

**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA**



**PERANCANGAN DAN PEMBUATAN VIDEO SWITCHER
UNTUK MONITORING GEDUNG BERBASIS
MIKROKONTROLER AT89S51**

SKRIPSI

**Disusun Oleh :
Rudy Prayitno
NIM 0217009**



MARET 2007

RECEIVED
MAY 15 1964
U.S. AIR FORCE
OFFICE OF THE
SECRETARY

RECEIVED
MAY 15 1964
U.S. AIR FORCE
OFFICE OF THE
SECRETARY

RECEIVED
MAY 15 1964
U.S. AIR FORCE
OFFICE OF THE
SECRETARY

RECEIVED

RECEIVED
MAY 15 1964
U.S. AIR FORCE
OFFICE OF THE
SECRETARY

RECEIVED

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN VIDEO SWITCHER UNTUK
MONITORING GEDUNG BERBASIS MIKROKONTROLER AT89S51

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

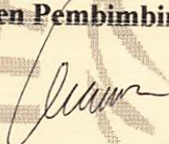
Disusun Oleh :


RUDY PRAYITNO
NIM : 0217009

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II


Ir. Usman Djuanda, MM
NIP.P.1018700143


I Komang Somawirata, ST, MT
NIP.Y.1030100361

Mengetahui



Jurusan Teknik Elektro S-1


Ir. F. Yudi Limpraptono, M.T.
NIP. Y. 1039500274

KONSENTRASI TEKNIK ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

2007



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Rudy Prayitno
NIM : 02.17.009
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Video Switcher
Untuk Monitoring Gedung Berbasis Mikrokontroler
AT89S51.
Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Jum'at
Tanggal : 16 Maret 2007
Dengan Nilai : A (83,05) *Bef*



PANITIA UJIAN SKRIPSI

KETUA

Ir. Mochtar Asroni, MSME
NIP.Y. 1018100036

SEKRETARIS

Ir. F. Yudi Limpraptono, M.T.
NIP.Y. 1039500274

ANGGOTA PENGUJI

PENGUJI I

Ir. Sidik Noertjahjono, M.T.
NIP. 1028700167

PENGUJI II

Sotyohadi, S.T, M.T.

27/3 '07

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas semua limpahan rahmat dan hidayah-Nya yang diberikan kepada penulis, sehingga penulis bisa menyelesaikan skripsi dengan judul ” *Perancangan dan Pembuatan Video Switcher Untuk Monotoring Gedung Berbasis Mikrokontroler AT89S51*” dengan baik dan tepat pada waktunya.

Banyak pihak yang telah membantu dalam penulisan skripsi ini, oleh karena itu penulis ingin mengucapkan rasa terima kasih yang sebesar – besarnya kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
3. Bapak Ir. F.Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S1 Konsentrasi Teknik Elektronika.
4. Bapak Ir. Usman Djuanda, MM dan Bapak I Komang Somawirata, ST, MT selaku Dosen Pembimbing yang telah memberikan bimbingan, petunjuk, dan saran sehingga skripsi ini bisa selesai dengan baik dan tepat pada waktunya.
5. Bapak Ir. Yusuf Ismail Nahkoda, MT. selaku Sekretaris Jurusan Teknik Elektro S1.
6. Seluruh dosen di Institut Teknologi Nasional Malang yang telah memberikan bekal ilmu kepada penulis selama penulis mengikuti kuliah.

Akhir kata, penulis berharap semoga skripsi ini bermanfaat, untuk pengembangan penelitian, khususnya di Jurusan Teknik Elektro Institut Teknologi Nasional Malang dan bisa bermanfaat bagi seluruh masyarakat pada umumnya.

Malang, Maret 2007

Penulis

PERANCANGAN DAN PEMBUATAN *VIDEO SWITCHER*
UNTUK MONITORING GEDUNG BERBASIS
MIKROKONTROLER AT89S51

Rudy Prayitno

Konsentrasi Teknik Elektronika, Jurusan Teknik Elektro S1, Fakultas Teknologi
Industri, Institut Teknologi Nasional Malang

Abstrak:

Perkembangan teknologi pada saat ini berkembang dengan cepat yakni berkembangnya teknologi *video* pada saat ini, kita juga dapat memanfaatkannya sebagai salah satu alat untuk memonitor gedung, perkantoran atau rumah. Skripsi ini membahas tentang monitoring beberapa gedung dengan menggunakan kamera yang akan ditampilkan oleh sebuah televisi. Proses perpindahan dari kamera satu dengan yang lain bisa dilakukan dengan cara otomatis dan manual. Pergerakan kamera juga dapat diatur secara manual dan otomatis. Untuk menggerakkan kamera digunakan motor stepper. Pengendalian dari keseluruhan sistem ini menggunakan mikrokontroler.

Kata Kunci: Kamera (CCTV), Mikrokontroler, Sensor Getar, Relay, Alarm.

DAFTAR ISI

HALAMAN JUDUL

LEMBAR PERSETUJUAN

BERITA ACARA UJIAN SKRIPSI

KATA PENGANTAR i

ABSTRAK iii

DAFTAR ISI iv

DAFTAR GAMBAR vii

DAFTAR TABEL ix

BAB I. PENDAHULUAN

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Tujuan 2

1.4 Batasan Masalah 2

1.5 Metodologi 3

1.6 Sistematika 4

BAB II. TEORI PENUNJANG

2.1 Mikrokontroler AT89S51 5

2.1.1 Pendahuluan 5

2.1.2 Arsitektur AT89S51 6

2.1.3 Konfigurasi Pin-Pin Mikrokontroler AT89S51 8

2.1.4 Organisasi Memory	11
2.1.4.1 Program Memory Internal	11
2.1.4.2 Data Memory (RAM) Internal	12
2.1.4.3 SFR (Special Function Register)	14
2.1.4.4 Program Status Word	14
2.1.4.5 PCON (Power Control)	15
2.1.4.6 Sistem Interupt	15
2.1.4.7 Timer / Counter	17
2.1.4.8 Metode Pengalamatan	18
2.2 IC ULN 2003A	20
2.3 LCD (<i>Liquid Crystal Display</i>)	22
2.4 Relay	23
2.5 Push button	25
2.6 Sensor Getar	26
2.7 Alarm	27
2.8 Motor Stepper	27

BAB III. PERENCANAAN DAN PEMBUATAN ALAT

3.1 Perencanaan Perangkat Keras	35
3.1.1 Rangkaian LCD (<i>Liquid Cristal Display</i>)	39
3.1.2 Mikrokontroler (MCU)	39
3.1.3 Sensor Getar	41
3.1.4 Rangkaian Driver <i>Buzzer</i>	42
3.1.5 Rangkaian Driver Motor Stepper	42

3.1.6 Sistem Pemilih Untuk Mengaktifkan Kamera	44
3.1.7 Tombol	46
3.2 Perancangan Perangkat Lunak (Software)	47
3.2.1 Diagram Alir	48
3.2.1.1 Diagra Alir Utama	48
3.2.1.2 Diagram Alir Layanan Interrupt	52

BAB IV PENGUJIAN ALAT

4.1 Pengujian Rangkaian Sensor Getar	53
4.2 Pengujian Rangkaian ULN2003	55
4.3 Pengujian Pemilihan Kamera Dan Motor Stepper	57
4.4 Pengujian sistem Secara Keseluruhan	66

BAB V PENUTUP

5.1 Kesimpulan	67
5.2 Saran	68

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Blok Diagram AT89S51	7
Gambar 2.2 Pin-Pin <i>AT89S51</i>	9
Gambar 2.3 Ilustrasi Pembagian Ruang RAM AT89S51	13
Gambar 2.4 Skema Mendefinisikan PSW	14
Gambar 2.5 Skema Mendefinisikan PCON	15
Gambar 2.6 Kegunaan Interrupt <i>Enable Register</i>	16
Gambar 2.7 Konfigurasi Dan Guna TMOD	17
Gambar 2.8 Pin-Pin Koneksi Dalam IC ULN 2003	20
Gambar 2.9 Rangkaian Darlington didalam IC ULN 2003A.....	21
Gambar 2.10 Pin – pin pada LCD M1632	22
Gambar 2.11 Gambar Jenis Kontak pada Relay	25
Gambar 2.12 Jenis Relay sederhana.....	25
Gambar 2.13 Simbol <i>Limit switch</i>	26
Gambar 2.14 Tipe-tipe Motor Stepper	27
Gambar 2.15 Struktur Motor Stepper.....	29
Gambar 2.16 Prinsip Kerja Motor Stepper.....	30
Gambar 2.17 Rangkaian Driver Untuk Berbagai Tipe Motor Stepper	31
Gambar 2.18 Macam-macam Eksitasi	34
Gambar 3.1 Diagram blok <i>video switcher</i> untuk monitoring gedung.....	36
Gambar 3.2 Simulasi alat	38
Gambar 3.3 Rangkaian LCD	39
Gambar 3.4 Rangkaian Minimum Sistem AT89S51	40

Gambar 3.5 Rangkaian Sensor Getar	41
Gambar 3.6 Rangkaian Driver <i>Buzzer</i>	42
Gambar 3.7 Rangkaian Driver Motor Stepper	43
Gambar 3.8 Rangkaian Sistem Pemilih Untuk Mengaktifkan Kamera.....	45
Gambar 3.9 Rangkaian Tombol/ <i>Limit switch</i>	46
Gambar 4. 1Pengujian Rangkaian Sensor Getar	54
Gambar 4.2 Pengujian rangkaian ULN 2003	56
Gambar 4.3 Rangkaian pemilih kamera dan motor stepper	58
Gambar 4.4 Hasil pengujian pemilih kamera dan motor stepper	59

DAFTAR TABEL

Tabel 2.1 Pengaturan RS0-RS1 <i>Select Register Bank</i>	14
Tabel 2.2 Alamat Sumber Interupsi	16
Tabel 2.3 <i>Mode Operasi Timer/Counter</i>	18
Tabel 2.4 Fungsi Pin LCD.....	23
Tabel 2.5 . Macam-macam Eksitasi	32
Tabel 4.1 Hasil Pengujian Rangkaian Sensor Getar	54
Tabel 4.2 Hasil pengukuran output mikrokontroler dan output ULN	56
Tabel 4.3. Hasil pengujian rangkaian pemilih kamera dan motor stepper	59
Tabel 4.4. Pengujian motor stepper	60

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi pada saat ini berkembang dengan cepat yakni berkembangnya teknologi *video* pada saat ini, kita juga dapat memanfaatkannya sebagai salah satu alat untuk memonitor gedung, perkantoran atau rumah. Dengan menggunakan kamera yang berada pada beberapa lokasi sehingga juga diperlukan beberapa monitor atau televisi untuk menampilkan gambar yang telah diambil gambarnya oleh kamera tersebut. Jika digunakan beberapa *televisi*, biaya yang dikeluarkan sangat besar oleh karena itu dibuatlah suatu alat yaitu *video switcher* yang dapat menampilkan semua tampilan gambar hanya dengan sebuah *televisi* secara bergantian, sehingga kita dapat menghemat biaya dengan menggunakan *video switcher* ini.

1.2. Rumusan Masalah

Dari latar belakang diatas, maka dapat dibuat suatu rumusan masalah sebagai berikut:

1. Bagaimana mengaktifkan *video switcher* secara bergantian antara kamera satu dengan yang lain secara otomatis.
2. Bagaimana mengaktifkan *video switcher* secara bergantian antara kamera satu dengan yang lain secara manual.

3. Bagaimana membuat perangkat keras(*hardware*) dan perangkat lunak(*software*) dari alat *video switcher* dengan menggunakan mikrokontroler.

Berdasarkan beberapa rumusan masalah di atas, maka judul yang ingin diangkat adalah **“Perancangan dan Pembuatan *Video Switcher* untuk *Monitoring* Gedung Berbasis Mikrokontroler AT89S51.”**

1.3. Tujuan

Tujuan dari pembuatan *video switcher* untuk monitoring gedung yang berbasis mikrokontroler AT89S51 adalah merancang dan membuat alat yang dapat memonitor beberapa lokasi dengan secara bergantian, tetapi hanya membutuhkan satu buah *televisi*. Sehingga dalam pembuatan alat ini dapat memudahkan *monitoring* gedung.

1.4. Batasan Masalah

Didalam penulisan skripsi ini, agar pembahasan masalah sesuai dengan yang diinginkan dan tidak menyimpang dari apa yang semula dirumuskan, maka penulis menyertakan batasan masalah sebagai berikut :

1. Pembuatan *video switcher* dengan menggunakan empat buah input dari kamera yang akan *diswitch* sehingga menghasilkan sebuah output.
2. Alat ini Menggunakan Mikrokontroler AT89S51.
3. Alat ini Tidak dapat mengirimkan sinyal audio ke TV.
4. Kamera yang digunakan adalah CCTV(tidak berwarna).
5. Pergerakan motor hanya 108⁰

6. Tidak membahas masalah kamera dan transmisi data kamera.
7. Tidak membahas masalah *power supply*.
8. Alat yang dirancang dan dibuat dalam bentuk simulasi.

1.5. Metodologi

Tugas laporan akhir ini bersifat aplikatif dimana adanya perencanaan dan pembuatan alat. Berikut ini adalah langkah-langkah yang perlu dilaksanakan untuk pembuatan alat ini:

1. Study literature

Memperoleh data-data pokok dan penunjang dengan cara membaca dan mempelajari buku atau jurnal literatur yang ada hubungannya dengan penyusunan skripsi ini.

2. Perancangan Tiap Blok

Berhubungan dengan prinsip kerja alat yang diinginkan serta komponen-komponen yang digunakan sehingga setelah digabungkan akan terbentuk sistem kerja sesuai alat yang direncanakan.

3. Pembuatan Software

Membuat listing program guna mengaktifkan dan menjalankan komponen-komponen yang dipakai.

4. Pengujian terhadap alat yang dibuat.

5. Analisa

Dari hasil pengujian diatas kemudian dibuat suatu kesimpulan terhadap alat yang dibuat.

1.6. Sistematika

Adapun sistematika dari penyusunan laporan skripsi ini adalah :

BAB I PENDAHULUAN

Berisi latar belakang belakang, rumusan masalah, tujuan, pembatasan masalah, metodologi penulisan, serta sistematika penyusunan dan pembuatan alat.

BAB II TEORI PENUNJANG

Berisi tentang teori – teori dasar yang memiliki relevansi sebagai dasar perencanaan dan pembuatan.

BAB III PERENCANAAN DAN PEMBUATAN ALAT

Berisi tentang perencanaan *hardware* dan *software*.

BAB IV PENGUJIAN ALAT

Berisi tentang data hasil pengujian peralatan yang telah dibuat secara keseluruhan.

BAB V PENUTUP

Berisi kesimpulan dan saran dari tugas akhir ini



BAB II

TEORI PENUNJANG

2.1. Mikrokontroler AT89S51

2.1.1 Pendahuluan

Mikrokontroller bisa dipandang sebagai sebuah mini komputer yang terintegrasi dalam sebuah *chip*. Didalam satu *chip* mikrokontroler sudah terdapat bagian-bagian seperti dalam sebuah komputer. Bagian-bagian itu antara lain ; ALU (*Arithmetic Logic Unit*), PC (*Program Counter*), SP (*Stack Pointer*), Register, ROM (*Read Only Memory*), RAM (*Random Acces Memory*), Paralel I/O, Serial I/O, *Counter* dan sebuah rangkaian *Clock*.

Seperti sebuah mikroprosesor, mikrokontroler adalah sebuah perangkat serbaguna, yang fungsi kerjanya dapat ditentukan melalui sebuah perangkat lunak yang mendeskripsikan sebuah sistem yang diinginkan.

Pada saat ini terdapat banyak keluarga mikrokontroler salah satunya adalah keluarga MCS51. Salah satu tipe mikrokontroler yang termasuk dalam keluarga MCS51 adalah AT89S51 buatan *Atmel*.

AT89S51 adalah mikrokontroler keluaran *atmel* dengan 4K *byte Flash PEROM (Programmable and Erasable Read Only Memory)*, AT89S51 merupakan memori dengan teknologi *nonvolatile memory*, artinya isi memori tersebut dapat diisi ulang ataupun dihapus berulang kali.

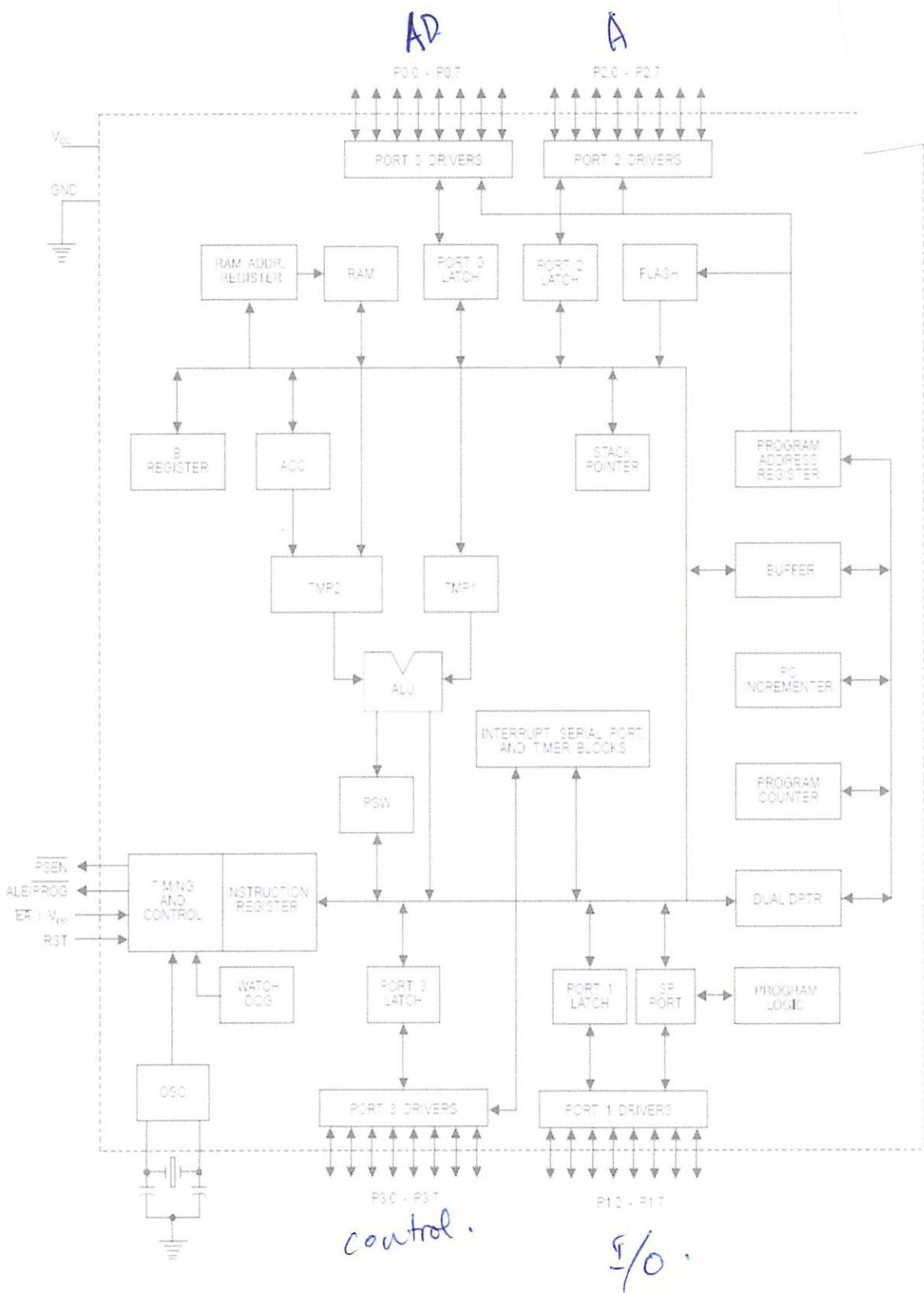
Memori ini biasa digunakan untuk menyimpan instruksi (Perintah) berstandar MCS – 51 code sehingga memungkinkan mikrokontroler ini untuk

bekerja dalam mode *Single Chip Operation* (Mode Operasi Keping Tunggal) yang tidak memerlukan *Eksternal Memori* (Memori luar) untuk menyimpan *source code* tersebut.

2.1.2. Arsitektur AT89S51

Mikrokontroler AT89S51 secara umum memiliki:

- CPU 8bit
- *Memory*
- Port I/O yang dapat diprogram
- *Timer dan Counter*
- Sumber *Interrupt*
- Program serial yang dapat diprogram
- *Osilator dan clock*



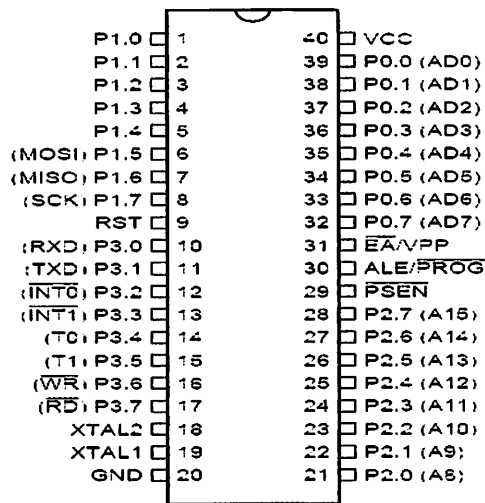
Gambar 2.1 Blok Diagram AT89S51

Arsitektur mikrokontroler AT89S51 adalah sebagai berikut:

1. CPU (Central Processing Unit) 8-bit dengan register A (*Accumulator*) dan B.
2. 16-bit program *counter* (PC) dan data *pointer* (DTPR).
3. 8-bit program status *word* (PSW).
4. 4-bit *stack pointer* (SP).
5. 5 Kbyte internal EPROM.
6. 128 Byte internal RAM
 - 4 bank register, masing- masing berisi 8 register.
 - 16 Byte yang dapat dialamati pada bit *level*.
 - 80 Byte *general purpose memory data*.
7. 32 pin *input- output* tersusun atas PO-P3. masing-masing 8-bit.
8. 2 buah 16-bit *Timer/counter*.
9. *Receiver/Tansmitter* data serial *Full Duplex*: SBUF.
10. Control Register, yaitu: TCON, TMOP, SCON, PCON, PCON, IP, dan IE.
11. 5 buah sumber interupsi (2buah sumber *interrup external* dan 3 buah sumber *interup internal*).
12. *Osilator* dan *Clock Internal*.

2.1.3. Konfigurasi Pin - Pin Mikrokontroler AT89S51

Untuk konfigurasi dari pin-pin kaki IC AT89S51 dapat dilihat pada gambar 2.2 dibawah ini.



Gambar 2.2 Pin – Pin AT89S51

Fungsi dari tiap- tiap pena adalah sebagai berikut:

1. VCC (*Suplay* tegangan).
2. GND (*Ground*).
3. PORT 0

Merupakan *port input- output* dua arah dan dikonfigurasi sebagai *multipleks* dua bus alamat rendah (A0 – A7) dan data selama pengaksesan program memori dan data memori *external*.

4. PORT 1

Merupakan port input – output dengan *pull up internal* yang dapat dipergunakan untuk berganti keperluan. Port 1 juga menerima alamat *byte* rendah selama program EPROM maupun ROM didalamnya

5. PORT 2

Merupakan *port input – output* dengan *pull up internal*. Port 2 juga merupakan *bus* alamat *byte* tinggi selama adanya akses ke memori program luar dan selama mengakses memori data *eksternal* yang menggunakan pengalamatan 16 bit. MOVX @DPTR digunakan dalam pengaksesan memori yang menggunakan pengalamatan 8 bit (MOVX @ R1), port 2 mengeluarkan isi dari P2 SFR (*Special Function Generator*). Port 2 juga dapat berfungsi mengeluarkan sebagai *bus* alamat atas selama pemrograman EPROM dan saat verifikasi ROM dan EPROM pada mikrokontroler yang dilengkapi ROM atau EPROM dikepingnya.

6. PORT 3

Merupakan *port input-output* dengan *internal pull-up*.

Port 3 juga memiliki fungsi khusus, yaitu:

- RXD (P3.0) : *Port input serial*.
- TXD (P3.1) : *Port output serial*.
- INT0 (P3.2) : *Interrupt 0 external*.
- INT1 (P3.3) : *Interrupt 1 external*.
- T0 (P3.4) : *Input external timer 0*.
- T1 (P3.5) : *Input external timer 1*.
- WR (P3.6) : *Strobe tulis data memory external*.
- RD (P3.7) : *Strobe baca data memory external*.

7. RST

Kondisi *high* pada pin ini selama dua siklus ketika *osilator* bekerja akan mereset mikrokontroler

8. ALE/ PROG

Pulsa output ALE digunakan untuk proses *'latching' byte address* rendah (A0- A7) selama pengaksesan ke *external memory*. Pin ini juga digunakan untuk memasukkan pulsa program (prog) selama pemrograman.

9. PSEN

PSEN (*Program Store Enable*) merupakan sinyal baca untuk memori program *eksternal*.

9. EA/VPP

External address enable EA digroundkan jika mengakses *memory external*. Untuk mengakses *memory internal* maka dihubungkan ke VCC.

4. X-TALL1 dan X-TALL2

Kaki ini dihubungkan dengan kristal bila menggunakan *osilator internal*. X-TALL1 merupakan *input inverting osilator amplifier* sedangkan X_TALL2 merupakan *output inverting osilator amplifier*.

2.1.4. Organisasi Memory

Di dalam AT89S51 ruang alamat telah dibedakan untuk program *memory* dan data *memory*.

2.1.4.1 Program Memory Internal

AT89S51 memiliki program *memory internal* sebesar 4 Kbyte dengan ruangan alamat 0000H- 0FFFH. Jika alamat-alamat program lebih tinggi dari pada 0FFFH, yang melebihi kapasitas ROM *internal* menyebabkan AT89S51 secara otomatis mengambil *code byte* dari program *memory external*. *Code byte* juga

dapat diambil hanya dari *external memory* dengan alamat 0000H-FFFFH dengan cara menghubungkan pin EA ke *ground*.

2.1.4.2. Data Memory (RAM) Internal

Ruangan alamat bahwa *memory data (RAM) internal* dengan kapasitas 128 *byte* yaitu: 00H-7FH yang terbagi atas 3 daerah, yaitu:

- Empat *bank* register

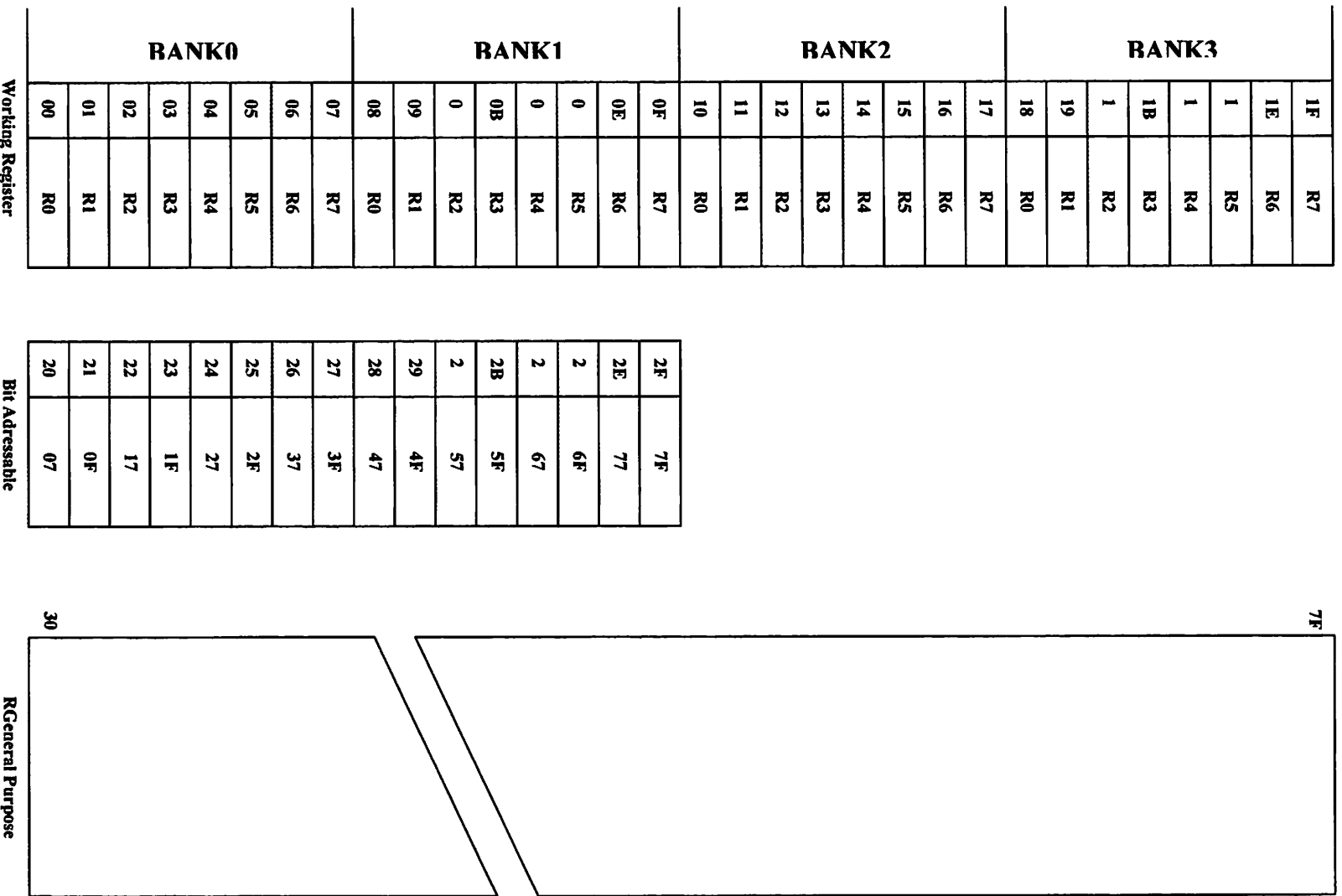
Setiap *bank* terdiri dari 8 register (R0-R7) sehingga jumlah register untuk keempat *bank* register menjadi 32 buah register yang menempati ruang alamat 00H-1FH. Mengaktifkan salah satu *bank* register dapat dilakukan dengan mengatur RS0-RS1 pada program *status word* (PSW).

- *Bit Addressable*

Terdiri dari 16 *byte* yang berada pada alamat 20H-2FH. Masing- masing 128 bit lokasi ini dapat dialamati secara langsung.

- *Stratc Pad Area*

Terdiri atas 80 *byte* yang menempati alamat 30H-7FH. Yang dapat dialamati secara langsung dan digunakan untuk keperluan umum (*general purpose*) misalnya digunakan untuk lokasi *attack*.



Gambar 2.3 Ilustrasi Pembagian Ruang RAM AT89SS51

Tabel 2.1 Pengaturan RS0-RS1 *Select Register Bank*

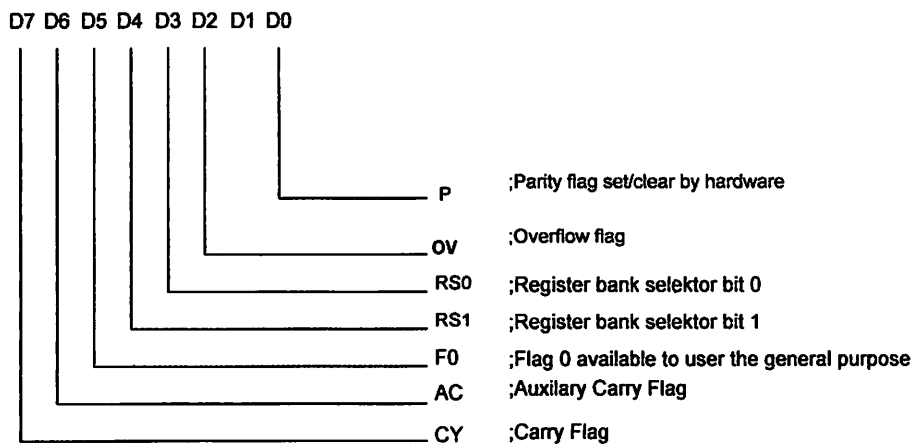
RS1	RS0	Select Register Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

2.1.4.3. SFR (*Special Function Register*)

Untuk operasi AT89S51 yang tidak menggunakan alamat *internal* RAM (00H-7FH) dilakukan oleh SFR yang beraddress 80H- FFH, tetapi tidak semua *address* tersebut digunakan sebagai SFR, hanya *address* tertentu saja.

