekuensi rendah dan group frekuensi tinggi dari *tone* DTMF yang dipilih. Bagian decoder menggunakan teknik perhitungan secara digital untuk mendeteksi dan mengkodekan 16 pasangan *tone* DTMF menjadi kode 4 bit.



Gambar 2.3. Pin MT 8870

Tabel 2.4. Penjelasan tiap pin pada IC MT 8870

▪ Kaki 1 (IN+)	:	Merupakan non inverting operasional <i>amplifier (input)</i> .
▪ Kaki 2 (IN-)	:	Merupakan inverting <i>amplifier</i> .
▪ Kaki 3 (GS)	:	Berfungsi mengubah penguatan dari penguat diferensial dengan memasang resistor umpan balik secara <i>external</i> .
▪ Kaki 4 (Vref)	:	Tegangan referensi (<i>output</i>) dalam keadaan normal sebesar $VDD/2$, digunakan untuk memberi bias input.
▪ Kaki 5 (INH)	:	Dalam keadaan logika tinggi untuk mencegah pendeteksian bunyi yang diwakili karakter A,B,C,dan D.
▪ Kaki 6 (PWDN)	:	<i>Power Down (input)</i> dalam kondisi logika tinggi dan mencegah kerja osilator.
▪ Kaki 7 (osc 1)	:	Clock (<i>input</i>).
▪ Kaki 8 (osc 2)	:	Clock (<i>output</i>) kristal 3,579545 Mhz dihubungkan pada osc 1 dan osc 2.
▪ Kaki 9 (Vss)	:	<i>Ground (input)</i> .
▪ Kaki 10 (TOE)	:	<i>Three State Output Enable</i> , sebagai kaki <i>enable</i> Q1-Q4 jika diberi logika tinggi.
▪ Kaki 11-14 (Q1-Q4)	:	Saat TOE <i>enable</i> , Q1-Q4 akan menghasilkan kode

		sesuai dengan <i>tone</i> terakhir yang diterima.
▪ Kaki 15 (STD)	:	<i>Delayed Steering</i> , berkondisi logika tinggi jika telah/sedang mencatat pasangan <i>tone</i> yang diterima dan <i>output</i> akan berlogika rendah ketika tegangan pada St/Gt turun dibawah Vtst.
▪ Kaki 16 (Est)	:	Berlogika tinggi ketika digital algoritma mendeteksi pasangan <i>tone</i> yang <i>valid</i> (sinyal kondisi). EST berlogika rendah ketika sinyal kondisi hilang.
▪ Kaki 17 (St/Gt)	:	<i>Steering Input / Board Tone (output)</i> . Jika terdeteksi ada tegangan yang lebih besar dari Vtst terdeteksi pada St menyebabkan peralatan mencatat pasangan <i>tone</i> yang terdeteksi dan memperbaharui <i>output</i> jika tegangan lebih kecil dari Vtst terjadi penyegaran (<i>fresh</i>) untuk menerima pasangan <i>tone</i> baru <i>output</i> GT digunakan untuk mereset pengendali <i>time</i> konstan eksternal.
▪ Kaki 18 (Vdd)	:	Positif <i>power supply</i> (+5 V).

Tabel 2.5. Tabel fungsi frekuensi DTMF

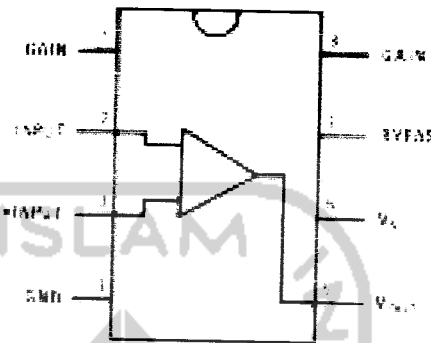
F _{LOW}	F _{HIGH}	No	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	+	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0

Pada tabel fungsi diatas, frekuensi-frekuensi sebuah DTMF dibagi menjadi dua kelompok, yaitu frekuensi rendah dan frekuensi tinggi. Setiap nada yang dikeluarkan oleh sebuah *tone* menghasilkan nilai-nilai biner tertentu yang akan diproses oleh mikrokontroler. Misalkan nada “2”, mempunyai nilai heksa dan data biner “2”, maka pada port kaki MT 8870 hanya Q2 yang mempunyai nilai logika tinggi (+5V), sedangkan yang lainnya berlogika rendah.

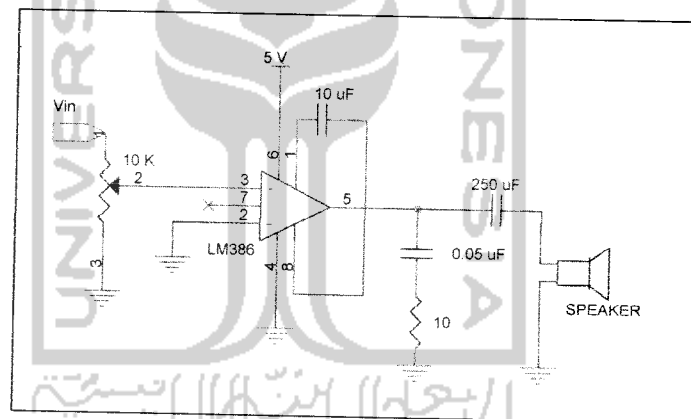
2.4. Penguat Sinyal (LM 386)

LM 386 merupakan *power amplifier*, yang memberikan penguatan dengan tegangan kerja yang kecil. Pada perancangan sistem ini digunakan rangkain *power amplifier* dengan menggunakan LM 386 yang berfungsi untuk mendapatkan penguatan frekuensi *tone* (sinyal DTMF) yang diterima HP

penerima sebesar 200 kali penguatan, sehingga sinyal DTMF tersebut dapat lebih mudah dibaca dan diterjemahkan oleh rangkaian penerima DTMF (IC MT 8870) yang selanjutnya data akan dikirim ke sistem minimum mikrokontroler AT89S51 untuk diolah menjadi data set *password* simulasi pengendali pintu irigasi.



Gambar 2.4. IC LM 386

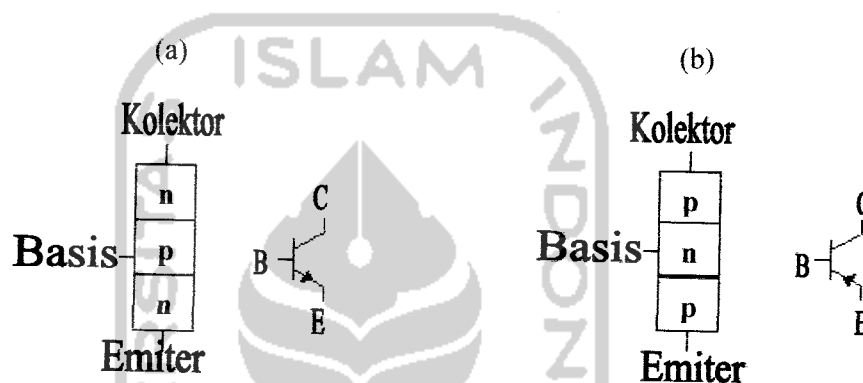


Gambar 2.5. Aplikasi LM 386 untuk penguat

2.5. Transistor sebagai saklar

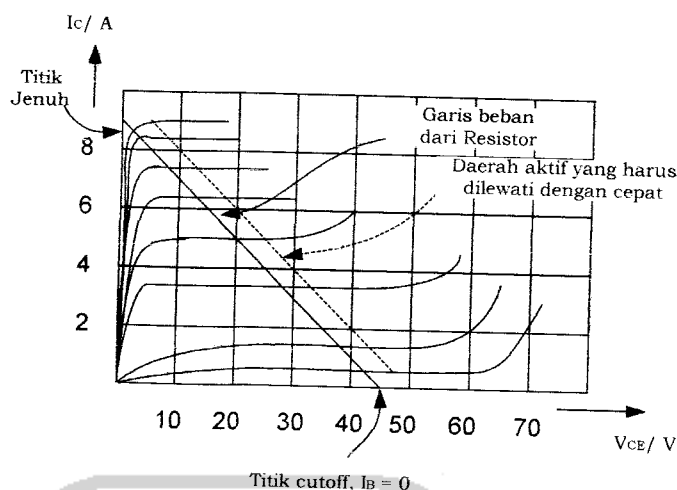
Transistor dalam tugas akhir ini digunakan sebagai saklar dan penguat arus. Pada umumnya transistor digunakan sebagai penguat, baik penguat arus, penguat tegangan, maupun penguat daya. Transistor memiliki 3 elektroda, yaitu : emitor, basis, dan kolektor. Pemasangan dari kaki-kaki elektroda ini tidak boleh

salah, jika terdapat kesalahan dalam pemasangannya, maka rangkaian tidak akan bekerja, bahkan akan merusak komponen itu sendiri. Transistor terdiri dari semikonduktor kristal tunggal (*germanium* atau *silicon*) dimana lapisan tipis jenis P diselipkan antara dua lapisan N yang disebut transistor jenis NPN. Transistor dapat pula terdiri dari lapisan N yang diselipkan diantara dua lapisan P yang disebut transistor jenis PNP. Simbol transistor dapat dilihat pada gambar dibawah ini :



Gambar 2.6. (a) Transistor NPN (b) Transistor PNP

Prinsip kerja dari transistor adalah arus bias basis yang kecil mengatur besar arus kolektor – emitor. Pemberian arus yang tepat akan menyebabkan transistor bekerja secara optimal. Dari hukum Kirchoff dapat diketahui bahwa arus yang masuk ke suatu titik akan sama jumlahnya dengan arus yang keluar, sehingga $I_E = I_C + I_B$. Karena arus I_B sangat kecil atau $I_B \ll I_C$, maka $I_E = I_C$. Hubungan antara arus basis (I_B), arus kolektor (I_C) dan tegangan kolektor- emitor adalah sebagai berikut :



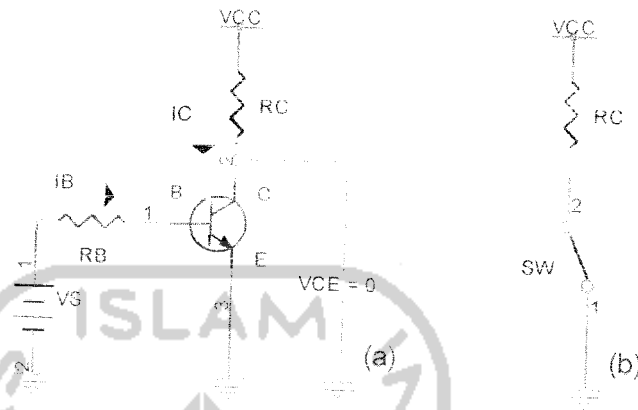
Gambar 2.7. Karakteristik beban DC Transistor
(Sumber : Richard Blocher : 2003)

Dari kurva diatas, dapat diketahui daerah kerja transistor, yaitu daerah saturasi, daerah aktif, dan daerah *cut-off*. Agar transistor dapat berfungsi sebagai saklar, maka transistor harus dioperasikan dalam dua keadaan, yaitu keadaan jenuh (saturasi) dan keadaan *cut-off*.

Pengaplikasian transistor sebagai saklar berarti transistor dioperasikan pada salah satu titik saturasi atau titik sumbat, tapi tidak di tempat-tempat sepanjang garis beban. Apabila transistor berada dalam keadaan saturasi, transistor seolah-olah merupakan sebuah saklar tertutup. Apabila transistor tersumbat (*cut off*), maka transistor ini berfungsi seperti sebuah saklar yang terbuka (Petruzella: 2001).

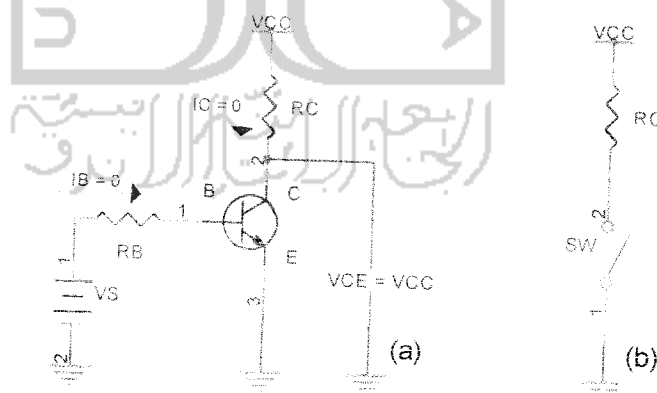
Penggunaan transistor sebagai saklar adalah dengan memanfaatkan daerah jenuh (saturasi) dan daerah mati (*cut off*) transistor. Ketika transistor pada daerah saturasi maka arus mengalir tanpa halangan dari kolektor menuju emitor dan $V_{CE} \approx 0$, sedangkan arus *colector* jenuh I_c (saturasi) sama dengan V_{cc}/R_c . Kondisi

ini menyerupai sebagai saklar pada kondisi tertutup (ON). Untuk membuat kondisi transistor konduksi diperlukan arus yang sangat besar atau minimal $I_B > I_C/\beta$.



Gambar 2.8. (a) Transistor dalam keadaan konduktif. (b) Ekuivalen saklar tertutup

Pada kondisi transistor non konduktif (*cut off*), berlaku ketentuan $V_{CE} \approx V_{CC}$ dan $I_C \approx 0$. Pada kondisi demikian menyerupai saklar pada kondisi terbuka (*off*). Kondisi non konduktif didapat dengan cara tidak memberikan bias pada basis, $I_B = 0$ atau pada basis diberi tegangan mundur terhadap emitor.



Gambar 2.9.(a) Transistor dalam keadaan non konduktif. (b) Ekuivalen saklar terbuka

Perhitungan kondisi saklar secara teori adalah sebagai berikut.

1). Kondisi *cut off*

$$V_{CE} = V_{CC} - I_C \cdot R_C \quad (2.1)$$

Karena kondisi *cut off*, $I_C = 0$ (kondisi ideal) maka:

$$V_{CE} = V_{CC} - 0 \cdot R_C$$

$$V_{CE} = V_{CC} \quad (2.2)$$

Besar arus basis I_B adalah

$$I_B = I_C / \beta$$

$$I_B = 0 / \beta$$

$$I_B = 0 \quad (2.3)$$

2). Kondisi saturasi atau jenuh

$$V_{CE} = V_{CC} - I_C \cdot R_C$$

Karena kondisi saturasi $V_{CE} = 0$ (kondisi ideal) atau $V_{CE} = 0,3$ Volt.

$$\text{Maka } I_C = V_{CC} / R_C \quad (2.4)$$

Besar tahanan basis R_B untuk mendapatkan arus basis I_B pada kondisi benar-benar saturasi adalah :

$$R_B = V_{BB} - V_{BE} / I_B \text{ sat} \quad (2.5)$$

Besar arus basis I_B saturasi adalah :

$$\beta \cdot I_B > I_C \text{ atau } I_B \text{ sat} > I_C / \beta \quad (2.6)$$

Bila besar tegangan masukan basis melalui rangkaian pembagi tegangan maka

besar tegangan basis V_B dan tahanan basis R_B adalah :

$$R_p = R_1 // R_2$$

$$V_B = (R_1 V_1) / (R_1 + R_2) \quad (2.7)$$

Keterangan:

V_{CC} : Tegangan Sumber/Catu Daya

V_{CE} : Tegangan Colector Emitor

V_{BE} : Tegangan Basis Emitor

V_B : Tegangan Basis

I_C : Arus Colector

I_B : Arus Basis

I_E : Arus Emitor

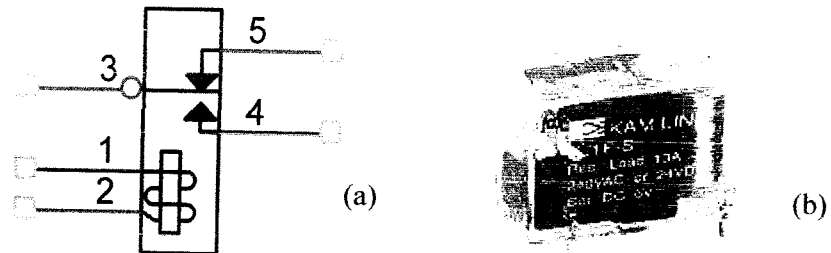
β : hfe (Faktor Penguatan DC)

2.6. Relay

Relay adalah sebuah alat elektromagnetik yang dapat mengubah kontak - kontak saklar sewaktu alat ini menerima sinyal listrik. Alat ini tersusun atas sebuah kumparan kawat beserta sebuah inti besi lunak. Fungsi utama *relay* adalah mengontrol arus yang lebih besar dalam rangkaian dengan arus kecil yang melewati kumparan.

Pada prinsipnya *relay* terdiri dari kumparan kawat penghantar, jika kumparan itu diberi arus listrik, maka inti (*core*) kumparan yang terbuat dari besi lunak akan menjadi magnet. Adanya magnet buatan ini akan menarik besi lunak pengungkit, sehingga mendekati magnet buatan, akibatnya kontak hubung akan berpindah dari satu posisi kontak keposisi yang lain. Dengan berpindahnya posisi kontak ini maka jadilah sebuah saklar. Banyaknya kontak hubung ini bervariasi, sehingga menghasilkan saklar SPST (*Single Pole Singel Throw*) , SPDT (*Single Pole Double Throw*), DPST (*Double Pole Singel Throw*), DPDT (*Double Pole*

Double Throw), tergantung banyaknya kontak yang bisa digerakkan oleh pengungkit.



Gambar 2.10. (a) Simbol *Relay* SPDT (b). *Relay* SPDT

Sewaktu arus kontrol melewati kumparan, inti besi lunak akan dimagnetisasi, armatur ditarik oleh inti yang dimagnetisasi. Gerakan armatur ini akan menutup kontak 3 dengan kontak 4 dan akan membuka kontak 3 dengan kontak 5. Dengan kata lain gerakan armatur tadi telah mengubah posisi kontak 3. Kontak-kontak ini dapat digunakan mengontrol arus yang lebih besar dalam rangkaian. Pada Gambar 2.10 dipakai untuk menggambarkan *relay* dalam diagram rangkaian. Simbol ini terdiri dari sebuah kumparan dan 2 set kontak, satu biasanya terbuka (*normally open* atau NO), lainnya biasanya tertutup (*normally close* atau NC). Sewaktu arus melewati kumparan, kontak NO tertutup, sebaliknya kontak NC terbuka.

Pada badan relay dituliskan kemampuan arus dan tegangan maksimal yang dapat dibebankan sehingga dapat digunakan sesuai dengan kebutuhan. Berikut merupakan sifat-sifat dari *relay*, antara lain:

- a. Untuk mengoperasikan *relay*, kuat arus yang dibutuhkan sudah ditentukan pabrik. Pada *relay* yang hambatannya kecil membutuhkan arus yang besar, sedangkan *relay* yang hambatannya besar membutuhkan arus yang kecil.

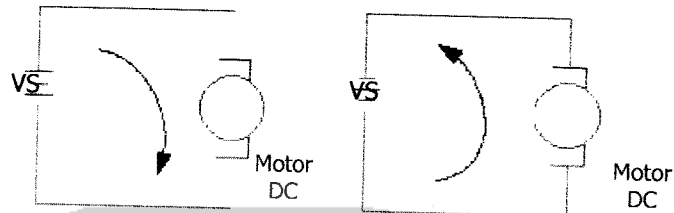
- b. Besarnya hambatan kumparan ditentukan oleh tebal kawat yang dipakai.
- c. Tegangan yang diperlukan *relay* adalah hasil kali antara kuat arus dan hambatan.
- d. Daya yang diperlukan *relay* adalah hasil kali antara tegangan dan kuat arus.

Relay dalam tugas akhir ini digunakan untuk mengaktifkan motor berputar ke kanan atau kiri. *Relay* merupakan piranti elektronik yang dioperasikan berdasarkan variasi masukan, untuk mengontrol piranti-piranti lain yang dihubungkan dengan keluaran. Asas kerja piranti ini adalah menimbulkan medan magnet yang dikendalikan oleh sinyal listrik. *Relay* berfungsi untuk memutuskan dan mengalirkan arus listrik yang dikontrol dengan memberikan tegangan dan arus pada koilnya. Pada tugas akhir ini digunakan *relay type* SJ4089 6 Vdc. *Relay* ini memiliki nilai hambatan lilitan 220Ω dan mampu dialiri arus maksimal 100 mA. Saklar internalnya mampu bekerja pada tegangan 220 Vac atau 120 Vdc dengan arus beban maksimal 3 A. Karena tegangan maupun arus keluaran mikrokontroler tidak memenuhi untuk menghidupkan *relay*, maka dibutuhkan sebuah transistor sebagai gainnya. Untuk melindungi *relay* dan transistor dari tegangan kejut, maka dipasang dioda yang dioperasikan secara langsung pada kaki lilitan *relay*.

2.7. Motor DC

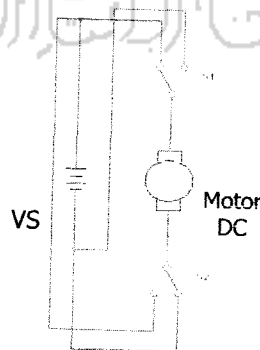
Untuk menutup dan membuka simulasi pintu irigasi, diperlukan motor untuk melakukannya. Dalam tugas akhir ini digunakan motor DC dilengkapi *gearbox* . Untuk memutar motor ke kanan atau ke kiri dilakukan dengan

membalik polaritasnya. Motor ini bekerja pada tegangan 24 V dan membutuhkan arus 250 mA. Keunggulan dari motor ini adalah putarannya kuat, sehingga dimungkinkan dapat mengangkat beban yang lebih. Untuk memperlambat putaran motor cukup dilakukan dengan mengurangi tegangan input.



Gambar 2.11. Dasar pengaturan arah putaran motor

Dari gambar diatas, dapat dilihat bahwa agar perubahan polaritas motor dapat dilakukan dengan mudah, maka digunakan dua buah saklar seperti terlihat pada Gambar 2.12 Kedua saklar harus berada pada posisi saling berlawanan. Apabila S1 (saklar 1) berada di posisi kiri (terhubung dengan positif) maka S2 (saklar 2) harus berada di posisi kanan (terhubung dengan negatif) dan demikian pula sebaliknya dengan perubahan yang serempak. Dalam perancangan sistem simulasi pintu irigasi ini, fungsi saklar S1 dan S2 akan diganti dengan transistor dan dikendalikan oleh mikrokontroler.



