

# SKRIPSI

**PERANCANGAN DAN PEMBUATAN MODEL SISTEM  
KEAMANAN PINTU GERBANG PADA PABRIK BERBASIS  
MIKROKONTROLER AT89S51 DIKOMUNIKASIKAN KE PC  
MELALUI USB**



Disusun Oleh :  
**SAIFUL ALIM**  
03.17.032

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

**MARET 2008**

REKAM KORBAN

REKAM KORBAN TERBUKA WISDOMAT KORBAN  
- REKAM KORBAN TERBUKA WISDOMAT  
REKAM KORBAN TERBUKA WISDOMAT  
REKAM KORBAN TERBUKA WISDOMAT 2-1

REKAM KORBAN  
REKAM KORBAN  
REKAM KORBAN

REKAM KORBAN

REKAM KORBAN TERBUKA WISDOMAT KORBAN KE BC  
REKAM KORBAN TERBUKA WISDOMAT KORBAN KE BC  
REKAM KORBAN TERBUKA WISDOMAT KORBAN KE BC

REKAM KORBAN

## LEMBAR PERSETUJUAN

# PERANCANGAN DAN PEMBUATAN MODEL SISTEM KEAMANAN PINTU GERBANG PADA PABRIK BERBASIS MIKROKONTROLER AT89S51 DIKOMUNIKASIKAN KE PC MELALUI USB

## SKRIPSI

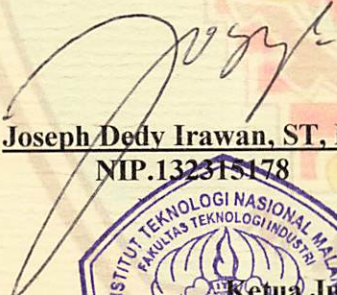
*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

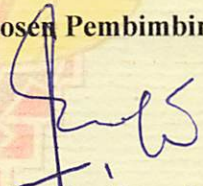
Disusun Oleh :  
SAIFUL ALIM  
03.17.032

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

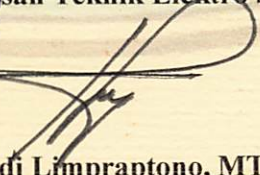
  
Joseph Dedy Irawan, ST, MT  
NIP.132315178

  
Ir. Eko Nurcahyo  
NIP. Y. 1028700172



Mengetahui

Ketua Jurusan Teknik Elektro S-1

  
Ir. F. Yudi Limpraptono, MT  
NIP.Y. 1039500274

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

MARET 2008



**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Saiful Alim  
NIM : 03.17.032  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Masa Bimbingan : 15 Desember 2007 s/d 15 Juni 2008  
Judul Skripsi : Perancangan dan Pembuatan Model Sistem Keamanan  
Pintu Gerbang Pada Pabrik Berbasis Mikrokontroler  
AT89S51 Dikomunikasikan Ke PC Melalui USB

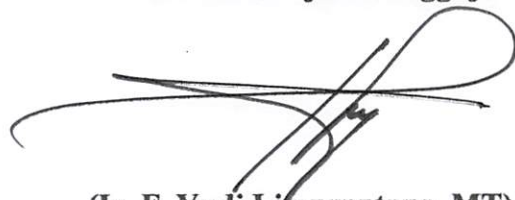
Dipertahankan di hadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Sabtu  
Tanggal : 15 Maret 2008  
Dengan Nilai : 78,2 (B+) *84*



**Ketua Majelis Penguji**

**(Ir. Mochtar Asroni, MSME)**  
NIP. Y. 1018100036



**Sekretaris Majelis Penguji**

**(Ir. F. Yudi Limpraptono, MT)**  
NIP.Y. 1039500274

**Penguji I**

**(Ir. Sidik Noertjahjono, MT)**  
NIP.Y. 10287700163

**Penguji II**

**(Ir. Teguh Herbasuki, MT)**  
NIP. Y. 1038900209

## ABSTRAKSI

### PERANCANGAN DAN PEMBUATAN MODEL SISTEM KEAMANAN PINTU GERBANG PADA PABRIK BERBASIS MIKROKONTROLER AT89S51 DIKOMUNIKASIKAN KE PC MELALUI USB

Saiful Alim

03.17.032

Jurusan Teknik Elektro S1 – Institut Teknologi Nasional Malang  
Jln. Raya Karanglo Km 2 Malang  
m4zt3r\_20@yahoo.co.id