2.1.4.4. Program Status *Word*

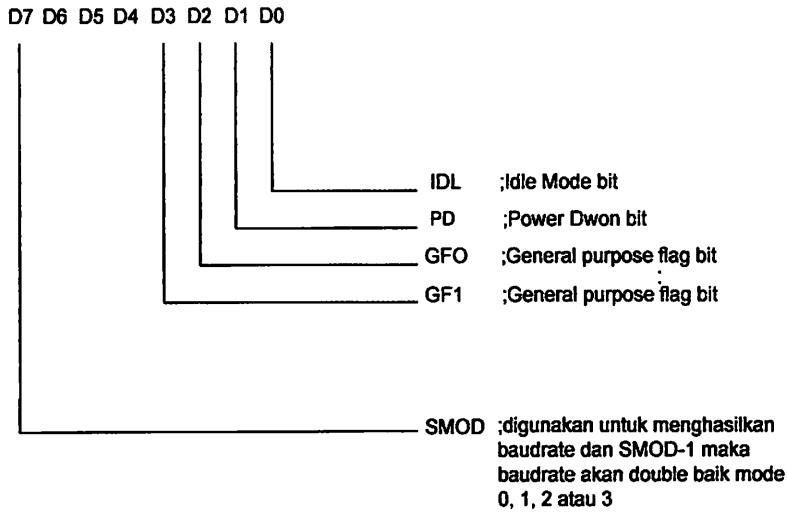
Register ini terletak di alamat D0H. Cara mendefenisikannya adalah sebagai berikut:



Gambar 2.4 Skema Medefinisikan PSW

2.1.4.5. PCON (*Power Control*)

Register ini terletak pada alamat 87H. Cara mendefinisikan adalah sebagai berikut:



Gambar 2.5 Skema Mendefinisikan PCON

2.1.4.6. Sistem *Interupt*

Mikrokontroler AT89S51 mempunyai 5 buah sumber interupt yang dapat mengakibatkan permintaan interupt, yaitu: INT0, INT1, T1, T2 dan *port* serial.

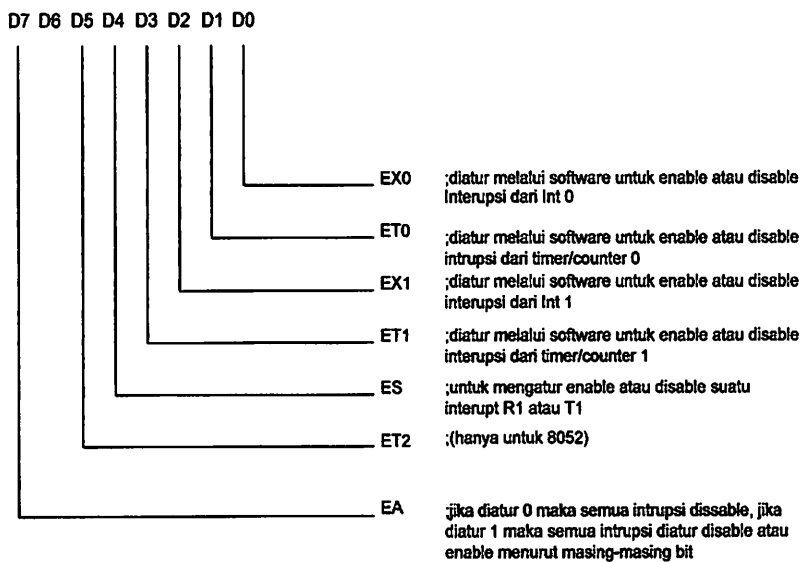
Saat terjadi interup mikrokontroler secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interupt *service* selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Dua sumber external adalah INT0, INT1, dimana kedua interupsi *eksternal* akan aktif atau aktif transisi tergantung isi IT0 dan IT1 pada register TCON. Interupsi T0 dan T1 aktif pada saat timer yang sesuai mengalami *roll over*. Interupsi serial dibangkitkan dengan melakukan operasi OR pada R1 dan T1. Tiap – tiap sumber interupsi dapat *enable* atau *disable* secara *software*.

Tingkat prioritas semua sumber interupsi dapat diprogram sendiri-sendiri dengan set atau *clear* bit pada SFRS IP (*Interrupt Priority*).

Tabel 2.2 Alamat Sumber Interupsi

Sumber Interupt	Alamat Awal
<i>Power On Reset</i>	0000h
Interupt Luar 0 (INT 0)	0003h
Pewaktu/ Pencacah 0 (T0)	000Bh
Interup Luar 1 (INT 1)	0013h
Pewaktu/ Pencacah 1 (T1)	001B
Port I/O Serial	0023h

Register yang berperan dalam mengatur aktif tidaknya interupsi adalah *interrupt enable register*, berikut ini adalah susunan dari bit-bit beserta kegunaannya:



Gambar 2.6 Kegunaan Interupt Enable Register

2.1.4.7. Timer/ Counter

Pengendalian kerja dari *timer/counter* dilakukan dengan pengaturan register yang berhubungan dengan kerja dari *Timer/counter* yaitu melalui sebuah *timer/counter mode control*.

Untuk mengaktifkan *timer/counter* yang meliputi penentuan fungsi sebagai *timer* atau sebagai *counter* serta pemilihan *mode* operasi dapat diatur melalui TMOD yang beralamat pada 89H.

- Gate

Bila *gate* = 1, *Timer/counter x enable* hanya pada saat pin INTx tinggi dan TRx 1,. Saat *gate* 0, *timer/counter enable* jika bit TRx 1

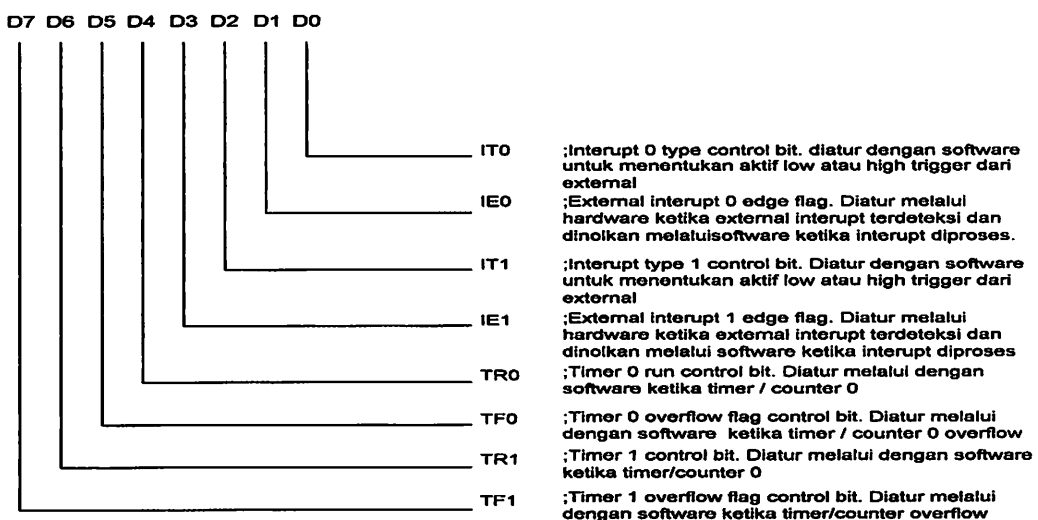
- CT

Jika bit C/T = 0, maka *timer/counter x* akan berfungsi sebagai *timer*. Jika C/T = 1, maka *timer/counter x* akan beroperasi sebagai *counter*.

- M1 dan M2

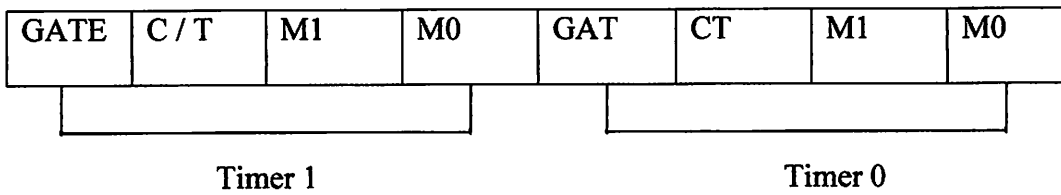
Menentukan *mode*.

Konfigurasi yang dimaksud adalah sebagai berikut:



Gambar 2.7 Konfigurasi dan Guna TMOD

TMOD : Timer/Counter Mode Control Register



Tabel 2.3 Mode Operasi Timer/Counter

M1	M0	Operating Mode
0	0	<i>Timer 13 bit</i>
0	1	<i>Timer/Counter 16 bit</i>
1	0	<i>8 bit Auto reload Timer /Counter</i>
1	1	<i>TL0 dari Timer adalah 8 Bit Timer/Counter dikendalikan oleh kontrol bit timer 0. TH0 adalah 8 bit yang dikendalikan oleh Timer 1 control bit</i>

2.1.4.8. Metode Pengalamatan

1. Pengalamatan Bit (*Direct Bit Addressing*)

Pengalamatan langsung tiap bit ini hanya dilakukan pada lokasi RAM *internal* yaitu 20H-2FH, d *port 1, port 2 port 3*, TCON register, SCON register, IE register, PSW register, ACC dan ACC dan B register.

2. Pengalamatan Tak Langsung

Pada pengalamatan tak langsung, instruksi menunjukkan suatu register yang isinya adalah alamat dari *operand*, *eksternal* dan *internal* RAM dapat dialamati secara tidak langsung. Register alamat untuk data dengan lebar 8 bit dapat berupa R0 dan R1 yang digunakan untuk memilih angka register atau *stack pointer*. Register alamat untuk data, dengan lebar 16 bit digunakan data *pointer* DPTR.

3. Pengalamatan Berindeks

Yang dapat diakses dengan pengalamatan *berindeks* hanya memori program. *Mode* ini dimaksudkan untuk membaca *look-up table program*.

4. Konstanta Immediat

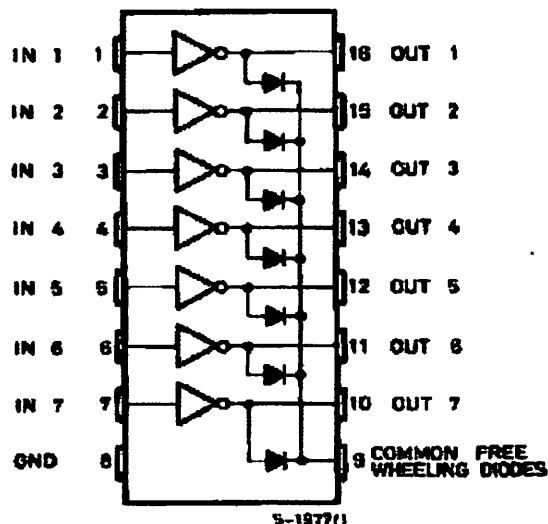
Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung, dilakukan dengan menggunakan tanda #.

Contoh: `Mov a# 100`

2.2. IC ULN 2003A

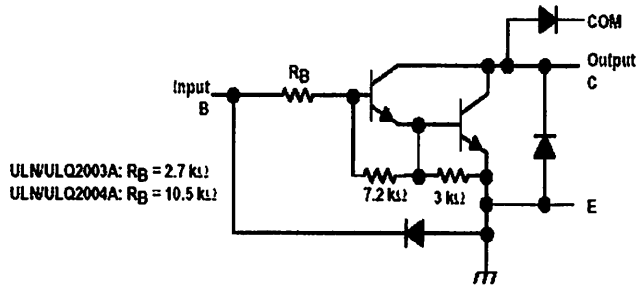
Pada IC ULN 2003A didalamnya terdapat rangkaian penguat Darlington. Untuk satu IC ULN2003A terdapat 7 pasang rangkaian Darlington NPN yang tersusun dalam rangkaian *common catoda*. Rangkaian Darlington ini digunakan sebagai saklar. Pada masing-masing rangkaian Darlington arus kolektornya sebesar 500mA. Rangkaian Darlington yang ada di dalam IC ULN 2003A dapat diparalel guna untuk kebutuhan arus yang besar. Karenanya IC ini dapat diaplikasikan untuk *driver* relay, *driver* lampu, *driver display* dan *logic buffer*. Pada skripsi ini rangkaian Darlington yang ada pada IC ULN 2003A digunakan sebagai *driver* relay yang digunakan untuk mengendalikan putaran motor Steper.

Adapun pin-pin koneksi yang ada dalam IC ULN 2003A dapat dilihat pada gambar 2.8 berikut ini:



Gambar 2.8 Pin-Pin Koneksi Dalam IC ULN 2003A

Sedangkan gambar untuk setiap rangkaian Darlington pada IC ULN 2003A dapat dilihat pada gambar 2.9



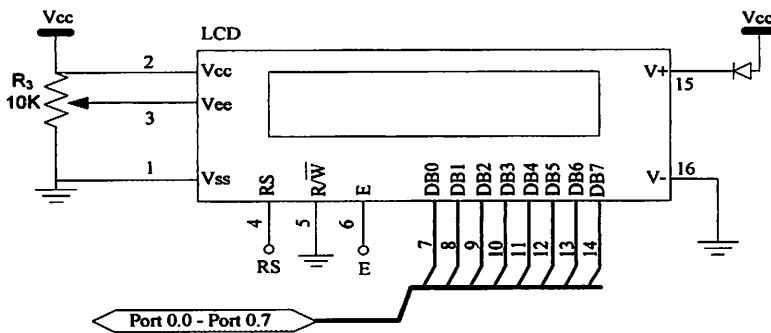
Gambar 2.9 Rangkaian Darlington didalam IC ULN 2003A.

Untuk *driver* penggerak *relay* digunakan IC ULN2003A dan *relay* sebagai komponen utamanya, IC ULN2003A dapat terpicu dengan tegangan 5 Volt dengan tegangan hubung sebesar 100 Volt dan arus maksimum yang diperbolehkan sebesar 500mA dengan suhu kerja dari -20°C sampai 80°C dengan data yang ada diatas maka IC ULN2003A mampu digunakan untuk menghidupkan dan mematikan *relay* yang hanya memiliki tegangan maksimal sebesar 12 Volt dengan resistansi kumparan sebesar 400 Ω jadi dapat diketahui arus *relay* sebesar:

$$\begin{aligned}
 \text{Dimana: } I_{\text{relay}} &= \frac{V_{CC}}{R_{\text{relay}}} \\
 &= \frac{12}{400} \\
 &= 30 \text{ mA}
 \end{aligned}$$

2.3. LCD (*Liquid Crystal Display*).

Pada perancangan ini penulis menggunakan penampil berupa LCD atau *Liquid Crystal Display* yaitu suatu tumpukan tipis atau sel yang terdiri dari atas dua lembar kaca yang kedua sisinya tertutup rapat, dimana diantara kedua lembar kaca tersebut diberikan bahan kristal cair (*Liquid Crystal*) yang tembus cahaya.



Gambar 2.10 Pin – pin pada LCD

Tabel 2.4 Fungsi Pin LCD

PIN	Jumlah Terminal	I/O	Tujuan	Fungsi
1	2	3	4	5
DB ₀ -DB ₃	4	I/O	MCU	4 bit bus data <i>lower tristate</i> dua arah, dapat dibaca/tulis terhadap MCU
DB ₄ -DB ₇	4	I/O	MCU	4 bit bus data <i>upper tristate</i> dua arah dapat dibaca/tulis terhadap MCU, DB ₇ juga sebagai bussy flag
E	1	Input	MCU	Sinyal tanda mulai operasi yang berfungsi sebagai alat baca/tulis
R/W	1	Input	MCU	0: Tulis, 1; Baca
RS	1	Input	MCU	Sinyal seleksi register ; 0: register instruksi (tulis); 1: Data register (baca/tulis)
Vee	1	-	Power Supply	Pengatur kontras LCD
Vcc	1	-	Power Supply	+ 5 Volt
Vss	1	-	Power Suplly	0 Volt (<i>Ground</i>)

2.4. Relay

Relay adalah suatu perangkat *switch* (saklar) yang dioperasikan oleh kumparan yang berada di dalamnya. Relay pada umumnya digunakan untuk menyambung atau memutuskan antara suatu bagian yang lain dalam suatu rangkaian elektronik, selain itu juga dimaksudkan untuk mengisolasi switching antara catu daya tinggi dan catu daya rendah. Kerugian yang ditemui pada relay yaitu adanya tanggapan pada respon (*response time*) saat on ataupun off relative lambat serta adanya efek induksi balik sesaat setelah relay off.

Pada relay terdapat beberapa susunan kontak :

1. Normal terbuka (Normally Open)

Kontak kontak tertutup pada saat relay dioperasikan atau ada arus kuat melalui kumparan.

2. Normal tertutup (Normally Close)

Kontak – kontak terbuka saat relay terbuka saat relay dioperasikan atau ada arus yang kuat melalui kumparan.

Cara kerja relay pada dasarnya adalah apabila ada arus yang masuk melalui kumparan maka pada kumparan akan terjadi induksi magnetic. Induksi magnetic tersebut nantinya akan menarik pegas kontak untuk merubah posisi awalnya menjadi terhubung kebagian yang diinginkan. Setelah arus berhenti maka tidak terjadi induksi sehingga kontak akan kembali ke posisi awal.

Macam – Macam Relay

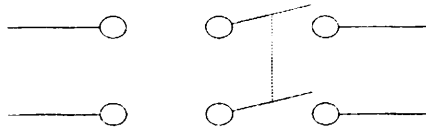
1) Single Pin Single Terminal (SPST)



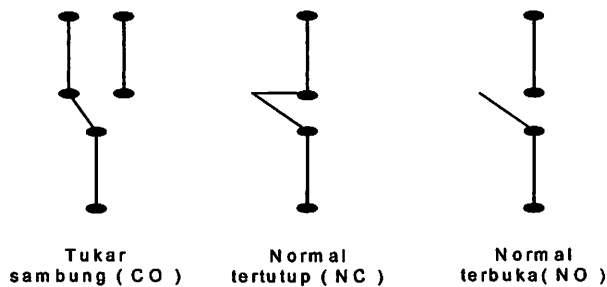
2) Single Pin Dual Terminal (SPDT)



3) Dual Pin Dual Terminal (DPDT)



Gambar 2.11. Gambar Jenis Kontak pada Relay

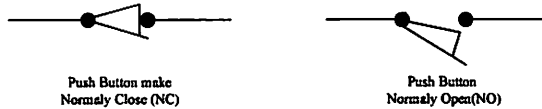


Gambar 2.12 Jenis Relay sederhana

2.5. Limit switch.

Limit switch merupakan sebuah saklar yang bekerja karena ada suatu gesekan atau sentuhan, *limit switch* kita tempatkan sesuai dengan kebutuhan kita agar tungkai *limit switch* dapat tersentuh benda-benda yang akan kita gerakkan. *Limit swich* mempunyai beberapa bagian mekanisme yaitu pengungkit dan roda pembuang, jika pengungkit dan roda pembuang tersebut tersentuh/bergesekan dengan benda atau mekanis lain maka *limit switch* akan ON dan akan menggerakkan lengan pengungkit dalam suatu kontak.

Ada beberapa tipe *limit switch* yaitu *Normaly Open* (NO) dan *Normaly Close* (NC). *Normaly Open* (NO) adalah kondisi dimana *limit switch* berfungsi sebagai pemutus, sedangkan pada *Normaly Close* (NC) *limit switch* akan berfungsi sebagai penghubung. Simbol dari *limit switch* adalah sebagai berikut :



Gambar 2.13 Simbol *Limit switch*

2.6. Sensor Getar

Sensor getar merupakan sebuah saklar yang bekerja karena adanya sebuah sentuhan atau getaran. Pemasangan sensor getar ini harus tepat dan reliable serta dapat memberikan respon yang cepat dengan memperhatikan faktor-faktor yang menyangkut tingkat kekritisian seperti ukuran. Sensor getar ini mempunyai beberapa bagian mekanik yaitu sebuah per. Sensor getar ini berfungsi hampir sama dengan sensor, dimana jika ada beban yang menyentuh per maka hubungan akan berpindah menghantarkan aliran listrik (seperti saklar).

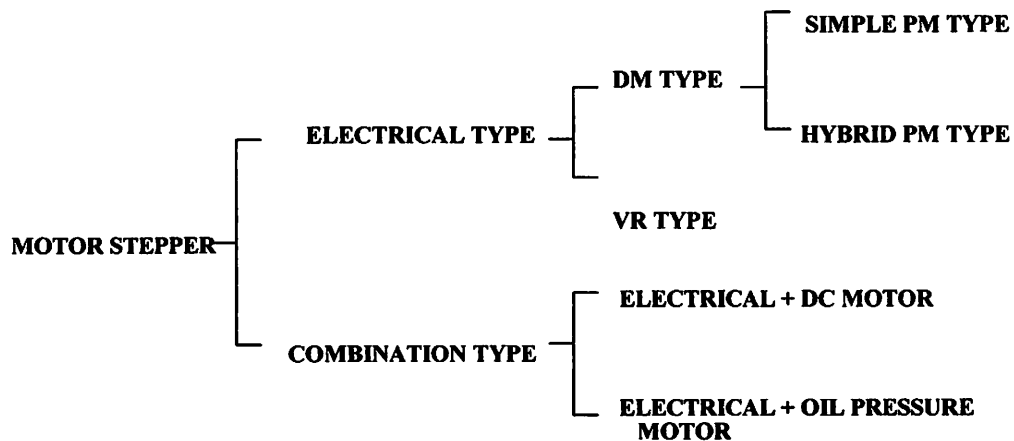
2.7. Alarm

Alarm adalah suatu alat yang dapat mengubah sinyal listrik menjadi sinyal suara. Penggunaannya cukup mudah yaitu dengan memberikan tegangan input maka alarm akan mengeluarkan bunyi. Frekuensi suara yang di keluarkan oleh alarm yaitu antara 1-5 KHz.

2.8. Motor stepper

Motor stepper atau motor langkah adalah motor listrik yang dirancang untuk penggunaan pada sistem kontrol digital langsung (Direct Digital Control), dimana sinyal yang dihasilkan berasal dari sistem digital, seperti: mikrokomputer.

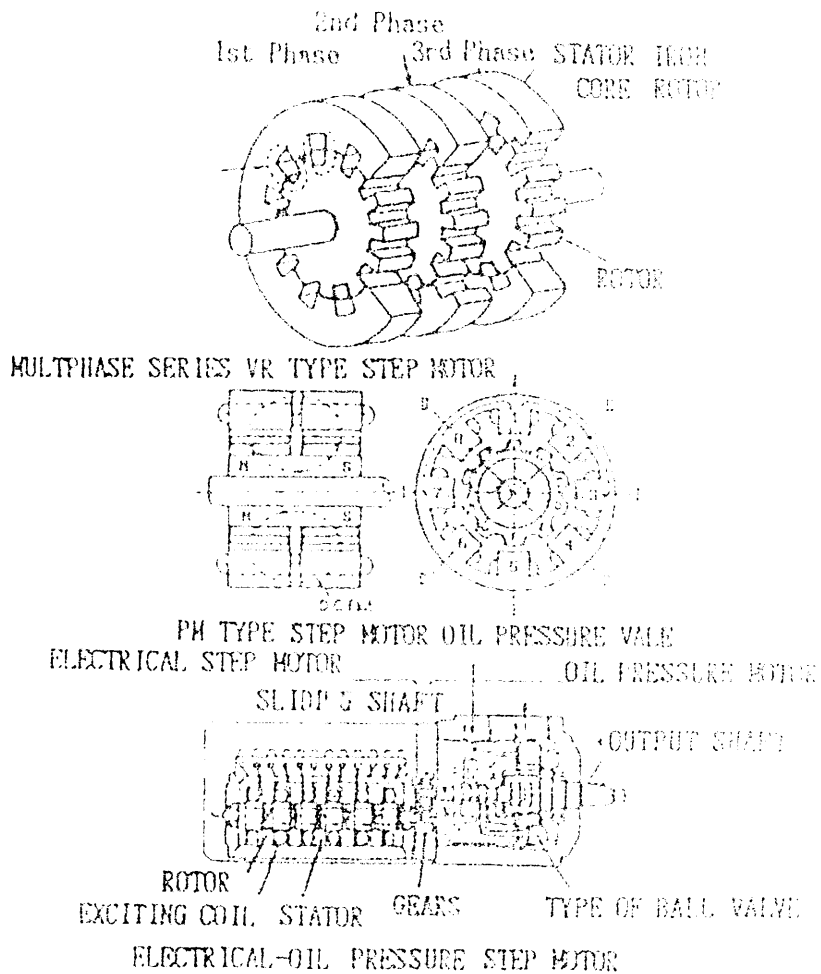
Motor stepper berputar dengan tahapan (step) yang tetap dari satu posisi ke posisi yang lain. Besar pergeseran step tergantung dari konstruksi motor. Besar tahapan atau step ini disebut dengan derajat step atau step angel (SA). Angka ini berkisar: 1,8 - 2,5 - 3,75 - 7,5 - 15 dan 30⁰.



Gambar 2.14 Tipe-tipe Motor Stepper

Berdasarkan prinsip dan strukturnya, motor stepper dapat dibagi menjadi beberapa tipe seperti yang ditunjukkan pada gambar 2.14 diatas.

Motor stepper mempunyai struktur yang lebih sederhana dibandingkan dengan motor-motor listrik yang lain, terutama rotornya yang terbuat dari besi magnet permanen. Pada gambar 2.15 berikut ini menggambarkan struktur motor stepper tipe VR, hybrid PM dan electrical oil pressure.



Gambar 2.15 Struktur Motor Stepper

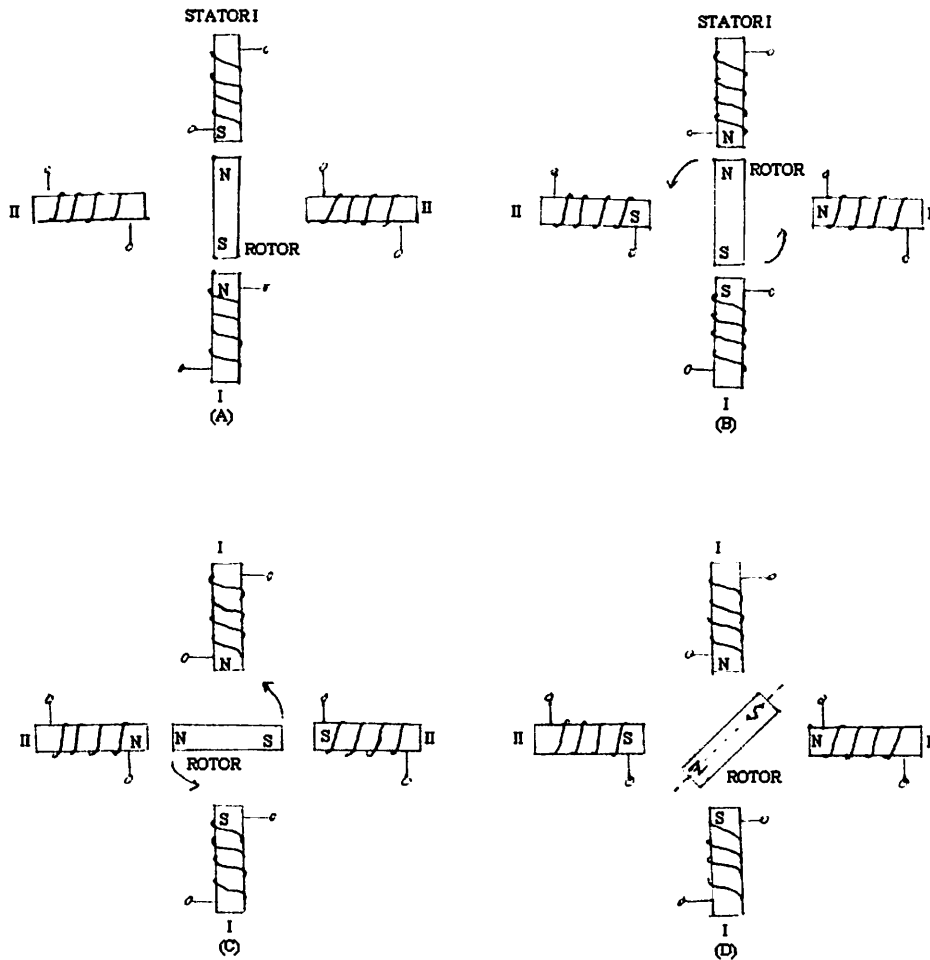
Step angel berhubungan dengan jumlah gigi (teeth) dari stator (N_s) dan rotor (N_r).

$$\text{Step Angel} = 360^\circ \frac{(N_s - N_r)}{(N_s \times N_r)}$$

$$\text{Jumlah Step/Rev} = \frac{(N_s \times N_r)}{(N_s - N_r)} \text{ atau}$$

$$\text{Jumlah Step . Rev} = \frac{360^\circ}{\text{StepAngel}}$$

Untuk mempermudah memahami prinsip kerja motor stepper, dianggap rotor memiliki 2 kutub dan statornya 4 kutub. Prinsip utama berbagai tipe motor stepper pada umumnya sama dan prinsip kerjanya dapat dianalisa dengan bantuan gambar 2.16 berikut ini:

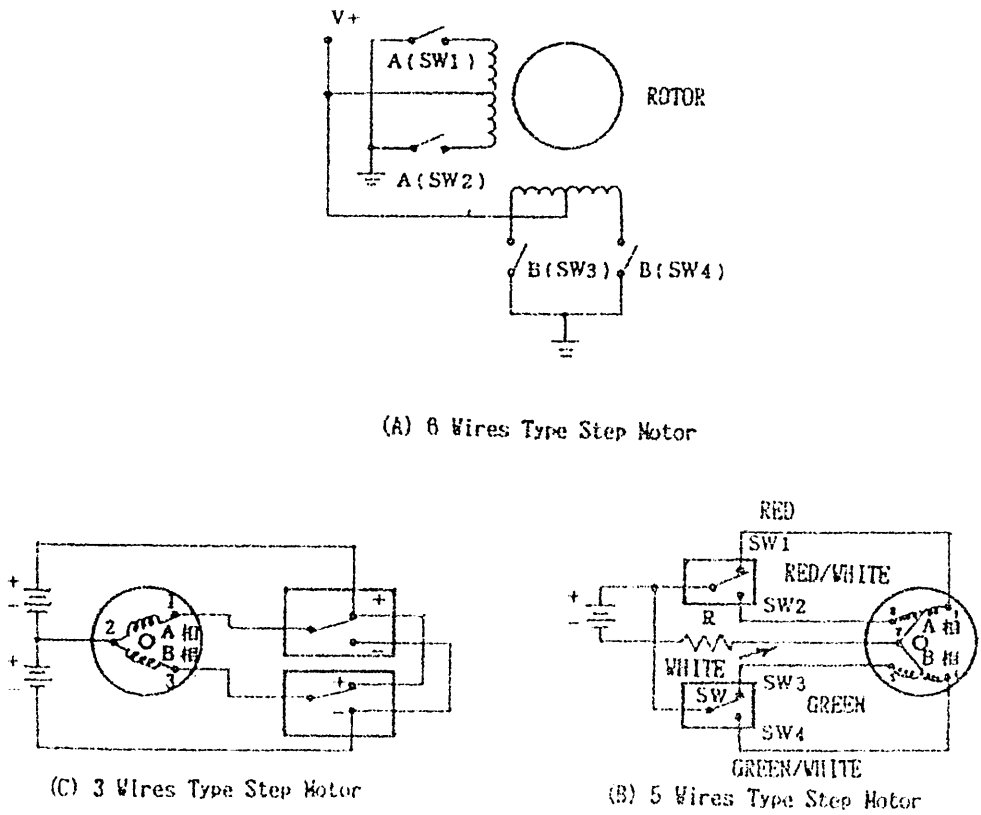


Gambar 2.16 Prinsip Kerja Motor Stepper

Dilapangan sering dijumpai motor stepper dengan macam sebagai berikut:

1. tipe 3 – kabel
2. tipe 5 – kabel
3. tipe 6 – kabel

Pada gambar 2.17 berikut ini menunjukkan tipe-tipe tersebut diatas dengan rangkaian drivernya. Switch pada gambar tersebut menggunakan transistor switching.