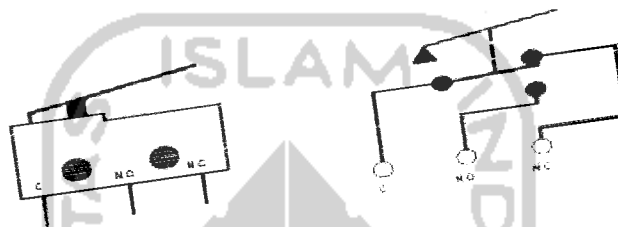
Gambar 2.12. Pengaturan arah motor dengan menggunakan saklar

2.8. Micro Switch

Micro Switch merupakan salah satu jenis saklar yang dioperasikan secara *manual*. Seperti jenis saklar yang lain, *micro switch* berfungsi untuk memutuskan atau menghubungkan suatu jalur aliran listrik. Terdapat dua macam *micro switch* yaitu NO (*Normally Open*) dan NC (*Normally Close*).

Adapun bentuk dan simbol dari *micro switch* seperti pada gambar dibawah

ini :



Gambar 2.13. Sensor *micro switch*

Saklar NO (*Normally open*) akan menghubungkan rangkaian ketika ditekan, dan kembali pada posisi terbuka ketika dilepas. Sebaliknya saklar NC (*Normally close*) akan membuka rangkaian apabila saklar ditekan, dan kembali pada posisi menutup ketika saklar dilepas. Dengan kriteria seperti *push button*, *micro switch* sangat tepat jika digunakan dalam aplikasi *keypad* seperti halnya terlihat pada aplikasi kalkulator, dan lain sebagainya.

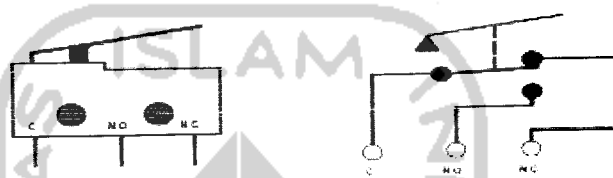
2.9. Software

Sistem mikrokontroler membutuhkan perangkat lunak (*software*) untuk bekerja. *Software* yang ada merupakan pusat pengolahan data dari alat yang

2.8. Micro Switch

Micro Switch merupakan salah satu jenis saklar yang dioperasikan secara *manual*. Seperti jenis saklar yang lain, *micro switch* berfungsi untuk memutuskan atau menghubungkan suatu jalur aliran listrik. Terdapat dua macam *micro switch* yaitu NO (*Normally Open*) dan NC (*Normally Close*).

Adapun bentuk dan simbol dari *micro switch* seperti pada gambar dibawah ini :



Gambar 2.13. Sensor *micro switch*

Saklar NO (*Normally open*) akan menghubungkan rangkaian ketika ditekan, dan kembali pada posisi terbuka ketika dilepas. Sebaliknya saklar NC (*Normally close*) akan membuka rangkaian apabila saklar ditekan, dan kembali pada posisi menutup ketika saklar dilepas. Dengan kriteria seperti *push button*, *micro switch* sangat tepat jika digunakan dalam aplikasi *keypad* seperti halnya terlihat pada aplikasi kalkulator, dan lain sebagainya.

2.9. Software

Sistem mikrokontroler membutuhkan perangkat lunak (*software*) untuk bekerja. *Software* yang ada merupakan pusat pengolahan data dari alat yang

dirancang. Untuk merancang program mikrokontroler, ada beberapa langkah yang diperlukan yaitu :

1. Menentukan algoritma program

Algoritma program merupakan urutan-urutan proses yang akan digunakan untuk mengoperasikan suatu sistem.

2. Diagram alir (*flow chart*)

Diagram alir merupakan simbol dari urutan proses yang sudah dibuat pada algoritma program. Tujuan dari pembuatan diagram alir ini adalah untuk mempermudah penyusunan program dan membantu pemogram dalam melakukan penelusuran kesalahan.

3. Penyusunan program

Program dibentuk dari sekumpulan intruksi-intruksi. Intruksi disimpan sebagai logika 1 dan logika 0 yang membentuk bahasa mesin dari mikrokontroler. Intruksi bahasa mesin didesain untuk memasukkan data kedalam mikrokontroler, memanipulasi data dan mengeluarkan data dari mikrokontroler.

BAB III PERANCANGAN SISTEM

3.1. Prinsip kerja

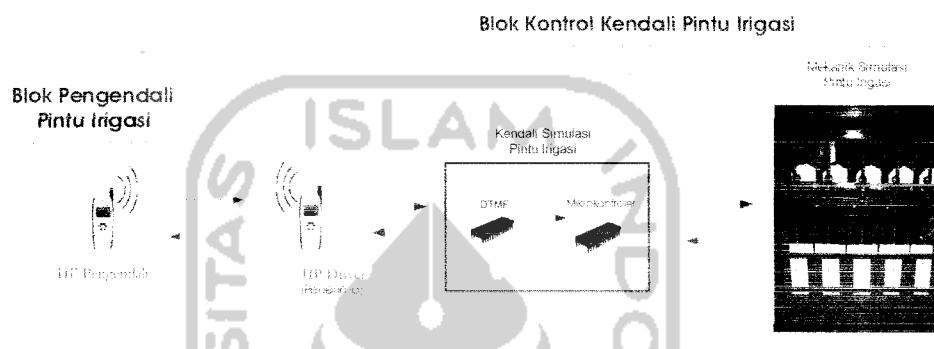
Simulasi pengendali pintu irigasi merupakan pengendali untuk mengendalikan buka dan tutup pintu air irigasi. Sistem simulasi bekerja berdasarkan *password*. Sistem terdiri dari dua bagian, blok pengendali dan blok yang dikendalikan (blok simulasi pintu irigasi). Blok pengendali menggunakan *mobile phone*, dimana memanfaatkan sinyal DTMF (*Dual Tone Multiple Frequency*) *mobile phone* sebagai data *password*. Blok yang dikendalikan terdiri dari bagian penerima data, pengkode data (penerjemah data), pengolah data, dan mekanik sistem pengendali pintu irigasi.

Pengendali akan mengirim *password* ke blok yang dikendalikan. Diblok ini, data *password* diterima, diterjemahkan atau dikodekan dan diolah untuk selanjutnya akan menggerakkan sistem mekanik. Dengan memanfaatkan *mobile phone* dan rangkaian pengolah data, maka sistem simulasi pintu irigasi dapat melakukan fungsi kerja, diantaranya:

1. Mengendalikan pintu irigasi jarak jauh, dapat dilakukan dimanapun dan kapanpun dengan memperhatikan keadaan jaringan telekomunikasi ditempat untuk pengendali.
2. Dapat membuka atau menutup pintu irigasi sesuai keinginan dengan variasi *password* yang dibuat.

3. Dapat mengetahui level ketinggian air, dengan mengirimkan sinyal dari blok kontrol kendali pintu irigasi ke pengendali.
4. Dapat menutup pintu secara otomatis jika kondisi air kosong.

Di bawah ini adalah gambar blok diagram sistem simulasi pengendali pintu irigasi :

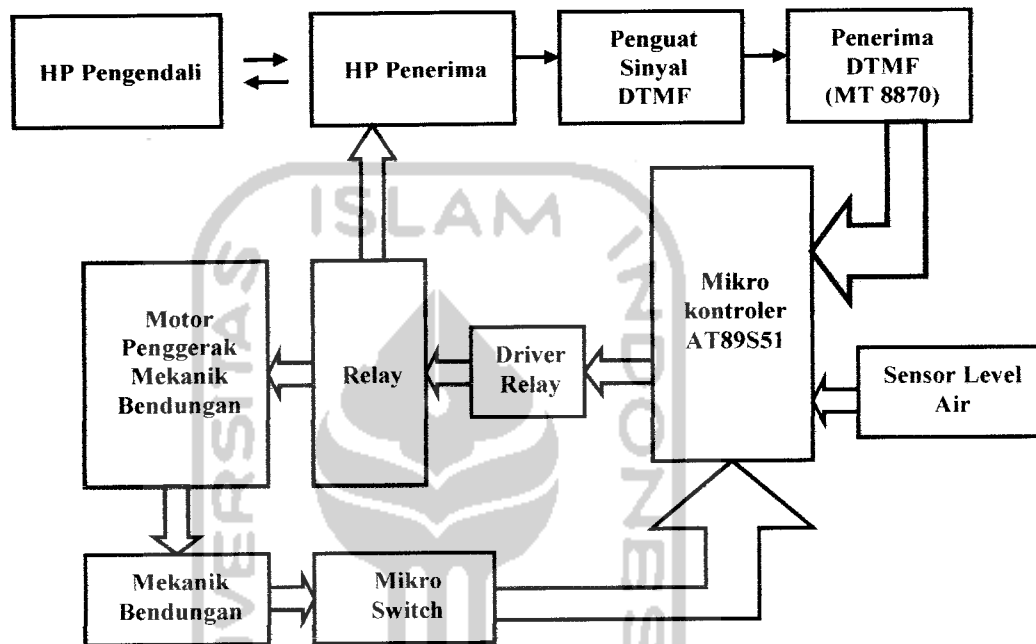


Gambar 3.1. Blok diagram simulasi pengendali pintu irigasi

Hp pengendali mengirimkan sinyal dan memberi input sinyal DTMF ke hp driver (penerima). Dari hp penerima, sinyal masuk ke kendali pintu irigasi. Pada blok kendali, data input dipakai untuk mengontrol blok mekanik sistem simulasi pintu irigasi yang dirancang. Blok mekanik memberi data ketinggian air ke blok kendali, data akan diolah dan dikirim ke hp penerima. Hp penerima akan mengirimkan sinyal ke hp pengendali yang memberitahukan tentang kondisi pada blok mekanik sistem pintu irigasi.

3.2. Perancangan Perangkat Keras

Perancangan perangkat keras (*hardware*) dari Simulasi Pengendali Pintu Irigasi Dengan Menggunakan DTMF Mobile Phone Berbasis Mikrokontroler AT89S51 ditunjukkan pada blok diagram berikut :



Gambar 3.2. Blok diagram perangkat keras (*hardware*)

Berdasarkan blok diagram diatas, setelah alat diaktifkan maka Mikrokontroler akan mendeteksi kondisi level air dengan menggunakan sensor level air. Jika level bawah mendeteksi tidak adanya air, maka mikrokontroler akan memerintahkan untuk menurunkan / menutup semua pintu air secara otomatis. Jika sensor level bawah mendeteksi adanya air, maka mikrokontroler akan mendeteksi keadaan sensor level atas. Apabila kondisi sensor atas mendeteksi adanya air, maka mikrokontroler akan memerintahkan *hand phone* penerima untuk memanggil

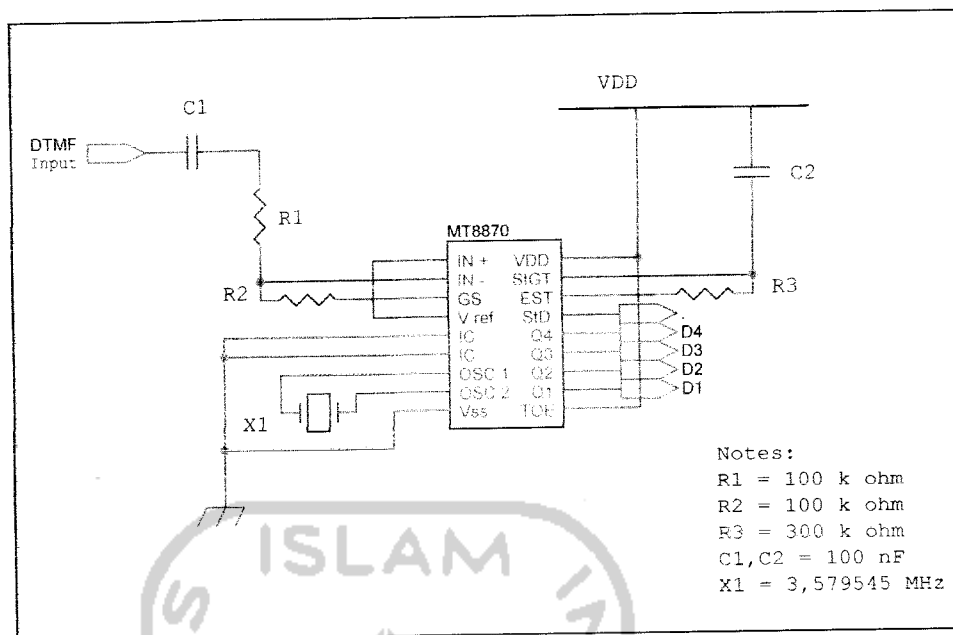
(*call*) *hand phone* pengendali untuk memberitahukan bahwa kondisi air pada bendungan irigasi dalam keadaan penuh / sedang meluap.

Hp pengendali adalah *hand phone* yang difungsikan untuk mengendalikan kondisi pintu pada bendungan irigasi. Jika ingin menaikkan (membuka) dan menurunkan (menutup) pintu irigasi, maka perlu memasukkan *password* yang telah ditentukan. Sinyal DTMF yang berasal dari *hand phone* pengendali kemudian diterima oleh *hand phone* penerima. Karena sinyal yang berasal dari *hand phone* masih terlalu lemah, maka dikuatkan dengan menggunakan rangkaian penguat sinyal (LM 386). *Output* dari penguat inilah yang digunakan oleh MT 8870 sebagai sinyal masukan untuk dikonversi ke dalam data digital yang bisa dibaca dan diterjemahkan oleh sistem mikrokontroler.

Data digital yang berasal dari hasil konversi MT 8870, digunakan untuk mengaktifkan motor yang digunakan untuk menggerakkan pintu irigasi sesuai dengan *password* yang telah ditentukan. Kemudian sensor *micro switch* berfungsi untuk membatasi gerak pintu irigasi pada posisi atas dan bawah / untuk menghentikan putaran motor pengendali pintu irigasi apabila telah menyentuh saklar (*micro switch*). Sehingga keadaan motor tidak selalu aktif yang dapat menyebabkan kerusakan pada motor itu sendiri dan mekanik pintu irigasi.

3.2.1. Penerima DTMF (*DTMF Receiver*)

Rangkaian ini berguna untuk menterjemahkan frekuensi *tone* yang berupa gabungan dua *tone* yang berbeda menjadi kode biner yang dapat dibaca dan diterjemahkan oleh sistem mikrokontroler. Rangkaian ini menggunakan kristal 3,579545 MHz.



Gambar 3.3. Rangkaian penerima DTMF

Antara Gs dan IN- terdapat resistor yang sama nilainya, hal ini bertujuan untuk mendapatkan nilai penguatan satu. Nilai penguatan selain ditentukan oleh kedua R tersebut, tetapi juga tergantung pada R3. Untuk membendung arus searah dan hanya arus bolak - balik yang masuk kerangkaian, maka dipasang C1. Untuk nilai R dan C bisa dilihat pada data sheet IC MT 8870.

Kaki Std merupakan *output* yang menandakan ada atau tidak data masuk ke rangkaian penerima DTMF. Jika ada sinyal yang masuk, maka kaki Std akan berlogika satu. Jika tidak ada sinyal yang masuk maka kaki Std berlogika "0" (0 V). Kaki 10 yaitu TOE merupakan pengatur data pada Q1-Q4. Agar data pada Q1-Q4 dapat diambil oleh mikrokontroler, maka TOE harus dihubungkan dengan VCC. Bila TOE dihubungkan dengan ground, MT 8870 tidak dapat memberi data pada kaki Q1- Q4.

3.2.2. Sistem Minimum Mikrokontroler AT89S51

Pengendali utama alat ini adalah mikrokontroler AT89S51. Rangkaian sistem minimum ini memerlukan sebuah mikrokontroler AT89S51, osilator kristal dan rangkaian reset serta *power supply*. Dengan rangkaian ini mikrokontroler sudah siap bekerja sesuai dengan program yang ada dalam *flash* memorinya. Rangkaian osilator pada sistem ini digunakan oleh mikrokontroler sebagai sinyal denyut (*clock*). Frekuensi *clock* inilah yang menentukan kecepatan eksekusi yang akan dijalankan. Frekuensi *clock* maksimum yang diperbolehkan adalah 33MHz. Dalam perancangan ini, periode siklus mesin yang diinginkan adalah 1 μ s. Dengan menggunakan 2 buah kapasitor 30 pF dan sebuah XTAL, maka besarnya frekuensi XTAL (f) yang diperlukan yaitu:

$$\begin{aligned} f &= 12 / t && (12 = \text{ketetapan frekuensi kristal mikrokontroler}) \\ &= 12 / 10^{-6} \text{ s} \\ &= 12 \text{ MHz} \end{aligned}$$

Sedangkan untuk membangkitkan sinyal reset, maka diperlukan sebuah kapasitor elektrolit 10 μ F/25 V dan resistor tetap dengan nilai tahanan sebesar 10 K Ω serta sebuah *push button switch*. Tegangan *reset* minimal (V_{RST}) untuk AT89S51 yaitu sebesar 0,7 V_{CC} (3,5V). Sehingga tegangan pada kapasitor (V_C) adalah $V_C = V_{CC} - V_{RST} = 1,5V$. Dengan tegangan $V_C = '1,5V'$, maka diperoleh besarnya waktu *reset*, yaitu :

$$\begin{aligned} V_C &= V_{CC} (1 - e^{-t/RC}) \\ 1,5 &= 5 (1 - e^{-t/10^4 \cdot 10^{-5}}) \\ 1,5 &= 5 (1 - e^{-10t}) \\ 1,5 &= 5 - 5e^{-10t} \\ 5e^{-10t} &= 3,5 \end{aligned}$$

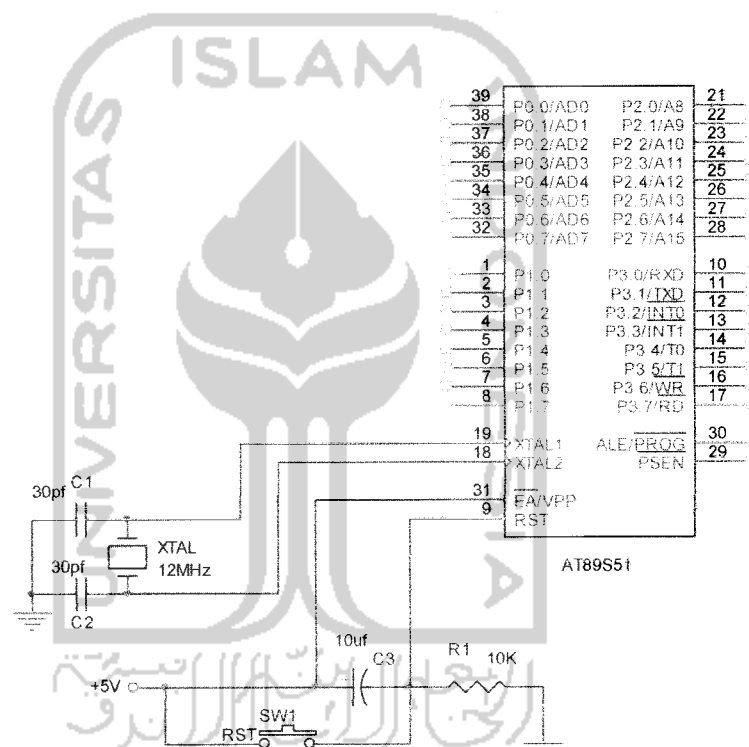
$$e^{-10t} = 0,7$$

$$-10t = \ln 0,7$$

$$-10t = -0,35667$$

$$t = 3,5667 \text{ ms}$$

Waktu *reset* minimal yang diperlukan oleh mikrokontroler AT89S51 yaitu sebesar 2 siklus *clock* ($2\mu\text{s}$ untuk kristal 12 MHz). Jadi rangkaian *reset* ini telah memenuhi syarat. Rangkaian sistem minimum mikrokontroler AT89S51 ditunjukkan pada gambar berikut :

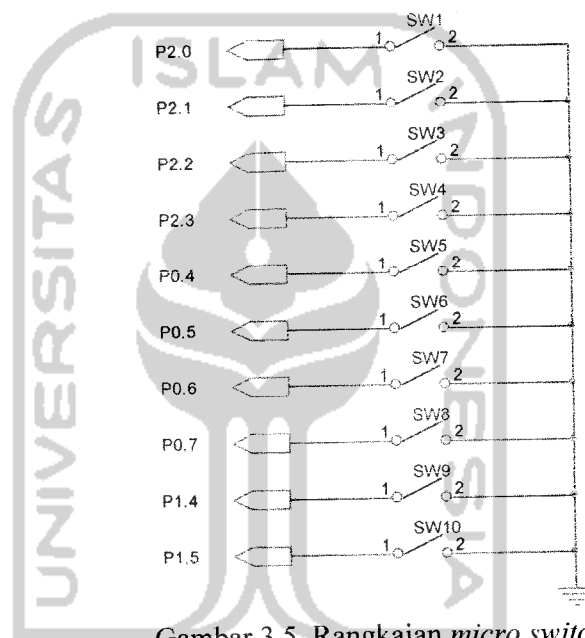


Gambar 3.4. Rangkaian sistem minimum mikrokontroller AT89S51

3.2.3. Rangkaian Sensor Pembatas dengan *Micro Switch*

Rangkaian sensor pembatas dalam alat ini digunakan untuk membatasi gerakan pintu pada bendungan irigasi yang digerakkan oleh motor atau bisa pula dikatakan untuk menghentikan putaran motor pengendali pintu irigasi apabila

telah menyentuh saklar (*micro switch*). Sensor pembatas ini dipasang dibagian atas dan bagian bawah mekanik sistem pintu irigasi. Sehingga untuk 5 pintu irigasi membutuhkan 10 *micro switch* (5 untuk batas atas dan 5 untuk batas bawah). Dari rangkaian sensor yang ditunjukkan pada gambar 3.5, maka kondisi idle *outputnya* adalah berlogika *high*, dan apabila mendeteksi adanya pintu, maka kondisi saklar akan tertutup sehingga terhubung dengan *ground* yang mengakibatkan *outputnya* akan berlogika *low* '0'.



Gambar 3.5. Rangkaian *micro switch*

3.2.4. Rangkaian penggerak Motor

Rangkaian ini merupakan rangkaian yang mengendalikan gerak motor. Motor dikendalikan untuk berputar ke kanan atau kiri. Motor didalam simulasi pintu irigasi ini berguna untuk membuka dan menutup pintu irigasi. Untuk bergerak ke kanan atau kiri maka dilakukan dengan mengubah polaritas dari kutub-kutub motor. Pada perancangan rangkaian ini digunakan transistor dan

relay, *relay* yang digunakan adalah dengan tipe SJ4089, yaitu merupakan *relay* semikonduktor yang cocok untuk penggunaan pada beban AC 220VAC dan beban DC. *Relay* jenis ini mempunyai tegangan kerja 6VDC dengan besarnya resistansi lilitannya " $R_C = 150\Omega$ " dan kemampuan arus beban sebesar 5A.