Dosen Pembimbing : I. Joseph Dedy Irawan,ST,MT  
Ir. Eko Nurcahyo

**Kata Kunci** : : Mikrokontroler AT89S51, USB, Sistem Keamanan.

Sebuah pabrik yang memiliki aktivitas produksi dalam jumlah besar, pasti memiliki sejumlah kendaraan yang cukup banyak untuk melaksanakan proses distribusi, setiap hari, kendaraan-kendaraan tersebut keluar dan masuk pabrik dengan jumlah, tujuan dan waktu yang berbeda-beda. Untuk mengawasi kendaraan tersebut, tentu kurang baik jika hanya ditangani oleh satu atau dua orang penjaga gerbang saja. Suatu teknik komunikasi data serial dapat dilakukan antara mikrokontroler AT89S51 dengan computer melalui USB. Alat ini menjadi salah satu solusi untuk lebih mengoptimalkan peran karyawan dalam hal ini petugas pengawas kendaraan di pintu gerbang.

Pada perancangan ini digunakan *hardware* yang meliputi IC mikrokontroler AT89S51, LCD, keypad, motor DC, modul FT232BM, dan detektor yang terdiri dari *infrared*, *photodiode*, *limit switch*. Dan *software* yang terdiri dari *software* pada mikrokontroler dengan menggunakan bahasa C dan software pada PC dengan menggunakan program Borlan Delphi.

Prinsip kerja dari sistem ini adalah tombol *keypad* ditekan, menghasilkan data *input* ke mikrokontroler. Data *input* diolah dan ditampilkan pada LCD. Kemudian data dikirim secara serial melalui USB ke PC menggunakan modul FT232BM. Data *input* dibandingkan dengan data pada database. Jika data *input* sesuai dengan data pada database, maka pintu akan terbuka. Setelah kendaraan melewati detektor, detektor *receiver* akan memberikan sinyal ke rangkaian kendali dan pintu akan menutup.

Dalam pengujian detektor, photodiode hanya mampu menerima sinyal dari *infrared* maksimal sejauh 40 cm. Error tegangan keluaran photodiode saat tidak terhalang adalah 18,4% dan error tegangan keluaran photodiode saat terhalang adalah 5,7%.

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran-Mu Ya Allah yang telah memberikan rahmat dan hidayah-Nya, sehingga saya dapat menyelesaikan skripsi yang berjudul **“Perancangan Dan Pembuatan Model Sistem Keamanan Pintu Gerbang Pada Pabrik Berbasis Mikrokontroler AT89S51 Dikomunikasikan Ke PC Melalui USB”** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Mochtar Asroni, MSME selaku selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I.
5. Bapak Ir. Eko Nurcahyo selaku Dosen Pembimbing II.
6. Kedua orang tuaku dan kakak-kakakku yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
7. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu

Penulis mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2008

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PERSETUJUAN</b> .....	ii
<b>BERITA ACARA UJIAN SKRIPSI</b> .....	iii
<b>ABSTRAKSI</b> .....	iv
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiv
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah .....	2
1.4. Tujuan .....	3
1.5. Metodologi Penulisan .....	3
1.6. Sistematika Pembahasan .....	4
<b>BAB II TEORI DASAR</b> .....	5
2.1. Mikrokontroler AT89S51 .....	5
2.1.1. Pendahuluan .....	5
2.1.2. Perangkat Keras Mikrokontroler AT89S51 .....	6
2.1.3. Konfigurasi Pin Mikrokontroler AT89S51 .....	7
2.1.4. Organisasi Memori .....	12
2.1.5. Mode Pengalamatan .....	18



2.2. Modul FT232BM .....	20
2.2.1. Kabel USB .....	22
2.2.2. Sinyal USB.....	23
2.3. LED ( <i>Light Emitting Diode</i> ) Infra Merah .....	24
2.4. Photodiode .....	25
2.5. Motor DC .....	26
2.5.1.Prinsip Kerja Motor DC .....	26
2.5.2. Pengendali Arah Putaran Motor DC .....	28
2.6. Relay .....	28
2.7. Keypad .....	30
2.8. LCD M1632 .....	31
2.9. Limit Switch .....	32
2.10. Schmitt Trigger .....	33
2.11. IC 74LS164 .....	35
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>38</b>
3.1. Pendahuluan .....	38
3.2. Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	40
3.2.1. Mikrokontroler AT89S51.....	40
3.2.1.1.Rangkaian Clock.....	41
3.2.2.2. Rangkaian Reset.....	42
3.2.4.3. Konfigurasi Kaki Mikrokontroler AT89S51 .	44
3.2.2. Rangkaian LCD M1632 .....	45
3.2.3. Rangkaian Driver Motor DC.....	47
3.2.4. Rangkaian Keypad .....	49