Gambar 2.17 Rangkaian Driver Untuk Berbagai Tipe Motor Stepper

Eksitasi 1-2 fasa disebut dengan mode hal-step, sehingga step angel akan dua kali lebih kecil. Sebagai contoh, jika kumparan motor stepper memiliki step angel $1,8^{\circ}$ dan dikendalikan dengan eksitasi 1-2 fasa, maka gerak per step akan $0,9^{\circ}$. Dari ketiga macam eksitasi diatas, eksitasi 2 fasa mempunyai torsi yang relatif lebih besar.

Tabel 2.3 Macam-macam Eksitasi

STEP	SW 1	SW 2	SW 3	SW 4
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	OFF	OFF	ON	OFF
4	OFF	OFF	OFF	ON
5	ON	OFF	OFF	OFF
6	OFF	ON	OFF	OFF
7	OFF	OFF	ON	OFF
8	OFF	OFF	OFF	ON

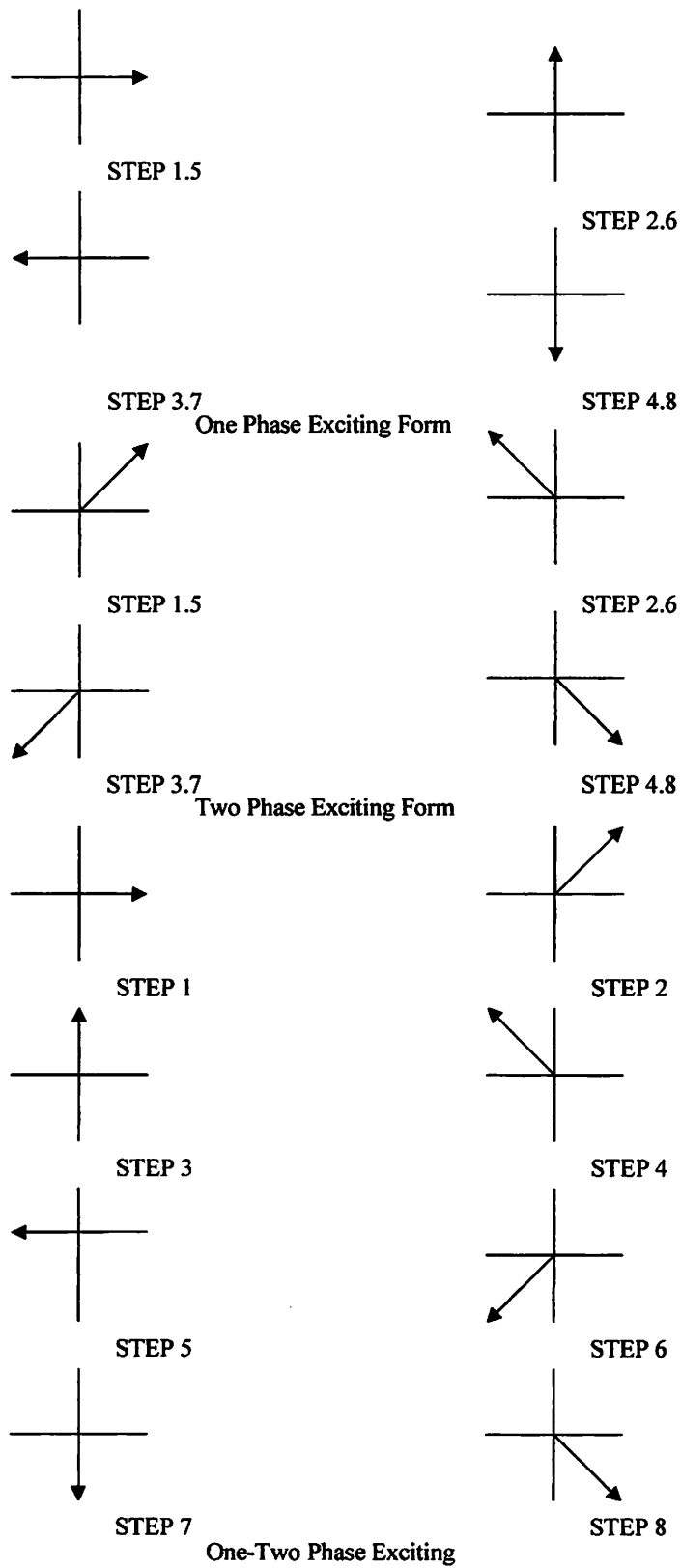
(A) One Phase Exciting Form

STEP	SW 1	SW 2	SW 3	SW 4
1	ON	ON	OFF	OFF
2	OFF	ON	ON	OFF
3	OFF	OFF	ON	ON
4	ON	OFF	OFF	ON
5	ON	ON	OFF	OFF
6	OFF	ON	ON	OFF
7	OFF	OFF	ON	ON
8	ON	OFF	OFF	ON

(B) Tow Phase Exciting Form

STEP	SW 1	SW 2	SW 3	SW 4
1	ON	OFF	OFF	OFF
2	ON	ON	OFF	OFF
3	OFF	ON	OFF	OFF
4	OFF	ON	ON	OFF
5	OFF	OFF	ON	OFF
6	OFF	OFF	ON	ON
7	OFF	OFF	OFF	ON
8	ON	OFF	OFF	ON

(C) One-Two Phase Exciting Form



Gambar 2.18 Macam-macam Eksitasi

BAB III

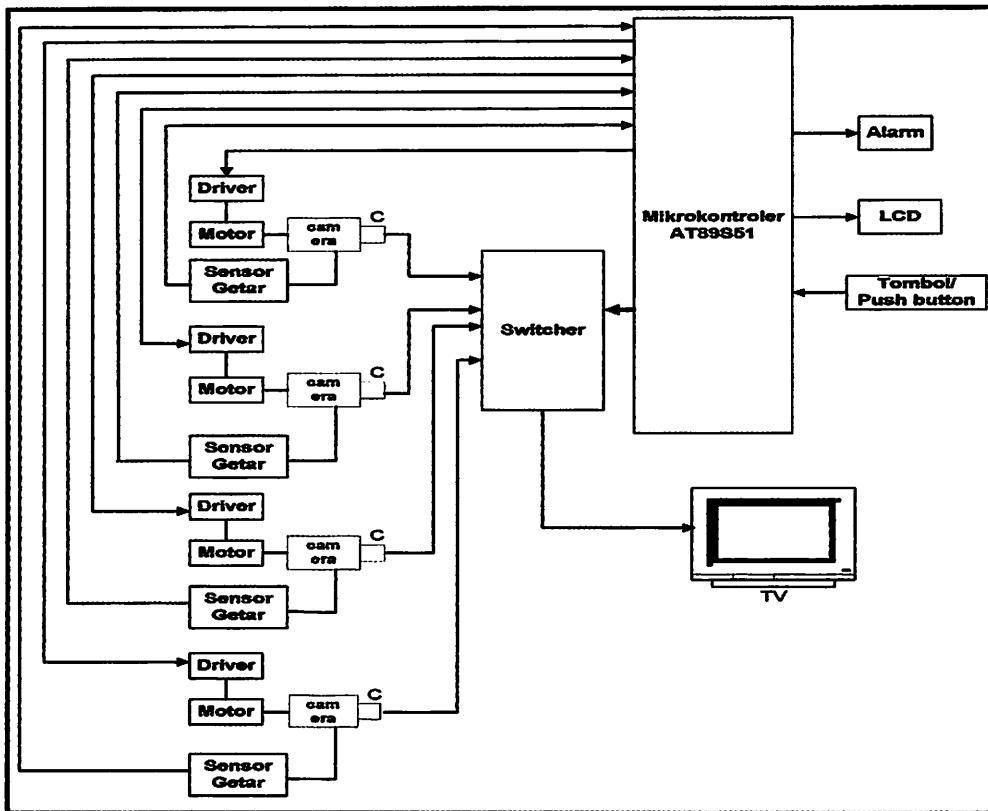
PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai peralatan yang direncanakan dan akan direalisasikan sebagaimana fungsinya. Adapun perencanaan dan pembuatan alat meliputi : perencanaan dan pembuatan perangkat keras serta perangkat lunak.

3.1. Perencanaan Perangkat keras

Pembuatan *video switcher* untuk monitoring gedung ini hanya dalam bentuk simulasi. Pada dasarnya perancangan meliputi dua pokok pembahasan, perangkat keras dan perangkat lunaknya.

Komponen utama dari perangkat ini adalah mikrokontroler AT89S51 yang berfungsi sebagai unit pengendali otomatisasi pada *video switcher*. Berikut adalah diagram blok *video switcher* untuk monitoring gedung:

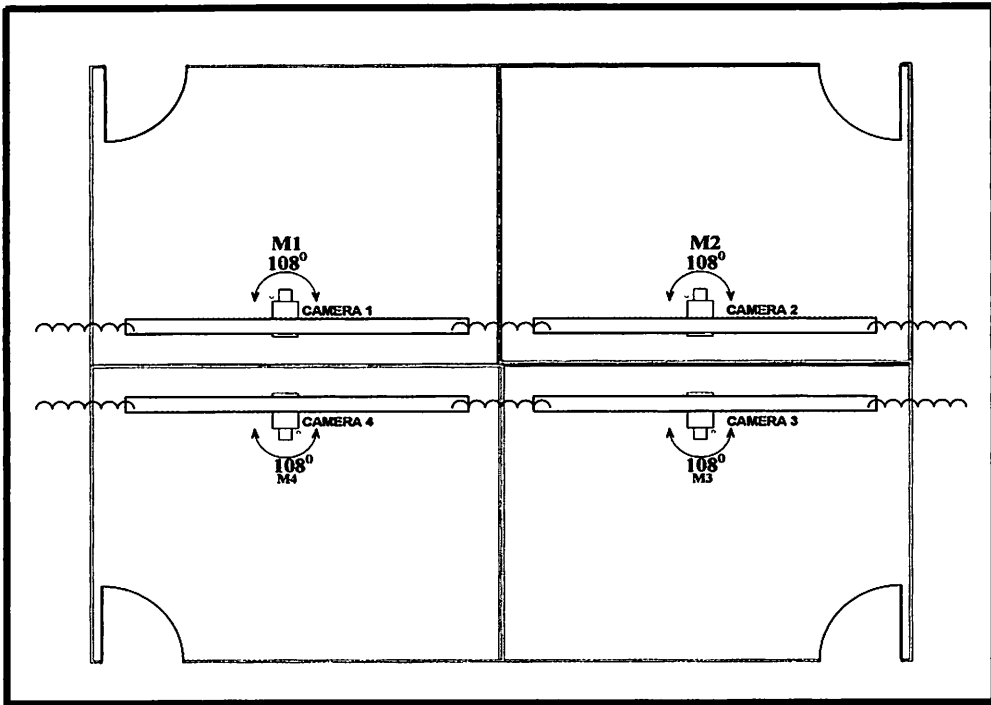


Gambar 3.1 Diagram blok *video switcher* untuk monitoring gedung

Dari diagram blok diatas, maka prinsip kerja dari alat *video switcher* untuk monitoring gedung yang berbasis mikrokontroler AT89S51 sebagai berikut:

- Kamera disini digunakan sebagai untuk mengambil gambar atau inputan yang akan ditampilkan oleh sebuah TV.
- Motor disini digunakan untuk menggerakkan kamera kekanan dan kekiri, motor ini berputar 108° .
- TV disini berfungsi untuk menghasilkan gambar yang telah diambil oleh Kamera, sedangkan gambar yang dihasilkan oleh TV tergantung dari banyaknya kamera.

- *Switcher* disini berfungsi untuk *menswitching* kamera secara otomatis yang dikendalikan oleh sebuah mikrokontroller. Pada saat proses *switching* ini kamera yang akan dimulai adalah dari kamera1 sampai kamera 4, kemudian sebaliknya pada saat kamera sampai dikamera4 kamera akan kembali lagi kekamera1. *Switcher* disini juga bisa dikendalikan secara manual.
- LCD disini digunakan untuk menampilkan kondisi pada *video switcher* yakni kondisi otomatis atau manual. Serta juga dapat menampilkan kondisi kamera yang mana telah digunakan, tetapi hanya pada saat manual.
- Sensor getar disini digunakan untuk mengetahui apabila kamera telah mendapat gangguan seperti merusak pada kamera.
- Alarm disini berfungsi sebagai tanda apabila ada gangguan pada kamera.
- Tombol/*Limit switch* disini digunakan untuk mengaktifkan *video switcher* secara otomatis atau manual, serta tombol disini juga untuk memilih kamera antara kamera satu dengan kamera yang lain secara manual. Tombol disini juga dapat dapat menggerakkan kamera yaitu gerak kekiri dan kekanan secara manual apabila *video switcher* bekerja secara manual.
- Mikrokontroler disini berfungsi sebagai pusat pengendali dari keseluruhan alat *video switcher*.



Gambar 3.2 Simulasi alat

Keterangan:

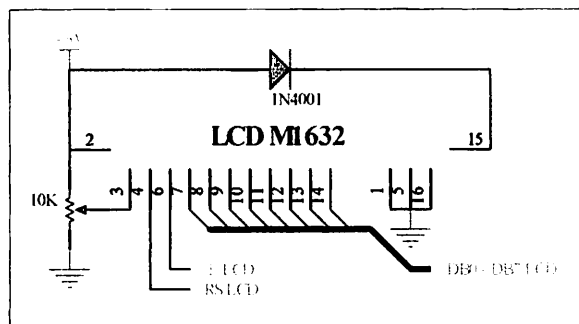
1. Kamera 1, 2, 3 dan 4 berfungsi untuk mengambil gambar, yang dimana kamera ini mengambil gambar secara bergantian dengan otomatis dari kamera 1 sampai kamera 4, dan kamera ini bisa dijalankan secara manual. Pada simulasi untuk kamera yang digunakan adalah CCTV(tidakberwarna).
2. Sensor getar ini berfungsi sebagai indicator apabila terdapat gangguan terhadap kamera atau kerusakan pada kamera. Pada simulasi alat saya sensor getar ini ditaruh sebuah lempengan kayu, pada kayu tersebut diberi sebuah per yang ditengah-tengahnya diberi sebuah inti besi. Sensor getar pada simulasi alat saya hampir

sama prinsip kerjanya seperti saklar.

3. M1, M2, M3 dan M4 adalah motor penggerak untuk kamera, motor disini berfungsi untuk mengerakan kamera kekanan dan kekiri. Dimana motor ini berputar 108° .

3.1.1. LCD (Liquid Cristal Display)

LCD ini digunakan sebagai display yang akan menampilkan suatu kondisi kamera yaitu otomatis atau manual dan menampilkan kamera mana yang telah digunakan yang pada saat itu dalam kondisi manual. Dengan adanya penampil ini diharapkan alat lebih mudah dioperasikan. Data bus pada LCD akan dihubungkan pada Port 0 di mikrokontroler dan pin RS dan E akan dihubungkan di P2.0 dan P2.1.



Gambar 3.3 Rangkaian LCD

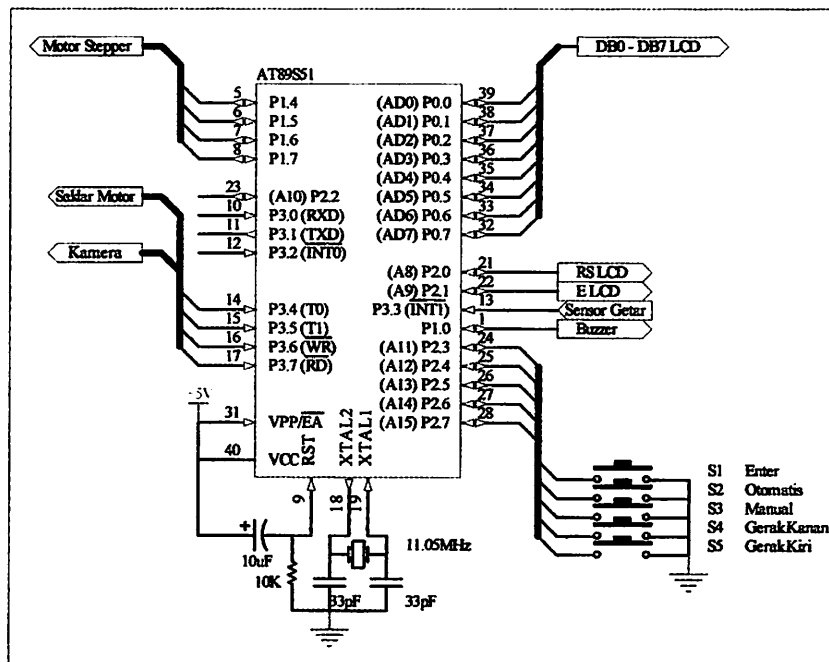
3.1.2 Perancangan Minimum Sistem AT89S51

Mikrokontroller yang digunakan dalam perancangan alat ini adalah IC mikrokontroler AT89S51. Mikrokontroler AT89S51 harus didukung oleh beberapa rangkaian lain agar dapat melakukan prosesnya, yaitu berupa rangkaian *clock* dan *reset*. Semua pengaturan dan pengolahan data dilakukan oleh

mikrokontroler. Gambar rangkaian secara keseluruhan dapat dilihat pada lampiran.

Penggunaan port Mikrokontroler AT89S51 pada rancangan ini adalah sebagai berikut :

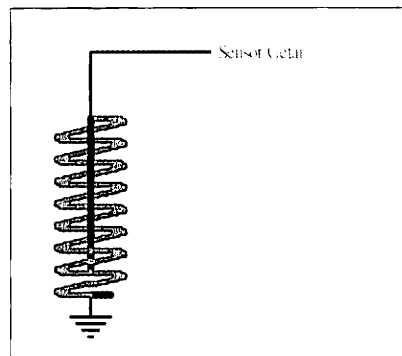
- 1) P1.4 – P1.7 adalah 4pin yang menggerakkan motor stepper.
- 2) P0.0 – P0.7 digunakan sebagai jalur data LCD.
- 3) P2.0 & P2.1 digunakan sebagai jalur kontrol LCD.
- 4) P2.3 – P2.7 digunakan sebagai tombol.
- 5) P3.3 digunakan sebagai sensor getar.
- 6) P3.4 – P3.7 digunakan untuk memilih kamera dan motor mana yang akan diaktifkan.
- 7) P1.0 digunakan untuk buzzer.



Gambar 3.4 Rangkaian Minimum Sistem AT89S51

3.1.3. Rangkaian Sensor Getar

Sensor getar berfungsi untuk mendeteksi adanya kerusakan pada kamera, sensor getar terdiri dari 2 bagian yaitu inti besi dan per. Inti besi diletakkan ditengah-tengah per, dan bagian ini besi dihubungkan kemikrokontroler.



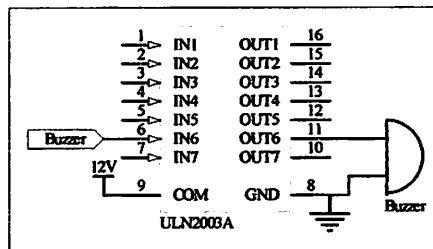
Gambar 3.5 Rangkaian Sensor Getar

Analisa rangkaian sensor getar:

Pada saat kondisi sensor tidak mendapatkan getaran atau guncangan(tidak aktif) maka outputan dari sensor getar tersebut berlogika high, sehingga intraf yang sudah set program tidak akan aktif. Pada saat sensor getar mendapat getaran atau guncangan(aktif) maka tegangan berlogika rendah (0V) akan mengalir keport 3.3(intraf 1). Karena port ini berlogika 0 maka alamat 13H akan aktif. maka alarm akan aktif.

3.1.4. Rangkaian Driver *Buzzer*

Bila sensor getar aktif maka *buzzer* juga akan aktif sebagai tanda bahwa ada kerusakan pada kamera. Pada perencanaan alat ini IC ULN 2003A adalah IC driver yang dikemas dalam chips sebagai inputan dari *buzzer*. Driver *buzzer* seperti gambar berikut :



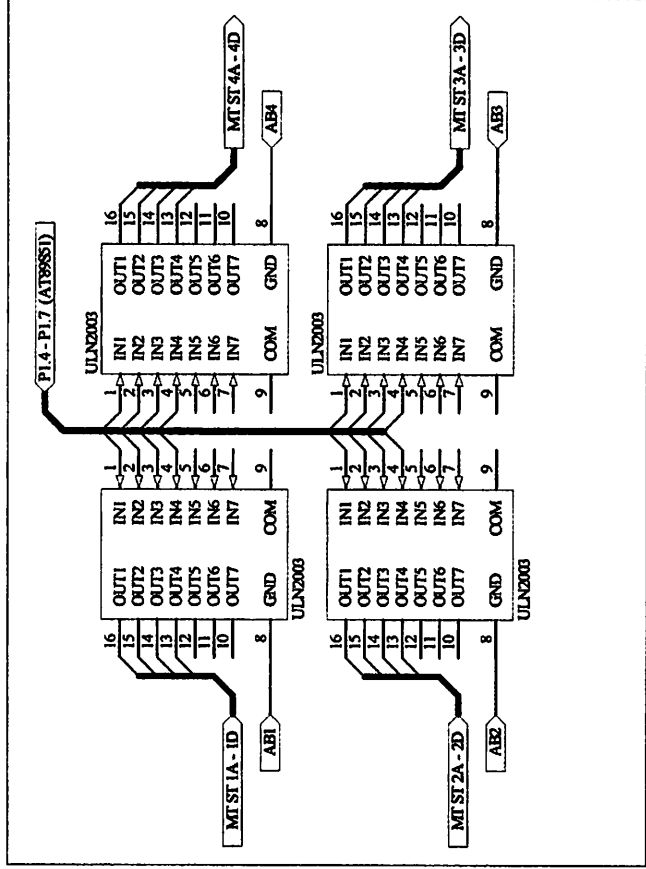
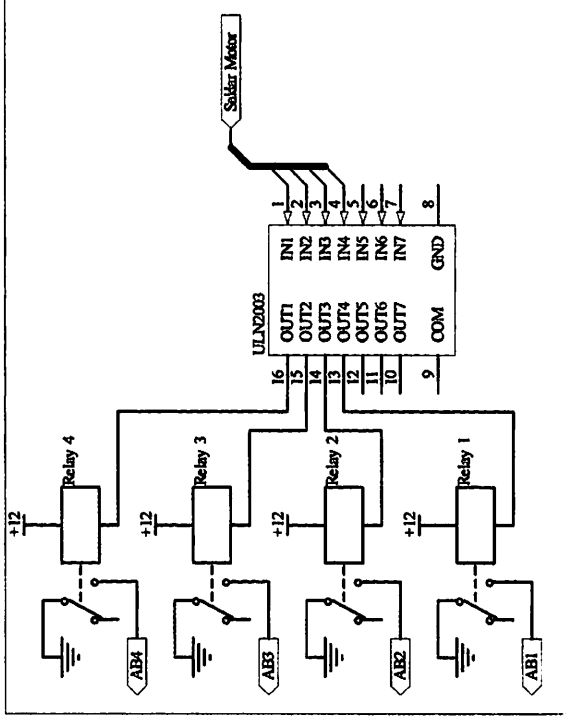
Gambar 3.6 Rangkaian Driver *Buzzer*

3.1.5. Rangkaian Driver Motor Stepper

Untuk rangkaian driver motor stepper digunakan IC ULN 2003A sebagai pengendali arah putaran mekanik motor stepper. Ic ULN 2003A mampu menghidupkan dan mematikan relay yang hanya memiliki tegangan maksimal sebesar 12 Volt dengan resistansi kumparan sebesar 380 Ω jadi dapat diketahui arus relay sebesar :

$$\begin{aligned} \text{Dimana : } I_{\text{relay}} &= \frac{VCC}{R_{\text{relay}}} \\ &= \frac{12}{380} = 31.57 \text{ mA} \end{aligned}$$

Jika relay membutuhkan arus sebesar 30 mA maka IC ULN2003A dapat menggerakkan relay tersebut, karena ULN 2003A mampu melewatkan arus maksimum sebesar 500 mA sesuai dengan data sheet. Dibawah ini adalah rangkaian driver relay motor stepper :



Gambar 3.7 Rangkaian Driver Motor Stepper

Analisa rangkaian motor stepper:

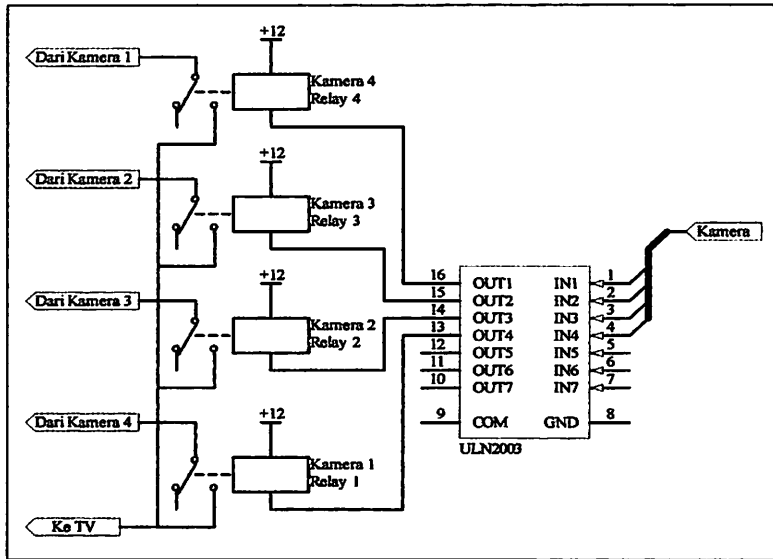
Pada common diberi tegangan sebesar 5V, dan untuk menggerakkan motor stepper per step maka pada pin 1-4 diberi logika rendah secara bergantian dan berurutan. Pada perencanaan yang diinginkan motor stepper ini berputar 108° , karena motor stepper yang digunakan memiliki $1,8^{\circ}$ per step. Maka perulangan untuk pemberian logika rendah pada ke4 pin pada motor stepper yaitu sebanyak 60 kali.

3.1.6. Sistem Pemilih Untuk Mengaktifkan Kamera

Sistem yang digunakan dalam proses pemilihan kamera yang akan ditampilkan pada sebuah televisi menggunakan sistem *switch* yang dikendalikan oleh tegangan (relay). Dimana relay ini tidak bisa langsung dihubungkan dengan mikrokontroler, maka diperlukan sebuah rangkaian *switch* yang dapat menghantarkan arus, yang arusnya sebesar 30ma yaitu sesuai dengan arus dari relay. Adapun *switch* yang digunakan adalah berupa IC ULN 2003A dimana memiliki arus sebesar 500ma. Adapun perhitungan arus relay dapat dilihat dibawah ini:

$$\begin{aligned} \text{Dimana : } I_{\text{relay}} &= \frac{VCC}{R.\text{relay}} \\ &= \frac{12}{380} = 31.57 \text{ mA} \end{aligned}$$

Jadi relay membutuhkan arus sebesar 30 mA maka IC ULN2003A dapat menggerakkan relay tersebut, karena ULN 2003A mampu melewati arus maksimum sebesar 500 mA sesuai dengan data sheet. Dibawah ini adalah rangkaian sistem pemilih untuk mengaktifkan kamera:



Gambar 3.8 Rangkaian Sistem Pemilih Untuk Mengaktifkan Kamera

Analisa sistem kerja pemilih untuk mengaktifkan kamera:

❖ Pada saat kondisi otomatis:

Adalah dengan cara memilih atau menekan tombol otomatis, dengan memilih atau menekan tombol ini maka program akan aktif pada sub program otomatis, dimana prosesnya kamera1 dan relay1 diaktifkan maka yang lain tidak aktif. Dan ini akan langsung mengaktifkan motor stepper untuk bisa bergerak kearah kanan sebanyak 60step dan akan kembali lagi kearah kiri sebanyak 60step. Setelah motor telah sampai 60step kearah kiri maka program akan mematikan kamera1 dan relay1 dan mengaktifkan kamera2 dan relay2 yang prosesnya sama dengan kamera1 dan relay1.

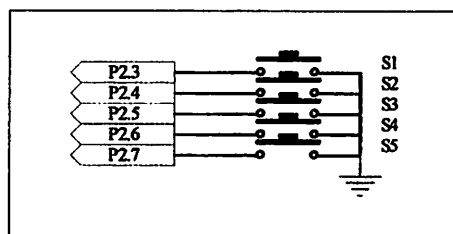
Pada saat proses sistem pemilih telah berada pada kamera4 dan relay4 maka kamera dan relay akan mengaktifkan kembali kamera1 dan relay1. Sehingga pada sistem pemilih kamera ini akan terus berjalan apabila sistem otomatis ini diaktifkan.

❖ Pada saat kondisi manual:

Dengan cara memilih atau menekan tombol manual maka mikrokontroler/sub program akan menjalankan sub program manual, dimana prosesnya adalah dengan cara memilih dan menekan tombol untuk menentukan kamera dan relay mana yang akan diaktifkan.

3.1.7. Push Button

Push button pada perancangan ini digunakan sebagai input dalam pemilihan menu. Adapun prinsip kerja dari tombol adalah apabila salah satu tombol ini ditekan maka tombol disini akan langsung terhubung pada ground sehingga keluaran dari tombol ini berlogika low yang kemudian digunakan sebagai input oleh mikrokontroler. Adapun rangkaian dari push button ini dapat dilihat pada gambar 3.9.



Gambar 3.9 Rangkaian Push Button

1. S1 digunakan untuk enter yang dihubungkan dengan port 2.3.
2. S2 digunakan untuk otomatis yang dihubungkan dengan port 2.4.
3. S3 digunakan untuk manual yang dihubungkan dengan port 2.5.
4. S4 digunakan untuk gerak kekanan yang dihubungkan dengan port 2.6.
5. S5 digunakan untuk gerak kekiri yang dihubungkan dengan port 2.7.