Untuk mengaktifkan *relay*, maka digunakan transistor yang berfungsi sebagai saklar elektronik. Transistor pada perancangan ini menggunakan transistor NPN tipe 2N2222 dengan besarnya $h_{fe} = '100'$ dan $I_C \text{ max} = '1A'$. Dengan besarnya resistansi $R_C = '150\Omega'$ pada kaki *collector* transistor, maka untuk mengaktifkan *relay* dibutuhkan arus *collector* I_C yaitu

$$I_C = \frac{V_S}{R_C} = \frac{6V}{150\Omega} = 40mA. \quad (6V = \text{tegangan sumber } relay)$$

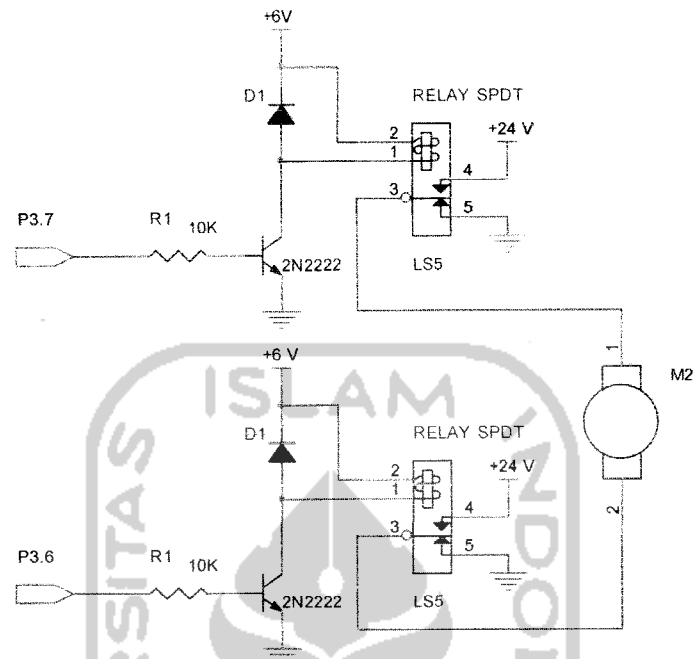
Ketika *relay* aktif, maka transistor berfungsi sebagai saklar tertutup/kondisi saturasi. Untuk membuat transistor saturasi, maka dibutuhkan $I_B \text{ sat}$. Sesuai dengan persamaan, $I_B \text{ sat} > \frac{I_C}{\beta}$, maka besarnya $I_B \text{ sat}$ yaitu: $I_B \text{ sat} > \frac{40mA}{100} > 0,4mA$. Dengan mengasumsikan $I_B \text{ sat} = '0,45mA'$, maka besarnya R_I

atau R_B dapat dicari dengan menggunakan persamaan 2.5 yaitu :

$$R_B = \frac{V_{BB} - V_{BE}}{I_B \text{ sat}} = \frac{5 - 0,7}{0,45mA} = 9,5 \text{ k}\Omega \text{ dibulatkan menjadi } 10 \text{ k}\Omega.$$

Pada perancangan ini motor yang digunakan adalah motor DC *gearbox* dengan tegangan kerja 24 V dan arus 250 mA. Motor DC jenis ini dapat mengangkat beban yang cukup berat dengan kecepatan motor yang sudah

direduksi dengan mekanik *gearbox* motor. Berikut aplikasi rangkaian yang digunakan.



Gambar 3.6. Rangkaian penggerak motor

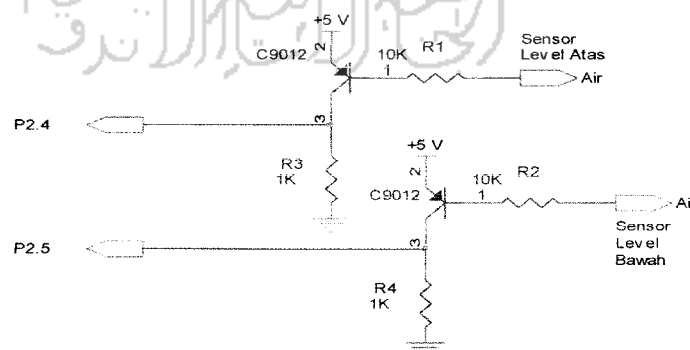
Dari rangkaian diatas, motor akan bergerak jika diberi logika '1' pada salah satu kaki basis transistor dan diberikan logika '0' untuk kaki basis lainnya. Sedangkan untuk kondisi "1 1" pada seluruh basis transistor tidak diperbolehkan. Kerena kondisi motor akan mendapat catu +24V pada kutub yang berlawanan. Sehingga akan menyebabkan drop tegangan pada rangkaian. Untuk lebih jelasnya berikut tabel pemberian input masukan untuk menggerakkan motor :

Tabel 3.1. Logika input rangkaian penggerak motor

P3.6	P3.7	Kondisi Motor
0	0	Tidak Bergerak
0	1	Bergerak ke atas
1	0	Bergerak ke bawah
1	1	Tidak diperbolehkan

3.2.1. Rangkaian Sensor Level Air

Rangkaian ini merupakan rangkaian yang digunakan untuk mengetahui keadaan level air pada bagian atas bendungan irigasi dan level air pada bagian bawah bendungan irigasi. Jika level air atas bendungan mendeteksi adanya air, maka keadaan ini digunakan oleh mikrokontroler untuk memanggil (call) ke no *hand phone* pengendali, dan jika level bawah tidak mendeteksi adanya air maka mikrokontroler akan memerintahkan untuk menutup semua pintu irigasi secara otomatis. Rangkaian sensor level air ini menggunakan transistor sebagai saklar dengan triger pada kaki basisnya yang dihubungkan dalam air yang merupakan ground pada rangkaian. Berikut aplikasi rangkaiannya :



Gambar 3.7. Rangkaian sensor level air

Besarnya arus yang mengalir pada *collector* I_C yaitu

$$I_C = \frac{V_S}{R_C} = \frac{5V}{1k\Omega} = 5mA. \text{ Sedangkan arus yang mengalir pada kaki basis dapat}$$

dihitung dengan rumus berikut:

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} = \frac{5 - 0,7}{10k\Omega} = 0,43mA$$

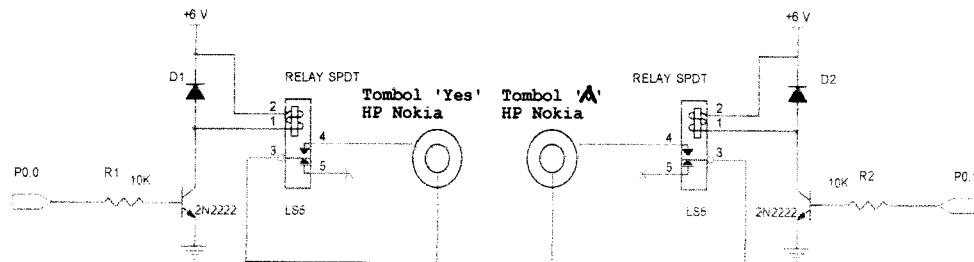
Dengan besarnya arus basis 0,43mA, maka transistor sudah dalam keadaan saturasi karena besarnya $I_B \cdot h_{fe} > I_C = 0,43mA \cdot 100 > 5mA = 43mA > 5mA$.

Dari gambar 3.7 diatas, ketika kondisi R1 dan R2 terhubung dengan ground air, maka kondisi transistor dalam keadan saturasi/on. Sehingga arus mengalir dari kaki emitor menuju kaki kolektor. Sehingga tegangan pada R3 atau R4 dalam kondisi *high* ('1'). Sedangkan jika R1 dan R2 tidak terhubung dengan air, maka kondisi transistor akan dalam keadaan *cut-off/off*. Artinya tidak ada arus triger pada kaki basis transistor. Sehingga tegangan pada R3 atau R4 dalam keadaan *low* ('0'). Dari kondisi *low* dan *high* pada kaki kolektor pada transistor ini digunakan oleh mikrokontroler untuk mengetahui kondisi level air atas dan level air bawah. Dimana kondisi *high* menandakan bahwa sensor mendeteksi adanya air dan kondisi *low* menandakan sensor tidak mendeteksi adanya air.

3.2.6. Rangkaian Pengendali *Hand phone* Penerima

Rangkaian pengendali handphone ini pada dasarnya sama seperti pada rangkaian penggerak motor, yang menggunakan *relay*. *Relay* pada rangkaian ini ini berfungsi untuk menghubungkan saklar yang terdapat pada tombol *hand phone* atau dengan kata lain menggantikan fungsi saklar *hand phone*. Saklar dalam hal

ini adalah saklar “Yes” dan “^” (tombol navigasi atas) yang terdapat pada *hand phone* penerima, dengan seri Nokia 1108. Berikut aplikasi rangkaian yang digunakan.



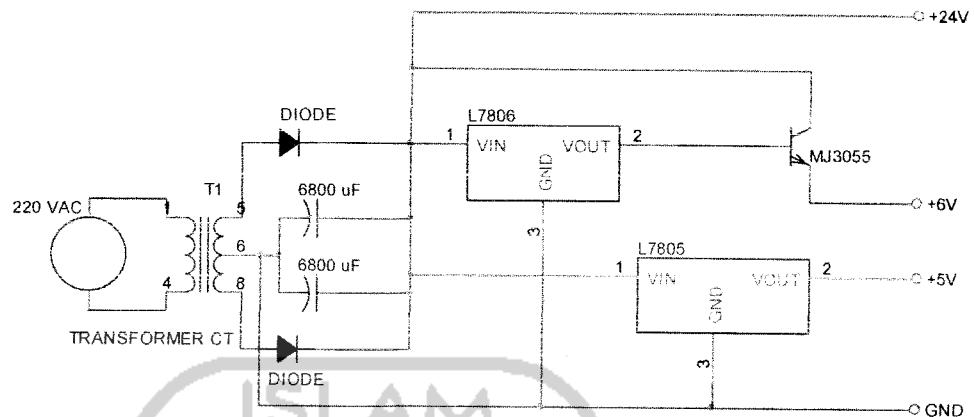
Gambar 3.8. Rangkaian pengendali *hand phone* penerima

Dari rangkaian diatas, untuk menggantikan saklar yang terdapat pada *hand phone* penerima, maka dilakukan dengan memberikan masukan berupa logika ‘1’ sehingga posisi saklar dalam keadaan tertutup atau terhubung. Sedangkan untuk memutus kondisi saklar, dilakukan dengan memberikan logika ‘0’ pada kaki basis transistor dari port mikrokontroler. Dimana pengoperasian untuk mengaktifkan *hand phone* disesuaikan dengan pengoperasian *hand phone* nokia seri yang telah diset no akhir yang akan dihubungi.

3.2.7. Rangkaian *Power Supply*

Setiap rangkaian elektronik tentunya membutuhkan catu daya, sehingga perancangan catu daya menjadi hal yang sangat penting, agar rangkaian ini dapat memberikan kebutuhan arus dan tegangan yang sesuai. Selain arus dan tegangan yang sesuai, hal lain yang perlu diperhatikan adalah kestabilan dari tegangan dan arus tersebut. Pada alat ini memerlukan tegangan catu +5V, +6V, +24V dan 220

VAC. Rangkaian catu daya yang digunakan pada alat ini dapat dilihat pada gambar berikut.



Gambar 3.9. Rangkaian catu daya (*power supply*)

Rangkaian catu daya diatas menggunakan transformator CT 5A dan pada keluaran 24 AC dipasang dua buah dioda berfungsi sebagai penyearah gelombang penuh dan mengubah tegangan AC menjadi DC serta kapasitor $13600\mu\text{F}$ digunakan untuk memperhalus tegangan DC yang dihasilkan. Dari tegangan DC yang berasal dari dioda dan kapasitor, maka untuk memperoleh tegangan +5V digunakan IC LM7805, tegangan +6V digunakan IC LM7806, dan tegangan +24V langsung setelah dioda penyearah.

Pada tegangan +6V digunakan transistor MJ3055 sebagai penguat arus. Karena arus yang dihasilkan pada IC LM7806 hanya sebesar 1A. Sehingga kurang cukup untuk memenuhi tegangan +6V yang digunakan untuk *supply* banyak *relay*. Sedangkan pada tegangan +24V dilangsungkan pada dioda penyearah agar tegangan yang dihasilkan untuk menggerakkan motor menghasilkan arus yang maksimal sehingga gerak motor akan maksimal kecepatannya.

3.3. Perancangan Perangkat Lunak

Pada perancangan alat ini diperlukan perancangan perangkat lunak untuk menjalankannya. *Software* atau perangkat lunak digunakan untuk memberikan langkah-langkah yang harus dilakukan CPU. Bahasa yang digunakan untuk memprogram mikrokontroler adalah bahasa *assembly*.

Pada perancangan *software* ini juga dilengkapi dengan sistem *password* yang digunakan untuk mengontrol kerja dari motor penggerak yang difungsikan untuk menggerakkan pintu bendungan. Untuk sistem *password* yang digunakan pada alat ini dapat dilihat pada tabel berikut :

SETTING PASSWORD PENGENDALI PINTU IRIGASI :

PASSWORD 3 DIGIT

1	2	3
----------	----------	----------

SETTING PASSWORD :

1. Banyaknya motor yang diaktifkan
2. Jenis keadaan
3. Naik / Turun / Berhenti
 1. Naik
 2. Turun
 3. Berhenti

PINTU IRIGASI :

1	2	3	4	5
----------	----------	----------	----------	----------

Kemungkinan Pintu dapat membuka & menutup secara acak :

Untuk 1 Pintu (1 motor) :

0	:	1
1	:	2
2	:	3
3	:	4
4	:	5

Untuk 2 Pintu (2 motor) :

0	:	1, 2	5	:	2, 4
1	:	1, 3	6	:	2, 5
2	:	1, 4	7	:	3, 4
3	:	1, 5			
4	:	2, 3			

Untuk 3 Pintu (3 motor) :

0	:	1, 2, 3	5	:	1, 4, 5
1	:	1, 2, 4	6	:	2, 3, 4
2	:	1, 2, 5	7	:	2, 3, 5
3	:	1, 3, 4	8	:	2, 4, 5
4	:	1, 3, 5	9	:	3, 4, 5

Untuk 4 Pintu (4 motor) :

0	:	1, 2, 3, 4
1	:	1, 2, 3, 5
2	:	1, 2, 4, 5
3	:	1, 3, 4, 5
4	:	2, 3, 4, 5

Untuk 5 Pintu (5 motor) :

0	:	1, 2, 3, 4, 5
---	---	---------------

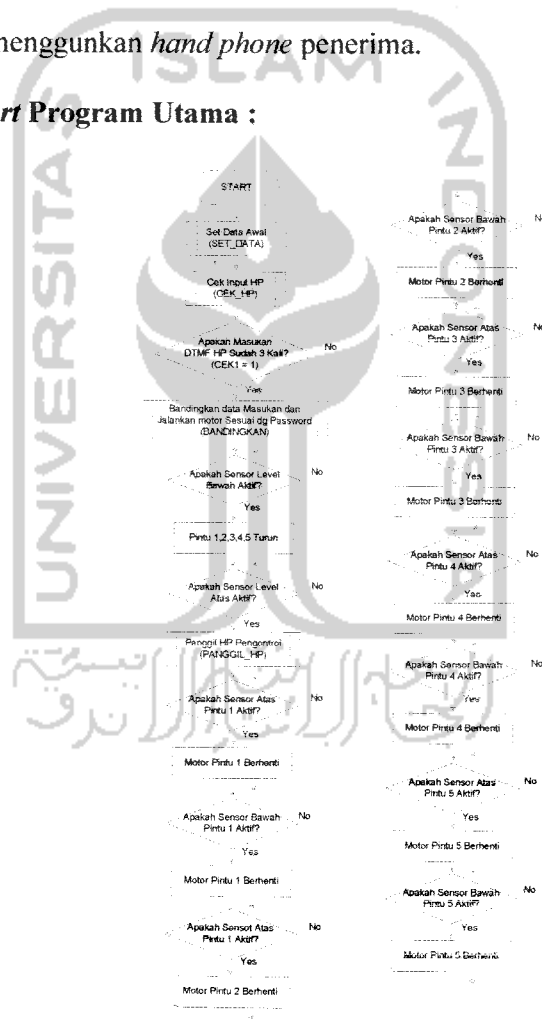
Software pada alat ini dibagi menjadi satu program utama yang merupakan program inti dari keseluruhan alat ini. Selain itu juga berisi *subroutin-subroutin* program, antara lain *subroutin* Cek_Hp yang digunakan untuk mengecek kondisi masukan DTMF HP, dan subrutin Bandingkan yang digunakan untuk membandingkan hasil password dari *hand phone* Pengendali. Tetapi untuk

penjelasnya pada laporan ini ditekankan pada program utama dan program untuk mengecek kondisi DTMF *hand phone* penerima

3.3.1. Program Utama

Program utama ini merupakan program yang mencakup seluruh program yang ada pada alat ini. Dimana program ini berisi program yang digunakan untuk mengecek masukan DTMF *hand phone* pengendali, mengecek kondisi semua sensor dan mengaktifkan motor pada pintu irigasi serta memanggil *hand phone* pengendali menggunakan *hand phone* penerima.

▪ Flowchart Program Utama :

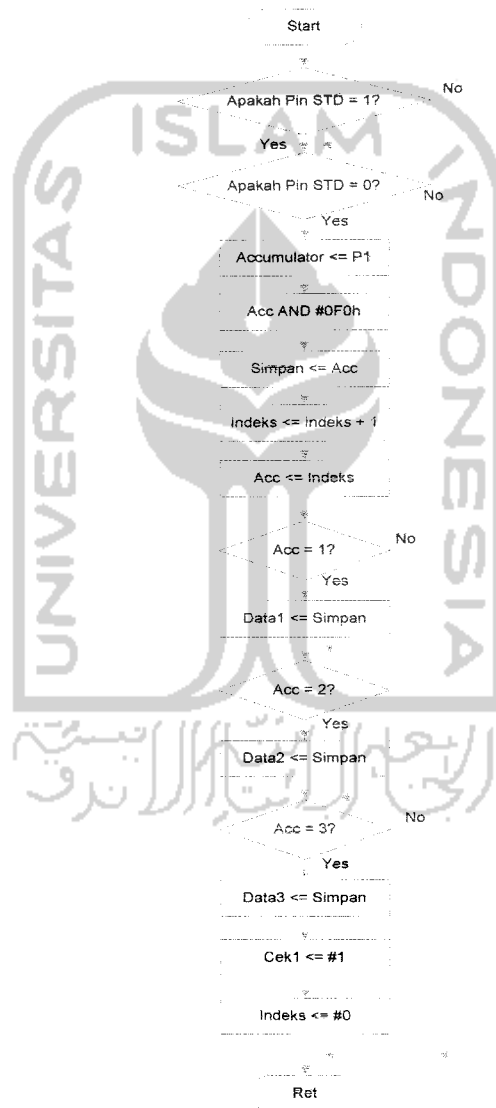


Gambar 3.10. Flowchart program utama (detail lihat lampiran)

3.3.2. Subroutin Program CEK_HP

Pada *subroutin* ini berisi program yang digunakan untuk mengecek kondisi masukan DTMF pada *hand phone* penerima serta mengecek apakah masukan *password* tersebut sudah dilakukan sebanyak 3 kali. Berikut untuk lebih jelasnya perhatikan diagram alir *subroutin* program CEK_HP.

▪ **Flowchart Subroutin Program Cek_HP :**

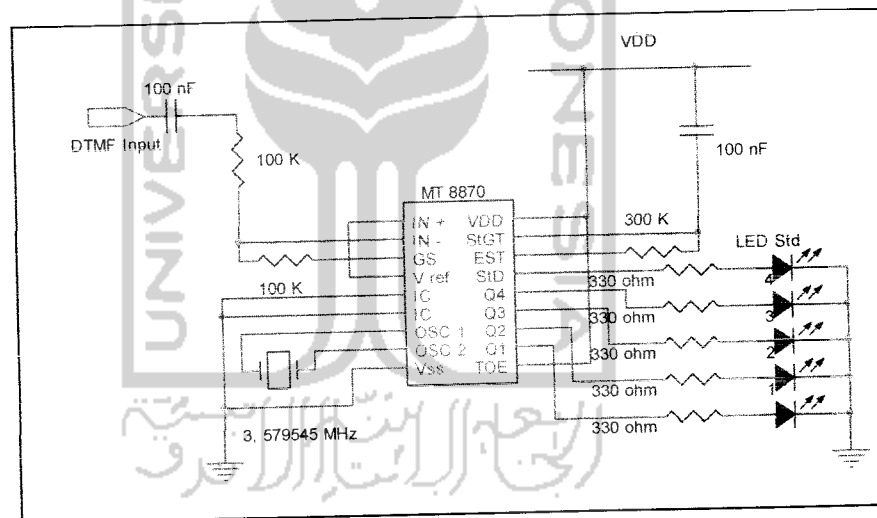


Gambar 3.11. Flowchart subroutin cek_Hp

BAB IV ANALISA DAN PEMBAHASAN

4.1. Pengujian Rangkaian Penerima DTMF

Untuk dapat melakukan pengujian terhadap unit blok ini, maka blok rangkaian dihubungkan dengan perangkat *mobile phone*. Pada perangkat *mobile phone* diaktifkan terlebih dahulu nada tombolnya, kemudian dihubungkan dengan blok penerima DTMF melalui jalur *handsfree*. Output dari IC DTMF (Q1- Q4) dihubungkan dengan LED. Input dimasukkan mulai dari tombol 1,2,3, dan seterusnya sampai tombol #. Berikut rangkaian yang digunakan untuk pengujian penerima DTMF.