3.2.5. Rangkaian Detektor.....	49
3.2.6. Perancangan USB .....	51
3.3. Perancangan Perangkat Lunak ( <i>software</i> ) .....	52
3.3.1. Diagram Alir Software Pada Mikrokontroler .....	53
3.3.2. Diagram Alir Program Komputer .....	55
<b>BAB IV PENGUJIAN ALAT .....</b>	<b>56</b>
4.1. Tujuan .....	56
4.2. Pengujian Perangkat Keras ( <i>Hardware</i> ) .....	56
4.2.1. Pengujian Detektor.....	56
4.2.1.1. Tujuan .....	56
4.2.1.2. Alat Yang Digunakan .....	56
4.2.1.3. Prosedur Pengujian .....	57
4.2.1.4. Hasil Pengujian .....	57
4.2.3. Pengujian LCD.....	60
4.2.3.1. Tujuan .....	60
4.2.3.2. Alat Yang Digunakan .....	60
4.2.3.3. Prosedur Pengujian .....	60
4.2.3.4. Hasil Pengujian .....	61
4.2.4. Pengujian Modul FT232BM .....	62
4.2.4.1. Tujuan .....	62
4.2.4.2. Alat Yang Digunakan .....	63
4.2.4.3. Prosedur Pengujian .....	63
4.2.4.4. Hasil Pengujian .....	68
4.2.5. Pengujian Keseluruhan Alat.....	69

4.2.5.1. Tujuan .....	69
4.2.5.2. Prosedur Pengujian .....	69
4.2.5.3. Hasil Pengujian .....	69
<b>BAB V PENUTUP .....</b>	<b>72</b>
5.1. Kesimpulan .....	72
5.2. Saran .....	72
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN-LAMPIRAN</b>	

## DAFTAR GAMBAR

2.1 Diagram Blok Mikrokontroler AT89S51 .....	7
2.2 Konfigurasi Pin-pin AT89S51 .....	7
2.3 Osilator Eksternal.....	10
2.4 Modul FT232BM .....	21
2.5 Konektor USB .....	22
2.6 Konektor Pin USB Tipe A .....	23
2.7a Perkabelan USB Kecepatan Rendah .....	24
2.7b Perkabelan USB Kecepatan Penuh .....	24
2.8 Struktur LED Infrared .....	25
2.9 Simbol Photodiode.....	26
2.10 Kaidah Tangan Kanan .....	27
2.11 Cara Kerja Motor DC.....	27
2.12 Pengendali Arah Putaran Motor DC .....	28
2.13 Cara Kerja Relay .....	29
2.14 Relay SPST .....	29
2.15 Relay SPDT .....	29
2.16 Relay DPST .....	30
2.17 Relay DPDT .....	30
2.18 Keypad Matrik 4x4 .....	31
2.19 Schmitt Trigger untuk pembentukan gelombang .....	33
2.20 Karakteristik schmitt trigger .....	34
2.21 Karakteristik Inverter Schmitt Trigger .....	34

2.22	Inverter Schmitt Trigger IC 74LS14 .....	35
2.23	Diagram Detail Register Geser 8 bit 74LS164 .....	35
2.24	Konfigurasi Kaki-kaki 74LS164 .....	36
3.1	Blok Diagram Sistem .....	38
3.2	Rangkaian Clock .....	41
3.3	Rangkaian Reset.....	43
3.4	Konfigurasi Kaki-kaki AT89S51 .....	44
3.5	Rangkaian LCD .....	45
3.6	Rangkaian Driver Motor.....	47
3.7	Blok Diagram Hubungan Keypad dengan Mikrokontroler .....	49
3.8	Rangkaian Detektor.....	50
3.9	Pemakaian Pin Pada Port Modul FT232BM .....	51
4.1	Tegangan Keluaran Photodiode Saat Tidak Terhalang .....	57
4.2	Tegangan Keluaran Photodiode Saat Terhalang.....	57
4.3	Tegangan Keluaran Schmitt Trigger Saat Detektor Tidak Terhalang..	59
4.4	Tegangan Keluaran Schmitt Trigger Saat Detektor Terhalang.....	59
4.5	Rangkaian Pengujian LCD .....	61
4.6	Hasil Pengujian LCD .....	61
4.7	Pengukuran Tegangan Awal LCD .....	62
4.8	Pengukuran Tegangan Saat melewati Diode .....	62
4.9	Blok Pengujian Modul FT232BM .....	63
4.10	PC Mendeteksi <i>Hardware</i> Baru .....	64
4.11	Perintah Untuk Memilih Cara Penginstalan .....	64
4.12	Perintah Memberitahu Lokasi Driver Modul USB .....	65