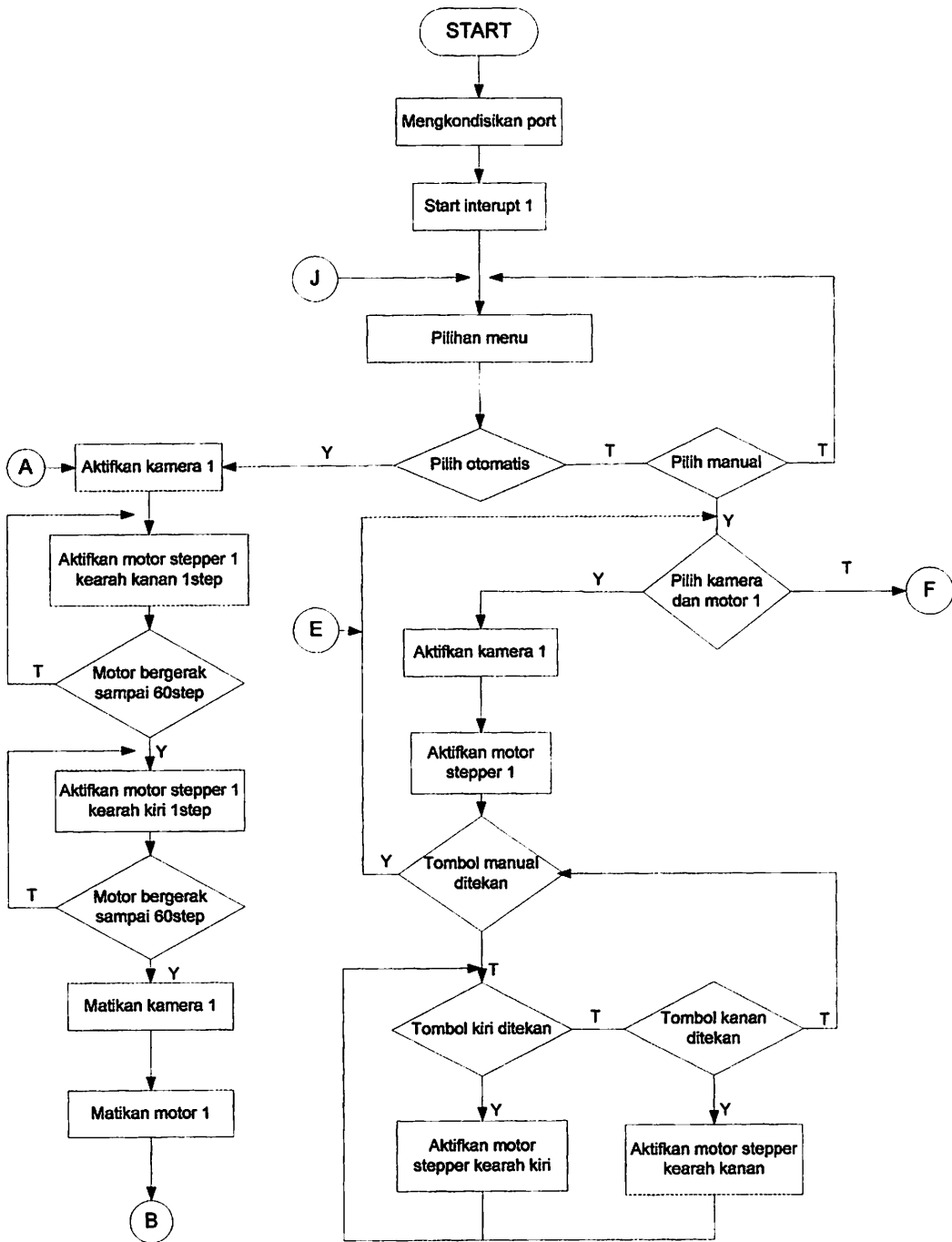
3.2. Perancangan Perangkat Lunak (Software)

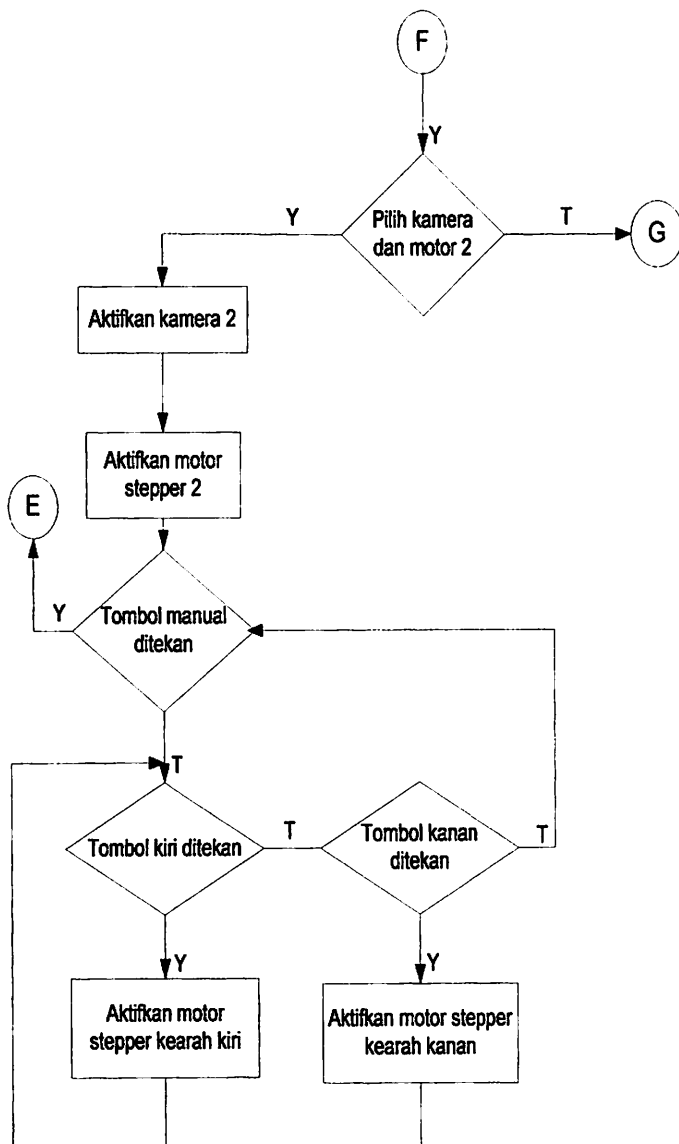
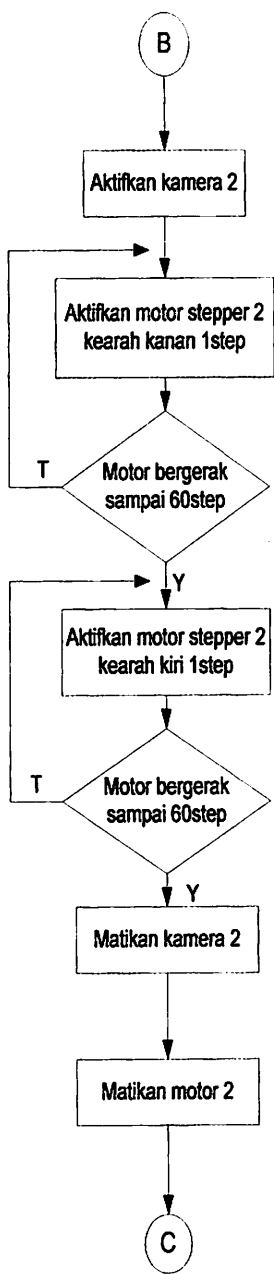
Untuk mendukung *hardware* yang sudah dibuat, maka dibutuhkan perangkat lunak (*software*) supaya perangkat keras tersebut bisa berjalan sesuai dengan tujuan. Mikrokontroler dapat mengendalikan seluruh sistem apabila ada urutan instruksi yang mendefinisikan secara jelas urutan kerja yang harus dilaksanakan. Dalam perancangan alat ini perangkat lunak yang digunakan adalah bahasa pemrograman *assembler*.

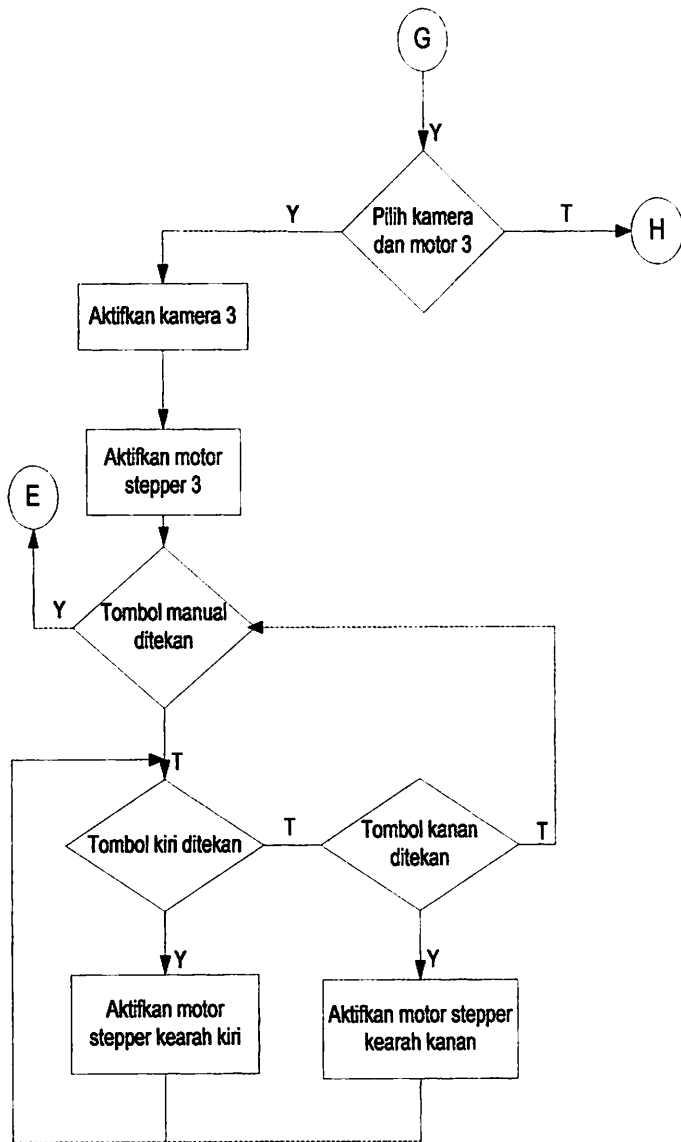
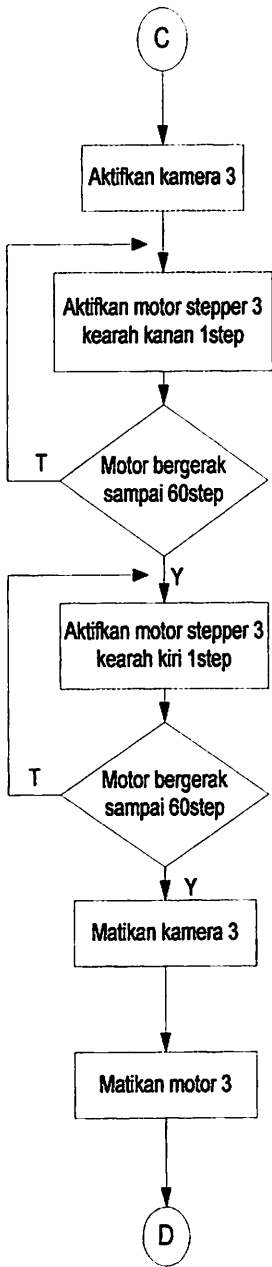
Sebelum membuat perangkat lunak, terlebih dahulu dibuat diagram alir (*flowchart*) dari proses yang akan dibuat supaya memudahkan dalam pembuatan perangkat lunak (*software*).

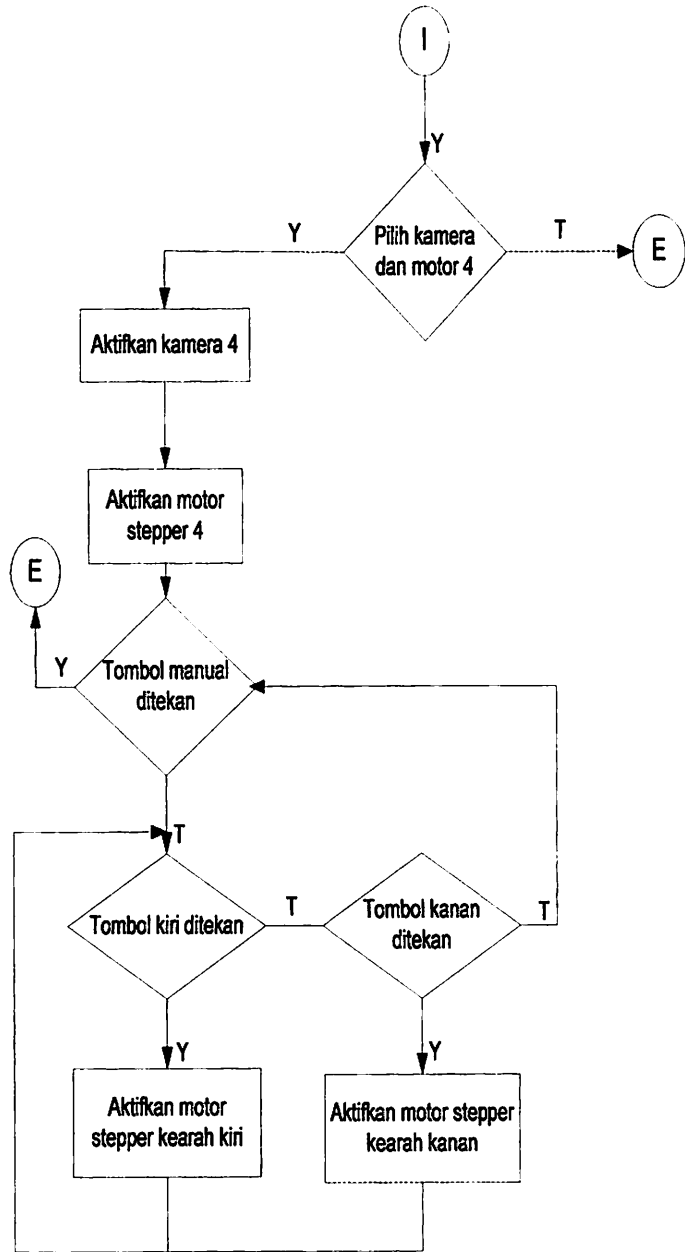
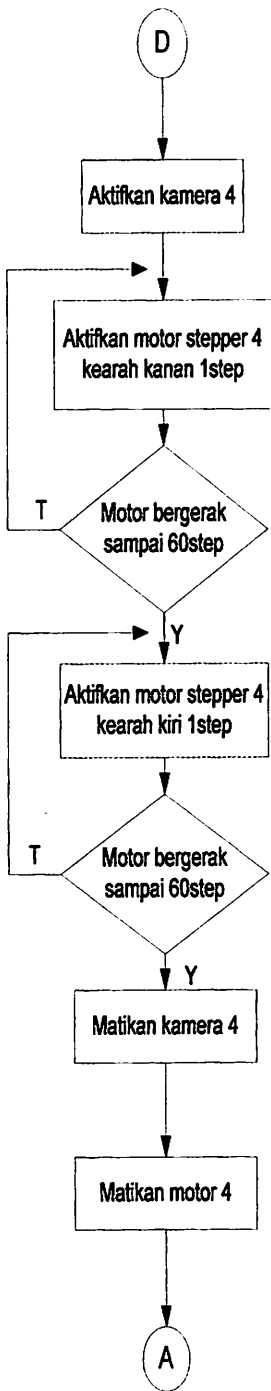
3.2.1. Diagram Alir

3.2.1.1 Diagram alir utama

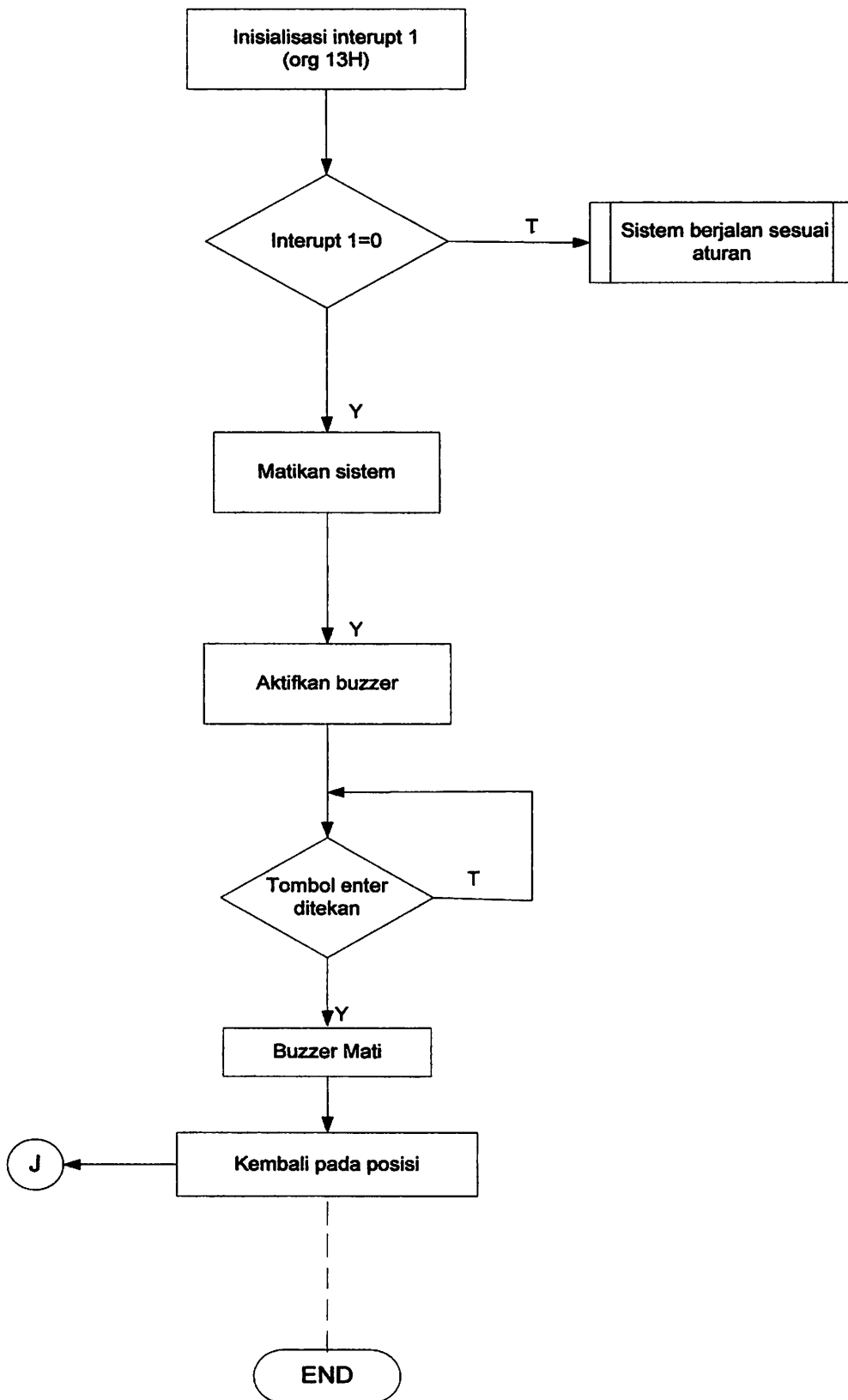








3.2.1.2 Diagram Alir Layanan Interupt



BAB IV

PENGUJIAN ALAT

Pengujian alat ini dilakukan untuk mengetahui kinerja dari keseluruhan sistem rangkaian. Jadi pada tahap ini akan diketahui nilai-nilai serta parameter-parameter dari setiap bagian yang menyusun sistem secara keseluruhan.

4.1. Pengujian Rangkaian Sensor Getar

4.1.1. Tujuan pengujian

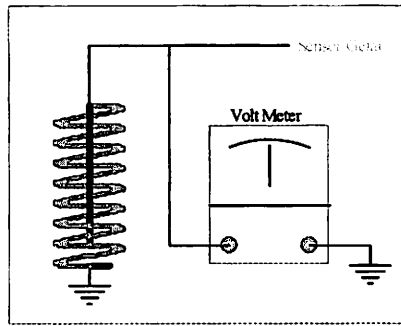
Untuk mengetahui apakah rangkaian sensor getar yang telah dibuat dapat bekerja sesuai dengan yang direncanakan.

Peralatan yang digunakan

1. Multimeter digital.
2. Rangkaian sensor getar.
3. Catu daya 5V.

4.1.2. Prosedur Pengujian

1. Membuat rangkaian seperti pada gambar 4-1
2. Pengukuran dilakukan dalam dua kondisi yaitu saat tidak ada getaran dan ada getaran.
3. Memasukkan hasil pengukuran pada tabel 4.1



Gambar 4.1 Pengujian Rangkaian Sensor Getar

4.1.3. Hasil Analisa Pengujian

Tabel 4.1.

Hasil Pengujian Rangkaian Sensor Getar

Kondisi	Tegangan (volt)
Ada Getaran	0,01 volt
Tidak Ada Getaran	4,30volt

Dari hasil pengujian rangkaian sensor getar pada tabel 4.1. diatas, maka dapat diketahui bahwa pada saat sensor digetar maka inti besi akan mengenai per, sehingga outputan dari sensor getar berlogika low dan merupakan inputan ke mikrokontroller, sebaliknya pada saat kondisi tidak ada getaran maka inti besi tidak mengenai per sehingga berlogika high sebagai inputan mikrokontroller.

4.2. Pengujian Rangkaian ULN 2003

4.2.1. Tujuan pengujian

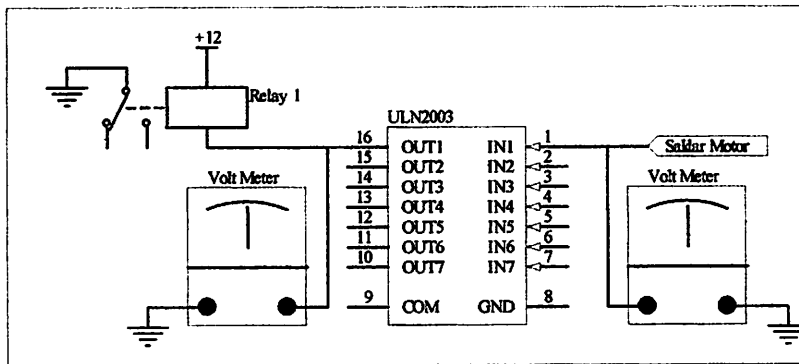
Untuk mengetahui apakah rangkaian ULN 2003 yang digunakan dapat berfungsi dengan yang direncanakan.

4.2.2 Peralatan yang digunakan

1. Rangkaian ULN 2003.
2. Multimeter digital
3. Catu daya 5V

4.2.3. Prosedur Pengujian

1. Alat dirangkai seperti dalam Gambar 4-3.
2. Mengukur logika tinggi dan rendah pada outputan mikrokontroller yang menuju inputan ULN.
3. Mengukur outputan ULN pada saat logika tinggi dan rendah dan merupakan inputan relay.



Gambar 4.2 Pengujian rangkaian ULN 2003

4.2.4. Hasil Analisa Pengujian

Tabel 4.2.
Hasil pengukuran output mikrokontroller dan output ULN

NO	Logika	Output mikrokontroller	Output ULN	Kondisi relay
1	1	4,70 volt	0,79 volt	Aktif
2	0	0,06 volt	4,66 volt	Tidak aktif

Dari hasil pengujian didapatkan pada waktu mikrokontroller memberi logika “1” tegangannya sebesar 4,70 volt. Besarnya tegangan tersebut diubah oleh IC ULN menjadi logika low yaitu sebesar 0,79 volt relay akan menjadi aktif. Pada saat mikrokontroller memberi logika “0” tegangannya sebesar 0,06 volt. Besar tegangan tersebut diubah oleh IC ULN menjadi logika high yaitu sebesar 4,66 volt relay akan menjadi tidak aktif.

4.3.Pengujian Pemilihan Kamera Dan Motor Stepper

4.3.1. Tujuan pengujian

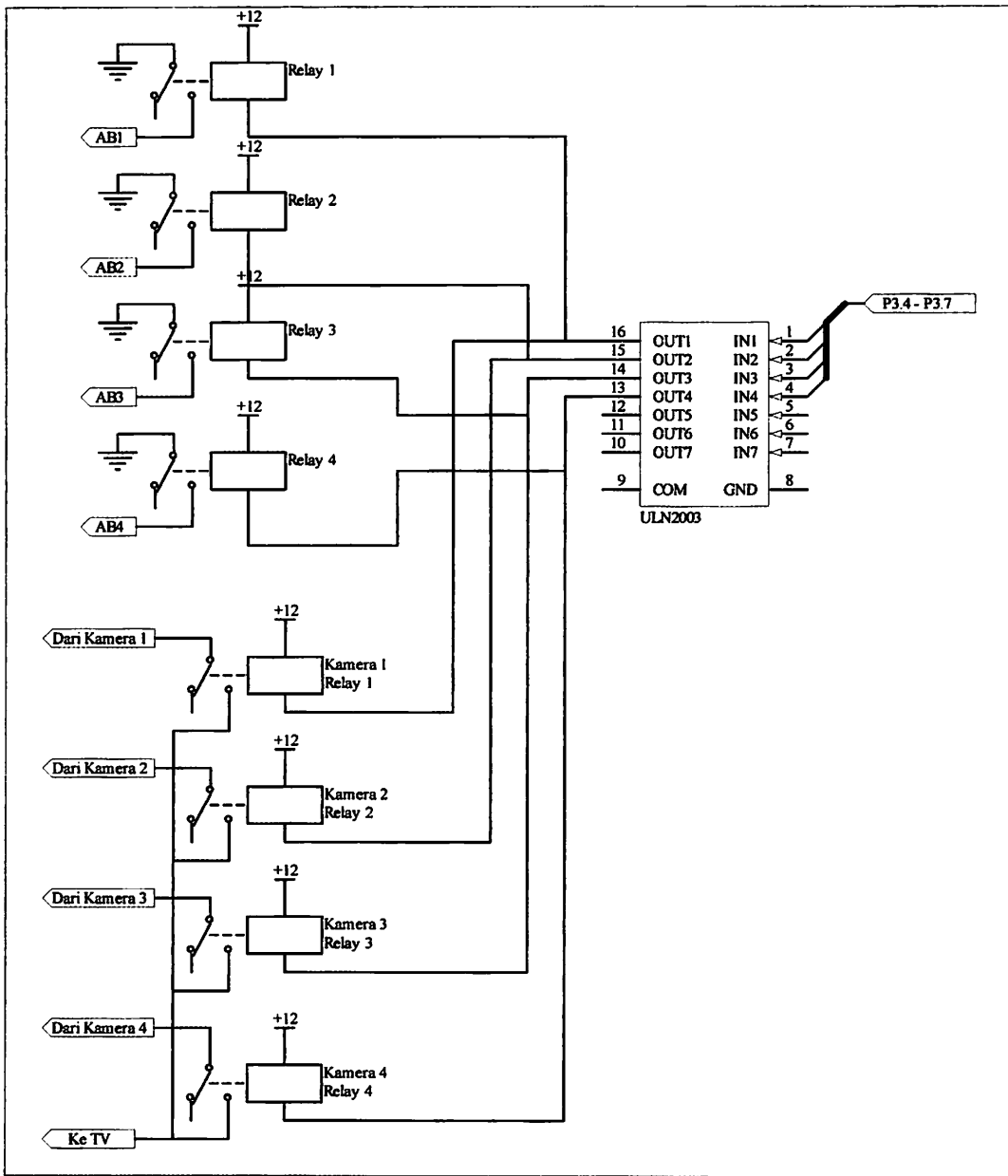
Untuk mengetahui apakah switch yang digunakan sudah bisa memilih kamera dan motor stepper yang digunakan.

4.3.2. Peralatan yang digunakan

1. Televisi
2. kamera CCTV(tidak berwarna).
3. Motor stepper.
4. Power supply kamera 9V.
5. Catu daya 5V.

4.3.3. Prosedur pengujian

1. Alat dirangkai seperti gambar 4.3
2. Memberikan catu daya ke kamera, ketelevisi dan kerangkaian switch.
3. Memasukkan inputan kerangkaian switch yang kameranya yang akan dipilih.



Gambar 4.3 Rangkaian pemilih kamera dan motor stepper

4.3.4. Hasil Analisa Pengujian



Gambar 4.4 Hasil pengujian pemilih kamera dan motor stepper

Tabel 4.3

Hasil pengujian rangkaian pemilih kamera dan motor stepper

NO	DATA INPUTAN				KETERANGAN
	P3.7	P3.6	P3.5	P3.4	
1	0	0	0	1	Kamera dan motor 1 aktif
2	0	0	1	0	Kamera dan motor 2 aktif
3	0	1	0	0	Kamera dan motor 3 aktif
4	1	0	0	0	Kamera dan motor 4 aktif

Dari hasil pengujian rangkaian kamera dan motor stepper pada tabel 4.3 diatas, maka dapat diketahui bahwa kamera dan motor 1 aktif apabila port 3.4

diberi logika 1. Kamera dan motor 2 aktif apabila port 3.5 diberi logika 1. Kamera dan motor 3 aktif apabila port 3.6 diberi logika 1. Kamera dan motor 4 aktif apabila port 3.7 diberi logika 1.