Gambar 4.1. Rangkaian uji unit penerima DTMF

Dengan menggunakan rangkaian diatas, maka diperoleh data hasil pengujian seperti pada Tabel 4.1 berikut :

Tabel 4.1. Data hasil pengujian rangkaian penerima DTMF

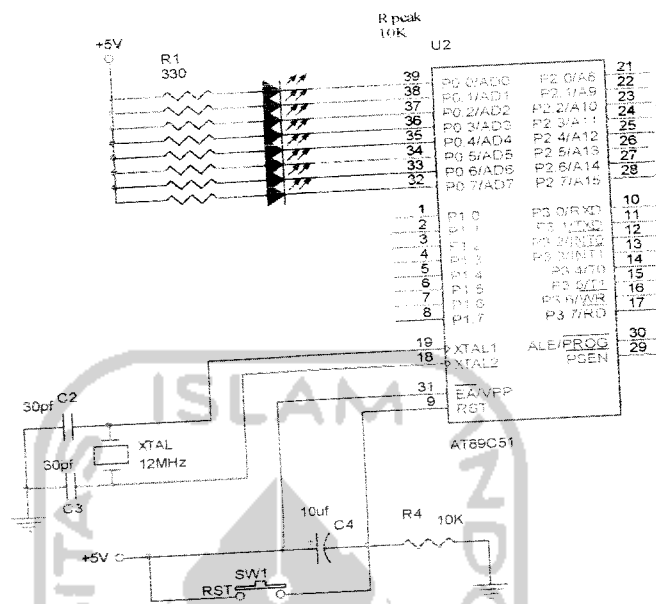
Digit	TOE	LED Std	LED 4	LED 3	LED 2	LED 1	Tegangan output (Volt)			
							Q4	Q3	Q2	Q1
1	H	1	0	0	0	1	≈ 0	≈ 0	≈ 0	4,9
2	H	1	0	0	1	0	≈ 0	≈ 0	4,9	≈ 0
3	H	1	0	0	1	1	≈ 0	≈ 0	4,9	4,9
4	H	1	0	1	0	0	≈ 0	4,9	≈ 0	≈ 0
5	H	1	0	1	0	1	≈ 0	4,9	≈ 0	4,9
6	H	1	0	1	1	0	≈ 0	4,9	4,9	≈ 0
7	H	1	0	1	1	1	≈ 0	4,9	4,9	4,9
8	H	1	1	0	0	0	4,9	≈ 0	≈ 0	≈ 0
9	H	1	1	0	0	1	4,9	≈ 0	≈ 0	4,9
0	H	1	1	0	1	0	4,9	≈ 0	4,9	≈ 0
*	H	1	1	0	1	1	4,9	≈ 0	4,9	4,9
#	H	1	1	1	0	0	4,9	4,9	≈ 0	≈ 0

Rangkaian ini menerjemahkan frekuensi yang berbeda dari nada tombol *mobile phone* menjadi data digital. Proses inisialisasi DTMF mengaktifkan TOE untuk menerima sinyal DTMF, sedangkan Std menandakan ada atau tidak sinyal DTMF yang masuk ke IC MT 8870. Std aktif *high* (+5 V) jika ada sinyal DTMF yang masuk. Dari Tabel 4.1 di atas dapat dilihat bahwa, TOE harus selalu aktif *high* (dihubungkan dengan VCC) agar MT8870 bisa mengeluarkan *output*. Logika '1' menyatakan kondisi LED nyala (*on*), sedangkan logika '0' menyatakan kondisi LED mati (*off*).

4.2. Pengujian Sistem Minimum Mikrokontroler

Pengujian rangkaian ini dilakukan dengan memberikan program sederhana untuk menghidupkan led berjalan geser kiri pada P0. Kondisi untuk mengaktifkan led disini adalah dengan memberikan logika '0' karena katoda dari led yang

dihubungkan dengan mikrokontroler. Untuk lebih jelasnya perhatikan gambar rangkaian berikut:



Gambar 4.2. Rangkaian pengujian sistem minimum

Dengan menggunakan rangkaian diatas, kemudian mikrokontroler diberi program sederhana seperti berikut, maka led akan berjalan bergeser ke kiri dengan tundaan waktu yang tidak cukup lama yang terdapat pada *subroutin* TUNDA.

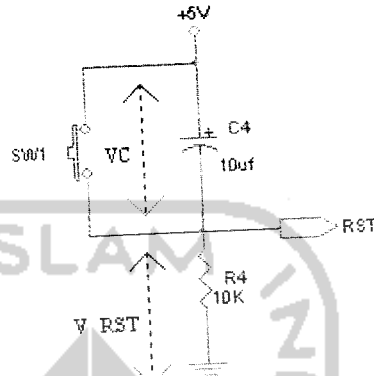
```

ORG 00H
MOV A, #0FEH
Mulai:
MOV P0, A
ACALL TUNDA
RL A
SJMP MULAI

TUNDA:
MOV R0, #05H
T1: MOV R1, #0FFH
T2: MOV R2, #0FFH
DJNZ R2, $
DJNZ R1, T2
DJNZ R0, T1
RET
END

```

Untuk membangkitkan sinyal *reset*, maka diperlukan sebuah kapasitor elektrolit 10 $\mu\text{F}/25\text{ V}$ dan resistor tetap dengan nilai tahanan sebesar 10 $\text{k}\Omega$ serta sebuah *push button switch*. Dalam rancangan alat ini rangkaian *reset* dapat bekerja secara manual maupun otomatis saat catu daya diaktifkan.



Gambar 4.3. Skema rangkaian *reset*

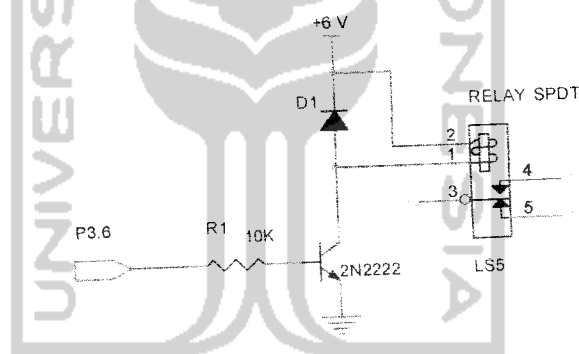
Berdasarkan gambar 4.3 ketika catu daya diaktifkan, kapasitor C4 akan berlogika '1'. Arus mengalir dari Vcc langsung ke kaki *reset* sehingga kaki tersebut berlogika '1'. Kemudian kapasitor terisi hingga tegangan pada kapasitor (Vc) mencapai Vcc, otomatis tegangan pada R4 atau tegangan V_{RST} turun menjadi 0 sehingga kaki *reset* akan berlogika '0' dan proses *reset* berakhir.

Jika saklar SW1 ditekan, *reset* bekerja secara manual dan aliran arus akan mengalir dari Vcc menuju kaki *reset*. Sehingga tegangan *reset* akan menjadi 5V. Tegangan 5V tersebut menyebabkan kaki *reset* berlogika '1' pada saat saklar tersebut ditekan. Saat saklar dilepas, aliran arus Vcc akan terhenti menuju ke nol sehingga logika pada kaki ini berubah menjadi '0' dan proses *reset* selesai.

internal mikrokontroler AT89S51. Sehingga untuk memberikan *input* diberi logika 0.

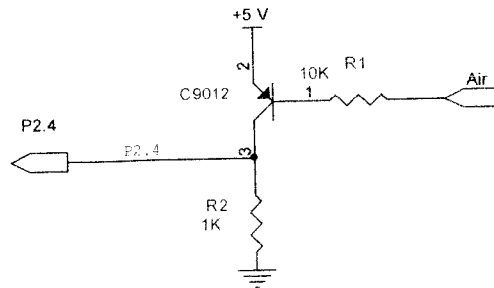
4.4. Pengujian Rangkaian Penggerak *Relay*

Pengujian rangkaian penggerak motor dan pengendali *hand phone* penerima dilakukan menguji keadaan *relay*. Karena pada dasarnya rangkaian tersebut merupakan rangkaian penggerak *relay* yang difungsikan untuk menggerakkan motor dan mengkatifkan *hand phone* penerima. Dengan memberikan logika '1' pada kaki basis transistor maka kondisi *relay* akan aktif dan dengan memberikan logika '0' maka kondisi *relay* akan mati. Berikut realisasi rangkaian pengujian penggerak *relay* yang digunakan.



Gambar 4.4. Rangkaian pengujian penggerak *relay*

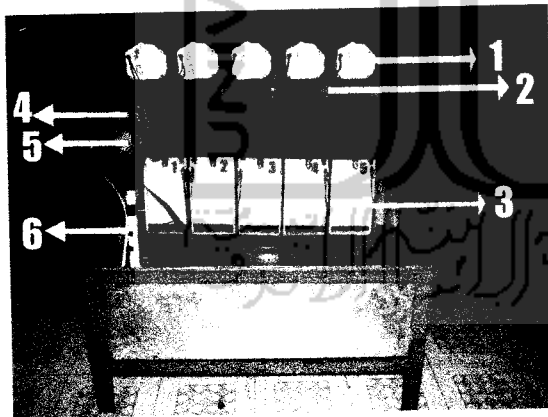
Dengan menggunakan rangkaian diatas, maka diperoleh data hasil pengujian rangkaian sebagai berikut.



Gambar 4.5. Rangkaian pengujian sensor level air

Dari gambar 4.5 rangkaian diatas, ketika kaki basis transistor terhubung dengan *ground*, maka transistor akan dalam kondisi saturasi/*ON*. Sedangkan kondisi transistor ketika kaki basis tidak terhubung dengan *ground*, maka transistor tersebut dalam keadaan *cut-off* atau seperti saklar tertutup. Sehingga arus yang mengalir pada basis dan *collector* adalah 0.

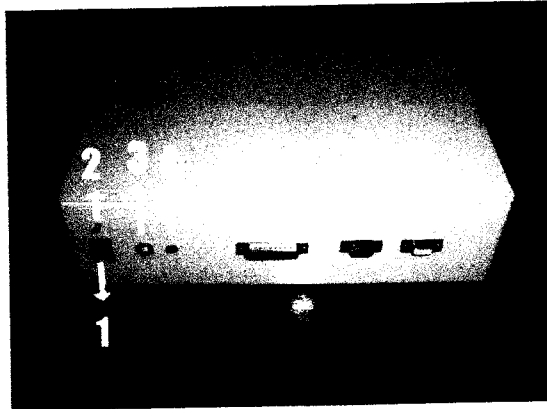
4.6. Gambar Sistem Simulasi Pengendali Pintu Irigasi



Keterangan :

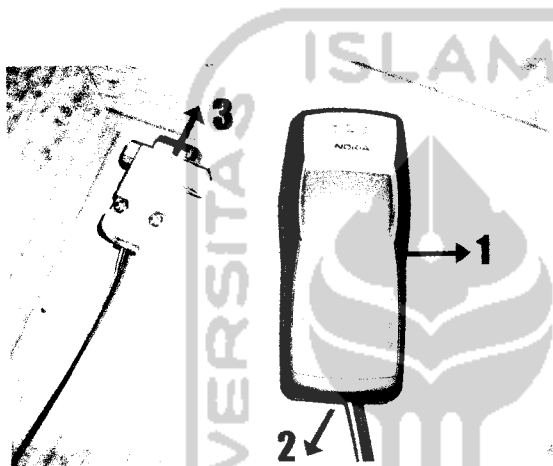
1. Motor Dc Gear Box
2. Mikro Switch
3. Simulasi Pintu Irigasi
4. Konektor Sensor Level Air (Input)
5. Konektor Driver Motor (Input)
6. Sensor Level Air

Gambar 4.6. Mekanik sistem simulasi pengendali pintu irigasi

**Keterangan :**

1. Saklar ON / OFF
2. LED Indikator
3. Tombol Reset
4. Input Handsfree
5. Konektor Driver Motor (Output)
6. Konektor Sensor Level Air (Output)
7. Konektor Driver Hp Penerima (Output)

Gambar 4.7. Blok kontrol kendali simulasi pintu irigasi

**Keterangan :**

1. Hp Penerima
2. Output Handsfree
3. Konektor Hp Penerima (Input)

Gambar 4.8. Hp Penerima Simulasi pengendali pintu irigasi

4.7. Perhitungan Arus dan Daya Sistem Simulasi Pintu Irigasi**4.7.1. Timing buka tutup pintu irigasi mulai dari awal perintah sampai**

terbuka dan tertutup penuh

Tabel 4.4. Timing buka tutup pintu irigasi

Timing pintu terbuka penuh	Timing pintu tertutup penuh
04 : 49	04 : 45

4.7.2. Perhitungan arus dan daya pada output sistem

$$V = 220 \text{ V}$$

$$P = V \times I$$

Arus pada keadaan sistem standby :

$$I = 0,03 \text{ A (pengukuran ampere meter)}$$

$$\begin{aligned} P &= 220 \text{ V} \times 0,03 \text{ A} \\ &= 6,6 \text{ W} = 7 \text{ W} \end{aligned}$$

Arus pada keadaan sistem kerja :

$$I = 0,06 \text{ A (pengukuran ampere meter)}$$

$$\begin{aligned} P &= 220 \text{ V} \times 0,06 \text{ A} \\ &= 13,2 \text{ W} = 13 \text{ W} \end{aligned}$$

4.7.3. Perhitungan arus dan daya pada output power supply (5 V, 6 V, 24 V)

Arus pada keadaan sistem standby :

$$\text{pada tegangan } 5 \text{ V, } I = 0 \text{ A (pengukuran ampere meter)}$$

$$\text{pada tegangan } 6 \text{ V, } I = 0 \text{ A (pengukuran ampere meter)}$$

$$\text{pada tegangan } 24 \text{ V, } I = 0 \text{ A (pengukuran ampere meter)}$$

$$P = V \times I = 0 \text{ W}$$

jadi daya pada keadaan sistem stand by = 0 W, karena arus pada keadaan sistem stand by = 0 A

Arus pada keadaan sistem kerja :

$$\text{pada tegangan } 5 \text{ V, } I = 0,5 \text{ mA (pengukuran ampere meter)}$$

$$P = 5 \text{ V} \times 0,5 \text{ mA}$$

$$= 2,5 \text{ mW} = 2,5 \cdot 10^{-3} \text{ W}$$

pada tegangan 6 V, $I = 0,6 \text{ mA}$ (pengukuran ampere meter)

$$P = 6 \text{ V} \times 0,6 \text{ mA}$$

$$= 3,6 \text{ mW} = 3,6 \cdot 10^{-3} \text{ W}$$

pada tegangan 24 V, $I = 130 \text{ mA}$ (pengukuran ampere meter 5 motor)

$$P = 24 \text{ V} \times 130 \text{ mA}$$

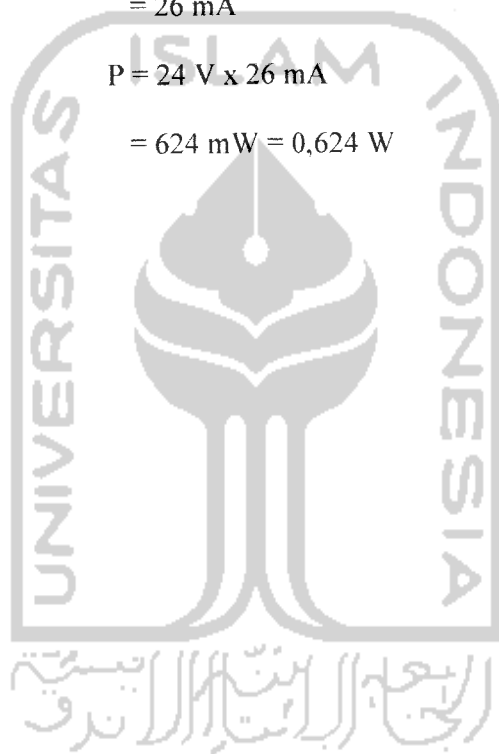
$$= 3120 \text{ mW} = 3,12 \text{ W}$$

I (untuk 1 motor) = $130 \text{ mA} / 5$

$$= 26 \text{ mA}$$

$$P = 24 \text{ V} \times 26 \text{ mA}$$

$$= 624 \text{ mW} = 0,624 \text{ W}$$



BAB V PENUTUP

5.1. Kesimpulan

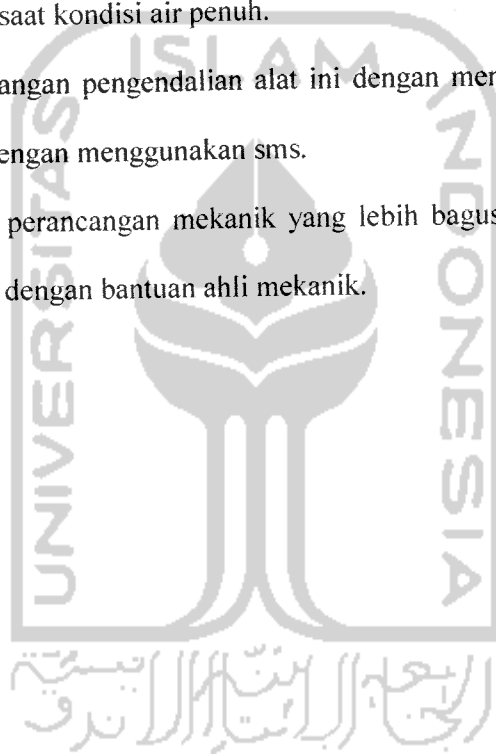
Berdasarkan perancangan dan hasil pengujian yang telah dilakukan terhadap “Simulasi Pengendali Pintu Irigasi Dengan Menggunakan DTMF (*Dual Tone Multiple Frequency*) Mobile Phone Berbasis Mikrokontroler AT89S51”, maka dapat disimpulkan bahwa:

1. Sistem ini terdiri dari perangkat keras (*hardware*) rangkaian catu daya, sistem minimum mikrokontroler AT89S51, rangkaian penerima DTMF, sensor pembatas dengan *micro switch*, pengendali *hand phone*, sensor level air, dan rangkaian penggerak motor.
2. Perangkat lunak (*software*) yang digunakan dalam sistem ini adalah bahasa *assembly* mikrokontroler AT89S51 yang berfungsi untuk mendeteksi kondisi sensor level air, sensor pembatas gerak motor dengan menggunakan *micro switch*, mengaktifkan motor pada pintu irigasi, mengecek kondisi masukan DTMF *hand phone* dan untuk memanggil (*call*) *hand phone* pengendali.
3. Berdasarkan pengujian yang telah dilakukan, secara keseluruhan sistem sudah sesuai dengan fungsi yang ditetapkan, yaitu dapat mengendalikan / mengontrol simulasi pintu irigasi dengan menggunakan DTMF *hand phone* serta memanggil *hand phone* pengendali dan membatasi gerak motor pada pintu irigasi secara otomatis menggunakan *micro switch* dan sensor level air

5.2. Saran

Karena keterbatasan kemampuan dan waktu, penyusun menyadari sistem yang dirancang ini belum sempurna sehingga ada beberapa hal yang dapat dikaji dan dikembangkan dari alat ini, yaitu :

1. Alat ini masih dalam bentuk simulasi, sehingga perlu pengembangan untuk realisasi pintu irigasi yang sebenarnya.
2. Sistem yang dirancang perlu dikembangkan agar dapat melakukan *call* berulang saat kondisi air penuh.
3. Pengembangan pengendalian alat ini dengan menggunakan *password* yang di kontrol dengan menggunakan sms.
4. Perlunya perancangan mekanik yang lebih bagus guna mengatasi keterbatasan pada alat dengan bantuan ahli mekanik.



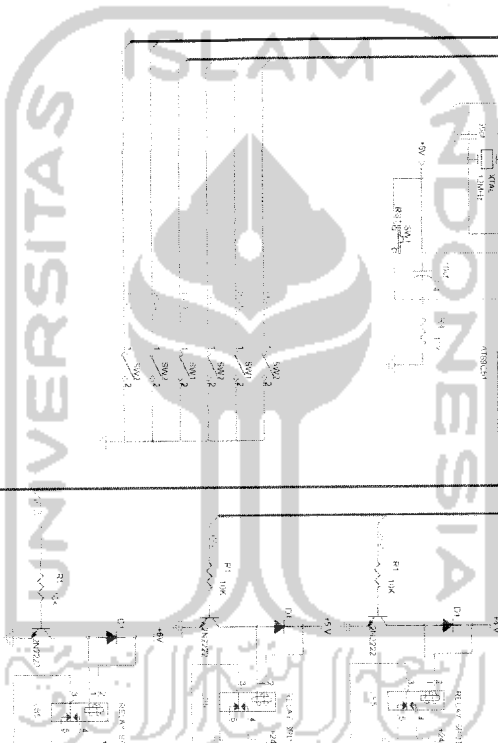
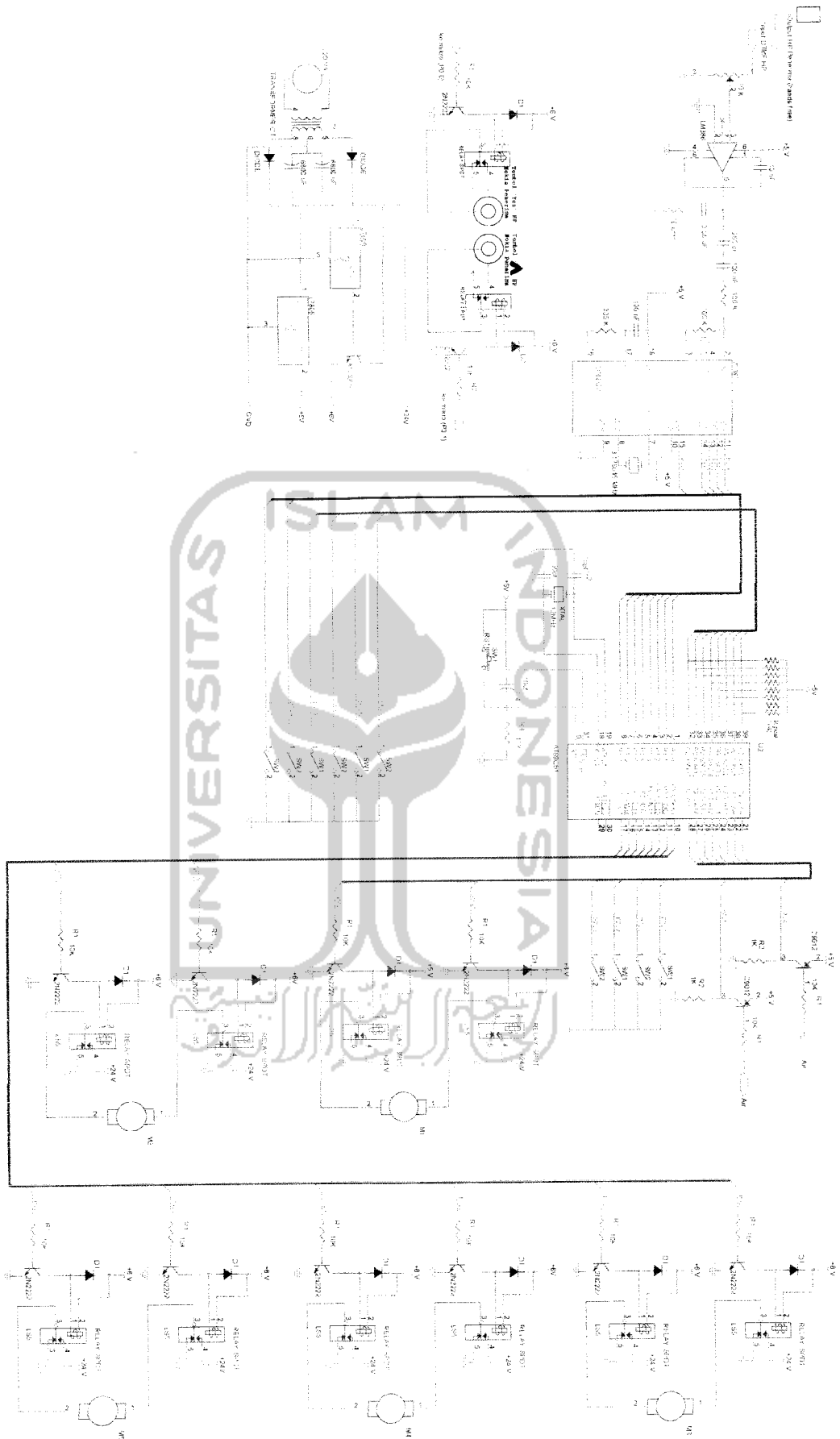
DAFTAR PUSTAKA

- Soedjono (1996). *Instalasi dan Alat-alat listrik*. Bandung: Titian ilmu.
- Coughlin F. Robert & Driscoll F. Frederic. 1985. *Penguat Operasional dan Rangkaian Terpadu Linier*. Jakarta: Erlangga.
- Sutanto Budhy. 2001. *Menghubungkan MCS51 ke Dunia Analog*.
<http://ALDS.stts.edu>.
- Nalwan Andi Paulus. 2003. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51*. Jakarta: PT Elex Media Komputindo.
- Malvino. 2003. *Prinsip-Prinsip Elektronika*. Jakarta: Salemba Teknika.
- Budiharto Widodo. 2005. *Perancangan Sistem dan Aplikasi Mikrokontroler*. Jakarta: PT Elex Media Komputindo
- Putra Eko Agfianto. 2002. *Belajar Mikrokontroler AT89C51/52/55*. Yogyakarta: Gava Media.
- Muchlas. 2005. *Rangkaian Digital*. Yogyakarta: Gava Media
- <http://alldatasheet.com>.
- <http://onsemi.com>.