Tabel 4.4

Pengujian motor stepper (berputar Kekakanan)

NO	DATA INPUTAN				KETERANGAN
	P1.7	P1.6	P1.5	P1.4	
1	0	0	0	1	Motor berputar 1,8 ⁰ kekanan
2	0	0	1	0	Motor berputar 3,6 ⁰ kekanan
3	0	1	0	0	Motor berputar 5,4 ⁰ kekanan
4	1	0	0	0	Motor berputar 7,2 ⁰ kekanan
5	0	0	0	1	Motor berputar 9 ⁰ kekanan
6	0	0	1	0	Motor berputar 10,8 ⁰ kekanan
7	0	1	0	0	Motor berputar 12,6 ⁰ kekanan
8	1	0	0	0	Motor berputar 14,4 ⁰ kekanan
9	0	0	0	1	Motor berputar 16,2 ⁰ kekanan
10	0	0	1	0	Motor berputar 18 ⁰ kekanan
11	0	1	0	0	Motor berputar 19,8 ⁰ kekanan
12	1	0	0	0	Motor berputar 21,6 ⁰ kekanan
13	0	0	0	1	Motor berputar 23,4 ⁰ kekanan
14	0	0	1	0	Motor berputar 25,2 ⁰ kekanan
15	0	1	0	0	Motor berputar 27 ⁰ kekanan

16	1	0	0	0	Motor berputar 28,8 ⁰ kekanan
17	0	0	0	1	Motor berputar 30,6 ⁰ kekanan
18	0	0	1	0	Motor berputar 32,4 ⁰ kekanan
19	0	1	0	0	Motor berputar 34,2 ⁰ kekanan
20	1	0	0	0	Motor berputar 36 ⁰ kekanan
21	0	0	0	1	Motor berputar 37,8 ⁰ kekanan
22	0	0	1	0	Motor berputar 39,6 ⁰ kekanan
23	0	1	0	0	Motor berputar 41,4 ⁰ kekanan
24	1	0	0	0	Motor berputar 43,2 ⁰ kekanan
25	0	0	0	1	Motor berputar 45 ⁰ kekanan
26	0	0	1	0	Motor berputar 46,8 ⁰ kekanan
27	0	1	0	0	Motor berputar 48,6 ⁰ kekanan
28	1	0	0	0	Motor berputar 50,4 ⁰ kekanan
29	0	0	0	1	Motor berputar 52,2 ⁰ kekanan
30	0	0	1	0	Motor berputar 54 ⁰ kekanan
31	0	1	0	0	Motor berputar 55,8 ⁰ kekanan
32	1	0	0	0	Motor berputar 57,6 ⁰ kekanan
33	0	0	0	1	Motor berputar 59,4 ⁰ kekanan
34	0	0	1	0	Motor berputar 61,2 ⁰ kekanan
35	0	1	0	0	Motor berputar 63 ⁰ kekanan
36	1	0	0	0	Motor berputar 64,8 ⁰ kekanan
37	0	0	0	1	Motor berputar 66,6 ⁰ kekanan
38	0	0	1	0	Motor berputar 68,4 ⁰ kekanan

39	0	1	0	0	Motor berputar 70,2 ⁰ kekanan
40	1	0	0	0	Motor berputar 72 ⁰ kekanan
41	0	0	0	1	Motor berputar 73,8 ⁰ kekanan
42	0	0	1	0	Motor berputar 75,6 ⁰ kekanan
43	0	1	0	0	Motor berputar 77,4 ⁰ kekanan
44	1	0	0	0	Motor berputar 79,2 ⁰ kekanan
45	0	0	0	1	Motor berputar 81 ⁰ kekanan
46	0	0	1	0	Motor berputar 82,8 ⁰ kekanan
47	0	1	0	0	Motor berputar 84,6 ⁰ kekanan
48	1	0	0	0	Motor berputar 86,4 ⁰ kekanan
49	0	0	0	1	Motor berputar 88,2 ⁰ kekanan
50	0	0	1	0	Motor berputar 90 ⁰ kekanan
51	0	1	0	0	Motor berputar 91,8 ⁰ kekanan
52	1	0	0	0	Motor berputar 93,6 ⁰ kekanan
53	0	0	0	1	Motor berputar 95,4 ⁰ kekanan
54	0	0	1	0	Motor berputar 97,2 ⁰ kekanan
55	0	1	0	0	Motor berputar 99 ⁰ kekanan
56	1	0	0	0	Motor berputar 100,8 ⁰ kekanan
57	0	0	0	1	Motor berputar 102,6 ⁰ kekanan
58	0	0	1	0	Motor berputar 104,4 ⁰ kekanan
59	0	1	0	0	Motor berputar 106,2 ⁰ kekanan
60	1	0	0	0	Motor berputar 108 ⁰ kekanan

Tabel 4.5**Pengujian Motor Stepper (berputar Kekiri)**

NO	DATA INPUTAN				KETERANGAN
	P1.7	P1.6	P1.5	P1.4	
1	1	0	0	0	Motor berputar 1,8 ⁰ kekiri
2	0	1	0	0	Motor berputar 3,6 ⁰ kekiri
3	0	0	1	0	Motor berputar 5,4 ⁰ kekiri
4	0	0	0	1	Motor berputar 7,2 ⁰ kekiri
5	1	0	0	0	Motor berputar 9 ⁰ kekiri
6	0	1	0	0	Motor berputar 10,8 ⁰ kekiri
7	0	0	1	0	Motor berputar 12,6 ⁰ kekiri
8	0	0	0	1	Motor berputar 14,4 ⁰ kekiri
9	1	0	0	0	Motor berputar 16,2 ⁰ kekiri
10	0	1	0	0	Motor berputar 18 ⁰ kekiri
11	0	0	1	0	Motor berputar 19,8 ⁰ kekiri
12	0	0	0	1	Motor berputar 21,6 ⁰ kekiri
13	1	0	0	0	Motor berputar 23,4 ⁰ kekiri
14	0	1	0	0	Motor berputar 25,2 ⁰ kekiri
15	0	0	1	0	Motor berputar 27 ⁰ kekiri
16	0	0	0	1	Motor berputar 28,8 ⁰ kekiri
17	1	0	0	0	Motor berputar 30,6 ⁰ kekiri
18	0	1	0	0	Motor berputar 32,4 ⁰ kekiri
19	0	0	1	0	Motor berputar 34,2 ⁰ kekiri

20	0	0	0	1	Motor berputar 36 ⁰ kekiri
21	1	0	0	0	Motor berputar 37,8 ⁰ kekiri
22	0	1	0	0	Motor berputar 39,6 ⁰ kekiri
23	0	0	1	0	Motor berputar 41,4 ⁰ kekiri
24	0	0	0	1	Motor berputar 43,2 ⁰ kekiri
25	1	0	0	0	Motor berputar 45 ⁰ kekiri
26	0	1	0	0	Motor berputar 46,8 ⁰ kekiri
27	0	0	1	0	Motor berputar 48,6 ⁰ kekiri
28	0	0	0	1	Motor berputar 50,4 ⁰ kekiri
29	1	0	0	0	Motor berputar 52,2 ⁰ kekiri
30	0	1	0	0	Motor berputar 54 ⁰ kekiri
31	0	0	1	0	Motor berputar 55,8 ⁰ kekiri
32	0	0	0	1	Motor berputar 57,6 ⁰ kekiri
33	1	0	0	0	Motor berputar 59,4 ⁰ kekiri
34	0	1	0	0	Motor berputar 61,2 ⁰ kekiri
35	0	0	1	0	Motor berputar 63 ⁰ kekiri
36	0	0	0	1	Motor berputar 64,8 ⁰ kekiri
37	1	0	0	0	Motor berputar 66,6 ⁰ kekiri
38	0	1	0	0	Motor berputar 68,4 ⁰ kekiri
39	0	0	1	0	Motor berputar 70,2 ⁰ kekiri
40	0	0	0	1	Motor berputar 72 ⁰ kekiri
41	1	0	0	0	Motor berputar 73,8 ⁰ kekiri

42	0	1	0	0	Motor berputar 75,6 ⁰ kekiri
43	0	0	1	0	Motor berputar 77,4 ⁰ kekiri
44	0	0	0	1	Motor berputar 79,2 ⁰ kekiri
45	1	0	0	0	Motor berputar 81 ⁰ kekiri
46	0	1	0	0	Motor berputar 82,8 ⁰ kekiri
47	0	0	1	0	Motor berputar 84,6 ⁰ kekiri
48	0	0	0	1	Motor berputar 86.4 ⁰ kekiri
49	1	0	0	0	Motor berputar 88.2 ⁰ kekiri
50	0	1	0	0	Motor berputar 90 ⁰ kekiri
51	0	0	1	0	Motor berputar 91.8 ⁰ kekiri
52	0	0	0	1	Motor berputar 93.6 ⁰ kekiri
53	1	0	0	0	Motor berputar 95.4 ⁰ kekiri
54	0	1	0	0	Motor berputar 97.2 ⁰ kekiri
55	0	0	1	0	Motor berputar 99 ⁰ kekiri
56	0	0	0	1	Motor berputar 100.8 ⁰ kekiri
57	1	0	0	0	Motor berputar 102.6 ⁰ kekiri
58	0	1	0	0	Motor berputar 104.4 ⁰ kekiri
59	0	0	1	0	Motor berputar 106.2 ⁰ kekiri
60	0	0	0	1	Motor berputar 108 ⁰ kekiri

Dari hasil pengujian yang telah dilakukan, terlihat bahwa motor stepper sudah bisa berfungsi seperti yang telah direncanakan.

4.4.Pengujian Sistem Secara Keseluruhan

Pengujian seluruh sistem dilakukan dengan membandingkan hasil rancangan dengan standart perancangan awal.

1. Tujuan :

Untuk mengetahui apakah sistem yang telah dibuat dapat bekerja sesuai dengan perencanaan. Pengujian dilakukan dengan menjalankan sistem secara keseluruhan, apakah sudah berjalan sesuai dengan perencanaan.

3. Proses Pengujian:

- Mengaktifkan kamera dan motor stepper pada masing-masing ruangan.
- Kemudian mengamati sistem kerja dari seluruh rangkaian.

4. Hasil pengujian :

Berdasarkan pengujian yang telah dilakukan terlihat bahwa pada saat kamera mengambil gambar dari masing-masing ruangan, kamera juga dapat berputar kekiri dan kekanan yang digerakkan oleh motor stepper. Dimana motor stepper ini dapat berputar kekiri dan kekanan dengan simpangan sudut paling jauh sebesar 108° . Dari gambar yang diperoleh dari masing-masing ruangan, dapat ditampilkan secara bergantian dengan cara otomatis dan manual.

BAB V

PENUTUP

Pada bab ini penulis akan menyampaikan kesimpulan dan saran setelah melakukan pengujian pada alat video switcher untuk monitoring gedung berbasis mikrokontroler AT89S51.

5.1 Kesimpulan

Setelah melakukan perencanaan, pembuatan dan pengamatan serta pengujian alat video switcher untuk monitoring gedung berbasis mikrokontroler AT89S51, maka pada Bab ini diberikan beberapa kesimpulan dan saran:

1. Untuk mendeteksi kerusakan kamera digunakan sensor getar. Apabila output dari sensor getar berlogika low, maka mikrokontroller akan mendeteksi bahwa telah terjadi kerusakan pada kamera. Output tegangan dari sensor getar pada kondisi high adalah 4,30 V, sedangkan pada kondisi Low 0,01 V.
2. Driver yang digunakan untuk menggerakkan relay adalah IC ULN2003. Tujuan penggunaan IC ULN2003 ini adalah supaya rangkaian mikrokontroler terpisah dengan tegangan relay. Output dari IC ULN2003 ini adalah kebalikan dari inputnya.
3. Untuk dapat mengaktifkan kamera1 sampai kamera4 digunakan relay dan ULN2003 sebagai kemudi, dimana sistem kerjanya adalah kamera dan motor 1 aktif apabila port 3.4 diberi logika 1. Kamera dan motor 2 aktif apabila port 3.5 diberi logika 1. Kamera

dan motor 3 aktif apabila port 3.6 diberi logika 1. Kamera dan motor 4 aktif apabila port 3.7 diberi logika 1.

4. Untuk melakukan perputaran kamera kekanan dan kekiri digunakan motor stepper, dimana 1Step motor stepper menghasilkan sudut $1,8^{\circ}$, dengan simpangan maksimal sudut 108° diperoleh 60step.

5.2 Saran

Dalam pembuatan skripsi ini masih terdapat beberapa kekurangan, sehingga untuk mencapai hasil yang lebih sempurna maka dapat diberikan saran-saran sebagai berikut :

1. *Video switcher* ini hanya bisa melakukan monitoring. Maka untuk kedepannya *video switcher* dikembangkan agar bisa merekam keadaan atau situasi didalam suatu gedung.
2. Pada pengembangan *video switcher* berikutnya diharapkan penampilan gambar dari seluruh ruangan dapat ditampilkan secara keseluruhan dalam sebuah layar televisi.
3. Untuk pengembangan berikutnya *video switcher* diharapkan dapat menampilkan kondisi kamera mana yang telah aktif pada saat proses *video switcher* dijalankan secara otomatis.
4. Pada pengembangan berikutnya *video switcher* ini juga diharapkan dapat menampilkan kondisi kamera mana yang mendapat kerusakan

5. Sistem akan bekerja dengan baik (stabil) jika digunakan komponen-komponen yang berkualitas tinggi.



DAFTAR PUSTAKA

1. Afgianto Eko Putra, Belajar mikrokontroler AT89S51/52/55 *Teori dan Aplikasi*. Gava Media: Yogyakarta, 2004.
2. *AT89S51 Data Sheet*. Download : ATMEL.
3. Seiko Instrument Inc. 1987. *user Manual LCD Module M1632*. Japan :
Seiko Japan Inc
4. *Data sheet ULN2003*.
5. Paul Malvino, Albert: *Prinsip-Prinsip Elektronika*, Salemba Teknika:
Jakarta, 2003.



INSTITUT TEKNOLOGI NASIONAL MALANG

LAMPIRAN



LEMBAR PERBAIKAN SKRIPSI

Nama Mahasiswa : Rudy Prayitno
NIM : 02.17.009
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Hari / Tanggal Ujian Skripsi : Jum'at / 16 Maret 2007

No	Materi Perbaikan	Paraf
1	Hal 46, Perancangan Limit Switch disempurnakan	
2	Hal 42, Arus 30mA Untuk Apa?	
3	Kesimpulan Tentang Motor Stepper	

Diperiksa / Disetujui :

PENGUJI I

Ir. Sidik Noertjahjono, M.T.
NIP. 1028700167

Mengetahui :

Dosen Pembimbing I

Ir. Usman Djuanda, MM.
NIP.P. 1018700143

Dosen Pembimbing II

I Komang Somawirata, S.T, M.T.
NIP.Y. 1030100361



LEMBAR PERBAIKAN SKRIPSI

Nama Mahasiswa : Rudy Prayitno
NIM : 02.17.009
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan Dan Pembuatan Video Switcher
Untuk Monitoring Gedung Berbasis
Mikrokontroler AT89S51
Hari / Tanggal Ujian Skripsi : Jum'at / 16 Maret 2007

No	Materi Perbaikan	Paraf
1	Hasil Pengujian Motor Stepper	<i>fai</i> 23/3 '07

Diperiksa / Disetujui:

PENGUJI II

Sohyohadi, S.T., M.T.

Mengetahui :

Dosen Pembimbing I

Ir. Usman Djuanda, MM
NIP.P. 1018700143

Dosen Pembimbing II

I Komang Somawirata, ST, MT
NIP.Y. 1030100361



FORMULIR BIMBINGAN SKRIPSI

Nama : Rudy Prayitno
NIM : 02 17 009
Masa Bimbingan : 10 Oktober 2006 - 10 April 2007
Judul Skripsi : Perancangan Dan Pembuatan Video Switcher Untuk Keamanan Gedung Berbasis Mikrokontroler AT 89S51

No	Tanggal	Uraian	Paraf Pembimbing
1		Das 1. Skematik Das 2. + Switcher Das 3. Skematik	
2		Das I, II, III OK.	
3		Das IV OK	
4		Das V Skematik	
5		ACE Switcher.	
6		ACE Komp. Ke.	
7			
8			
9			
10			

Malang,
Dosen Pembimbing

Ir. Usman Djuanda, MM.
NIP. P. 101 8700143



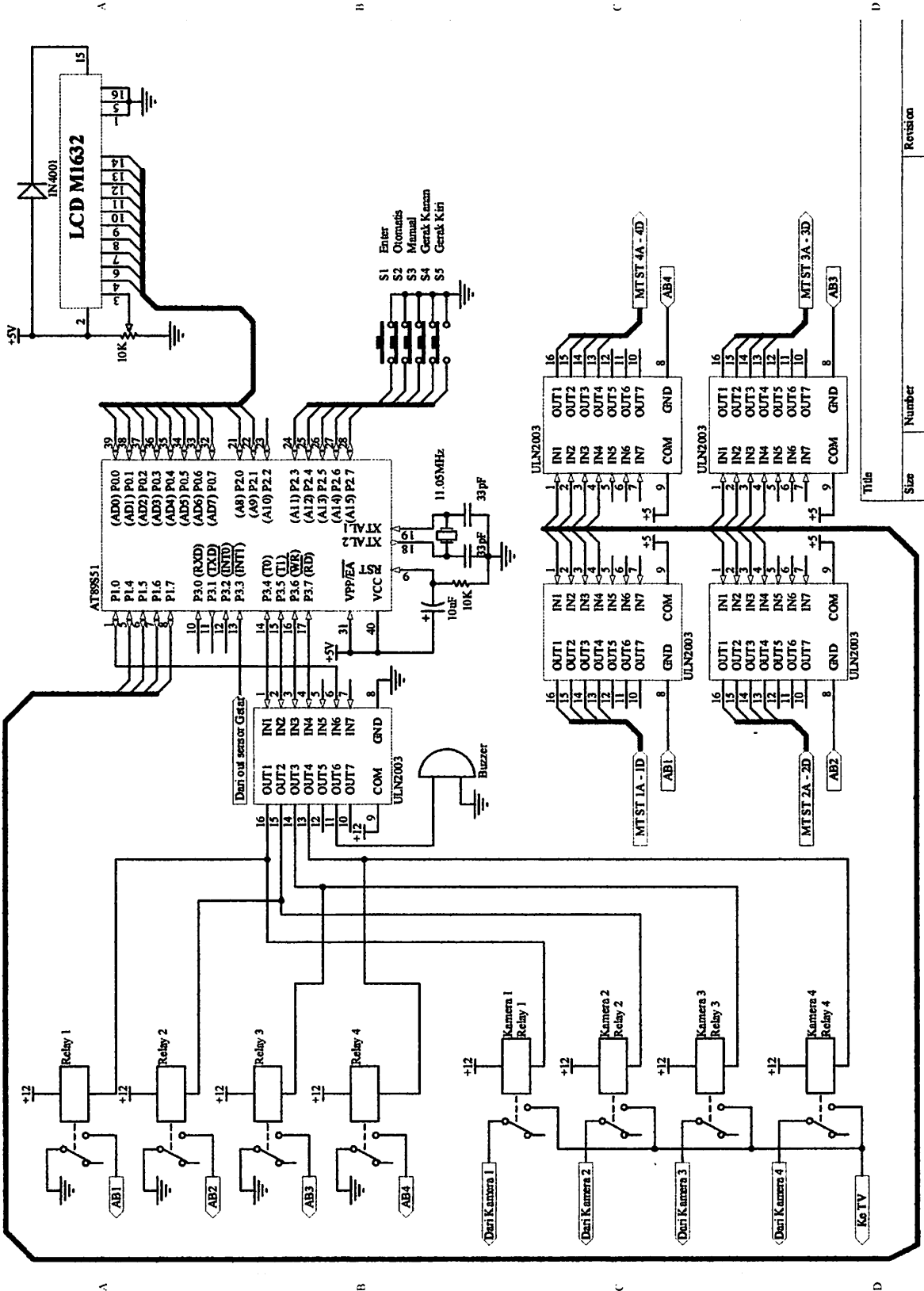
FORMULIR BIMBINGAN SKRIPSI

Nama : Rudy Prayitno
NIM : 02 17 009
Masa Bimbingan : 10 Oktober 2006 - 10 April 2007
Judul Skripsi : Perancangan Dan Pembuatan Video Switcher Untuk Keamanan Gedung Berbasis Mikrokontroler AT.89S51

No	Tanggal	Uraian	Paraf Pembimbing
1	31/07 /01	Konsultasi Bab I, Bab II & Bab III	
2	07/07 /02	Revisi Bab I, Bab II, Bab III	
3	12/07 /02	Revisi Bab IV & V	
4	17/07 /02	Revisi Abstrak, Kesimpulan	
5	21/07 /02	SKRIPSI Revisi Cat: Kesimpulan di perbaiki	
6			
7			
8			
9			
10			

Malang, 8-03-2007
Dosen Pembimbing

Komang Somawirata, ST, MT
NIP.Y. 1030100361



Size	Number	Revision
A		

Date	Sheet of
3/12/2007	3

File	Drawn By
E: Mandaun atipya\... \alurhukum_SCHD-OC	

3

2

1

4

3

2

A

B

C

D

A

B

C

D

```

1: ; LCD KONSTANTA
2: DISPCLR EQU 00000001B
3: BLINK EQU 00001101B
4: ENTRMOD EQU 00000110B
5: DISPON EQU 00001100B
6: CURSOR EQU 00011100B
7: FUNCSET EQU 00111000B
8: ;
9: ;DEVICE ADDRESS {LCD}
10: RS BIT P2.0 ;LCD
11: E BIT P2.1 ;LCD
12:
13: LOKASI EQU 47H
14: LOKASI1 EQU 48H
15: ORG 00h ; Reset vector
16: JMP MULAI ; Lompat ke Mulai
17: ORG 0013H ; Alamat interrupt
18: getar :
19: clr TR0 ; Beri logika 0 pada TR0
20: clr EA ; Semua interrupt nonaktif
21: mov TH0,#0 ; Isi TH0 dengan data 0
22: mov TL0,#0 ; Isi TL0 dengan data 0
23: ALAREM:
24: CLR P3.4 ; Beri logika 0 pada P3.4
25: CLR P3.5 ; Beri logika 0 pada P3.5
26: CLR P3.6 ; Beri logika 0 pada P3.6
27: CLR P3.7 ; Beri logika 0 pada P3.7
28: MOV DPTR,#MENU28 ; Isi DPTR dengan MENU28
29: LCALL PRINTSTRING1 ; Panggil Subroutine PRINTSTRING1
30: MOV DPTR,#MENU29 ; Isi DPTR dengan MENU29
31: LCALL PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
32: ACALL ROLL ; Panggil subroutine ROLL
33: KONDISI:
34: SETB P1.0 ; Beri logika 1 pada P1.0
35: JB P2.3,KONDISI ; Cek P2.3, jika 1 lompat ke KONDISI
36: CLR P1.0 ; Berilogika 0 pada P1.0
37: SETB EA ; Aktifkan semua interrupt
38: reti ; Kembali (Layanan interrupt selesai)
39:
40:
41: MULAI: CLR P1.0 ; Beri logika 0 pada P1.0
42: CLR P1.4 ; Beri logika 0 pada P1.4
43: CLR P1.5 ; Beri logika 0 pada P1.5
44: CLR P1.6 ; Beri logika 0 pada P1.6
45: CLR P1.7 ; Beri logika 0 pada P1.7
46: CLR P3.4 ; Beri logika 0 pada P3.4
47: CLR P3.5 ; Beri logika 0 pada P3.5
48: CLR P3.6 ; Beri logika 0 pada P3.6
49: CLR P3.7 ; Beri logika 0 pada P3.7
50: LCALL INIT ; Panggil subroutine INIT

```

```

51:          LCALL  INIT_LCD      ; Panggil subroutine INIT_LCD
52:
53: TUNDA_TAM:
54:          MOV    R7,#100        ; Isi R7 dengan data 100
55: LOOP10: MOV    R6,#100        ; Isi R6 dengan data 100
56: LOOP9:  MOV    R5,#100        ; Isi R5 dengan data 100
57:          DJNZ  R5,$            ; Kurangi R5 sampai nilainya 0
58:          DJNZ  R6,LOOP9       ; Kurangi R6, jika belum 0 lompat ke LOOP9
59:          DJNZ  R7,LOOP10      ; Kurangi R7, jika belum 0 lompat ke LOOP10
60:
61:          MOV    R7,#100        ; Isi R7 dengan data 100
62: LOOP12: MOV    R6,#100        ; Isi R6 dengan data 100
63: LOOP11: MOV    R5,#100        ; Isi R5 dengan data 100
64:          DJNZ  R5,$            ; Kurangi R5 sampai nilainya 0
65:          DJNZ  R6,LOOP11      ; Kurangi R6, jika belum 0 lompat ke LOOP11
66:          DJNZ  R7,LOOP12      ; Kurangi R7, jika belum 0 lompat ke LOOP12
67:          RET                    ; Kembali
68: INIT:
69:          mov   SP,#30H        ; Isi SP dengan data 30h
70:          mov   TCON,#0        ; Isi TCON dengan data 0
71:          setb  TR0            ; Start timer0
72:          mov   TMOD,#29h      ; Masukkan data 29h ke dalam register TMOD
73:          setb  EX1            ; priority
74:          Setb  IT1            ; set ext. int.1 to down
75:          Sctb  EX1            ; enable ext. int.1
76:
77:          call  delay          ; Panggil subroutine delay
78:          call  delay          ; Panggil subroutine delay
79:          Setb  EA            ; Aktifkan semua interrupt
80:          ret                    ; Kembali
81: INISIALISASI:
82:
83: ;*****
84: ;*  INISIALISASI LCD      *
85: ;*****
86: DELAY_INIT_LCD:
87:          MOV  R6,#20H          ; Isi R6 dengan data 20h
88:
89: DLY_LCD_LP:
90:
91:          MOV  R7,#0            ; Isi R7 dengan data 0
92:          DJNZ R7,$            ; Kurangi R7 sampai nilainya 0
93:          DJNZ R6,DLY_LCD_LP   ; Kurangi R6 dengan 1, jika belum 0 lompat ke DLY_LCD_LP
94:          RET                    ; Kembali
95:
96: INIT_LCD:
97:          SETB RS                ; Beri logika 1 pada RS
98:          ;SETB RW
99:          CLR  E                 ; Beri logika 0 pada E
100:         MOV  A,#DISPCLR        ; Pindahkan isi DISPCLR ke A

```

```

101:      LCALL CONTROLOUT      ; Panggil subroutine CONTROLOUT
102:      LCALL DELAY_INIT_LCD  ; Panggil subroutine DELAY_INIT_LCD
103:      MOV      A,#FUNCSET    ; Pindahkan isi FUNCSET ke A
104:      LCALL      CONTROLOUT  ; Panggil subroutine CONTROLOUT
105:      MOV      A,#DISPON     ; Pindahkan isi DISPON ke A
106:      LCALL      CONTROLOUT  ; Panggil subroutine CONTROLOUT
107:      MOV      A,#ENTRMOD    ; Pindahkan isi ENTRMOD ke A
108:      LCALL      CONTROLOUT  ; Panggil subroutine CONTROLOUT
109:      ;
110:      MOV      DPTR,#MENU    ; Isi DPTR dengan MENU
111:      LCALL      PRINTSTRING1 ; Panggil subroutine PRINTSTRING1
112:      MOV      DPTR,#MENU1   ; Isi DPTR dengan MENU1
113:      LCALL      PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
114:      CALL      TUNDA_TAM    ; Panggil subroutine TUNDA_TAM
115:
116:      MOV      DPTR,#MENU2   ; Isi DPTR dengan MENU2
117:      LCALL      PRINTSTRING1 ; Panggil subroutine PRINTSTRING1
118:      MOV      DPTR,#MENU3   ; Isi DPTR dengan MENU3
119:      LCALL      PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
120:      CALL      TUNDA_TAM    ; Panggil subroutine TUNDA_TAM
121: LANG1:
122:      MOV      DPTR,#MENU4   ; Isi DPTR dengan MENU4
123:      LCALL      PRINTSTRING1 ; Panggil subroutine PRINTSTRING1
124:      MOV      DPTR,#MENU5   ; Isi DPTR dengan MENU5
125:      LCALL      PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
126:      CALL      TUNDA_TAM    ; Panggil subroutine TUNDA_TAM
127:
128: ;*****
129:      Sub pilihan otomatis atau manual
130: ;*****
131: ATAS:
132:      JNB      P2.4,PILIHAN1
133:      JNB      P2.5,PILIHAN2
134:      JMP      ATAS
135:
136: ;*****
137:      Pilihan otomatis
138: ;*****
139:
140: PILIHAN1:
141:      MOV      DPTR,#MENU6   ; Isi DPTR dengan MENU6
142:      LCALL      PRINTSTRING1 ; Panggil subroutine PRINTSTRING1
143:      MOV      DPTR,#MENU7   ; Isi DPTR dengan MENU7
144:      LCALL      PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
145:      CALL      TUNDA_TAM    ; Panggil subroutine TUNDA_TAM
146: ATAS1:
147:      CLR      P3.4          ; Beri logika 0 pada P3.4
148:      CLR      P3.5          ; Beri logika 0 pada P3.5
149:      CLR      P3.6          ; Beri logika 0 pada P3.6
150:      CLR      P3.7          ; Beri logika 0 pada P3.7

```

```

151:      JNB      P2.3,ATAS2      ; Cek P2.3, jika 0 lompat ke ATAS2
152:      JNB      P2.5,PILIHAN2   ; Cek P2.5, jika 0 lompat ke PILIHAN2
153:      JMP      ATAS1           ; Lompat ke ATAS1
154:
155:
156: ;*****
157:                      Pilihan Manual
158: ;*****
159:
160: PILIHAN2:
161:      MOV      DPTR,#MENU8      ; Isi DPTR dengan MENU8
162:      LCALL   PRINTSTRING1     ; Panggil subroutine PRINTSTRING1
163:      MOV      DPTR,#MENU9      ; Isi DPTR dengan MENU9
164:      LCALL   PRINTSTRING2     ; Panggil subroutine PRINTSTRING2
165:      MOV      R1,#10001000B    ; Isi R1 dengan data 10001000b
166: ATAS3:
167:      CLR      P3.4            ; Beri logika 0 pada P3.4
168:      CLR      P3.5            ; Beri logika 0 pada P3.5
169:      CLR      P3.6            ; Beri logika 0 pada P3.6
170:      CLR      P3.7            ; Beri logika 0 pada P3.7
171:      JNB      P2.3,ATAS4A     ; Cek P2.3, jika 0 lompat ke ATAS4A
172:      JNB      P2.4,PILIHAN1   ; Cek P2.4, jika 0 lompat ke PILIHAN1
173:      JMP      ATAS3           ; Lompat ke ATAS3
174: ATAS4A:
175:      AJMP    ATAS4            ; Lompat ke ATAS4
176:
177:
178: ;*****
179:                      Pilihan Otomatis
180: ;*****
181: ATAS2:
182:      MOV      R2,#15           ; Isi R2 dengan data 15
183:      MOV      R3,#4            ; Isi R3 dengan data 4
184:      MOV      R0,#10000000B    ; Isi R0 dengan data 10000000b
185:      MOV      LOKASI,#0        ; Isi reg LOKASI dengan data 0
186:      MOV      LOKASI1,#0       ; Isi reg LOKASI1 dengan data 0
187:      mov     A,R0              ; Pindahkan isi R0 ke A
188:      mov     B,#00001111B     ; Isi B dengan data 00001111b
189:      ORL     A,B              ; OR kan A dengan B
190:      MOV     P3,A             ; Pindahkan isi A ke P3
191:
192: ENTER1:
193:
194:      SETB    P1.4            ; Beri logika 1 pada P1.4
195:      CLR     P1.5            ; Beri logika 0 pada P1.5
196:      clr     P1.6            ; Beri logika 0 pada P1.6
197:      CLR     P1.7            ; Beri logika 0 pada P1.7
198:      JNB     P2.5,PILIHAN2   ; Cek P2.5, jika 0 lompat ke PILIHAN2
199:      ACALL   ROLL            ; Panggil subroutine ROLL
200:      JNB     P2.5,PILIHAN2   ; Cek P2.5, jika 0 lompat ke PILIHAN2

```



```

201:
202:      CLR      P1.4          ; Beri logika 0 pada P1.4
203:      SETB     P1.5          ; Beri logika 1 pada P1.5
204:      clr       P1.6          ; Beri logika 0 pada P1.6
205:      CLR      P1.7          ; Beri logika 0 pada P1.7
206:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
207:      ACALL    ROLL          ; Panggil subroutine ROLL
208:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
209:
210:      CLR      P1.4          ; Beri logika 0 pada P1.4
211:      CLR      P1.5          ; Beri logika 0 pada P1.5
212:      SETB     P1.6          ; Beri logika 1 pada P1.6
213:      CLR      P1.7          ; Beri logika 0 pada P1.7
214:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
215:      ACALL    ROLL          ; Panggil subroutine ROLL
216:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
217:
218:      CLR      P1.4          ; Beri logika 0 pada P1.4
219:      CLR      P1.5          ; Beri logika 0 pada P1.5
220:      clr       P1.6          ; Beri logika 0 pada P1.6
221:      SETB     P1.7          ; Beri logika 1 pada P1.7
222:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
223:      ACALL    ROLL          ; Panggil subroutine ROLL
224:      JNB      P2.5,PILIHAN2 ; Cek P2.5, jika 0 lompat ke PILIHAN2
225:
226:      DJNZ     R2,ENTER1     ; Kurangi R2 dengan 1, jika belum 0 lompat ke ENTER1
227:      MOV      R2,#15        ; Isi R2 dengan data 15
228:
229: ;*****
230: ENTER2:
231:
232:      clr       P1.4          ; Beri logika 0 pada P1.4
233:      CLR      P1.5          ; Beri logika 0 pada P1.5
234:      clr       P1.6          ; Beri logika 0 pada P1.6
235:      SETB     P1.7          ; Beri logika 1 pada P1.7
236:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
237:      ACALL    ROLL          ; Panggil subroutine ROLL
238:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
239:
240:      CLR      P1.4          ; Beri logika 0 pada P1.4
241:      clr       P1.5          ; Beri logika 0 pada P1.5
242:      SETB     P1.6          ; Beri logika 1 pada P1.6
243:      clr       P1.7          ; Beri logika 0 pada P1.7
244:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
245:      ACALL    ROLL          ; Panggil subroutine ROLL
246:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
247:
248:      CLR      P1.4          ; Beri logika 0 pada P1.4
249:      SETB     P1.5          ; Beri logika 1 pada P1.5
250:      clr       P1.6          ; Beri logika 0 pada P1.6

```

```

251:      CLR      P1.7          ; Beri logika 0 pada P1.7
252:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
253:      ACALL    ROLL          ; Panggil subroutine ROLL
254:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
255:
256:      SETB     P1.4          ; Beri logika 1 pada P1.4
257:      CLR      P1.5          ; Beri logika 0 pada P1.5
258:      clr      P1.6          ; Beri logika 0 pada P1.6
259:      clr      P1.7          ; Beri logika 0 pada P1.7
260:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
261:      ACALL    ROLL          ; Panggil subroutine ROLL
262:      JNB      P2.5,PILIHAN2A ; Cek P2.5, jika 0 lompat ke PILIHAN2A
263:
264:
265:
266: ;*****
267:      DJNZ     R2,ENTER2      ; Kurangi R2 dengan 1, jika belum 0 lompat ke ENTER2
268:      MOV      R2,#15         ; Isi R2 dengan data 15
269:      MOV      A,R0          ; Pindahkan isi R0 ke A
270:      RR       A              ; A digeser kekanan
271:      MOV      R0,A          ; Pindahkan isi A ke R0
272:      mov      B,#00001111B   ; Isi B dengan data 00001111b
273:      ORL      A,B           ; OR kan A dengan B
274:      MOV      p3,A          ; Pindahkan isi A ke P3
275:      DJNZ     R3,ENTER1A     ; Kurangi R3 dengan 1, jika belum 0 lompat ke ENTER1A
276:      AJMP     ATAS2         ; Lompat ke ATAS2
277: ENTER1A:
278:      AJMP     ENTER1        ; Lompat ke ENTER1
279:
280:
281:
282:
283: PILIHAN1A:
284:      AJMP     PILIHAN1      ; Lompat ke PILIHAN1
285: PILIHAN2A:
286:      AJMP     PILIHAN2      ; Lompat ke PILIHAN2
287:
288: ;*****
289:      Pilihan Manual
290: ;*****
291:
292: ATAS4:
293:      MOV      DPTR,#MENU8    ; Isi DPTR dengan MENU8
294:      LCALL    PRINTSTRING1   ; Panggil subroutine PRINTSTRING1
295:      MOV      DPTR,#MENU9    ; Isi DPTR dengan MENU9
296:      LCALL    PRINTSTRING2   ; Panggil subroutine PRINTSTRING2
297:
298: ATAS5:
299:      JNB      P2.4,PILIHAN1A ; Cek P2.4, jika 0 lompat ke PILIHAN1A
300:      JNB      P2.6,MOTOR1    ; Cek P2.6, jika 0 lompat ke MOTOR1

```

```

301:      JNB      P2.7,MOTOR4      ; Cek P2.7, jika 0 lompat ke MOTOR4
302:
303:      JMP      ATAS5            ; Lompat ke ATAS5
304:
305:
306: MOTOR1:
307:      MOV      DPTR,#MENU10      ; Isi DPTR dengan MENU10
308:      LCALL   PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
309:      MOV      DPTR,#MENU11      ; Isi DPTR dengan MENU11
310:      LCALL   PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
311:      ACALL   ROLL              ; Panggil subroutine ROLL
312: ATAS6:
313:      JNB      P2.6,MOTOR2      ; Cek P2.6, jika 0 lompat ke MOTOR2
314:      JNB      P2.7,MOTOR4      ; Cek P2.7, jika 0 lompat ke MOTOR4
315:      JNB      P2.3,OKE_MOTOR1  ; Cek P2.3, jika 0 lompat ke OKE_MOTOR1
316:      JMP      ATAS6            ; Lompat ke ATAS6
317:
318: MOTOR2:
319:      MOV      DPTR,#MENU12      ; Isi DPTR dengan MENU12
320:      LCALL   PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
321:      MOV      DPTR,#MENU13      ; Isi DPTR dengan MENU13
322:      LCALL   PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
323:      ACALL   ROLL              ; Panggil subroutine ROLL
324: ATAS7:
325:      JNB      P2.6,MOTOR3      ; Cek P2.6, jika 0 lompat ke MOTOR3
326:      JNB      P2.7,MOTOR1      ; Cek P2.7, jika 0 lompat ke MOTOR1
327:      JNB      P2.3,OKE_MOTOR2  ; Cek P2.3, jika 0 lompat ke OKE_MOTOR2
328:      JMP      ATAS7            ; Lompat ke ATAS7
329:
330: MOTOR3:
331:      MOV      DPTR,#MENU14      ; Isi DPTR dengan MENU14
332:      LCALL   PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
333:      MOV      DPTR,#MENU15      ; Isi DPTR dengan MENU15
334:      LCALL   PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
335:      ACALL   ROLL              ; Panggil subroutine ROLL
336: ATAS8:
337:      JNB      P2.6,MOTOR4      ; Cek P2.6, jika 0 lompat ke MOTOR4
338:      JNB      P2.7,MOTOR2      ; Cek P2.7, jika 0 lompat ke MOTOR2
339:      JNB      P2.3,OKE_MOTOR3  ; Cek P2.3, jika 0 lompat ke OKE_MOTOR3
340:      JMP      ATAS8            ; Lompat ke ATAS8
341:
342:
343: MOTOR4:
344:      MOV      DPTR,#MENU16      ; Isi DPTR dengan MENU16
345:      LCALL   PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
346:      MOV      DPTR,#MENU17      ; Isi DPTR dengan MENU17
347:      LCALL   PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
348:      ACALL   ROLL              ; Panggil subroutine ROLL
349: ATAS9:
350:      JNB      P2.6,MOTOR1      ; Cek P2.6, jika 0 lompat ke MOTOR1

```

```

351:      JNB      P2.7,MOTOR3      ; Cek P2.7, jika 0 lompat ke MOTOR3
352:      JNB      P2.3,OKE_MOTOR4 ; Cek P2.3, jika 0 lompat ke OKE_MOTOR4
353:      JMP      ATAS9            ; Lompat ke ATAS9
354:
355: OKE_MOTOR1:
356:      MOV      DPTR,#MENU20      ; Isi DPTR dengan MENU20
357:      LCALL    PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
358:      MOV      DPTR,#MENU21      ; Isi DPTR dengan MENU21
359:      LCALL    PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
360:      ACALL    ROLL              ; Panggil subroutine ROLL
361:      SETB     P3.7              ; Beri logika 1 pada P3.7
362:      CLR      P3.6              ; Beri logika 0 pada P3.6
363:      clr      P3.5              ; Beri logika 0 pada P3.5
364:      clr      P3.4              ; Beri logika 0 pada P3.4
365:      CLR      P1.4              ; Beri logika 0 pada P1.4
366:      CLR      P1.5              ; Beri logika 0 pada P1.5
367:      clr      P1.6              ; Beri logika 0 pada P1.6
368:      clr      P1.7              ; Beri logika 0 pada P1.7
369:      AJMP     PUTAR            ; Lompat ke PUTAR
370:
371: OKE_MOTOR2:
372:      MOV      DPTR,#MENU22      ; Isi DPTR dengan MENU22
373:      LCALL    PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
374:      MOV      DPTR,#MENU23      ; Isi DPTR dengan MENU23
375:      LCALL    PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
376:      ACALL    ROLL              ; Panggil subroutine ROLL
377:      clr      P3.7              ; Beri logika 0 pada P3.7
378:      SETB     P3.6              ; Beri logika 1 pada P3.6
379:      clr      P3.5              ; Beri logika 0 pada P3.5
380:      clr      P3.4              ; Beri logika 0 pada P3.4
381:      CLR      P1.4              ; Beri logika 0 pada P1.4
382:      CLR      P1.5              ; Beri logika 0 pada P1.5
383:      clr      P1.6              ; Beri logika 0 pada P1.6
384:      clr      P1.7              ; Beri logika 0 pada P1.7
385:      AJMP     PUTAR            ; Lompat ke PUTAR
386:
387: OKE_MOTOR3:
388:      MOV      DPTR,#MENU24      ; Isi DPTR dengan MENU24
389:      LCALL    PRINTSTRING1      ; Panggil subroutine PRINTSTRING1
390:      MOV      DPTR,#MENU25      ; Isi DPTR dengan MENU25
391:      LCALL    PRINTSTRING2      ; Panggil subroutine PRINTSTRING2
392:      ACALL    ROLL              ; Panggil subroutine ROLL
393:      clr      P3.7              ; Beri logika 0 pada P3.7
394:      CLR      P3.6              ; Beri logika 0 pada P3.6
395:      SETB     P3.5              ; Beri logika 1 pada P3.5
396:      clr      P3.4              ; Beri logika 0 pada P3.4
397:      CLR      P1.4              ; Beri logika 0 pada P1.4
398:      CLR      P1.5              ; Beri logika 0 pada P1.5
399:      clr      P1.6              ; Beri logika 0 pada P1.6
400:      clr      P1.7              ; Beri logika 0 pada P1.7

```

```

401:      AJMP   PUTAR      ; Lompat ke PUTAR
402:
403: OKE_MOTOR4:
404:      MOV    DPTR,#MENU26 ; Isi DPTR dengan MENU26
405:      LCALL  PRINTSTRING1 ; Panggil subroutine PRINTSTRING1
406:      MOV    DPTR,#MENU27 ; Isi DPTR dengan MENU27
407:      LCALL  PRINTSTRING2 ; Panggil subroutine PRINTSTRING2
408:      ACALL  ROLL        ; Panggil subroutine ROLL
409:      CLR    P3.7        ; Beri logika 0 pada P3.7
410:      CLR    P3.6        ; Beri logika 0 pada P3.6
411:      CLR    P3.5        ; Beri logika 0 pada P3.5
412:      SETB   P3.4        ; Beri logika 1 pada P3.4
413:      CLR    P1.4        ; Beri logika 0 pada P1.4
414:      CLR    P1.5        ; Beri logika 0 pada P1.5
415:      CLR    P1.6        ; Beri logika 0 pada P1.6
416:      CLR    P1.7        ; Beri logika 0 pada P1.7
417:      AJMP   PUTAR      ; Lompat ke PUTAR
418:
419:
420: ATAS5A:
421:      CLR    P3.7        ; Beri logika 0 pada P3.7
422:      CLR    P3.6        ; Beri logika 0 pada P3.6
423:      CLR    P3.5        ; Beri logika 0 pada P3.5
424:      CLR    P3.4        ; Beri logika 0 pada P3.4
425:      AJMP   ATAS5      ; Lompat ke ATAS5
426: PUTAR:
427: ATAS10:
428:      JNB   P2.4,ATAS5A ; Cek P2.4, jika 0 lompat ke ATAS5A
429:      JNB   P2.5,ATAS5A ; Cek P2.5, jika 0 lompat ke ATAS5A
430:      JNB   P2.6,KIRI   ; Cek P2.6, jika 0 lompat ke KIRI
431:      JNB   P2.7,KANAN  ; Cek P2.7, jika 0 lompat ke KANAN
432:
433:      JMP   ATAS10      ; Lompat ke ATAS10
434: KIRI:
435:      MOV   A,R1        ; Pindahkan isi R1 ke A
436:      RR    A            ; A digeser kekanan
437:      MOV   R1,A        ; Pindahkan isi A ke R1
438:      MOV   B,#0F0H     ; Isi B dengan data 0F0h
439:      ANL  A,B          ; AND kan A dengan B
440:      MOV   P1,A        ; Pindahkan isi A ke P1
441:      ACALL ROLL        ; Panggil subroutine ROLL
442:      JMP   ATAS10      ; Lompat ke ATAS10
443: KANAN:
444:      MOV   A,R1        ; Pindahkan isi R1 ke A
445:      RL    A            ; A digeser ke kiri
446:      MOV   R1,A        ; Pindahkan isi A ke R1
447:      MOV   B,#0F0H     ; Isi B dengan data 0F0h
448:      ANL  A,B          ; AND kan A dengan B
449:      MOV   P1,A        ; Pindahkan isi A ke P1
450:      ACALL ROLL        ; Panggil subroutine ROLL

```

```

451:      JMP      ATAS10          ; Lompat ke ATAS10
452:
453:
454:
455: ;*****
456: ;^ KUMPULAN RUTIN PELAYANAN LCD ^
457: ;*****
458:
459: POSISI2_1:
460:      MOV      A,#1            ; Isi A dengan data 1
461: POSISI2:
462:      ADD      A,#11000000B    ; Tambahkan A dengan 11000000b
463:      SJMP     POSISI_SUB      ; Lompat ke POSISI_SUB
464:
465: POSISI1_1:
466:      MOV      A,#1            ; Isi A dengan data 1
467: POSISI1:
468:      ADD      A,#10000000B    ; Tambahkan A dengan 10000000b
469: POSISI_SUB:
470:      DEC      A                ; Kurangi A
471:      LCALL   CONTROLOUT      ; Panggil subroutine CONTROLOUT
472:      RET                          ; Kembali
473:
474: PRINTSTRING2:
475:      LCALL   POSISI2_1        ; Panggil subroutine POSISI2_1
476:      SJMP     PRINTSTRING     ; Lompat ke PRINTSTRING
477:
478: PRINTSTRING1:
479:      LCALL   POSISI1_1        ; Panggil subroutine POSISI1_1
480:
481: PRINTSTRING:
482:      SJMP     OUTSTRING       ; Lompat ke OUTSTRING
483: PRINTSTRINGLOOP:
484:      LCALL   DATAOUT         ; Panggil subroutine DATAOUT
485:      INC     DPTR              ; Tambahkan DPTR dengan 1
486:
487: OUTSTRING:
488:      CLR     A                 ; Nol kan isi A
489:      MOVC   A,@A+DPTR         ; Pindahkan isi alamat yang ditunjuk oleh DPTR+A ke A
490:      JNZ    PRINTSTRINGLOOP   ; Jika 1 lompat ke PRINTSTRINGLOOP
491:      RET                          ; Kembali
492:
493: CONTROLOUT:
494:      CPL    RS                 ; Komplemen RS (dibalik)
495:      ;CPL  RW
496:      CPL    E                 ; Komplemen E (dibalik)
497:      MOV    P0,A              ; Pindahkan isi A ke P0
498:      CPL    E                 ; Komplemen E (dibalik)
499:      ;CPL  RW
500:      CPL    RS                 ; Komplemen RS (dibalik)

```

```

501:      MOV P0,#0FFH      ; Isi P0 dengan data 0FFh
502:      SJMP LCD_OUT     ; Lompat ke LCD_OUT
503:
504: DATAOUT:
505:      ;CPL RW
506:      CPL E             ; Komplemen E (dibalik)
507:      MOV P0,A         ; Pindahkan isi A ke P0
508:      CPL E             ; Komplemen E (dibalik)
509:      ;CPL RW
510:
511: LCD_OUT:
512:      MOVX @DPTR,A     ; Pindahkan isi A ke alamat yang ditunjuk oleh DPTR
513:
514: DELAY_LCD:
515:      PUSH ACC         ; Pindahkan isi ACC ke stack pointer
516:      MOV A,#250       ; Isi A dengan data 250
517:      DJNZ ACC,$       ; Kurangi ACC sampai nilainya 0
518:      POP ACC          ; Pindahkan isi stack pointer ke ACC
519:      RET              ; Kembali
520:
521: Delay: Mov R4,#00h    ; Isi R4 dengan data 00h
522: Delay1: nop           ; No operation
523:      Djnz R4,Delay1   ; Kurangi R4 dengan 1, jika belum 0 lompat ke Delay1
524:      Ret              ; Kembali
525:
526: roll:
527:      MOV R7,#75       ; Isi R7 dengan data 50
528: LOOPAC: MOV R6,#50    ; Isi R6 dengan data 50
529: LOOPAD: MOV R5,#50    ; Isi R5 dengan data 50
530:      DJNZ R5,$        ; Kurangi R5 sampai nilainya 0
531:      DJNZ R6,LOOPAD   ; Kurangi R6 dengan 1, jika belum 0 lompat ke LOOPAC
532:      DJNZ R7,LOOPAC   ; Kurangi R7 dengan 1, jika belum 0 lompat ke LOOPAC
533:      RET              ; Kembali
534:
535: MENU :   DB ' RUDY PRAYITNO ',0
536: MENU1:   DB 'NIM : 02.17.009 ',0
537: MENU2:   DB 'MASUKAN DATA ',0
538: MENU3:   DB 'SISTEM PENGAMAN ',0
539: MENU4:   DB 'ADA DUA SISTEM ',0
540: MENU5:   DB 'PILIHAN OTS/MNL ',0
541: MENU6:   DB 'PILIHAN ANDA ',0
542: MENU7:   DB 'SISTEM OTOMATIS ',0
543: MENU8:   DB 'PILIHAN ANDA ',0
544: MENU9:   DB 'SISTEM MANUAL ',0
545: MENU10:  DB 'SISTEM YG AKTIF ',0
546: MENU11:  DB 'KAMERA & MOTOR 1',0
547: MENU12:  DB 'SISTEM YG AKTIF ',0
548: MENU13:  DB 'KAMERA & MOTOR 2',0
549: MENU14:  DB 'SISTEM YG AKTIF ',0
550: MENU15:  DB 'KAMERA & MOTOR 3',0

```

551: MENU16: DB 'SISTEM YG AKTIF ',0
552: MENU17: DB 'KAMERA & MOTOR 4',0
553: MENU18: DB 'PILIH KMR & MTR ',0
554: MENU19: DB 'YG AKAN DI AKTIF',0
555: MENU20: DB 'KAMERA DAN MOTOR',0
556: MENU21: DB ' 1 AKTIF ',0
557: MENU22: DB 'KAMERA DAN MOTOR',0
558: MENU23: DB ' 2 AKTIF ',0
559: MENU24: DB 'KAMERA DAN MOTOR',0
560: MENU25: DB ' 3 AKTIF ',0
561: MENU26: DB 'KAMERA DAN MOTOR',0
562: MENU27: DB ' 4 AKTIF ',0
563: MENU28: DB ' SISTEM MATI ',0
564: MENU29: DB ' ALAREM AKTIF ',0
565: END

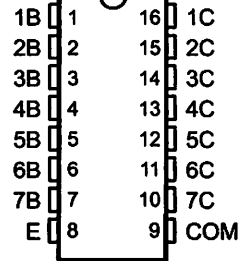
ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A HIGH-VOLTAGE HIGH-CURRENT DARLINGTON TRANSISTOR ARRAY

SLRS027F – DECEMBER 1976 – REVISED FEBRUARY 2003

The ULN2001A is obsolete and is no longer supplied.

- 500-mA-Rated Collector Current (Single Output)
- High-Voltage Outputs . . . 50 V
- Output Clamp Diodes
- Inputs Compatible With Various Types of Logic
- Relay-Driver Applications
- Designed to Be Interchangeable With Sprague ULN2001A Series

ULN2001A . . . D OR N PACKAGE
ULN2002A . . . N PACKAGE
ULN2003A, ULN2004A . . . D, N, OR NS PACKAGE
ULQ2003A, ULQ2004A . . . D OR N PACKAGE
(TOP VIEW)



description/ordering information

The ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, and ULQ2004A are high-voltage, high-current Darlington transistor arrays. Each consists of seven npn Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of a single Darlington pair is 500 mA. The Darlington pairs can be paralleled for higher current capability. Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. For 100-V (otherwise interchangeable) versions of the ULN2003A and ULN2004A, see the SN75468 and SN75469, respectively.

ORDERING INFORMATION

T _A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
-20°C to 70°C	PDIP (N)	Tube of 25	ULN2002AN	ULN2002AN
			ULN2003AN	ULN2003AN
			ULN2004AN	ULN2004AN
	SOIC (D)	Tube of 40	ULN2003AD	ULN2003A
			ULN2003ADR	
		Reel of 2500	ULN2004AD	ULN2004A
			ULN2004ADR	
	SOP (NS)	Reel of 2000	ULN2003ANSR	ULN2003A
ULN2004ANSR			ULN2004A	
-40°C to 85°C	PDIP (N)	Tube of 25	ULQ2003AN	ULQ2003A
			ULQ2004AN	ULQ2004AN
	SOIC (D)	Tube of 40	ULQ2003AD	ULQ2003A
			ULQ2003ADR	ULQ2003A
		Reel of 2500	ULQ2004AD	ULQ2004A
			ULQ2004ADR	ULQ2004A

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75285

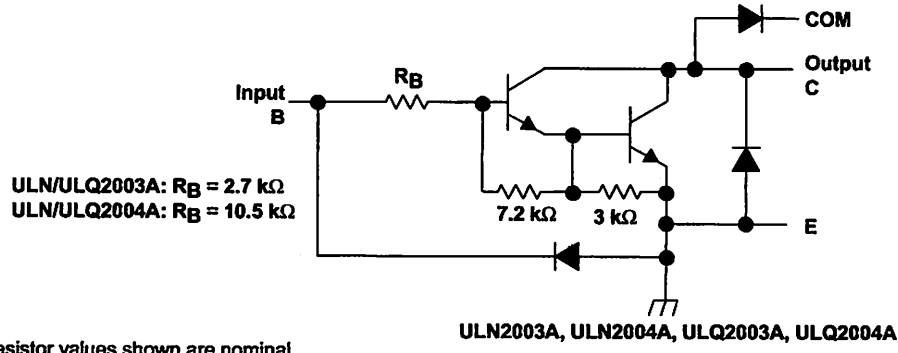
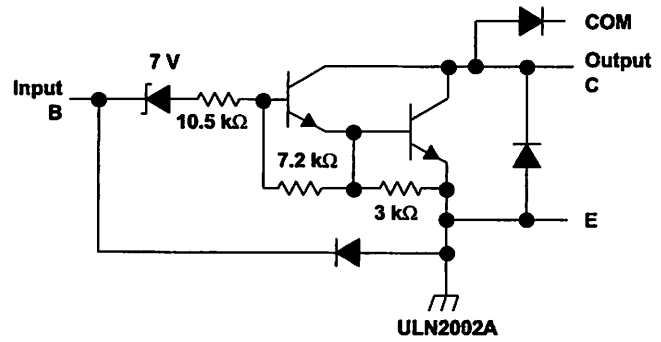
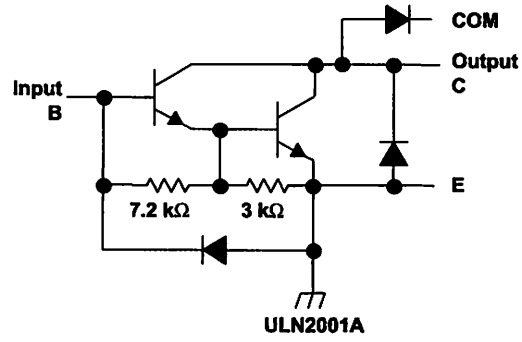
Copyright © 2003, Texas Instruments Incorporated

**ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
TRANSISTOR ARRAY**

The ULN2001A is obsolete
and is no longer supplied.

SLRS027F - DECEMBER 1976 - REVISED FEBRUARY 2003

schematics (each Darlington pair)



All resistor values shown are nominal.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A

H-VOLTAGE HIGH-CURRENT DARLINGTON TRANSISTOR ARRAY

027F – DECEMBER 1976 – REVISED FEBRUARY 2003

The ULN2001A is obsolete and is no longer supplied.

Absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Collector-emitter voltage	50 V
Clamp diode reverse voltage (see Note 1)	50 V
Input voltage, V_I (see Note 1)	30 V
Peak collector current (see Figures 14 and 15)	500 mA
Output clamp current, I_{OK}	500 mA
Total emitter-terminal current	-2.5 A
Operating free-air temperature range, T_A , ULN200xA	-20°C to 70°C
ULQ200xA	-40°C to 85°C
ULQ200xAT	-40°C to 105°C
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package	73°C/W
N package	67°C/W
NS package	64°C/W
Package thermal impedance, θ_{JC} (see Notes 4 and 5): D package	36°C/W
N package	54°C/W
Operating virtual junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	-65°C to 150°C

Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and normal operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not intended. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES:
- All voltage values are with respect to the emitter/substrate terminal E, unless otherwise noted.
 - Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\text{max}) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
 - The package thermal impedance is calculated in accordance with JESD 51-7.
 - Maximum power dissipation is a function of $T_J(\text{max})$, θ_{JC} , and T_C . The maximum allowable power dissipation at any allowable case temperature is $P_D = (T_J(\text{max}) - T_C)/\theta_{JC}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
 - The package thermal impedance is calculated in accordance with MIL-STD-883.

Typical characteristics, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS	ULN2001A			ULN2002A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
On-state input voltage	6	$V_{CE} = 2\text{ V}$, $I_C = 300\text{ mA}$						13	V
Collector-emitter saturation voltage	5	$I_I = 250\ \mu\text{A}$, $I_C = 100\text{ mA}$		0.9	1.1		0.9	1.1	V
		$I_I = 350\ \mu\text{A}$, $I_C = 200\text{ mA}$		1	1.3		1	1.3	
		$I_I = 500\ \mu\text{A}$, $I_C = 350\text{ mA}$		1.2	1.6		1.2	1.6	
Clamp forward voltage	8	$I_F = 350\text{ mA}$		1.7	2		1.7	2	V
Collector cutoff current	1	$V_{CE} = 50\text{ V}$, $I_I = 0$			50			50	μA
	2	$V_{CE} = 50\text{ V}$, $T_A = 70^\circ\text{C}$, $V_I = 6\text{ V}$			100			100	
Off-state input current	3	$V_{CE} = 50\text{ V}$, $T_A = 70^\circ\text{C}$, $I_C = 500\ \mu\text{A}$	50	65		50	65		μA
Input current	4	$V_I = 17\text{ V}$					0.82	1.25	mA
Clamp reverse current	7	$V_R = 50\text{ V}$, $T_A = 70^\circ\text{C}$			100			100	μA
		$V_R = 50\text{ V}$			50			50	
Static forward-current transfer ratio	5	$V_{CE} = 2\text{ V}$, $I_C = 350\text{ mA}$	1000						
Input capacitance		$V_I = 0$, $f = 1\text{ MHz}$		15	25		15	25	pF



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

**ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
TRANSISTOR ARRAY**

The ULN2001A is obsolete
and is no longer supplied.

SLRS027F – DECEMBER 1976 – REVISED FEBRUARY 2003

electrical characteristics, $T_A = 25^\circ\text{C}$ (unless otherwise noted) (continued)

PARAMETER	TEST FIGURE	TEST CONDITIONS	ULN2003A			ULN2004A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
$V_{I(on)}$ On-state input voltage	6	$V_{CE} = 2\text{ V}$	$I_C = 125\text{ mA}$					5	V
			$I_C = 200\text{ mA}$			2.4		6	
			$I_C = 250\text{ mA}$			2.7			
			$I_C = 275\text{ mA}$					7	
			$I_C = 300\text{ mA}$			3			
			$I_C = 350\text{ mA}$					8	
$V_{CE(sat)}$ Collector-emitter saturation voltage	5	$I_I = 250\ \mu\text{A}, I_C = 100\text{ mA}$		0.9	1.1		0.9	1.1	V
		$I_I = 350\ \mu\text{A}, I_C = 200\text{ mA}$		1	1.3		1	1.3	
		$I_I = 500\ \mu\text{A}, I_C = 350\text{ mA}$		1.2	1.6		1.2	1.6	
I_{CEX} Collector cutoff current	1	$V_{CE} = 50\text{ V}, I_I = 0$			50			50	μA
	2	$V_{CE} = 50\text{ V}, T_A = 70^\circ\text{C}$	$I_I = 0$		100			100	
			$V_I = 1\text{ V}$					500	
V_F Clamp forward voltage	8	$I_F = 350\text{ mA}$		1.7	2		1.7	2	V
$I_{I(off)}$ Off-state input current	3	$V_{CE} = 50\text{ V}, T_A = 70^\circ\text{C}$	$I_C = 500\ \mu\text{A}$	50	65		50	65	μA
I_I Input current	4	$V_I = 3.85\text{ V}$		0.93	1.35				mA
		$V_I = 5\text{ V}$				0.35	0.5		
		$V_I = 12\text{ V}$				1	1.45		
I_R Clamp reverse current	7	$V_R = 50\text{ V}$			50			50	μA
		$V_R = 50\text{ V}, T_A = 70^\circ\text{C}$			100			100	
C_i Input capacitance		$V_I = 0, f = 1\text{ MHz}$		15	25		15	25	pF



ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A

7-TRANSISTOR HIGH-VOLTAGE HIGH-CURRENT DARLINGTON

TRANSISTOR ARRAY

The ULN2001A is obsolete and is no longer supplied.

027F - DECEMBER 1976 - REVISED FEBRUARY 2003

Electrical characteristics over recommended operating conditions (unless otherwise noted)

PARAMETER	TEST FIGURE	TEST CONDITIONS	ULQ2003A			ULQ2004A			UNIT
			MIN	TYP	MAX	MIN	TYP	MAX	
On-state input voltage	6	$V_{CE} = 2\text{ V}$	$I_C = 125\text{ mA}$					5	V
			$I_C = 200\text{ mA}$			2.7		6	
			$I_C = 250\text{ mA}$			2.9			
			$I_C = 275\text{ mA}$					7	
			$I_C = 300\text{ mA}$			3			
			$I_C = 350\text{ mA}$					8	
Collector-emitter saturation voltage	5	$I_I = 250\text{ }\mu\text{A}$, $I_C = 100\text{ mA}$		0.9	1.2	0.9	1.1	V	
		$I_I = 350\text{ }\mu\text{A}$, $I_C = 200\text{ mA}$		1	1.4	1	1.3		
		$I_I = 500\text{ }\mu\text{A}$, $I_C = 350\text{ mA}$		1.2	1.7	1.2	1.6		
Collector cutoff current	1	$V_{CE} = 50\text{ V}$, $I_I = 0$			100		50	μA	
	2	$V_{CE} = 50\text{ V}$	$I_I = 0$				100		
							500		
Clamp forward voltage	8	$I_F = 350\text{ mA}$		1.7	2.3	1.7	2	V	
Off-state input current	3	$V_{CE} = 50\text{ V}$, $I_C = 500\text{ }\mu\text{A}$		65		50	65	μA	
Input current	4	$V_I = 3.85\text{ V}$		0.93	1.35			mA	
		$V_I = 5\text{ V}$				0.35	0.5		
		$V_I = 12\text{ V}$				1	1.45		
Clamp reverse current	7	$V_R = 50\text{ V}$, $T_A = 25^\circ\text{C}$			100		50	μA	
		$V_R = 50\text{ V}$			100		100		
Input capacitance		$V_I = 0$, $f = 1\text{ MHz}$		15	25	15	25	pF	

Switching characteristics, $T_A = 25^\circ\text{C}$

PARAMETER	TEST CONDITIONS	ULN2001A, ULN2002A, ULN2003A, ULN2004A			UNIT
		MIN	TYP	MAX	
Propagation delay time, low- to high-level output	See Figure 9		0.25	1	μs
Propagation delay time, high- to low-level output	See Figure 9		0.25	1	μs
High-level output voltage after switching	$V_S = 50\text{ V}$, $I_O \approx 300\text{ mA}$, See Figure 10	$V_S - 20$			mV

Switching characteristics over recommended operating conditions (unless otherwise noted)

PARAMETER	TEST CONDITIONS	ULQ2003A, ULQ2004A			UNIT
		MIN	TYP	MAX	
Propagation delay time, low- to high-level output	See Figure 9		1	10	μs
Propagation delay time, high- to low-level output	See Figure 9		1	10	μs
High-level output voltage after switching	$V_S = 50\text{ V}$, $I_O \approx 300\text{ mA}$, See Figure 10	$V_S - 500$			mV



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
**HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
 TRANSISTOR ARRAY**

The ULN2001A is obsolete
 and is no longer supplied.

SLRS027F - DECEMBER 1976 - REVISED FEBRUARY 2003

PARAMETER MEASUREMENT INFORMATION

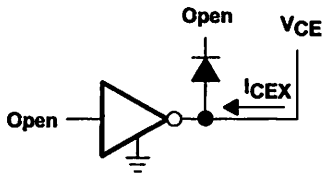


Figure 1. I_{CEX} Test Circuit

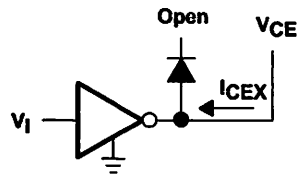


Figure 2. I_{CEX} Test Circuit

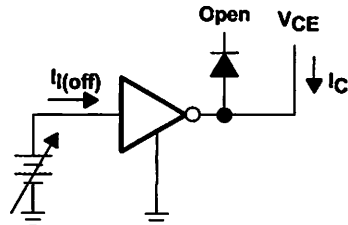


Figure 3. $I_{I(off)}$ Test Circuit

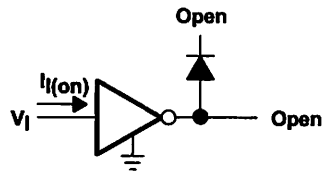
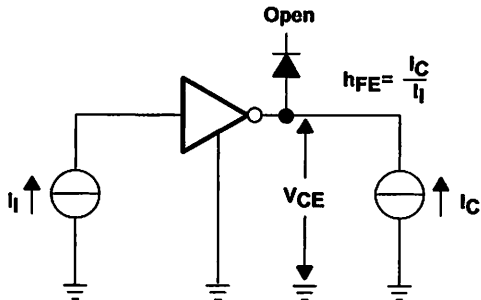


Figure 4. I_I Test Circuit



NOTE: I_I is fixed for measuring $V_{CE(sat)}$, variable for measuring h_{FE} .