LAMPURAN

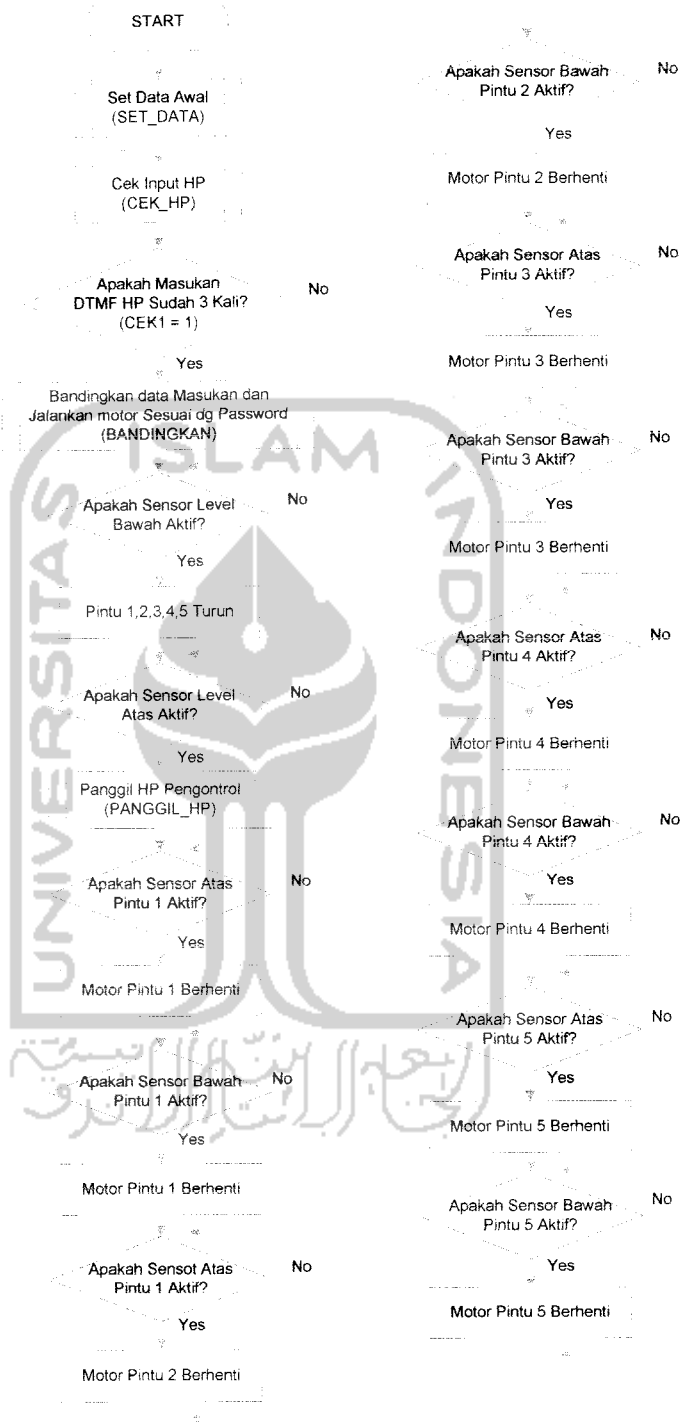


SUBJECT: ELECTRIC POWER SYSTEMS (EPS)

NAME: ...

NPM: ...

▪ **Flowchart Program Utama :**



**Program Simulasi Pengendali Pintu Irigasi Dengan Menggunakan DTMF
Mobile Phone Berbasis Mikrokontroler AT89S51 :**

```

s_pintu1_atas bit p2.0
s_pintu1_bawah bit p2.1
s_pintu2_atas bit p2.2
s_pintu2_bawah bit p2.3
s_pintu3_atas bit p0.4
s_pintu3_bawah bit p0.5
s_pintu4_atas bit p0.6
s_pintu4_bawah bit p0.7
s_pintu5_atas bit p1.4
s_pintu5_bawah bit p1.5
pintu1_atas bit p2.6
pintu1_bawah bit p2.7
pintu2_atas bit p3.0
pintu2_bawah bit p3.1
pintu3_atas bit p3.2
pintu3_bawah bit p3.3
pintu4_atas bit p3.4
pintu4_bawah bit p3.5
pintu5_atas bit p3.6
pintu5_bawah bit p3.7
std bit p1.6
s_level_atas bit p2.5
s_level_bawah bit p2.4
hp1 bit p0.0
hp2 bit p0.1
indeks equ 20h
simpan equ 21h
data1 equ 22h
data2 equ 23h
data3 equ 24h
data4 equ 25h
cek equ 26h
cek1 equ 27h
cek_hp equ 28h
detik equ 29h
detik1 equ 2ah
detik2 equ 2bh
ceking equ 2ch
detect equ 2dh
pintu1 equ 30h
pintu2 equ 31h
pintu3 equ 32h
pintu4 equ 33h
pintu5 equ 34h
TIM0 EQU -250
;

```

```

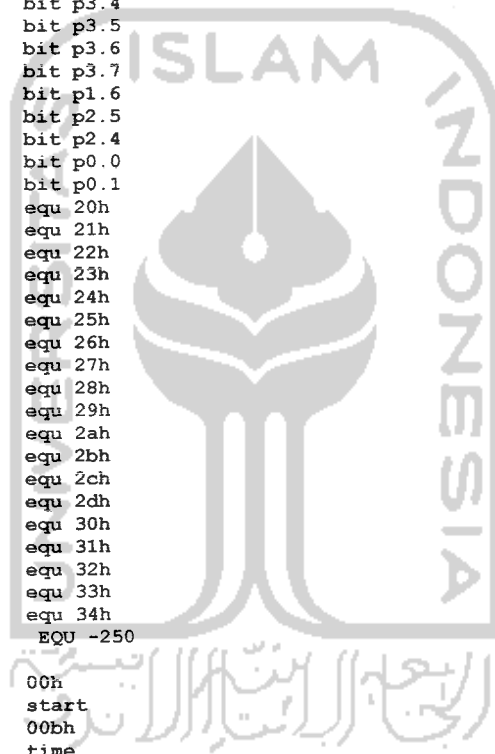
org 00h
sjmp start
org 00bh
sjmp time

```

```

time: push acc
inc detik1
mov a, detik1
cjne a, #250, time1
mov detik1, #00h
inc detik
mov a, detik
cjne a, #250, time1
setb cek_hp.1
mov detik, #00h
time1: pop acc
reti

```



```

start:
    acall    setdata
mulai:  acall    cek_hp
        mov     a,cek1
        cjne   a,#1,mulai1
        mov     cek1,#0
        lcall  bandingkan
        mov     ceking,#00h
        mov     detect,#00h
mulai1:
        jb     s_level_bawah,mulai2
        jb     ceking.0,mulai2
        acall  pintu1_turun
        acall  pintu2_turun
        acall  pintu3_turun
        acall  pintu4_turun
        acall  pintu5_turun
        setb   ceking.0
mulai2: jnb     s_level_atas,mulai3
        jb     cek.1,mulai3
        acall  panggil_hp
        setb   ceking.1
mulai3: mov     a,pintu1
        cjne   a,#1,mulai3a
        jb     s_pintu1_atas,mulai4
        sjmp   mulai3b
mulai3a:cjne   a,#2,mulai4
        jb     s_pintu1_bawah,mulai4
mulai3b:acall  pintu1_berhenti
mulai4: mov     a,pintu2
        cjne   a,#1,mulai4a
        jb     s_pintu2_atas,mulai5
        sjmp   mulai4b
mulai4a:cjne   a,#2,mulai5
        jb     s_pintu2_bawah,mulai5
mulai4b:acall  pintu2_berhenti
mulai5: mov     a,pintu3
        cjne   a,#1,mulai5a
        jb     s_pintu3_atas,mulai6
        sjmp   mulai5b
mulai5a:cjne   a,#2,mulai6
        jb     s_pintu3_bawah,mulai6
mulai5b:acall  pintu3_berhenti
mulai6: mov     a,pintu4
        cjne   a,#1,mulai6a
        jb     s_pintu4_atas,mulai7
        sjmp   mulai6b
mulai6a:cjne   a,#2,mulai7
        jb     s_pintu4_bawah,mulai7
mulai6b:acall  pintu4_berhenti
mulai7: mov     a,pintu5
        cjne   a,#1,mulai7a
        jb     s_pintu5_atas,mulai8
        sjmp   mulai7b
mulai7a:cjne   a,#2,mulai8
        jb     s_pintu5_bawah,mulai8
mulai7b:acall  pintu5_berhenti
mulai8: ajmp   mulai

cek_hp:
        jnb   std,terus3
        jb   std,$
        mov  a,p1
        anl  a,#0fh
        mov  simpan,a
        inc  indeks

```



```

konversi:
    mov     a, indeks
    cjne   a, #1, terus1
    mov     data1, simpan
terus1:  cjne   a, #2, terus2
    mov     data2, simpan
terus2:  cjne   a, #3, terus3
    mov     data3, simpan
    mov     cek1, #1
    mov     indeks, #0
terus3:  ret

baca_dtmf:
    jb     std, $
    mov     a, p1
    anl    a, #0fh
    mov     simpan, a
    dec    indeks
    mov     cek, #1
    ret

setdata:
    mov     p2, #0ffh
    mov     p0, #0ffh
    mov     p3, #0ffh
    mov     p1, #0ffh
    mov     cek, #00
    mov     cek1, #0
    mov     indeks, #0
    mov     simpan, #0
    mov     detik, #00h
    mov     detik1, #00h
    mov     detik2, #00h
    mov     cek_hp, #00h
    mov     ceking, #00h
    mov     detect, #00h
    mov     pintu1, #0
    mov     pintu2, #0
    mov     pintu3, #0
    mov     pintu4, #0
    mov     pintu5, #0
    ret

panggil_hp:
    setb   hp1
    setb   hp2
    clr    hp2
    acall  delay
    setb   hp2
    acall  delay
    acall  delay
    clr    hp1
    acall  delay
    setb   hp1
    acall  delay
    acall  aktifkan_pewaktu
    jnb   cek_hp.1, $
    clr    hp1
    acall  delay
    setb   hp1
    acall  delay
    acall  delay
    ret

AKTIFKAN PEWAKTU:
    mov     detik, #00h
    mov     detik1, #00h
    MOV     TMOD, #002H    ;SET MODE
    MOV     TH0, #TIMO
    SETB   ET0
    SETB   EA
    SETB   TR0
    RET

```

MATIKAN_PEWAKTU:

```
MOV    TCON,#00H
RET
```

```
;
; Digit 1: banyaknya motor yang diaktifkan
; Digit 2: jenis keadaan
; Digit 3: naik/turun/berhenti (1: naik, 2:turun, 3:berhenti)
;
; keadaan:
; 1 motor (m1=0, m2=1, m3=2, m4=3, m5=4)
; 2 motor (m1m2=0, m1m3=1, m1m4=2, m1m5=3, m2m3=4, m2m4=5, m2m5=6)
;          (m3m4=7)
; 3 motor (m1m2m3=0, m1m2m4=1, m1m2m5=2, m1m3m4=3, m1m3m5=4)
;          (m2m3m4=5, m2m3m5=6, m3m4m5=7, m2m4m5=8, m3m4m5=9)
; 4 motor (m1m2m3m4=0, m1m2m3m5=1, m1m2m4m5=2, m1m3m4m5=3
;          (m2m3m4m5=4)
; 5 motor (m1m2m3m4m5=0)
```

bandingkan:

```
mov    a,data1
cjne   a,#1,bd1
;
mov    a,data2
cjne   a,#10,bd01
mov    a,data3
cjne   a,#1,bd000
acall  pintu_naik
ret
bd000: cjne   a,#2,bd001
acall  pintu_turun
ret
bd001: cjne   a,#3,bd002
acall  pintu_berhenti
bd002: ret
;
bd01:  cjne   a,#1,bd02
mov    a,data3
cjne   a,#1,bd010
acall  pintu2_naik
ret
bd010: cjne   a,#2,bd011
acall  pintu2_turun
ret
bd011: cjne   a,#3,bd012
acall  pintu2_berhenti
bd012: ret
;
bd02:  cjne   a,#2,bd03
mov    a,data3
cjne   a,#1,bd020
acall  pintu3_naik
ret
bd020: cjne   a,#2,bd021
acall  pintu3_turun
ret
bd021: cjne   a,#3,bd022
acall  pintu3_berhenti
bd022: ret
;
bd03:  cjne   a,#3,bd04
mov    a,data3
cjne   a,#1,bd030
acall  pintu4_naik
ret
bd030: cjne   a,#2,bd031
acall  pintu4_turun
ret
bd031: cjne   a,#3,bd032
acall  pintu4_berhenti
bd032: ret
;
```

```

bd04:  cjne    a,#4,bd042
        mov     a,data3
        cjne    a,#1,bd040
        acall   pintu5_naik
        ret
bd040:  cjne    a,#2,bd041
        acall   pintu5_turun
        ret
bd041:  cjne    a,#3,bd042
        acall   pintu5_berhenti
bd042:  ret
;
; 2 motor (m1m2=0, m1m3=1, m1m4=2, m1m5=3, m2m3=4, m2m4=5, m2m5=6)
;         (m3m4=7)
;
bd11:  ljmp    bd2

bd1:    cjne    a,#2,bd11
        mov     a,data2
        cjne    a,#10,bd11
        mov     a,data3
        cjne    a,#1,bd100
        acall   pintu1_naik
        acall   pintu2_naik
        ret
bd100:  cjne    a,#2,bd101
        acall   pintu1_turun
        acall   pintu2_turun
        ret
bd101:  cjne    a,#3,bd102
        acall   pintu1_berhenti
        acall   pintu2_berhenti
bd102:  ret
;
bd11:   cjne    a,#1,bd12
        mov     a,data3
        cjne    a,#1,bd110
        acall   pintu1_naik
        acall   pintu3_naik
        ret
bd110:  cjne    a,#2,bd111
        acall   pintu1_turun
        acall   pintu3_turun
        ret
bd111:  cjne    a,#3,bd112
        acall   pintu1_berhenti
        acall   pintu3_berhenti
bd112:  ret
;
bd12:   cjne    a,#2,bd13
        mov     a,data3
        cjne    a,#1,bd120
        acall   pintu1_naik
        acall   pintu4_naik
        ret
bd120:  cjne    a,#2,bd121
        acall   pintu1_turun
        acall   pintu4_turun
        ret
bd121:  cjne    a,#3,bd122
        acall   pintu1_berhenti
        acall   pintu4_berhenti
bd122:  ret
;
bd13:   cjne    a,#3,bd14
        mov     a,data3
        cjne    a,#1,bd130
        acall   pintu1_naik
        acall   pintu5_naik
        ret
bd130:  cjne    a,#2,bd131

```

```

        acall  pintul_turun
        acall  pintu5_turun
        ret
bd131:  cjne   a,#3,bd132
        acall  pintul_berhenti
        acall  pintu5_berhenti
bd132:  ret
;
bd14:   cjne   a,#4,bd15
        mov    a,data3
        cjne   a,#1,bd140
        acall  pintu2_naik
        acall  pintu3_naik
        ret
bd140:  cjne   a,#2,bd141
        acall  pintu2_turun
        acall  pintu3_turun
        ret
bd141:  cjne   a,#3,bd142
        acall  pintu2_berhenti
        acall  pintu3_berhenti
bd142:  ret
;
bd15:   cjne   a,#5,bd16
        mov    a,data3
        cjne   a,#1,bd150
        acall  pintu2_naik
        acall  pintu4_naik
        ret
bd150:  cjne   a,#2,bd151
        acall  pintu2_turun
        acall  pintu4_turun
        ret
bd151:  cjne   a,#3,bd152
        acall  pintu2_berhenti
        acall  pintu4_berhenti
bd152:  ret
;
bd16:   cjne   a,#6,bd17
        mov    a,data3
        cjne   a,#1,bd160
        acall  pintu2_naik
        acall  pintu5_naik
        ret
bd160:  cjne   a,#2,bd161
        acall  pintu2_turun
        acall  pintu5_turun
        ret
bd161:  cjne   a,#3,bd162
        acall  pintu2_berhenti
        acall  pintu5_berhenti
bd162:  ret
;
bd17:   cjne   a,#7,bd18
        mov    a,data3
        cjne   a,#1,bd170
        acall  pintu3_naik
        acall  pintu4_naik
        ret
bd170:  cjne   a,#2,bd171
        acall  pintu3_turun
        acall  pintu4_turun
        ret
bd171:  cjne   a,#3,bd172
        acall  pintu3_berhenti
        acall  pintu4_berhenti
bd172:  ret

```

```
; 3 motor (m1m2m3=0, m1m2m4=1, m1m2m5=2, m1m3m4=3, m1m3m5=4)  
; (m1m4m5=5, m2m3m4=6, m2m3m5=7, m2m4m5=8, m3m4m5=9)  
;
```

```
bd12: ljmp bd3  
bd2:  cjne a,#3,bd12  
      mov  a,data2  
      cjne a,#10,bd21  
      mov  a,data3  
      cjne a,#1,bd200  
      acall pintu1_naik  
      acall pintu2_naik  
      acall pintu3_naik  
      ret
```

```
bd200: cjne a,#2,bd201  
       acall pintu1_turun  
       acall pintu2_turun  
       acall pintu3_turun  
       ret
```

```
bd201: cjne a,#3,bd202  
       acall pintu1_berhenti  
       acall pintu2_berhenti  
       acall pintu3_berhenti
```

```
bd202: ret  
;
```

```
bd21:  cjne a,#1,bd22  
       mov  a,data3  
       cjne a,#1,bd210  
       acall pintu1_naik  
       acall pintu2_naik  
       acall pintu4_naik  
       ret
```

```
bd210: cjne a,#2,bd211  
       acall pintu1_turun  
       acall pintu2_turun  
       acall pintu4_turun  
       ret
```

```
bd211: cjne a,#3,bd212  
       acall pintu1_berhenti  
       acall pintu2_berhenti  
       acall pintu4_berhenti
```

```
bd212: ret  
;
```

```
bd22:  cjne a,#2,bd23  
       mov  a,data3  
       cjne a,#1,bd220  
       acall pintu1_naik  
       acall pintu2_naik  
       acall pintu5_naik  
       ret
```

```
bd220: cjne a,#2,bd221  
       acall pintu1_turun  
       acall pintu2_turun  
       acall pintu5_turun  
       ret
```

```
bd221: cjne a,#3,bd222  
       acall pintu1_berhenti  
       acall pintu2_berhenti  
       acall pintu5_berhenti
```

```
bd222: ret  
;
```

```
bd23:  cjne a,#3,bd24  
       mov  a,data3  
       cjne a,#1,bd230  
       acall pintu1_naik  
       acall pintu3_naik  
       acall pintu4_naik  
       ret
```

```
bd230: cjne a,#2,bd231  
       acall pintu1_turun  
       acall pintu3_turun
```

```

        acall pintu4_turun
        ret
bd231:  cjne   a,#3,bd232
        acall pintu_berhenti
        acall pintu3_berhenti
        acall pintu4_berhenti
bd232:  ret
;-----
bd24:   cjne   a,#4,bd25
        mov    a,data3
        cjne   a,#1,bd240
        acall pintu1_naik
        acall pintu3_naik
        acall pintu5_naik
        ret
bd240:  cjne   a,#2,bd241
        acall pintu1_turun
        acall pintu3_turun
        acall pintu5_turun
        ret
bd241:  cjne   a,#3,bd242
        acall pintu1_berhenti
        acall pintu3_berhenti
        acall pintu5_berhenti
bd242:  ret
;-----
bd25:   cjne   a,#5,bd26
        mov    a,data3
        cjne   a,#1,bd250
        acall pintu1_naik
        acall pintu4_naik
        acall pintu5_naik
        ret
bd250:  cjne   a,#2,bd251
        acall pintu1_turun
        acall pintu4_turun
        acall pintu5_turun
        ret
bd251:  cjne   a,#3,bd252
        acall pintu1_berhenti
        acall pintu4_berhenti
        acall pintu5_berhenti
bd252:  ret
;-----
bd26:   cjne   a,#6,bd27
        mov    a,data3
        cjne   a,#1,bd260
        acall pintu2_naik
        acall pintu3_naik
        acall pintu4_naik
        ret
bd260:  cjne   a,#2,bd261
        acall pintu2_turun
        acall pintu3_turun
        acall pintu4_turun
        ret
bd261:  cjne   a,#3,bd262
        acall pintu2_berhenti
        acall pintu3_berhenti
        acall pintu4_berhenti
bd262:  ret
;-----
bd27:   cjne   a,#7,bd28
        mov    a,data3
        cjne   a,#1,bd270
        acall pintu2_naik
        acall pintu3_naik
        acall pintu5_naik
        ret
bd270:  cjne   a,#2,bd271
        acall pintu2_turun