Figure 5. h_{FE} , $V_{CE(sat)}$ Test Circuit

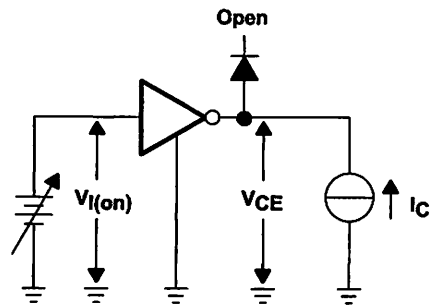


Figure 6. $V_{I(on)}$ Test Circuit

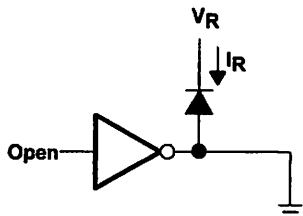


Figure 7. I_R Test Circuit

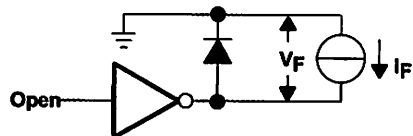


Figure 8. V_F Test Circuit

PARAMETER MEASUREMENT INFORMATION

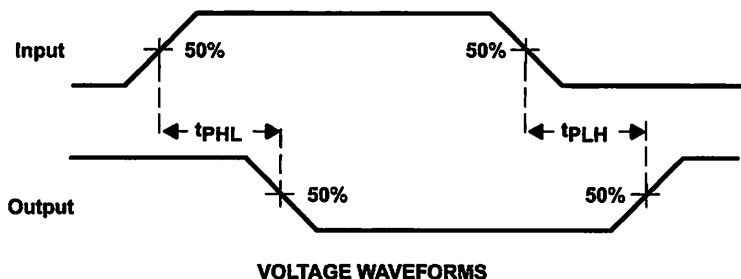
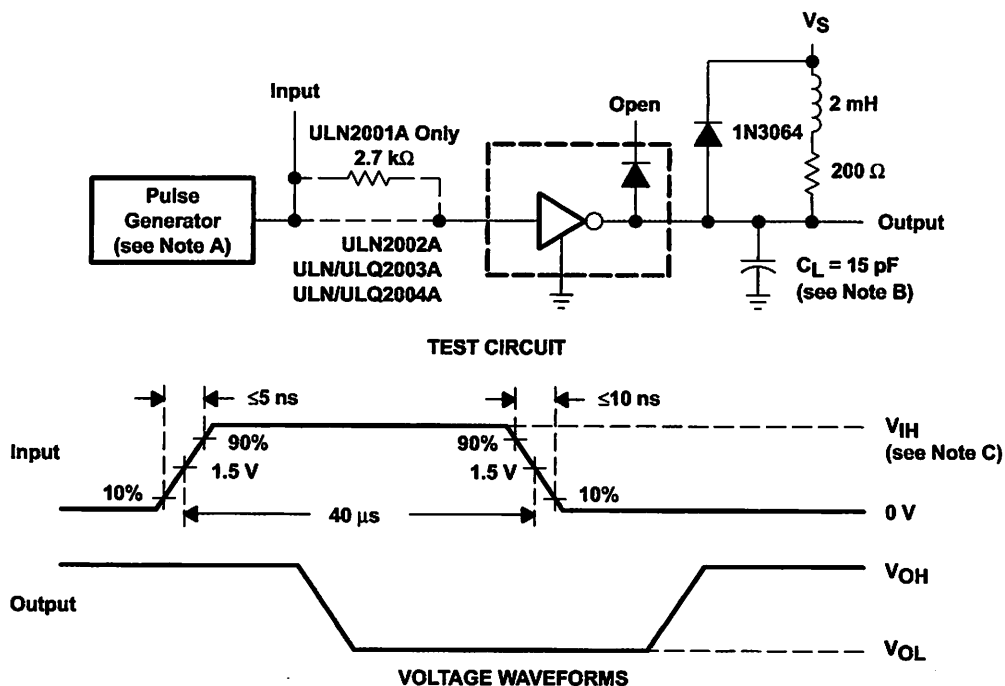


Figure 9. Propagation Delay-Time Waveforms



- ES: A. The pulse generator has the following characteristics: PRR = 12.5 kHz, $Z_O = 50 \Omega$.
 B. C_L includes probe and jig capacitance.
 C. For testing the ULN2001A, the ULN2003A, and the ULQ2003A, $V_{IH} = 3 \text{ V}$; for the ULN2002A, $V_{IH} = 13 \text{ V}$; for the ULN2004A and the ULQ2004A, $V_{IH} = 8 \text{ V}$.

Figure 10. Latch-Up Test Circuit and Voltage Waveforms

ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
**HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
 TRANSISTOR ARRAY**

The ULN2001A is obsolete
 and is no longer supplied.

SLRS027F – DECEMBER 1976 – REVISED FEBRUARY 2003

TYPICAL CHARACTERISTICS

**COLLECTOR-EMITTER
 SATURATION VOLTAGE
 VS
 COLLECTOR CURRENT
 (ONE DARLINGTON)**

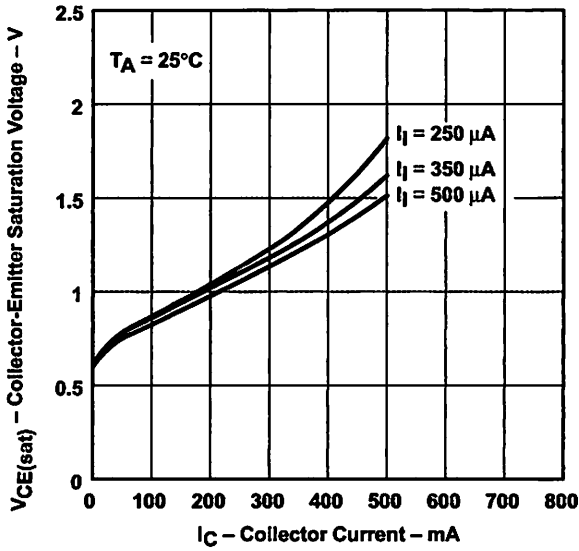


Figure 11

**COLLECTOR-EMITTER
 SATURATION VOLTAGE
 VS
 TOTAL COLLECTOR CURRENT
 (TWO DARLINGTONS IN PARALLEL)**

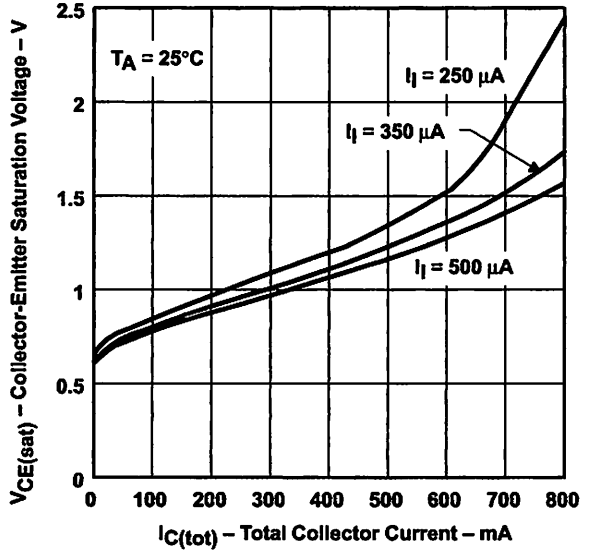


Figure 12

**COLLECTOR CURRENT
 VS
 INPUT CURRENT**

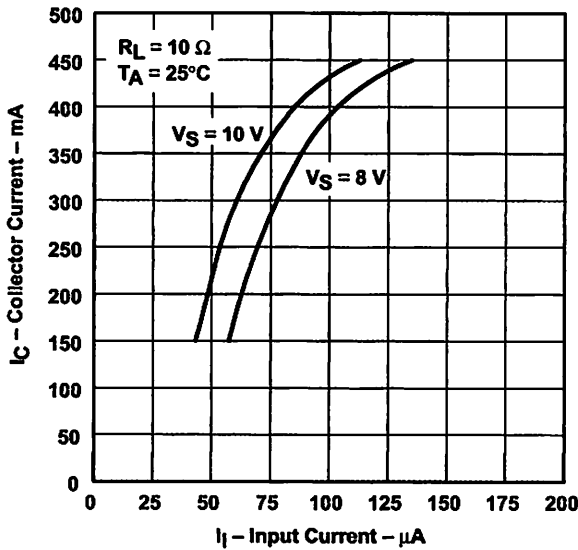


Figure 13



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
TRANSISTOR ARRAY

0027F - DECEMBER 1976 - REVISED FEBRUARY 2003

The ULN2001A is obsolete
 and is no longer supplied.

THERMAL INFORMATION

D PACKAGE
MAXIMUM COLLECTOR CURRENT
VS
DUTY CYCLE

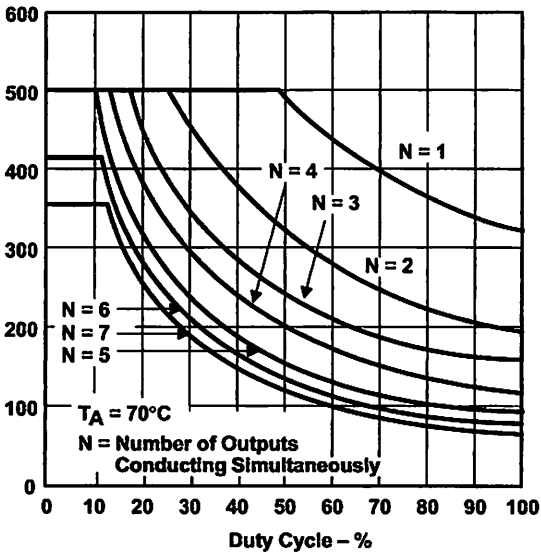


Figure 14

N PACKAGE
MAXIMUM COLLECTOR CURRENT
VS
DUTY CYCLE

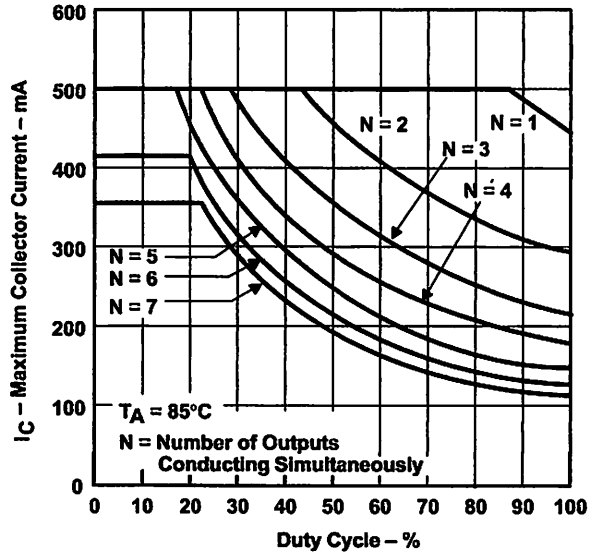


Figure 15



POST OFFICE BOX 855303 • DALLAS, TEXAS 75285

ULN2001A, ULN2002A, ULN2003A, ULN2004A, ULQ2003A, ULQ2004A
**HIGH-VOLTAGE HIGH-CURRENT DARLINGTON
 TRANSISTOR ARRAY**

The ULN2001A is obsolete
 and is no longer supplied.

SLRS027F - DECEMBER 1976 - REVISED FEBRUARY 2003

APPLICATION INFORMATION

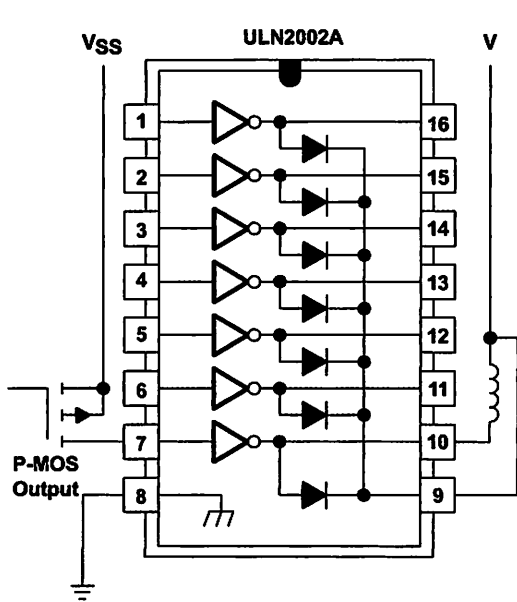


Figure 16. P-MOS to Load

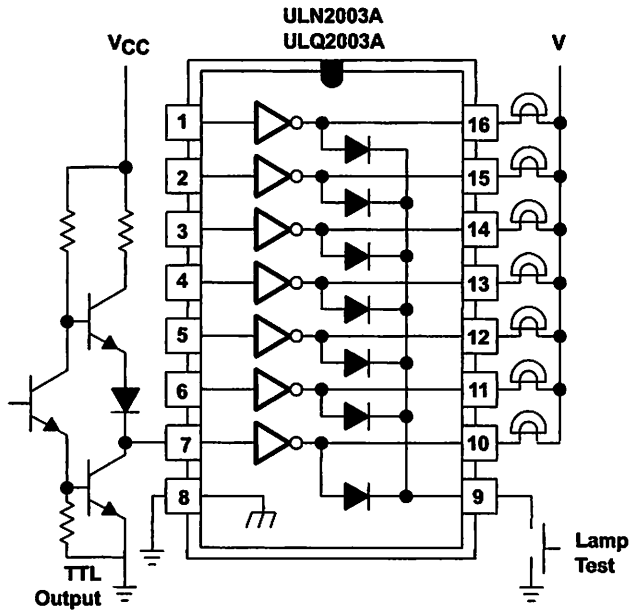


Figure 17. TTL to Load

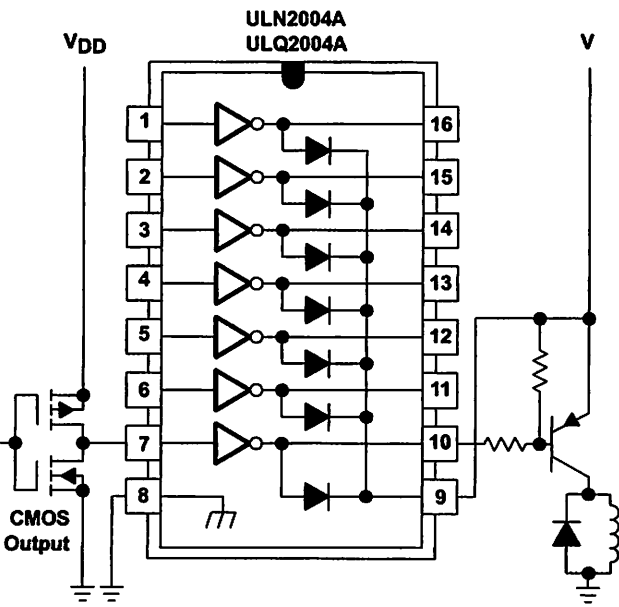


Figure 18. Buffer for Higher Current Loads

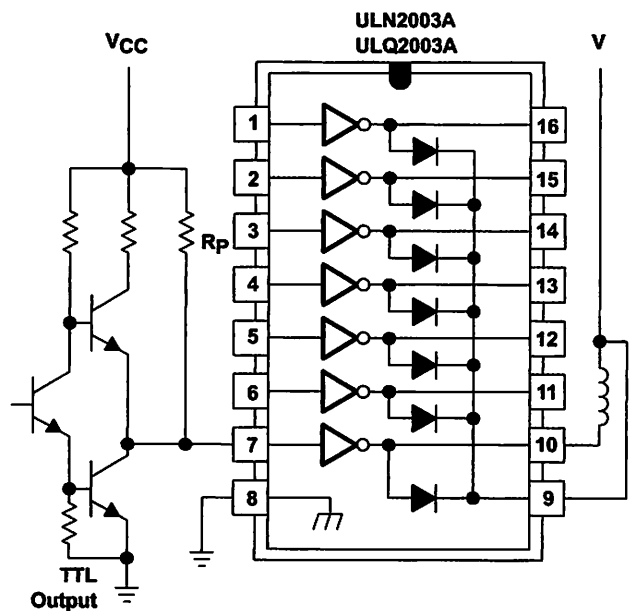


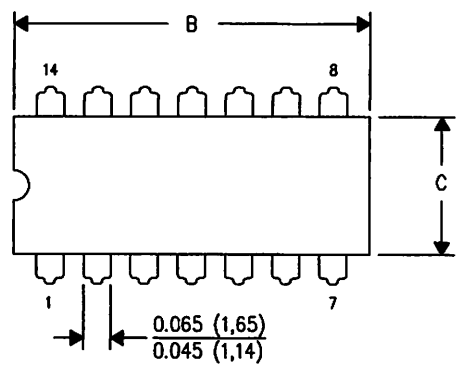
Figure 19. Use of Pullup Resistors
 to Increase Drive Current



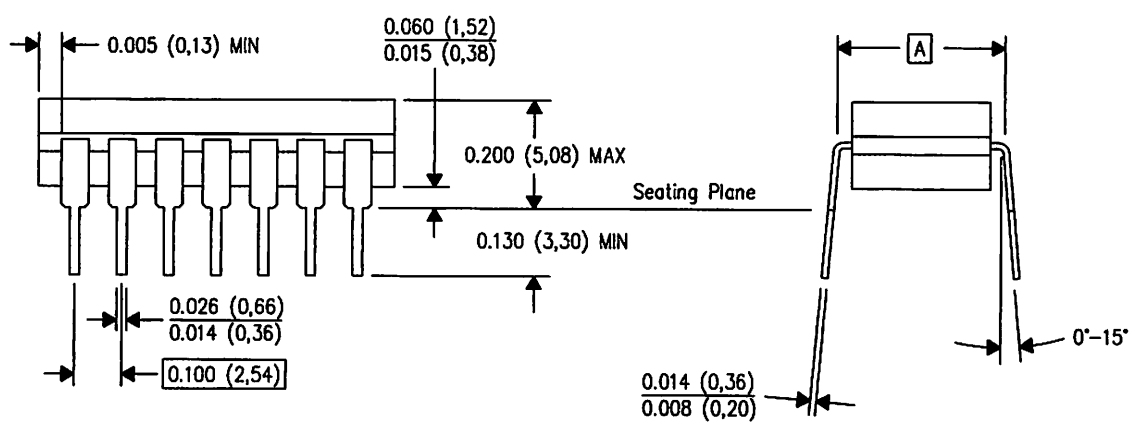
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

(R-GDIP-T**) LEADS SHOWN

CERAMIC DUAL IN-LINE PACKAGE



DIM \ PINS **	14	16	18	20
A	0.300 (7,62) BSC	0.300 (7,62) BSC	0.300 (7,62) BSC	0.300 (7,62) BSC
B MAX	0.785 (19,94)	.840 (21,34)	0.960 (24,38)	1.060 (26,92)
B MIN	—	—	—	—
C MAX	0.300 (7,62)	0.300 (7,62)	0.310 (7,87)	0.300 (7,62)
C MIN	0.245 (6,22)	0.245 (6,22)	0.220 (5,59)	0.245 (6,22)



4040083/F 03/03

- NOTES:
- A. All linear dimensions are in inches (millimeters).
 - B. This drawing is subject to change without notice.
 - C. This package is hermetically sealed with a ceramic lid using glass frit.
 - D. Index point is provided on cap for terminal identification only on press ceramic glass frit seal only.
 - E. Falls within MIL STD 1835 GDIP1-T14, GDIP1-T16, GDIP1-T18 and GDIP1-T20.

MECHANICAL

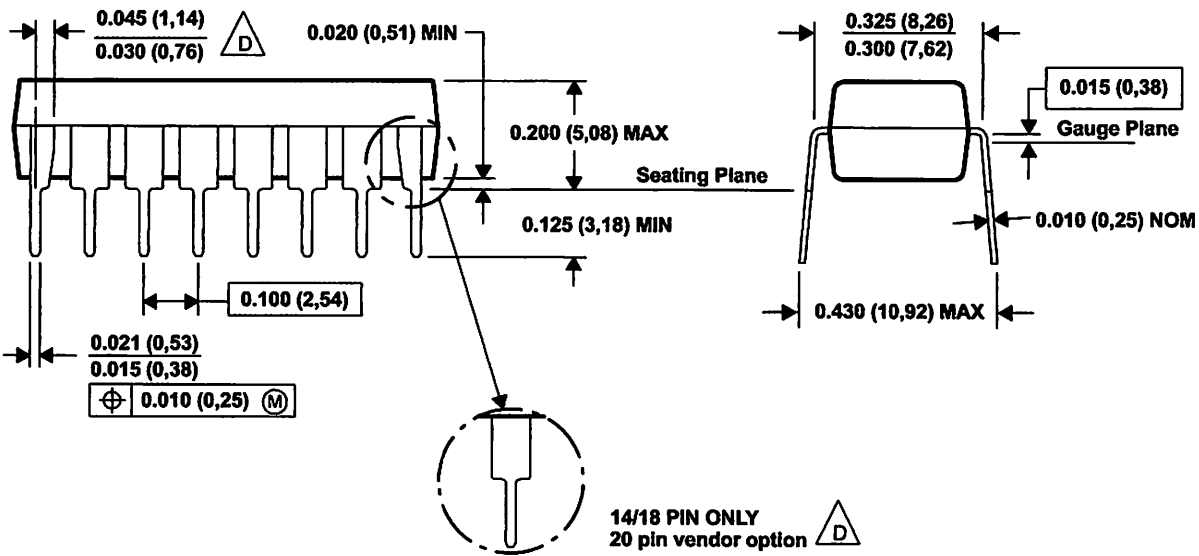
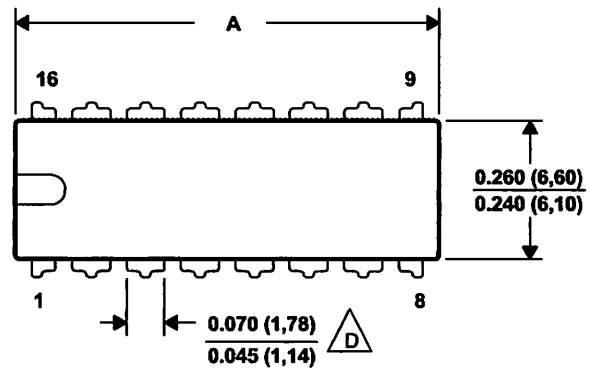
MPDI002C - JANUARY 1995 - REVISED DECEMBER 2002

N (R-PDIP-T)**

PLASTIC DUAL-IN-LINE PACKAGE

16 PINS SHOWN

PINS ** DIM	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
MS-100 VARIATION	AA	BB	AC	AD



14/18 PIN ONLY
20 pin vendor option $\triangle D$

4040049/E 12/2002

- NOTES: A. All linear dimensions are in inches (millimeters).
 B. This drawing is subject to change without notice.
 $\triangle C$ Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
 $\triangle D$ The 20 pin end lead shoulder width is a vendor option, either half or full width.

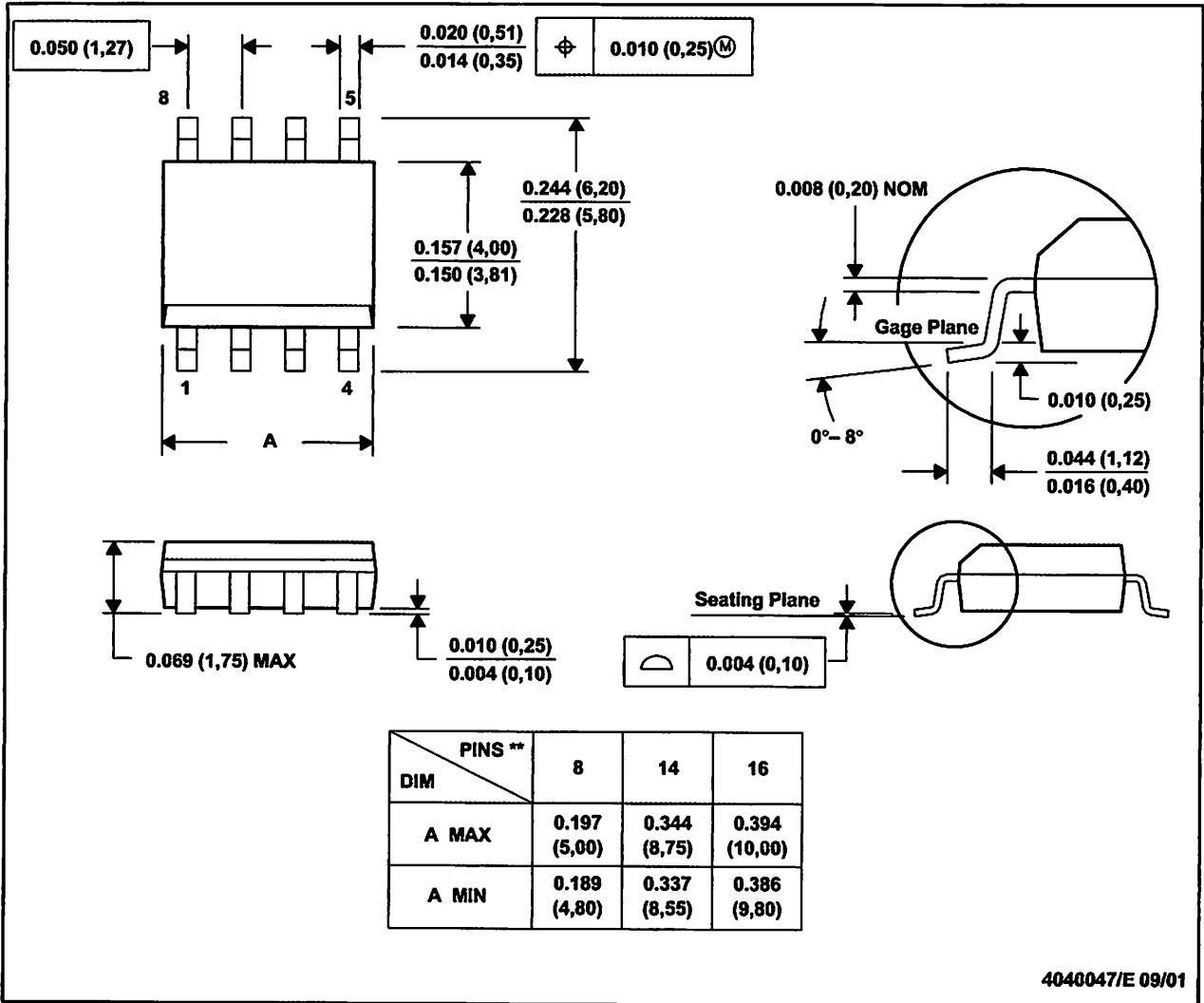
MECHANICAL DATA

MSOI002B - JANUARY 1995 - REVISED SEPTEMBER 2001

D (R-PDSO-G**)

PLASTIC SMALL-OUTLINE PACKAGE

8 PINS SHOWN



- NOTES: A. All linear dimensions are in inches (millimeters).
 B. This drawing is subject to change without notice.
 C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0,15).
 D. Falls within JEDEC MS-012

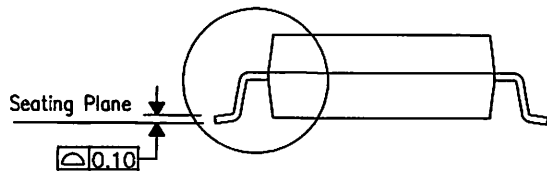
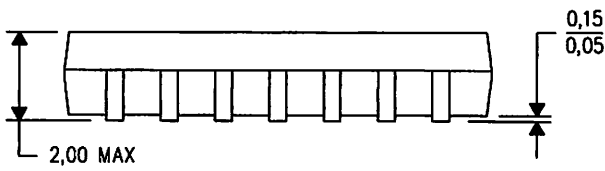
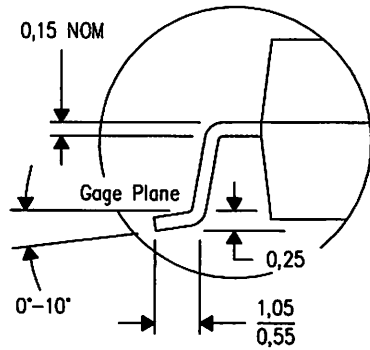
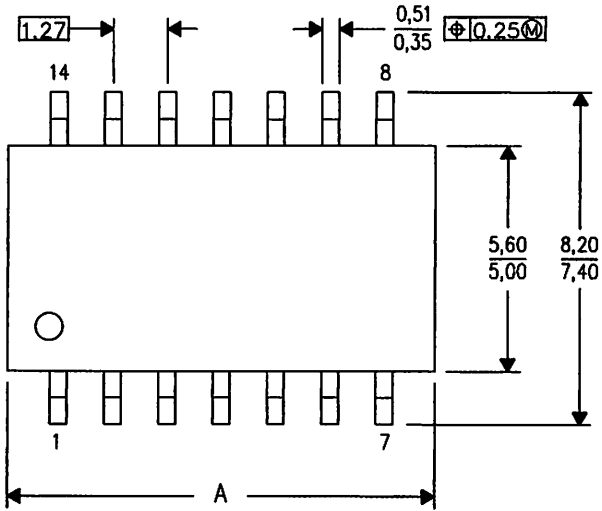


POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

MECHANICAL DATA

S (R-PDSO-G)**
PINS SHOWN

PLASTIC SMALL-OUTLINE PACKAGE



DIM \ PINS **	14	16	20	24
A MAX	10,50	10,50	12,90	15,30
A MIN	9,90	9,90	12,30	14,70

4040062/C 03/03

- NOTES:
- A. All linear dimensions are in millimeters.
 - B. This drawing is subject to change without notice.
 - C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

**Texas Instruments
Post Office Box 655303
Dallas, Texas 75265**

Features

Compatible with MCS-51® Products
4K Bytes of In-System Programmable (ISP) Flash Memory
Endurance: 1000 Write/Erase Cycles
3 to 5.5V Operating Range
Static Operation: 0 Hz to 33 MHz
Two-Level Program Memory Lock
8-bit Internal RAM
8 Programmable I/O Lines
Two 16-bit Timer/Counters
Five Interrupt Sources
Full Duplex UART Serial Channel
Power Idle and Power-down Modes
Rapid Recovery from Power-down Mode
Watchdog Timer
Two Data Pointers
Power-off Flag
Short Programming Time
In-System ISP Programming (Byte and Page Mode)

Description

The AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using advanced 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 single chip, the Atmel AT89S51 is a powerful microcontroller which provides a flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of internal RAM, 8 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-level interrupt architecture, a full duplex serial port, on-chip oscillator, and support 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 content but freezes the oscillator, disabling all other chip functions until the next external reset or hardware reset.



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

AT89S51

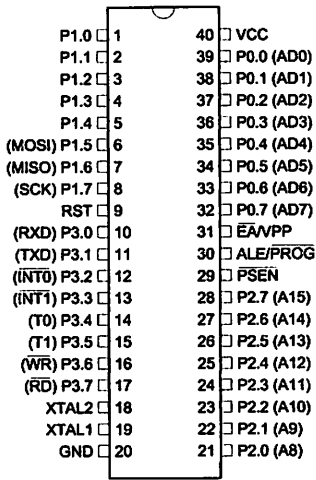
Rev. 2487A-10/01



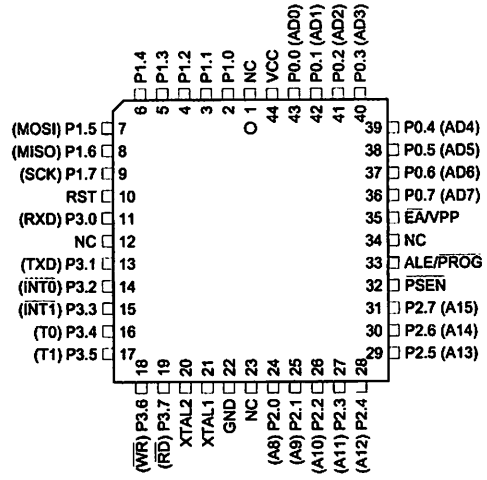


Configurations

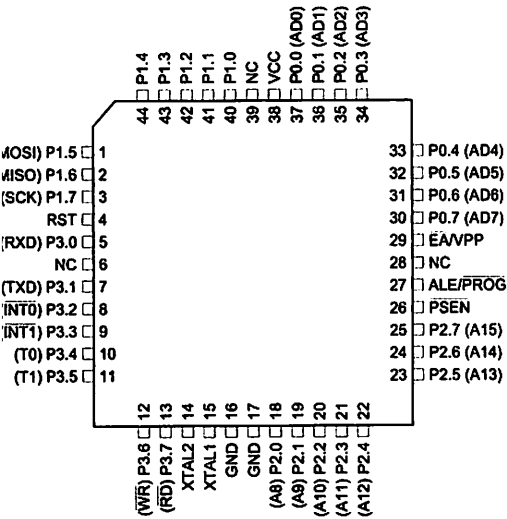
PDIP



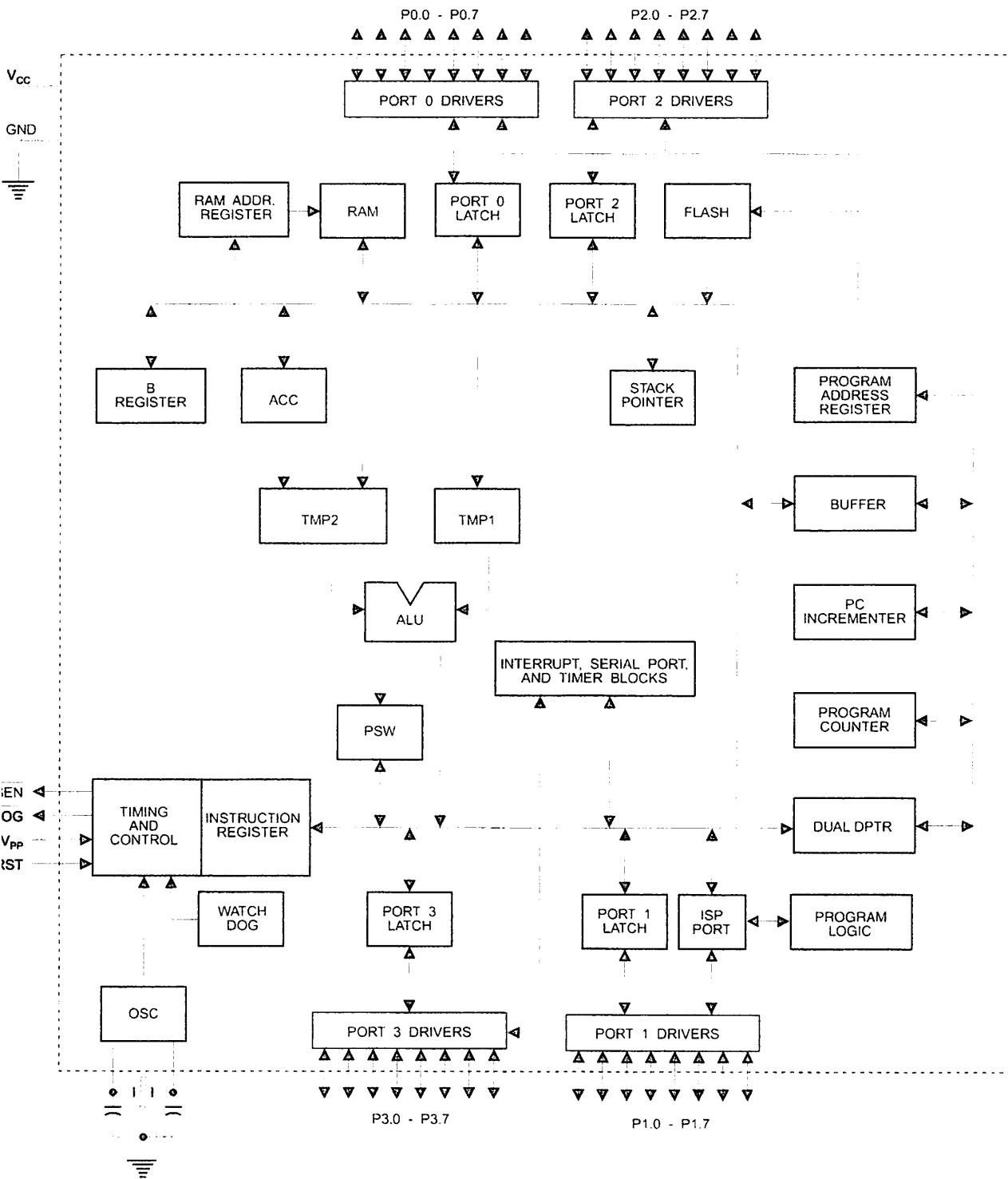
PLCC



TQFP



Block Diagram





Description

Supply voltage.

Ground.

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

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.

$\overline{\text{PSEN}}$

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.

PP

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.

.1

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

.2

Output from the inverting oscillator amplifier





Special
Function
Registers

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

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

1. AT89S51 SFR Map and Reset Values

								0FFH
B 00000000								0F7H
								0EFH
ACC 00000000								0E7H
								0DFH
PSW 00000000								0D7H
								0CFH
								0C7H
IP XX000000								0BFH
P3 11111111								0B7H
IE 0X000000								0AFH
P2 11111111		AUXR1 XXXXXXXX0				WDTRST XXXXXXXXX		0A7H
SCON 00000000	SBUF XXXXXXXXX							9FH
P1 11111111								97H
TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XX00000	87H

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

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

Table 2. AUXR: Auxiliary Register

AUXR	Address = 8EH							Reset Value = XXX00XX0B
Not Bit Addressable								
Bit	-	-	-	WDIDLE	DISRTO	-	-	DISALE
	7	6	5	4	3	2	1	0
-	Reserved for future expansion							
DISALE	Disable/Enable ALE							
	DISALE							
	Operating Mode							
	0	ALE is emitted at a constant rate of 1/6 the oscillator frequency						
	1	ALE is active only during a MOVX or MOVC instruction						
DISRTO	Disable/Enable Reset out							
	DISRTO							
	0	Reset pin is driven High after WDT times out						
	1	Reset pin is input only						
WDIDLE	Disable/Enable WDT in IDLE mode							
	WDIDLE							
	0	WDT continues to count in IDLE mode						
	1	WDT halts counting in IDLE mode						

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





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

Table 3. AUXR1: Auxiliary Register 1

AUXR1							
Address = A2H							
							Reset Value = XXXXXXX0B
Not Bit Addressable							
Bit	-	-	-	-	-	-	DPS
	7	6	5	4	3	2	1
-	Reserved for future expansion						
DPS	Data Pointer Register Select						
	DPS						
	0	Selects DPTR Registers DP0L, DP0H					
	1	Selects DPTR Registers DP1L, DP1H					

dry
nization

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.

am 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.

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.

ndog

time
led with
-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.

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

AT89S51

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

During Power-down 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.

T

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

Timer 0 and 1

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

Interrupts

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

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

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

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



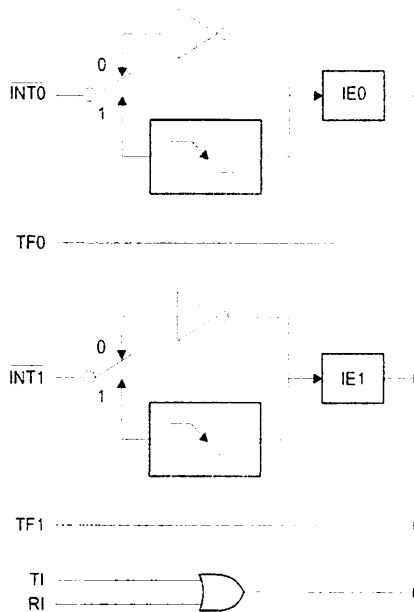
Table 4. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	-	-	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt. Enable Bit = 0 disables the interrupt.							

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
-	IE.6	Reserved
-	IE.5	Reserved
ES	IE.4	Serial Port interrupt enable bit
ET1	IE.3	Timer 1 interrupt enable bit
EX1	IE.2	External interrupt 1 enable bit
ET0	IE.1	Timer 0 interrupt enable bit
EX0	IE.0	External interrupt 0 enable bit

User software should never write 1s to reserved bits, because they may be used in future AT89 products.

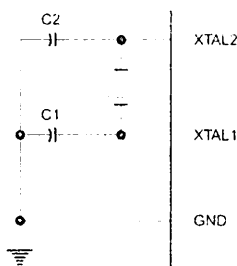
Figure 1. Interrupt Sources



Oscillator Characteristics

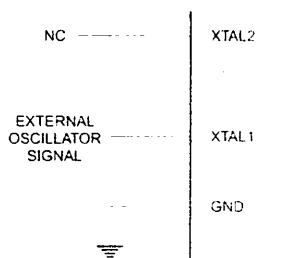
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

Figure 2. Oscillator Connections



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

Figure 3. External Clock Drive Configuration



Idle Mode

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

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

Power-down Mode

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





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

Mode	Program Memory	ALE	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

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

Table 6. Lock Bit Protection Modes

Program Lock Bits				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.

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

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

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

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V.
5. Pulse ALE/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 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. 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 $\overline{RDY/BSY}$ 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 \overline{READY} .

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

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

(000H) = 1EH indicates manufactured by Atmel

(100H) = 51H indicates 89S51

(200H) = 06H

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

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

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

Programming Flash – Parallel Mode

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

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

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

Serial Programming Algorithm

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

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





Power-off sequence (if needed):

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

Set RST to "L".

Turn V_{CC} power off.

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

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






Programming Instruction Set

Programming Interface – Parallel Mode

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

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

7. Flash Programming Modes

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

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

Figure 4. Programming the Flash Memory (Parallel Mode)

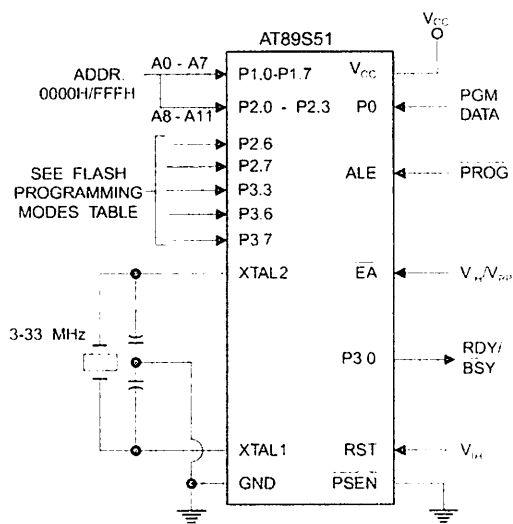
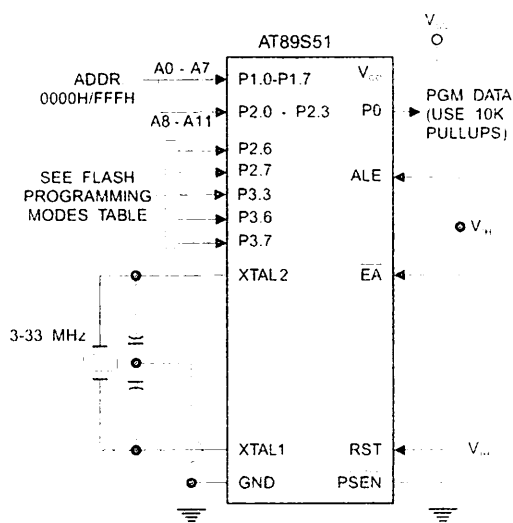


Figure 5. Verifying the Flash Memory (Parallel Mode)



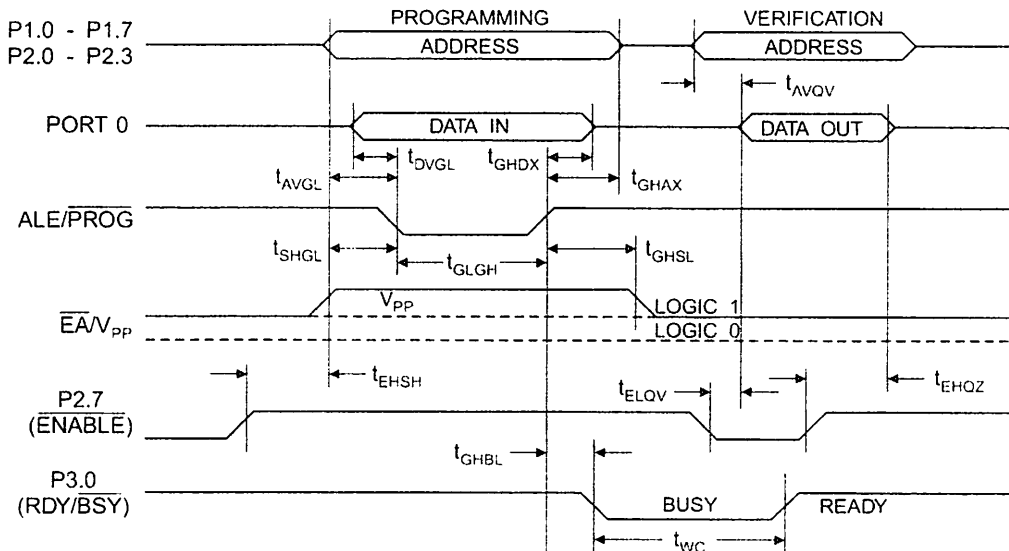


Flash Programming and Verification Characteristics (Parallel Mode)

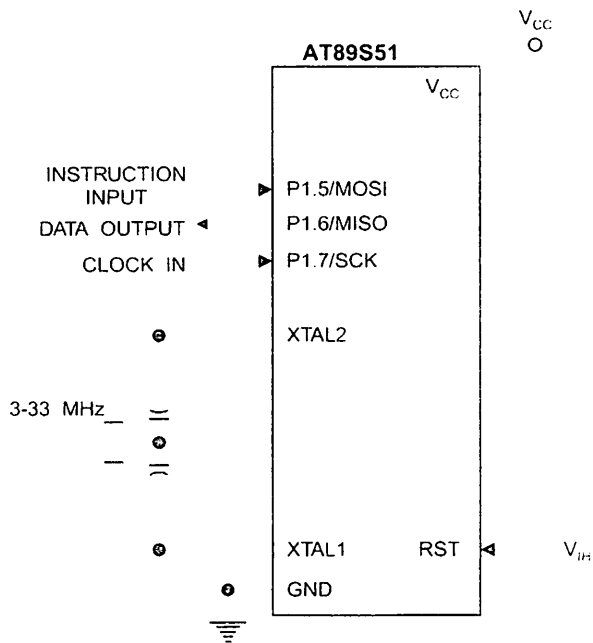
0°C to 30°C, $V_{CC} = 4.5$ to $5.5V$

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

6. Flash Programming and Verification Waveforms – Parallel Mode

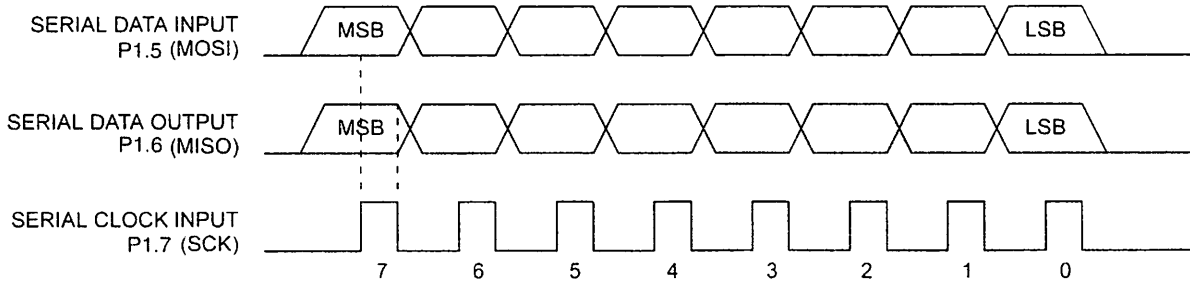


7. Flash Memory Serial Downloading



8. Programming and Verification Waveforms – Serial Mode

8.1. Serial Programming Waveforms





8. 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)	Enable Serial Programming while RST is high
Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Program Memory (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
Program Memory (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
Lock Bits ⁽²⁾	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Signature Bytes ⁽¹⁾	0010 1000	xxx A9 A8 A7 A6 A5 A4 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
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)
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)

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

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

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

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

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

Serial Programming Characteristics

Figure 9. Serial Programming Timing

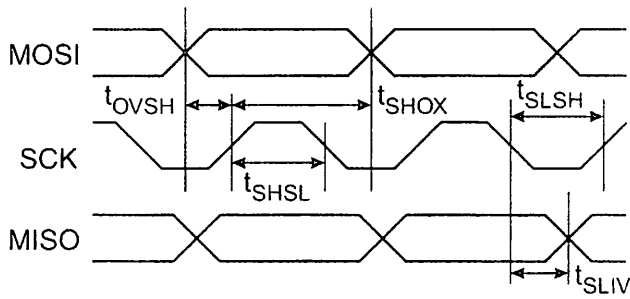


Table 9. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
f _{CL}	Oscillator Frequency	0		33	MHz
T _{CL}	Oscillator Period	30			ns
t _{SL}	SCK Pulse Width High	8 t _{CLCL}			ns
t _{SH}	SCK Pulse Width Low	8 t _{CLCL}			ns
t _{SH}	MOSI Setup to SCK High	t _{CLCL}			ns
t _{HOX}	MOSI Hold after SCK High	2 t _{CLCL}			ns
t _{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t _{USE}	Chip Erase Instruction Cycle Time			500	ms
t _{WC}	Serial Byte Write Cycle Time			64 t _{CLCL} + 400	μs



Absolute Maximum Ratings*

Operating Temperature.....	-55°C to +125°C
Storage Temperature.....	-65°C to +150°C
Voltage on Any Pin with respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
Maximum 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.

Characteristics

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
	Input Low Voltage	(Except $\bar{E}A$)	-0.5	$0.2 V_{CC}-0.1$	V
	Input Low Voltage ($\bar{E}A$)		-0.5	$0.2 V_{CC}-0.3$	V
	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC}+0.9$	$V_{CC}+0.5$	V
	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC}+0.5$	V
	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
	Output Low Voltage ⁽¹⁾ (Port 0, ALE, $\bar{P}SEN$)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
	Output High Voltage (Ports 1,2,3, ALE, $\bar{P}SEN$)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	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
	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
	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	μA
	Input Leakage Current (Port 0, $\bar{E}A$)	$0.45 < V_{IN} < V_{CC}$		± 10	μA
	Reset Pulldown Resistor		50	300	K Ω
	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
	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

- 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.
- Minimum V_{CC} for Power-down is 2V.

AT89S51

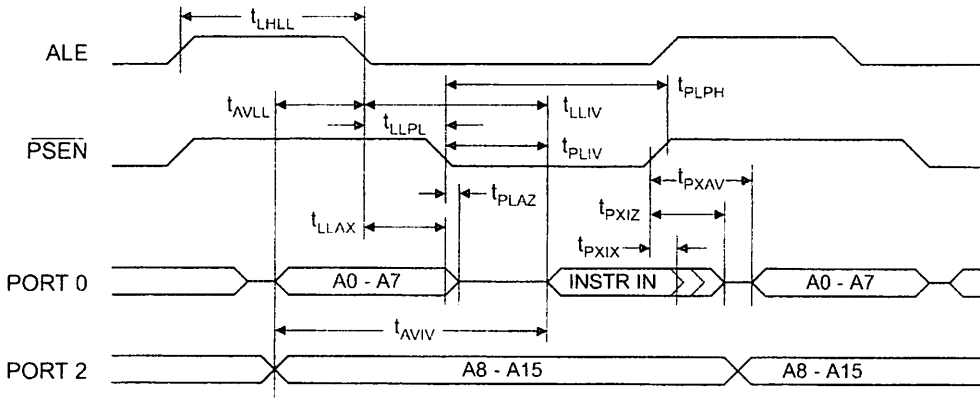
Characteristics

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

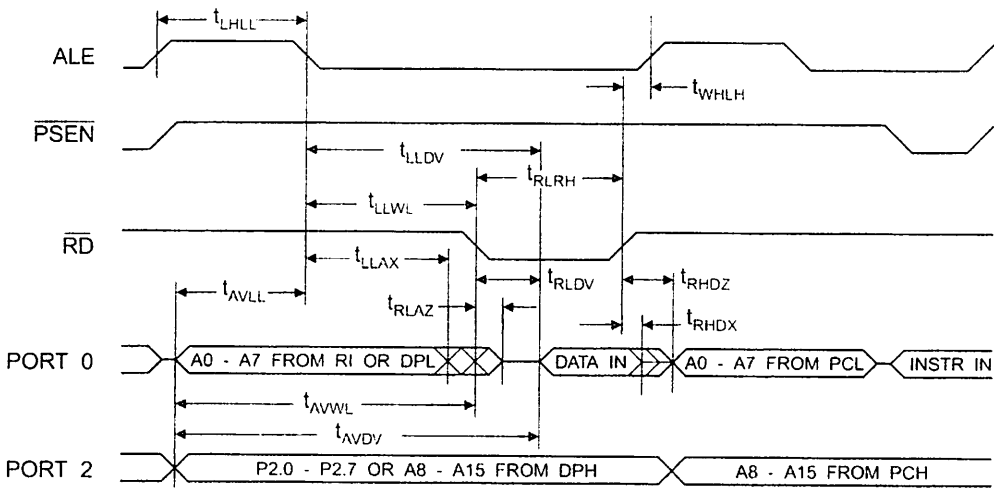
Internal Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
	Oscillator Frequency			0	33	MHz
	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
	Address Valid to ALE Low	43		$t_{\text{CLCL}}-25$		ns
	Address Hold After ALE Low	48		$t_{\text{CLCL}}-25$		ns
	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-25$		ns
	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-45$		ns
	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-60$	ns
	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-25$	ns
	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-80$	ns
	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
	Data Hold After $\overline{\text{RD}}$	0		0		ns
	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-30$		ns
	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-130$		ns
	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-25$		ns
	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-25$	$t_{\text{CLCL}}+25$	ns

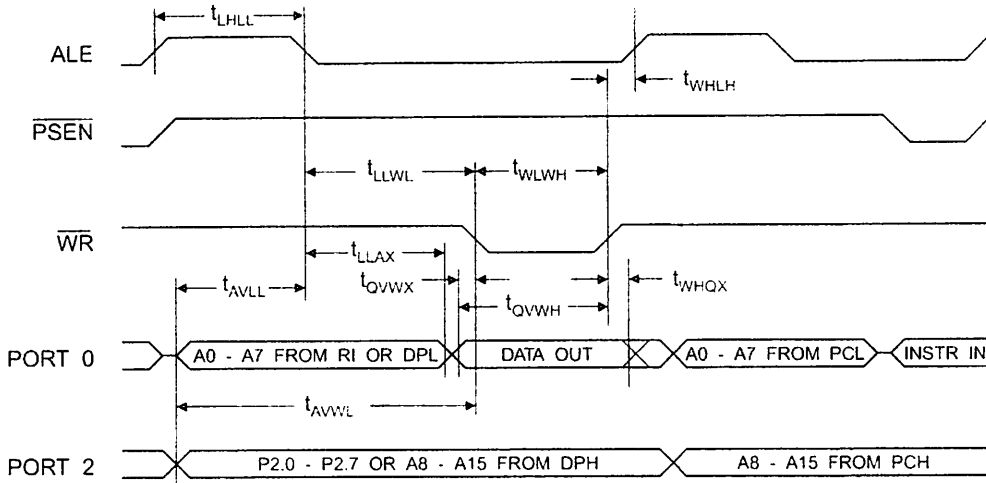
Internal Program Memory Read Cycle



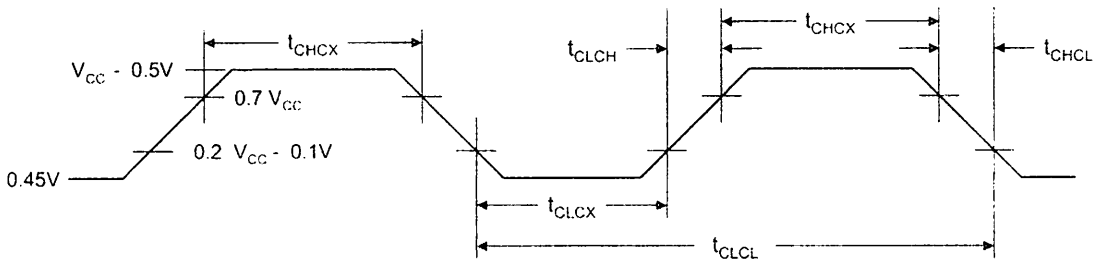
Internal Data Memory Read Cycle



Internal Data Memory Write Cycle



Internal Clock Drive Waveforms



Internal Clock Drive

Symbol	Parameter	Min	Max	Units
f _{osc}	Oscillator Frequency	0	33	MHz
T _{osc}	Clock Period	30		ns
t _{high}	High Time	12		ns
t _{low}	Low Time	12		ns
t _r	Rise Time		5	ns
t _f	Fall Time		5	ns



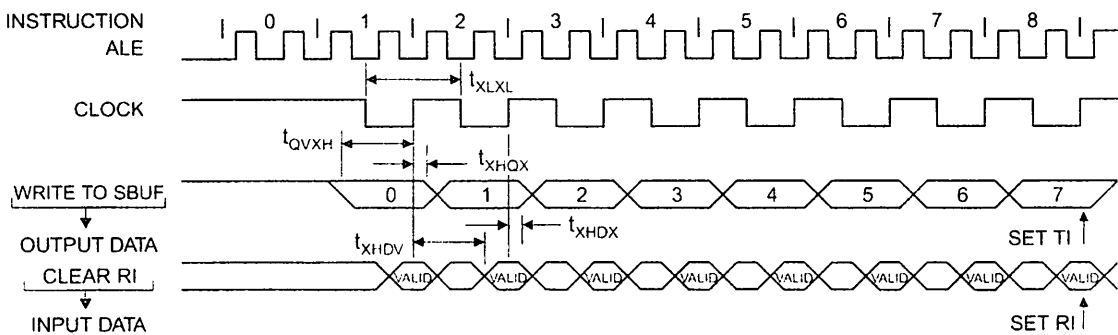


Serial Port Timing: Shift Register Mode Test Conditions

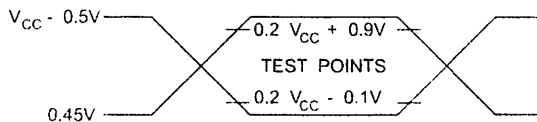
Values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = 80 pF .

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

Shift Register Mode Timing Waveforms

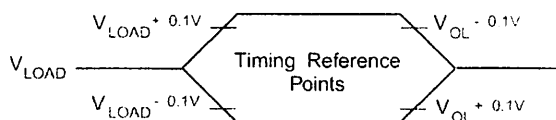


Testing Input/Output Waveforms⁽¹⁾



- 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.

Output Waveforms⁽¹⁾



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

Ordering Information

Order Code	Power Supply	Ordering Code	Package	Operation Range
4	4.0V to 5.5V	AT89S51-24AC	44A	Commercial (0° C to 70° C)
		AT89S51-24JC	44J	
		AT89S51-24PC	40P6	
		AT89S51-24AI	44A	Industrial (-40° C to 85° C)
		AT89S51-24JI	44J	
		AT89S51-24PI	40P6	
3	4.5V to 5.5V	AT89S51-33AC	44A	Commercial (0° C to 70° C)
		AT89S51-33JC	44J	
		AT89S51-33PC	40P6	

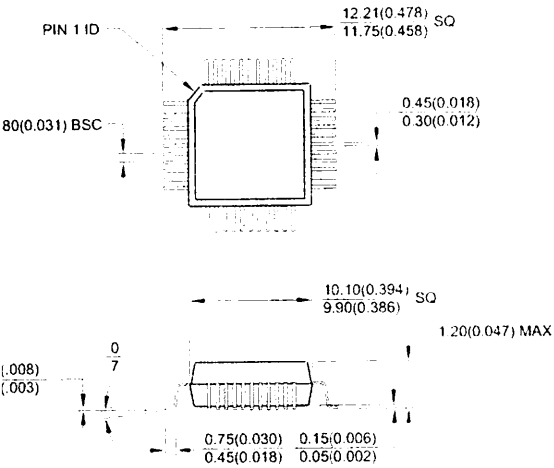
= Preliminary Availability

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

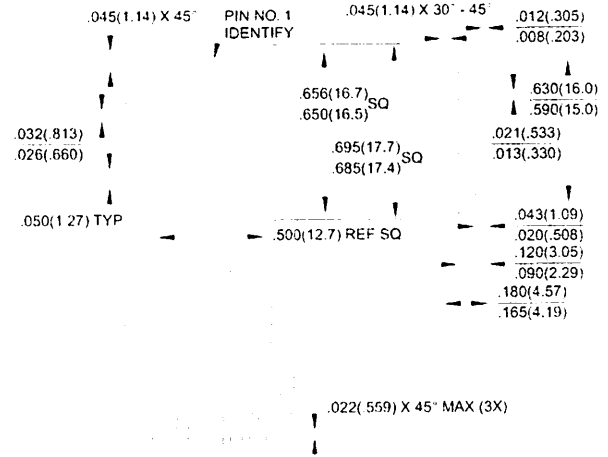


Mounting Information

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

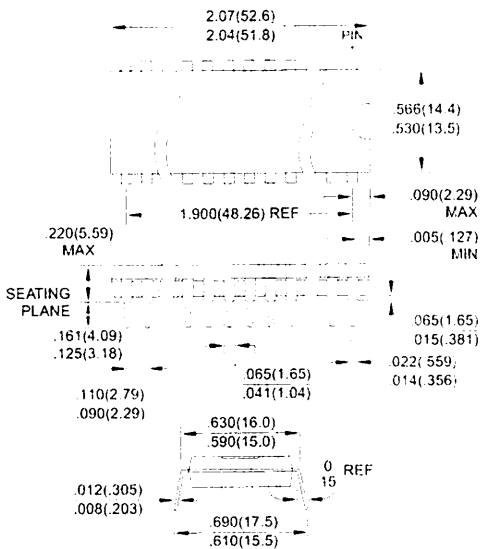


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)



Controlling dimension: millimeters

PDIP, 40-pin, 0.600" Wide, Plastic Dual In Line Package (PDIP)
Dimensions in Inches and (Millimeters)
IEEE STANDARD MS-011 AC





Atmel Headquarters

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

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

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

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

Atmel Product Operations

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

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

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

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

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

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

e-mail
literature@atmel.com

Web Site
<http://www.atmel.com>

Atmel Corporation 2001.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty as detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors that may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Atmel is the registered trademark of Atmel.

Intel is the registered trademark of Intel Corporation. Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

2487A-10/01/xM