```

```

        acall pintu3_turun
        acall pintu5_turun
        ret
bd271:  cjne   a,#3,bd272
        acall pintu3_berhenti
        acall pintu2_berhenti
        acall pintu5_berhenti
bd272:  ret
;=====
bd28:   cjne   a,#7,bd29
        mov    a,data3
        cjne   a,#1,bd280
        acall pintu2_naik
        acall pintu4_naik
        acall pintu5_naik
        ret
bd280:  cjne   a,#2,bd281
        acall pintu2_turun
        acall pintu4_turun
        acall pintu5_turun
        ret
bd281:  cjne   a,#3,bd282
        acall pintu2_berhenti
        acall pintu4_berhenti
        acall pintu5_berhenti
bd282:  ret
;=====
bd29:   cjne   a,#7,bd292
        mov    a,data3
        cjne   a,#1,bd290
        acall pintu3_naik
        acall pintu4_naik
        acall pintu5_naik
        ret
bd290:  cjne   a,#2,bd291
        acall pintu3_turun
        acall pintu4_turun
        acall pintu5_turun
        ret
bd291:  cjne   a,#3,bd292
        acall pintu3_berhenti
        acall pintu4_berhenti
        acall pintu5_berhenti
bd292:  ret
;=====
; 4 motor (m1m2m3m4=0, m1m2m3m5=1, m1m2m4m5=2, m1m3m4m5=3)
;          (m2m3m4m5=4)
;=====
bd13:   ljmp   bd4
bd3:    cjne   a,#4,bd13
        mov    a,data2
        cjne   a,#10,bd31
        mov    a,data3
        cjne   a,#1,bd300
        acall pintu1_naik
        acall pintu2_naik
        acall pintu3_naik
        acall pintu4_naik
        ret
bd300:  cjne   a,#2,bd301
        acall pintu1_turun
        acall pintu2_turun
        acall pintu3_turun
        acall pintu4_turun
        ret
bd301:  cjne   a,#3,bd302
        acall pintu1_berhenti
        acall pintu2_berhenti
        acall pintu3_berhenti
        acall pintu4_berhenti
bd302:  ret

```

```

;
bd31:  cjne  a,#1,bd32
       mov  a,data3
       cjne a,#1,bd310
       acall pintu1_naik
       acall pintu2_naik
       acall pintu3_naik
       acall pintu5_naik
       ret
bd310: cjne  a,#2,bd311
       acall pintu1_turun
       acall pintu2_turun
       acall pintu3_turun
       acall pintu5_turun
       ret
bd311: cjne  a,#3,bd312
       acall pintu1_berhenti
       acall pintu2_berhenti
       acall pintu3_berhenti
       acall pintu5_berhenti
bd312: ret
;

```

```

bd32:  cjne  a,#2,bd33
       mov  a,data3
       cjne a,#1,bd320
       acall pintu1_naik
       acall pintu2_naik
       acall pintu4_naik
       acall pintu5_naik
       ret
bd320: cjne  a,#2,bd321
       acall pintu1_turun
       acall pintu2_turun
       acall pintu4_turun
       acall pintu5_turun
       ret
bd321: cjne  a,#3,bd322
       acall pintu1_berhenti
       acall pintu2_berhenti
       acall pintu4_berhenti
       acall pintu5_berhenti
bd322: ret
;

```

```

bd33:  cjne  a,#3,bd34
       mov  a,data3
       cjne a,#1,bd330
       acall pintu1_naik
       acall pintu3_naik
       acall pintu4_naik
       acall pintu5_naik
       ret
bd330: cjne  a,#2,bd331
       acall pintu1_turun
       acall pintu3_turun
       acall pintu4_turun
       acall pintu5_turun
       ret
bd331: cjne  a,#3,bd332
       acall pintu1_berhenti
       acall pintu3_berhenti
       acall pintu4_berhenti
       acall pintu5_berhenti
bd332: ret
;

```

```

bd34:  cjne  a,#4,bd342
       mov  a,data3
       cjne a,#1,bd340
       acall pintu2_naik
       acall pintu3_naik
       acall pintu4_naik
       acall pintu5_naik

```



```

ret
bd340: cjne a,#2,bd341
        acall pintu2_turun
        acall pintu3_turun
        acall pintu4_turun
        acall pintu5_turun
ret
bd341: cjne a,#3,bd342
        acall pintu2_berhenti
        acall pintu3_berhenti
        acall pintu4_berhenti
        acall pintu5_berhenti
bd342: ret

```

```

; 5 motor (m1m2m3m4m5=1)

```

```

bd4:    cjne a,#5,bd5
        mov a,data2
        cjne a,#10,bd5
        mov a,data3
        cjne a,#1,bd500
        acall pintu1_naik
        acall pintu2_naik
        acall pintu3_naik
        acall pintu4_naik
        acall pintu5_naik
bd5:    ret
bd500:  cjne a,#2,bd501
        acall pintu1_turun
        acall pintu2_turun
        acall pintu3_turun
        acall pintu4_turun
        acall pintu5_turun
ret
bd501:  cjne a,#3,bd502
        acall pintu1_berhenti
        acall pintu2_berhenti
        acall pintu3_berhenti
        acall pintu4_berhenti
        acall pintu5_berhenti
bd502:  ret

```

```

;
delay:  mov r2,#05h
delay1: mov r3,#0ffh
delay2: mov r4,#0ffh
        djnz r4,$
        djnz r3,delay2
        djnz r2,delay1
ret

```

```

tunda:  mov r1,#255
        djnz r1,$
ret

```

```

pintul_turun:
setb    pintul_bawah
setb    pintul_atas
acall   delay
clr     pintul_bawah
mov     pintul,#2
ret

```

```

pintul_naik:
setb    pintul_atas
setb    pintul_bawah
acall   delay
clr     pintul_atas
mov     pintul,#1
ret

```

```

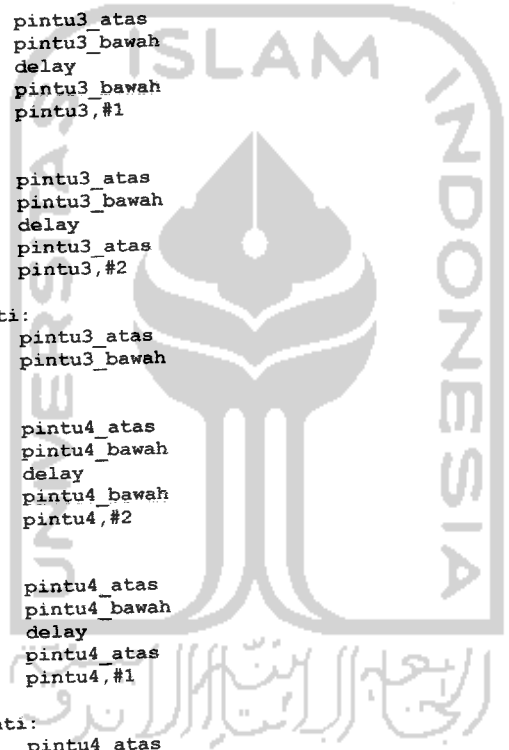
pintul_berhenti:

```

```

        setb    pintu1_atas
        setb    pintu1_bawah
        ret
pintu2_naik:
        setb    pintu2_atas
        setb    pintu2_bawah
        acall   delay
        clr     pintu2_bawah
        mov     pintu2,#1
        ret
pintu2_turun:
        setb    pintu2_atas
        setb    pintu2_bawah
        acall   delay
        clr     pintu2_atas
        mov     pintu2,#2
        ret
pintu2_berhenti:
        setb    pintu2_atas
        setb    pintu2_bawah
        ret
pintu3_naik:
        setb    pintu3_atas
        setb    pintu3_bawah
        acall   delay
        clr     pintu3_bawah
        mov     pintu3,#1
        ret
pintu3_turun:
        setb    pintu3_atas
        setb    pintu3_bawah
        acall   delay
        clr     pintu3_atas
        mov     pintu3,#2
        ret
pintu3_berhenti:
        setb    pintu3_atas
        setb    pintu3_bawah
        ret
pintu4_turun:
        setb    pintu4_atas
        setb    pintu4_bawah
        acall   delay
        clr     pintu4_bawah
        mov     pintu4,#2
        ret
pintu4_naik:
        setb    pintu4_atas
        setb    pintu4_bawah
        acall   delay
        clr     pintu4_atas
        mov     pintu4,#1
        ret
pintu4_berhenti:
        setb    pintu4_atas
        setb    pintu4_bawah
        ret
pintu5_turun:
        setb    pintu5_atas
        setb    pintu5_bawah
        acall   delay
        clr     pintu5_bawah
        mov     pintu5,#2
        ret
pintu5_naik:
        setb    pintu5_atas
        setb    pintu5_bawah
        acall   delay
        clr     pintu5_atas
        mov     pintu5,#1
        ret

```



```
pintu5_berhenti:  
    setb pintu5_atas  
    setb pintu5_bawah  
    ret  
end
```



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)
- Green (Pb/Halide-free) Packaging Option

1. 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

2487C-MICRO-03/05





2. Pin Configurations

2.1 40-lead PDIP

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
(MOSI) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

2.3 44-lead PLCC

(MOSI) P1.5	7	39	P0.4 (AD4)
(MISO) P1.6	8	38	P0.5 (AD5)
(SCK) P1.7	9	37	P0.6 (AD6)
RST	10	36	P0.7 (AD7)
(RXD) P3.0	11	35	EA/VPP
NC	12	34	NC
(TXD) P3.1	13	33	ALE/PROG
(INT0) P3.2	14	32	PSEN
(INT1) P3.3	15	31	P2.7 (A15)
(T0) P3.4	16	30	P2.6 (A14)
(T1) P3.5	17	29	P2.5 (A13)
(WR) P3.6	18	28	P2.4 (A12)
(RD) P3.7	19	27	P2.3 (A11)
XTAL2	20	26	P2.2 (A10)
XTAL1	21	25	P2.1 (A9)
GND	22	24	P2.0 (A8)
NC	23	23	NC
(A8) P2.0	24	24	P0.0 (AD0)
(A9) P2.1	25	25	P0.1 (AD1)
(A10) P2.2	26	26	P0.2 (AD2)
(A11) P2.3	27	27	P0.3 (AD3)
(A12) P2.4	28	28	P0.4 (AD4)
(A13) P2.5	29	29	P0.5 (AD5)
(A14) P2.6	30	30	P0.6 (AD6)
(A15) P2.7	31	31	P0.7 (AD7)
P1.0	32	32	VCC
P1.1	33	33	P0.0 (AD0)
P1.2	34	34	P0.1 (AD1)
P1.3	35	35	P0.2 (AD2)
P1.4	36	36	P0.3 (AD3)
P1.5	37	37	P0.4 (AD4)
P1.6	38	38	P0.5 (AD5)
P1.7	39	39	P0.6 (AD6)
P1.8	40	40	P0.7 (AD7)
P1.9	41	41	VCC
P1.10	42	42	P0.0 (AD0)
P1.11	43	43	P0.1 (AD1)
P1.12	44	44	P0.2 (AD2)
P1.13	45	45	P0.3 (AD3)
P1.14	46	46	P0.4 (AD4)
P1.15	47	47	P0.5 (AD5)
P1.16	48	48	P0.6 (AD6)
P1.17	49	49	P0.7 (AD7)
P1.18	50	50	VCC

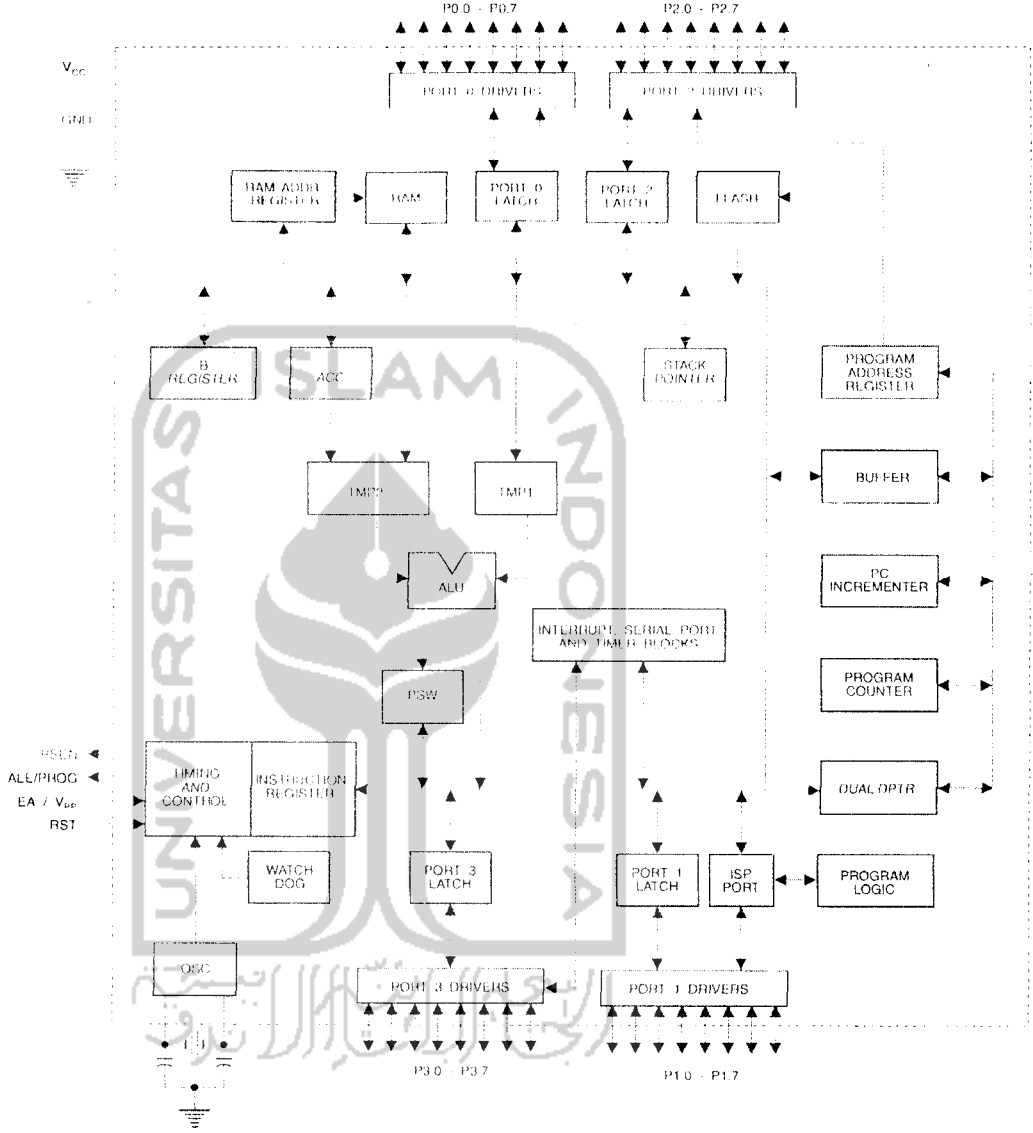
2.2 44-lead TQFP

(MOSI) P1.5	1	33	P0.4 (AD4)
(MISO) P1.6	2	32	P0.5 (AD5)
(SCK) P1.7	3	31	P0.6 (AD6)
RST	4	30	P0.7 (AD7)
(RXD) P3.0	5	29	EA/VPP
NC	6	28	NC
(TXD) P3.1	7	27	ALE/PROG
(INT0) P3.2	8	26	PSEN
(INT1) P3.3	9	25	P2.7 (A15)
(T0) P3.4	10	24	P2.6 (A14)
(T1) P3.5	11	23	P2.5 (A13)
(WR) P3.6	12	22	P2.4 (A12)
(RD) P3.7	13	21	P2.3 (A11)
XTAL2	14	20	P2.2 (A10)
XTAL1	15	19	P2.1 (A9)
GND	16	18	P2.0 (A8)
(A8) P2.0	17	17	P0.0 (AD0)
(A9) P2.1	18	18	P0.1 (AD1)
(A10) P2.2	19	19	P0.2 (AD2)
(A11) P2.3	20	20	P0.3 (AD3)
(A12) P2.4	21	21	P0.4 (AD4)
(A13) P2.5	22	22	P0.5 (AD5)
(A14) P2.6	23	23	P0.6 (AD6)
(A15) P2.7	24	24	P0.7 (AD7)
P1.0	25	25	VCC
P1.1	26	26	P0.0 (AD0)
P1.2	27	27	P0.1 (AD1)
P1.3	28	28	P0.2 (AD2)
P1.4	29	29	P0.3 (AD3)
P1.5	30	30	P0.4 (AD4)
P1.6	31	31	P0.5 (AD5)
P1.7	32	32	P0.6 (AD6)
P1.8	33	33	P0.7 (AD7)
P1.9	34	34	VCC
P1.10	35	35	P0.0 (AD0)
P1.11	36	36	P0.1 (AD1)
P1.12	37	37	P0.2 (AD2)
P1.13	38	38	P0.3 (AD3)
P1.14	39	39	P0.4 (AD4)
P1.15	40	40	P0.5 (AD5)
P1.16	41	41	P0.6 (AD6)
P1.17	42	42	P0.7 (AD7)
P1.18	43	43	VCC
P1.19	44	44	P0.0 (AD0)
P1.20	45	45	P0.1 (AD1)
P1.21	46	46	P0.2 (AD2)
P1.22	47	47	P0.3 (AD3)
P1.23	48	48	P0.4 (AD4)
P1.24	49	49	P0.5 (AD5)
P1.25	50	50	P0.6 (AD6)
P1.26	51	51	P0.7 (AD7)
P1.27	52	52	VCC

2.4 42-lead PDIP

RST	1	42	P1.7 (SCK)
(RXD) P3.0	2	41	P1.6 (MISO)
(TXD) P3.1	3	40	P1.5 (MOSI)
(INT0) P3.2	4	39	P1.4
(INT1) P3.3	5	38	P1.3
(T0) P3.4	6	37	P1.2
(T1) P3.5	7	36	P1.1
(WR) P3.6	8	35	P1.0
(RD) P3.7	9	34	VDD
XTAL2	10	33	PWRVDD
XTAL1	11	32	P0.0 (AD0)
GND	12	31	P0.1 (AD1)
PWRGND	13	30	P0.2 (AD2)
(A8) P2.0	14	29	P0.3 (AD3)
(A9) P2.1	15	28	P0.4 (AD4)
(A10) P2.2	16	27	P0.5 (AD5)
(A11) P2.3	17	26	P0.6 (AD6)
(A12) P2.4	18	25	P0.7 (AD7)
(A13) P2.5	19	24	EA/VPP
(A14) P2.6	20	23	ALE/PROG
(A15) P2.7	21	22	PSEN

3. Block Diagram





4. Pin Description

4.1 VCC

Supply voltage (all packages except 42-PDIP).

4.2 GND

Ground (all packages except 42-PDIP; for 42-PDIP GND connects only the logic core and the embedded program memory).

4.3 VDD

Supply voltage for the 42-PDIP which connects only the logic core and the embedded program memory.

4.4 PWRVDD

Supply voltage for the 42-PDIP which connects only the I/O Pad Drivers. The application board **MUST** connect both VDD and PWRVDD to the board supply voltage.

4.5 PWRGND

Ground for the 42-PDIP which connects only the I/O Pad Drivers. PWRGND and GND are weakly connected through the common silicon substrate, but not through any metal link. The application board **MUST** connect both GND and PWRGND to the board ground.

4.6 Port 0

Port 0 is an 8-bit open drain bi-directional 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.**

4.7 Port 1

Port 1 is an 8-bit bi-directional 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)

4.8 Port 2

Port 2 is an 8-bit bi-directional 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.

4.9 Port 3

Port 3 is an 8-bit bi-directional 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	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

4.10 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.

4.11 ALE/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 (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.

4.12 $\overline{\text{PSEN}}$

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

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

4.13 $\overline{\text{EA/VPP}}$

External Access Enable. $\overline{\text{EA}}$ 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, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ 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.

4.14 XTAL1

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

4.15 XTAL2

Output from the inverting oscillator amplifier

5. Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 5-1.

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 5-1. AT89S51 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0E7H
0E0H	ACC 00000000							0E7H
0D8H								0D7H
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				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 5-2. AUXR: Auxiliary Register

AUXR	Address = 8EH				Reset Value = XXX00XX0B			
Not Bit Addressable								
Bit	7	6	5	WDIDLE	DISRTO	2	1	DISALE
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 5-3. AUXR1: Auxiliary Register 1

AUXR1	Address = A2H							Reset Value = XXXXXX0B
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						

6. 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.

6.1 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.

6.2 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.

7. 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.

7.1 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 $98 \times TOSC$, 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.

7.2 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.

8. UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

9. 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, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

10. 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 10-1.

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.

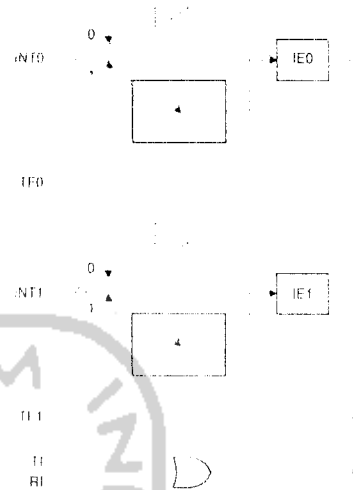
Note that Table 10-1 shows that bit positions IE.6 and IE.5 are unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

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 10-1. 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 10-1. Interrupt Sources



11. Oscillator Characteristics

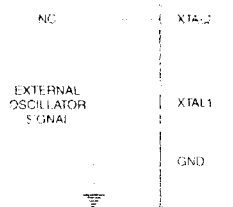
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11-1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 11-2. 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 11-1. Oscillator Connections



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

Figure 11-2. External Clock Drive Configuration



12. 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.

13. 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 ($\overline{INT0}$ 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 13-1. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	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



14. 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 Table 14-1.

Table 14-1. Lock Bit Protection Modes

Program Lock Bits				Protection Type
LB1	LB2	LB3		
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.

15. 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 Modes table (Table 17-1) and Figure 17-1 and Figure 17-2. 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/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 μ s. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

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/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.0 is pulled low after ALE goes high during programming to indicate \overline{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 AT89S51
 {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.

16. 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.

16.1 Serial Programming Algorithm

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

1. Power-up sequence
 - a. Apply power between VCC and GND pins.
 - b. 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.



- 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.

16.2 Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in the "Serial Programming Instruction Set" on page 20.

17. 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.

Most major worldwide programming vendors offer worldwide support for the Atmel AT89 micro-controller series. Please contact your local programming vendor for the appropriate software revision.

Table 17-1. 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	P1.7-0
												Address	
Write Code Data	5V	H	L		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		12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L		12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L		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		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:
- Each \overline{PROG} pulse is 200 ns - 500 ns for Chip Erase.
 - Each \overline{PROG} pulse is 200 ns - 500 ns for Write Code Data.
 - Each \overline{PROG} pulse is 200 ns - 500 ns for Write Lock Bits.
 - $\overline{RDY/BSY}$ signal is output on P3.0 during programming.
 - X = don't care.

Figure 17-1. Programming the Flash Memory (Parallel Mode)

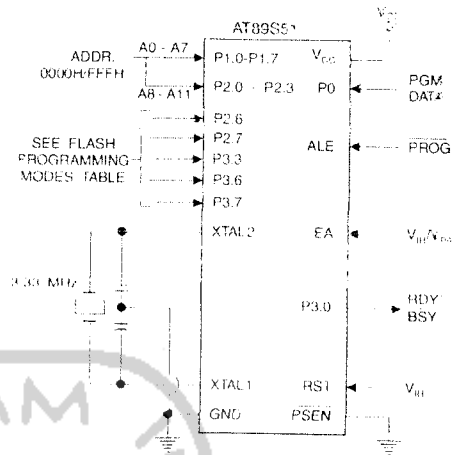
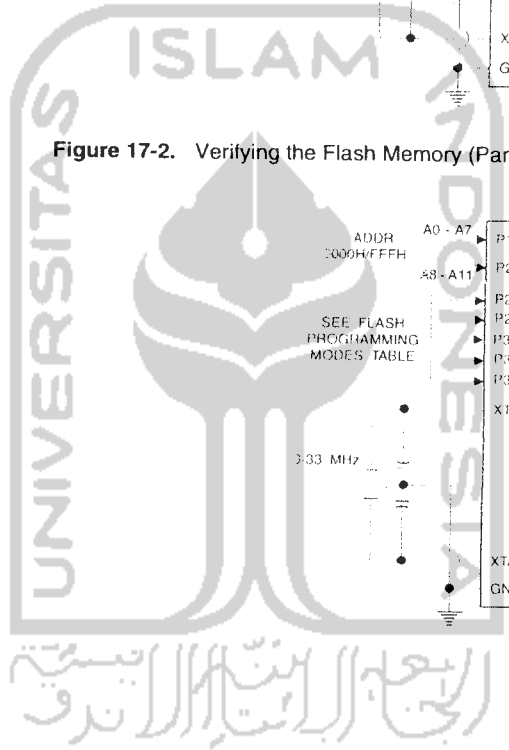
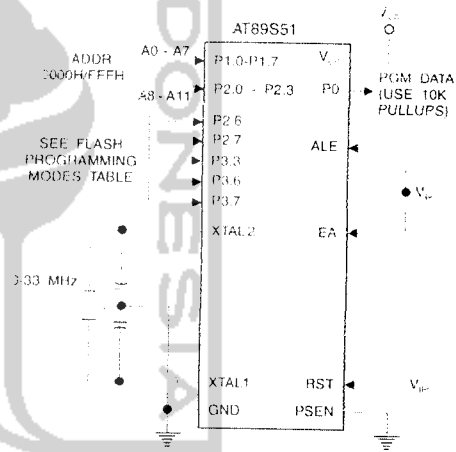


Figure 17-2. Verifying the Flash Memory (Parallel Mode)



18. Flash Programming and Verification Characteristics (Parallel Mode)

$T_A = 20^\circ\text{C}$ 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 $\overline{\text{PROG}}$ Low	$48 t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48 t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48 t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48 t_{CLCL}$		
t_{EHS}	P2.7 (ENABLE) High to V_{PP}	$48 t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	μs
t_{AVQV}	Address to Data Valid		$48 t_{CLCL}$	
t_{ELOV}	ENABLE Low to Data Valid		$48 t_{CLCL}$	
t_{EHOZ}	Data Float After ENABLE	0	$48 t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		50	μs

Figure 18-1. Flash Programming and Verification Waveforms – Parallel Mode

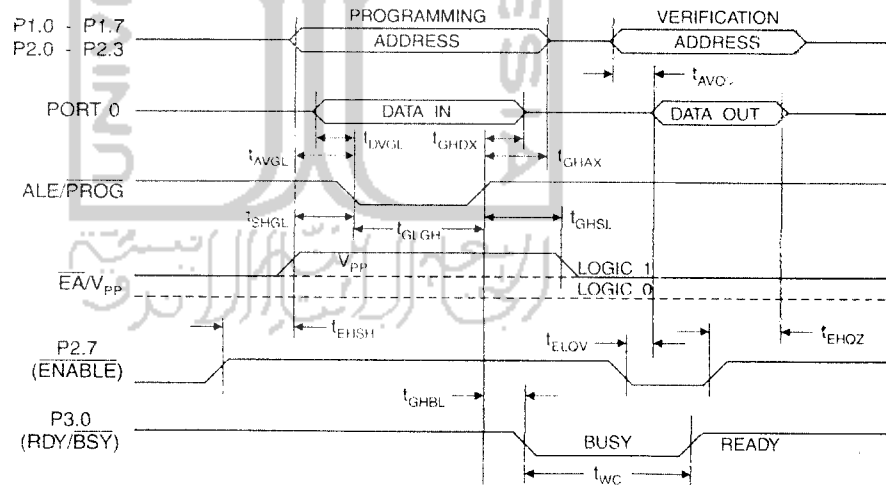
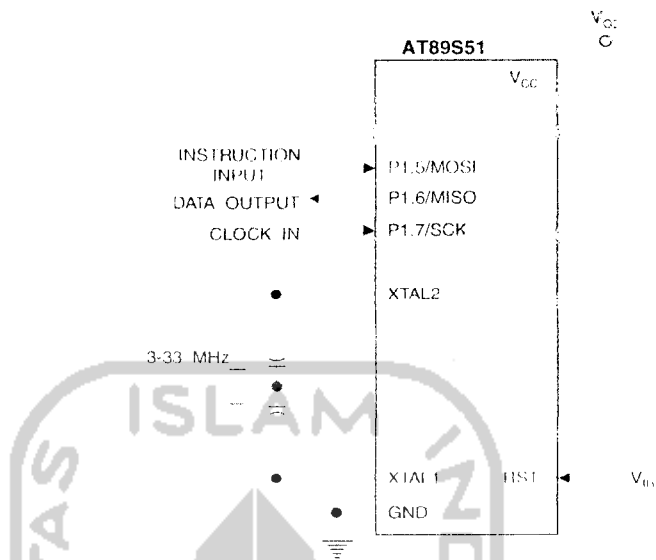
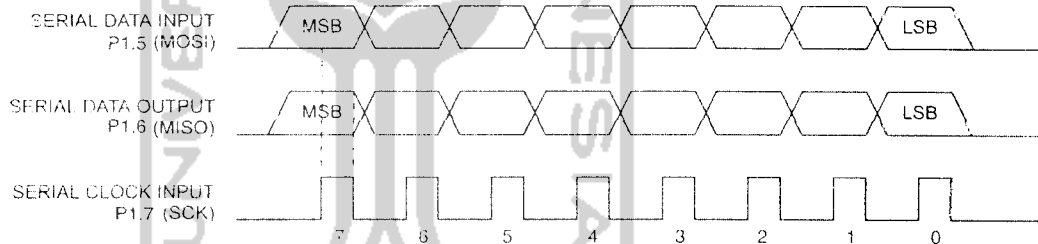


Figure 18-2. Flash Memory Serial Downloading



19. Flash Programming and Verification Waveforms – Serial Mode

Figure 19-1. Serial Programming Waveforms





20. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	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 ⁽¹⁾	1010 1100	1110 00B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxx B3 B2 B1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxxx A11 A10 A9 A8	A7 xxx xxx0	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)

Note: 1. 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, B2 = 1 → Mode 4, lock bit 3 activated

Each of the lock bit modes need 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.

21. Serial Programming Characteristics

Figure 21-1. Serial Programming Timing

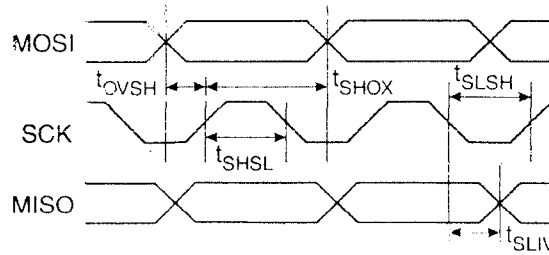


Table 21-1. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	3		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

22. 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.

23. DC Characteristics

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

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

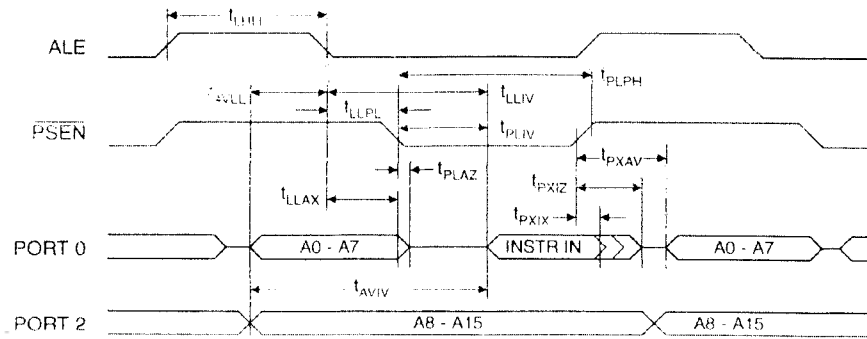
24. AC Characteristics

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

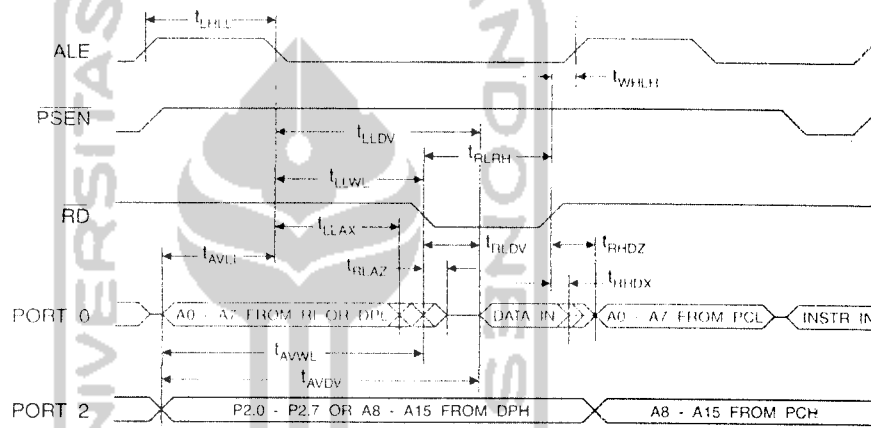
24.1 External Program and Data Memory Characteristics

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

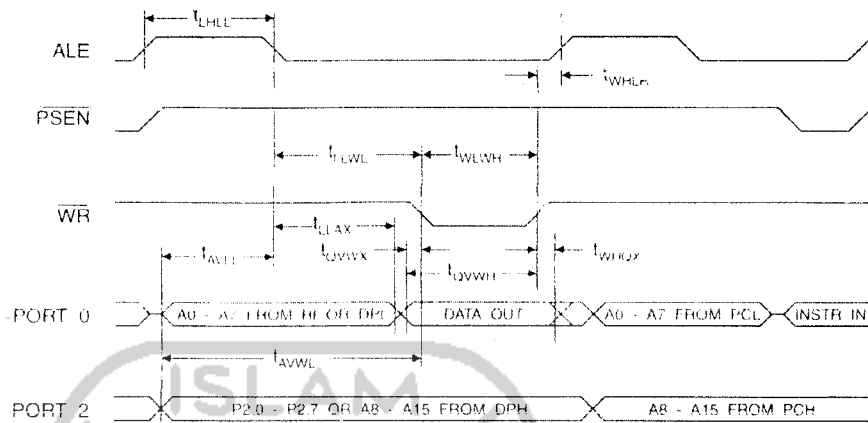
25. External Program Memory Read Cycle



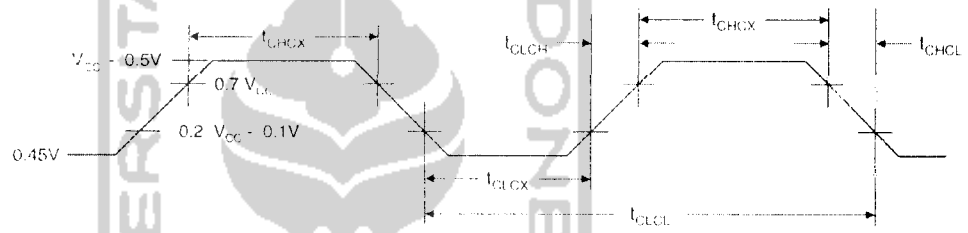
26. External Data Memory Read Cycle



27. External Data Memory Write Cycle



28. External Clock Drive Waveforms



29. 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

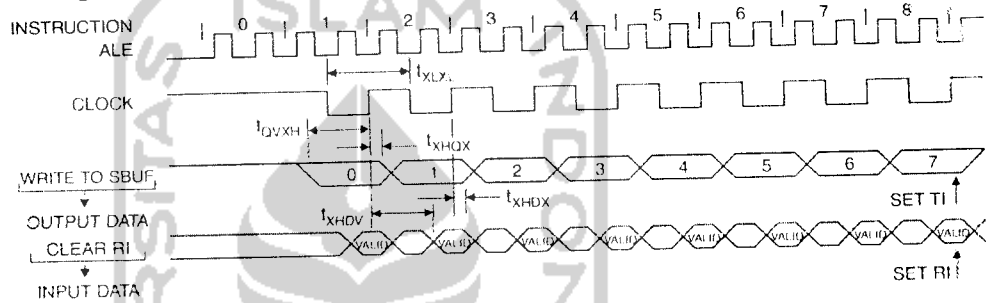


30. Serial Port Timing: Shift Register Mode Test Conditions

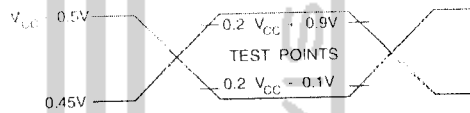
The values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = 80 pF .

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

31. Shift Register Mode Timing Waveforms

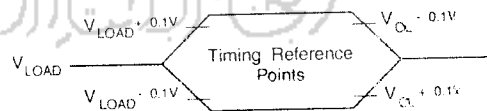


32. AC Testing Input/Output Waveforms⁽¹⁾



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.

33. 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.

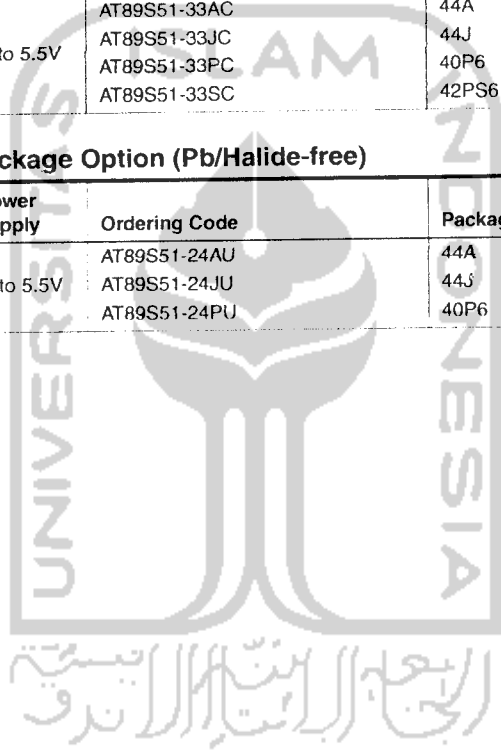
34. Ordering Information

34.1 Standard Package

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range		
24	4.0V to 5.5V	AT89S51-24AC	44A	Commercial (0°C to 70°C)		
		AT89S51-24JC	44J			
		AT89S51-24PC	40P6			
		AT89S51-24SC	42PS6			
		33	4.5V to 5.5V	AT89S51-24AI	44A	Industrial (-40°C to 85°C)
				AT89S51-24JI	44J	
				AT89S51-24PI	40P6	
				AT89S51-24SI	42PS6	
33	4.5V to 5.5V	AT89S51-33AC	44A	Commercial (0°C to 70°C)		
		AT89S51-33JC	44J			
		AT89S51-33PC	40P6			
		AT89S51-33SC	42PS6			

34.2 Green Package Option (Pb/Halide-free)

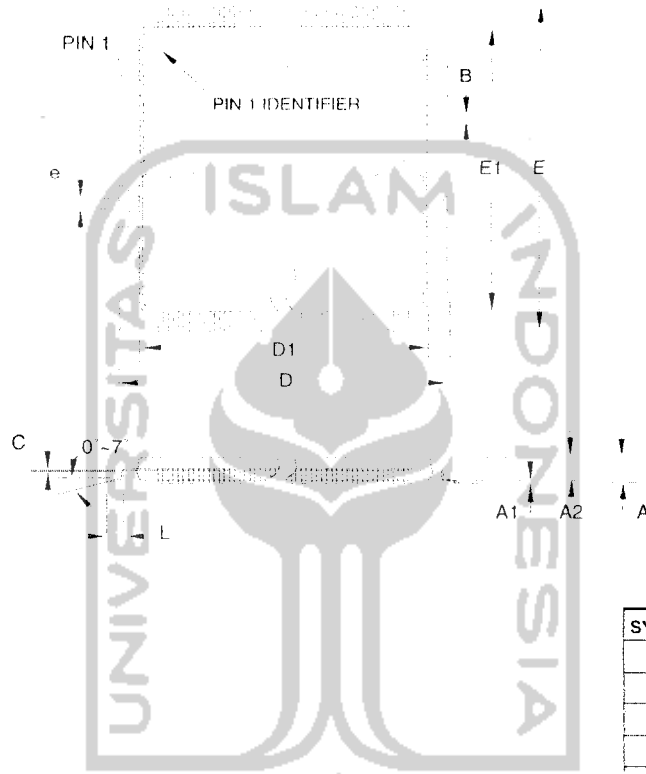
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AU	44A	Industrial (-40°C to 85°C)
		AT89S51-24JU	44J	
		AT89S51-24PU	40P6	



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)
42PS6	42-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

35. Packaging Information

35.1 44A – TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

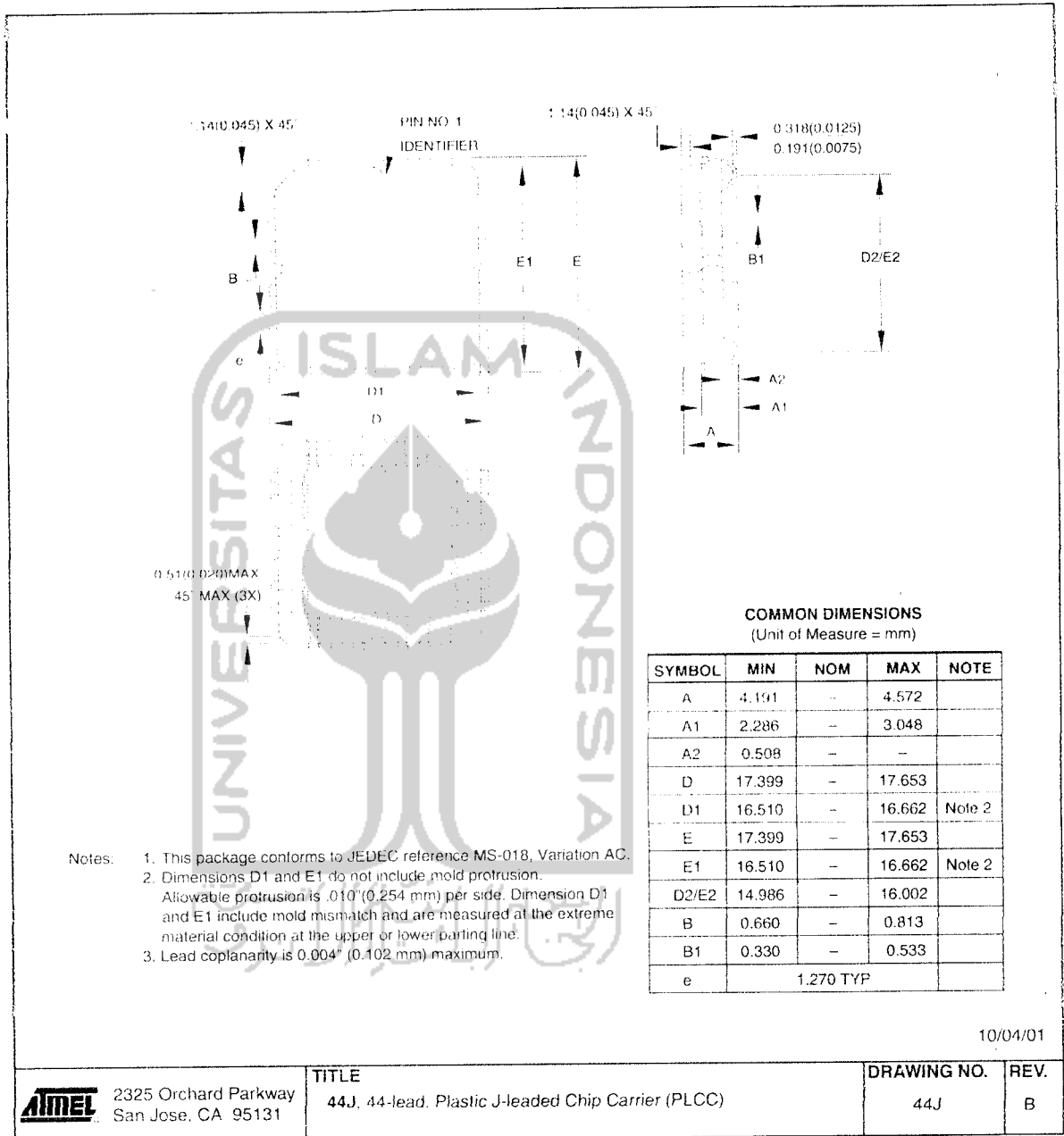
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

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

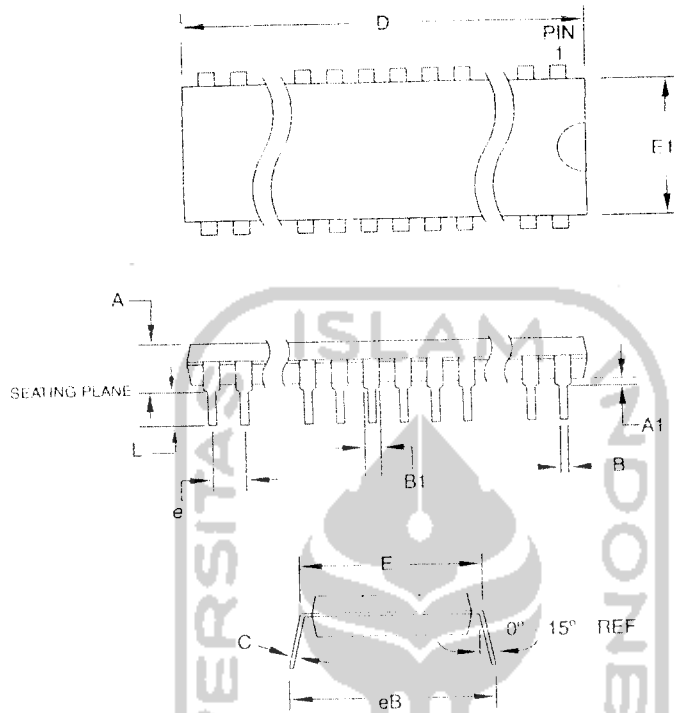
10/5/2001

2325 Orchard Parkway San Jose, CA 95131	TITLE	DRAWING NO.	REV.
	44A , 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	44A	B

35.2 44J – PLCC



35.3 40P6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

2325 Orchard Parkway San Jose, CA 95131	TITLE 40P6 , 40-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	DRAWING NO.	REV.
		40P6	B

35.4 42PS6 – PDIP

ISLAM UNIVERSITY INDONESIA

UNIVERSITAS ISLAM INDONESIA

Notes:

1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.83	
A1	0.51	-	-	
D	36.70	-	36.96	Note 2
E	15.24	-	15.88	
E1	13.46	-	13.97	Note 2
B	0.38	-	0.56	
B1	0.76	-	1.27	
L	3.05	-	3.43	
C	0.20	-	0.30	
eB	-	-	18.55	
e	1.78 TYP			

11/6/03

2325 Orchard Parkway San Jose, CA 95131	TITLE	DRAWING NO.	REV.
	42PS6, 42-lead (0.600"/15.24 mm Wide) Plastic Dual In-line Package (PDIP)	42PS6	A



Atmel Corporation

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

Regional Headquarters

Europe

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

Asia

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

Japan

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

Atmel Operations

Memory

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

Microcontrollers

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

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

ASIC/ASSP/Smart Cards

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

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

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

RF/Automotive

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

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

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

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

Literature Requests

www.atmel.com/literature

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

© Atmel Corporation 2005. All rights reserved. Atmel®, logo and combinations thereof, and others, are registered trademarks, and Everywhere You AreSM and others are the trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Printed on recycled paper.

2487C-MICRO-03/05 /xM

LM386 Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.

The inputs are ground referenced while the output automatically biases to one-half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

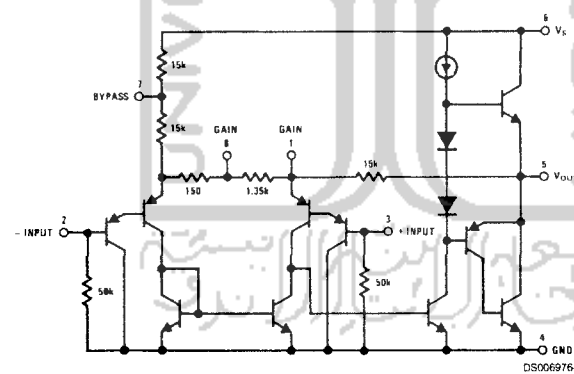
Features

- Battery operation
- Minimum external parts
- Wide supply voltage range: 4V–12V or 5V–18V
- Low quiescent current drain: 4mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion: 0.2% ($A_v = 20$, $V_{CC} = 6V$, $R_L = 8\Omega$, $P_{O1} = 125mW$, $f = 1kHz$)
- Available in 8 pin MSOP package

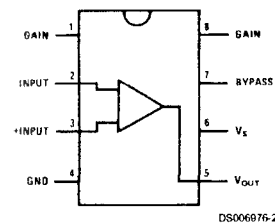
Applications

- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams



Small Outline,
Molded Mini Small Outline,
and Dual-In-Line Packages



Top View
Order Number LM386M-1,
LM386MM-1, LM386N-1,
LM386N-3 or LM386N-4
See NS Package Number
M08A, MUA08A or N08E

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (LM386N-1, -3, LM386M-1)	15V
Supply Voltage (LM386N-4)	22V
Package Dissipation (Note 3) (LM386N)	1.25W
(LM386M)	0.73W
(LM386MM-1)	0.595W
Input Voltage	±0.4V
Storage Temperature	-65°C to +150°C
Operating Temperature	0°C to +70°C
Junction Temperature	+150°C
Soldering Information	

Dual-In-Line Package

Soldering (10 sec)

+260°C

Small Outline Package

(SOIC and MSOP)

Vapor Phase (60 sec)

+215°C

Infrared (15 sec)

+220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Thermal Resistance

 θ_{JC} (DIP)

37 °C/W

 θ_{JA} (DIP)

107 °C/W

 θ_{JC} (SO Package)

35 °C/W

 θ_{JA} (SO Package)

172 °C/W

 θ_{JA} (MSOP)

210 °C/W

 θ_{JC} (MSOP)

56 °C/W

Electrical Characteristics (Notes 1, 2) $T_A = 25^\circ\text{C}$

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage (V_S)					
LM386N-1, -3, LM386M-1, LM386MM-1		4		12	V
LM386N-4		5		18	V
Quiescent Current (I_Q)	$V_S = 6\text{V}, V_{IN} = 0$		4	8	mA
Output Power (P_{OUT})					
LM386N-1, LM386M-1, LM386MM-1	$V_S = 6\text{V}, R_L = 8\Omega, \text{THD} = 10\%$	250	325		mW
LM386N-3	$V_S = 9\text{V}, R_L = 8\Omega, \text{THD} = 10\%$	500	700		mW
LM386N-4	$V_S = 16\text{V}, R_L = 32\Omega, \text{THD} = 10\%$	700	1000		mW
Voltage Gain (A_V)	$V_S = 6\text{V}, f = 1\text{kHz}$		26		dB
	10 μF from Pin 1 to B		46		dB
Bandwidth (BW)	$V_S = 6\text{V}, \text{Pins 1 and 8 Open}$		300		kHz
Total Harmonic Distortion (THD)	$V_S = 6\text{V}, R_L = 8\Omega, P_{OUT} = 125\text{mW}$ $f = 1\text{kHz}, \text{Pins 1 and 8 Open}$		0.2		%
Power Supply Rejection Ratio (PSRR)	$V_S = 6\text{V}, f = 1\text{kHz}, C_{BYPASS} = 10\mu\text{F}$ Pins 1 and 8 Open, Referred to Output		50		dB
Input Resistance (R_{IN})			50		k Ω
Input Bias Current (I_{IB})	$V_S = 6\text{V}, \text{Pins 2 and 3 Open}$		250		nA

Note 1: All voltages are measured with respect to the ground pin, unless otherwise specified.

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and 1) a thermal resistance of 107°C/W junction to ambient for the dual in-line package and 2) a thermal resistance of 172°C/W for the small outline package.

Application Hints

GAIN CONTROL

To make the LM386 a more versatile amplifier, two pins (1 and 8) are provided for gain control. With pins 1 and 8 open the 1.35 k Ω resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35 k Ω resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitively coupling a resistor (or FET) from pin 1 to ground.

Additional external components can be placed in parallel with the internal feedback resistors to tailor the gain and frequency response for individual applications. For example, we can compensate poor speaker bass response by frequency shaping the feedback path. This is done with a series RC from pin 1 to 5 (paralleling the internal 15 k Ω resistor). For 6 dB effective bass boost: $R = 15$ k Ω , the lowest value for good stable operation is $R = 10$ k Ω if pin 8 is open. If pins 1 and 8 are bypassed then R as low as 2 k Ω can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

INPUT BIASING

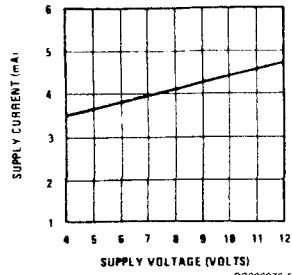
The schematic shows that both inputs are biased to ground with a 50 k Ω resistor. The base current of the input transistors is about 250 nA, so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than 250 k Ω it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than 10 k Ω , then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

When using the LM386 with higher gains (bypassing the 1.35 k Ω resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1 μ F capacitor or a short to ground depending on the dc source resistance on the driven input.

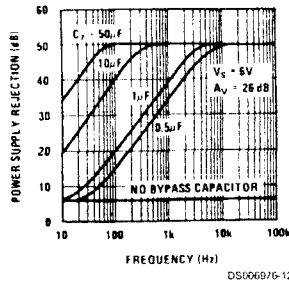


Typical Performance Characteristics

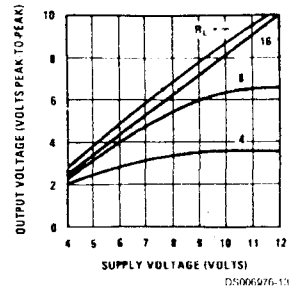
Quiescent Supply Current vs Supply Voltage



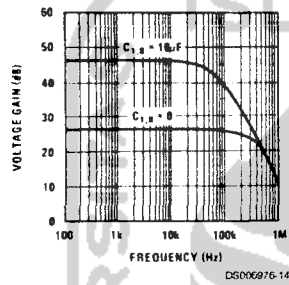
Power Supply Rejection Ratio (Referred to the Output) vs Frequency



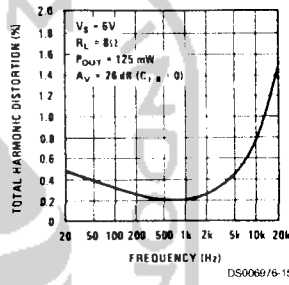
Peak-to-Peak Output Voltage Swing vs Supply Voltage



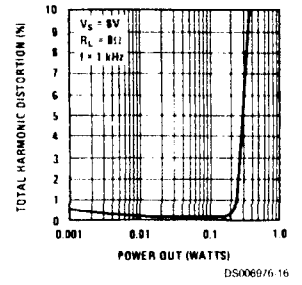
Voltage Gain vs Frequency



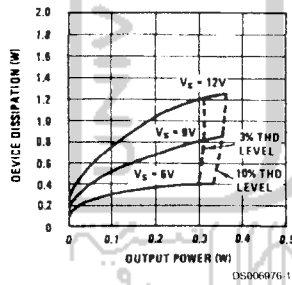
Distortion vs Frequency



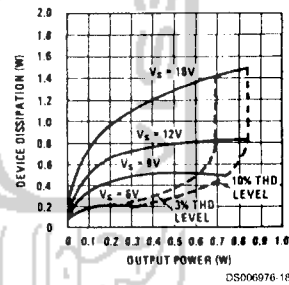
Distortion vs Output Power



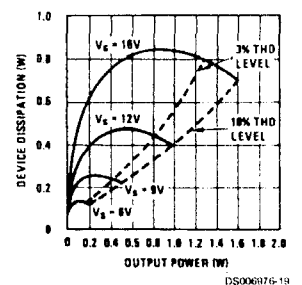
Device Dissipation vs Output Power—4 Ohm Load



Device Dissipation vs Output Power—8 Ohm Load

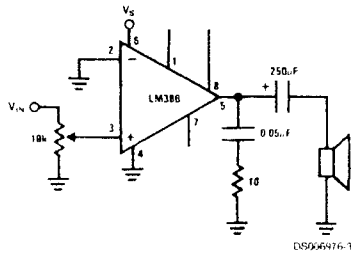


Device Dissipation vs Output Power—16 Ohm Load



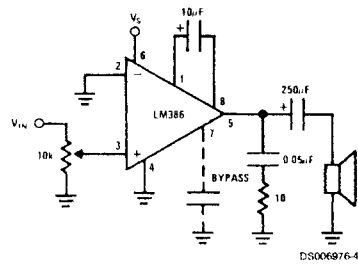
Typical Applications

Amplifier with Gain = 20
Minimum Parts



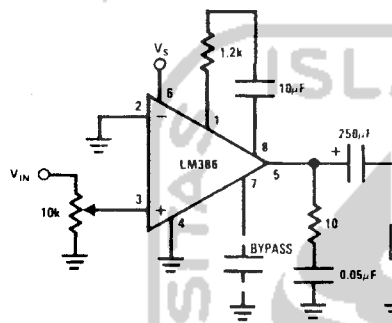
DS006976-1

Amplifier with Gain = 200



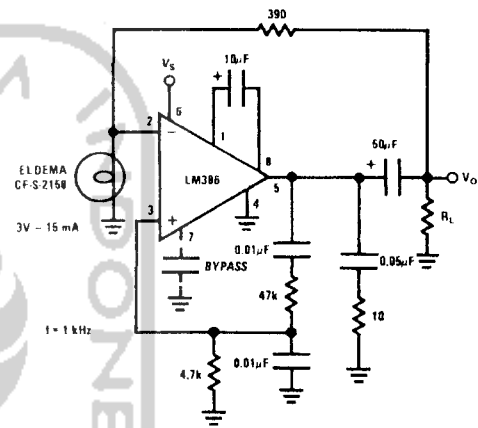
DS006976-4

Amplifier with Gain = 50



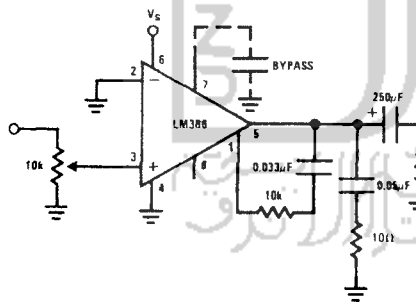
DS006976-6

Low Distortion Power Wienbridge Oscillator



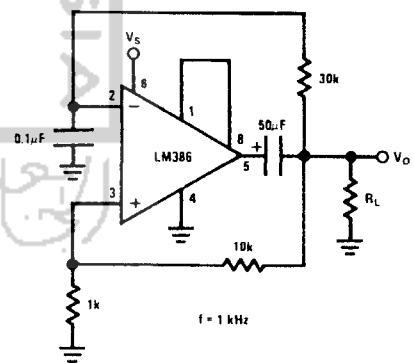
DS006976-7

Amplifier with Bass Boost



DS006976-8

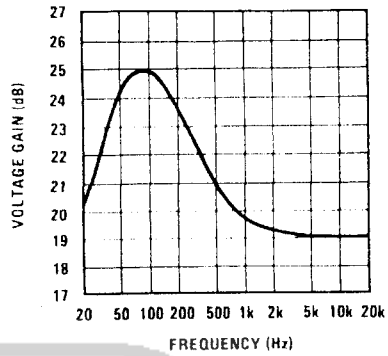
Square Wave Oscillator



DS006976-9

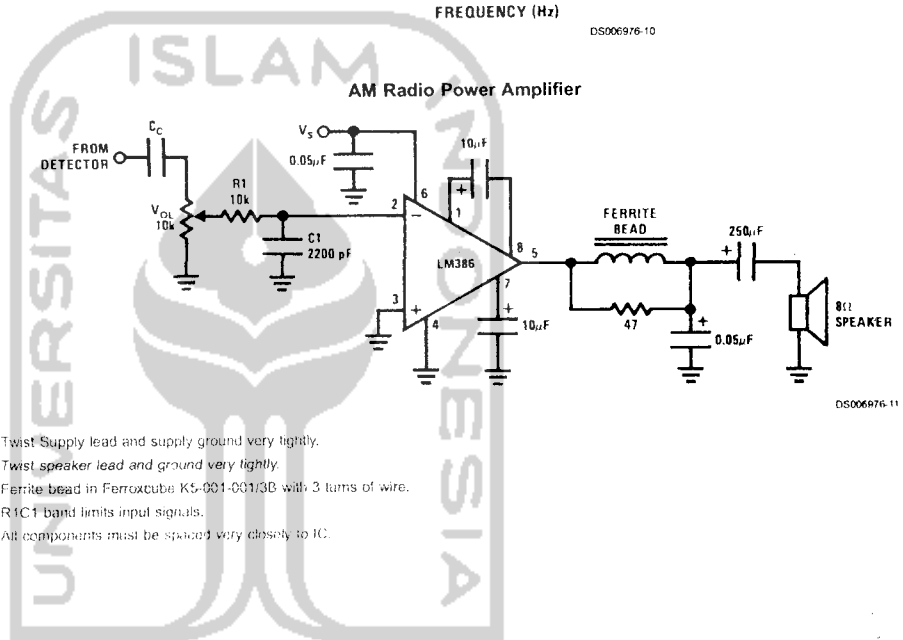
Typical Applications (Continued)

Frequency Response with Bass Boost



DS006976-10

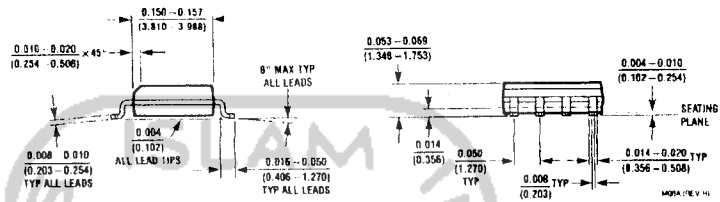
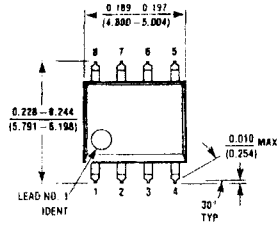
AM Radio Power Amplifier



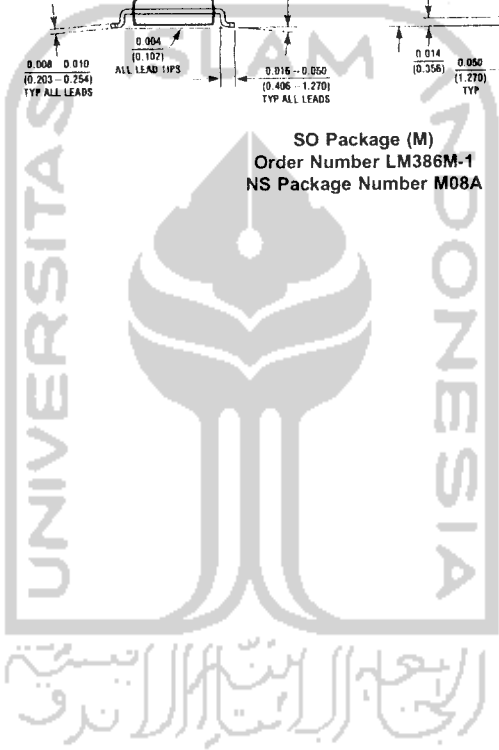
DS006976-11

- Note 4: Twist supply lead and supply ground very tightly.
- Note 5: Twist speaker lead and ground very tightly.
- Note 6: Ferrite bead in Ferroxcube K5-001-001/3D with 3 turns of wire.
- Note 7: R1C1 band limits input signals.
- Note 8: All components must be spaced very closely to IC.

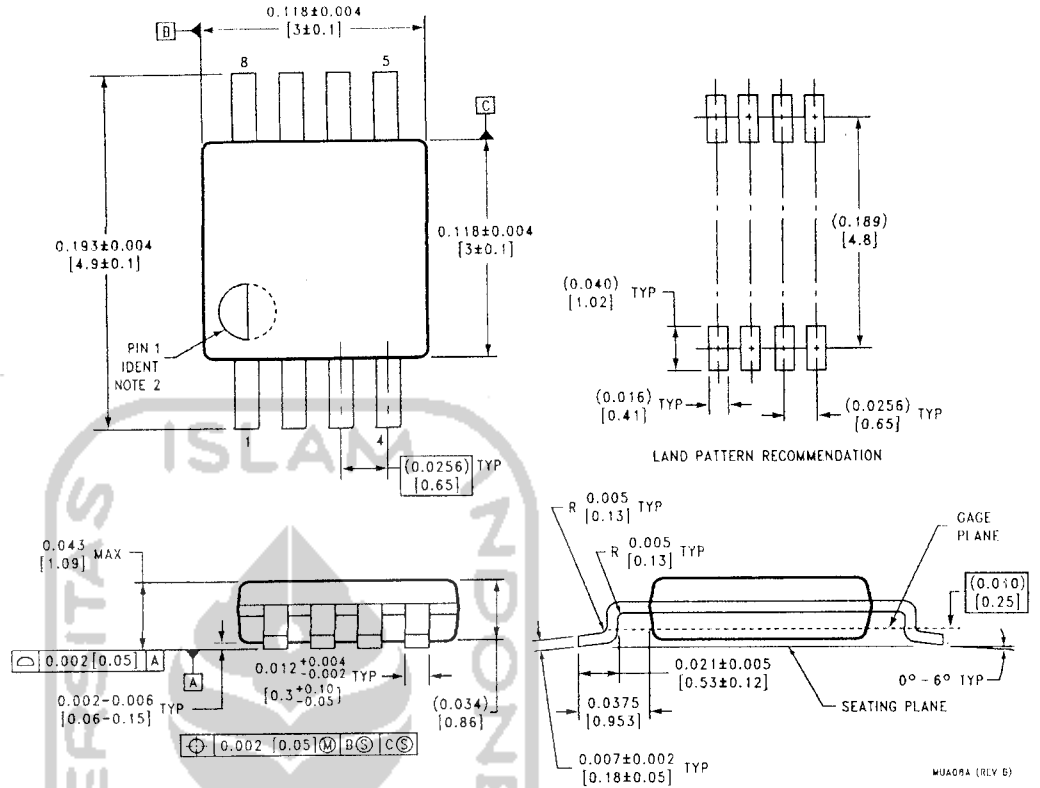
Physical Dimensions inches (millimeters) unless otherwise noted



SO Package (M)
Order Number LM386M-1
NS Package Number M08A



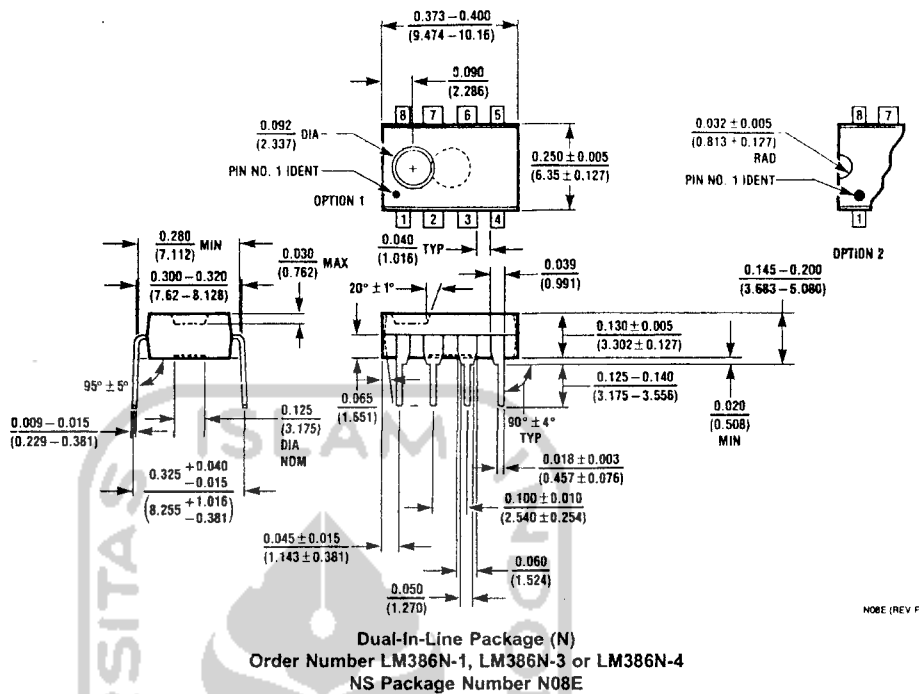
Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



8-Lead (0.118" Wide) Molded Mini Small Outline Package
Order Number LM386MM-1
NS Package Number MUA08A

MUA08A (REV 6)

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



LIFE SUPPORT POLICY

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

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

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65 2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7580
Fax: 81-3-5639-7507