4.13 Lokasi Driver .....	65
4.14 Proses Transfer Driver USB.....	66
4.15 Proses Penginstalan Selesai.....	66
4.16 Tampilan Delphi Saat Setting Komponen Com Port .....	68
4.17 Tampilan Saat Program Dijalankan .....	68
4.18 Nyala LED Indikator Pada Modul FT232BM .....	69
4.19 Tampilan PC Saat ID Dan Password Di Terima.....	70
4.20 Model Pintu Gerbang .....	70

## DAFTAR TABEL

2.1 Fungsi-fungsi Alternatif Port 1 .....	8
2.2 Fungsi Khusus Pada Port 3 .....	9
2.3 Special Function Register .....	13
2.4 Fungsi Kaki-kaki M1632.....	32
4.1 Hasil Pengukuran Tegangan Keluaran Photodiada.....	58
4.2 Hasil Perhitungan Tegangan Keluaran Photodiada .....	58
4.3 Hasil Pengujian Detektor Photodiada .....	63
4.4 Hasil Pengujian Rangkaian LCD .....	62
4.5 Spesifikasi Model Pintu Gerbang.....	71

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Sebuah pabrik yang memiliki aktivitas produksi dalam jumlah besar, pasti memiliki sejumlah kendaraan yang cukup banyak untuk melaksanakan proses distribusi. Setiap hari, kendaraan-kendaraan tersebut keluar dan masuk pabrik dengan jumlah, tujuan dan waktu yang berbeda-beda. Untuk mengawasi sejumlah kendaraan tersebut, tentu kurang baik jika hanya ditangani oleh satu atau dua orang penjaga gerbang saja.

Beberapa kekurangan yang mungkin dapat terjadi misalnya, penjaga harus memeriksa setiap sopir yang membawa kendaraan masuk atau keluar pabrik. Selain itu juga penjaga harus membuka dan menutup pintu gerbang setiap kali ada kendaraan yang hendak masuk maupun keluar pabrik. Kemudian, kemungkinan kekurangan yang lain yaitu keamanan kurang terjamin.

Untuk menanggulangi hal tersebut, pihak perusahaan harus menambah pengeluaran untuk membayar sejumlah petugas penjaga gerbang. Selain itu, faktor keamanan yang kemungkinan dapat terganggu akibat keluar dan masuknya kendaraan kurang tertib yang bisa menimbulkan kerugian yang cukup berarti bagi sebuah perusahaan. Dengan demikian, sangat berarti sekali jika proses pengawasan setiap kendaraan yang keluar dan masuk pabrik dibantu dengan penerapan teknologi, agar efisiensi dan proses pengawasannya lebih terstruktur dan lebih baik.



Di lain hal, suatu teknik komunikasi data serial dapat dilakukan antara mikrokontroler AT89S51 dengan komputer melalui USB. Berdasarkan permasalahan tersebut, penulis mencoba untuk membuat suatu prototipe sistem keamanan pintu gerbang pada pabrik berbasis mikrokontroler AT89S51 dikomunikasikan ke PC melalui USB. Alat ini menjadi salah satu solusi untuk lebih mengoptimalkan peran karyawan dalam hal ini petugas pengawas kendaraan di pintu gerbang, dikarenakan sebagian tugas lainnya digantikan oleh sistem terprogram.

## **1.2. Rumusan Masalah**

Dalam perancangan dan pembuatan prototipe sistem keamanan pintu gerbang pada pabrik berbasis mikrokontroler dikomunikasikan ke PC melalui USB dapat dirumuskan beberapa masalah yang akan dibahas yaitu:

1. Bagaimana merancang dan membuat rangkaian yang menunjang dalam metode ini.
2. Bagaimana merancang dan membuat perangkat lunak atau *software* pada mikrokontroler yang mengendalikan semua kerja sistem.

## **1.3 Batasan Masalah**

Agar pembahasan dari perancangan dan pembuatan prototipe sistem keamanan pintu gerbang pada pabrik berbasis mikrokontroler dikomunikasikan ke PC melalui USB ini tidak terlalu meluas maka penyusun perlu membuat batasan-batasan masalah yang meliputi :

1. Tidak membahas jenis pintu gerbang yang digunakan.

2. Alat ini di buat hanya untuk pintu masuk pabrik saja karena untuk pintu keluar pabrik mekanismenya juga sama.
3. Tidak membahas secara detail tentang teori *database*.
4. Kabel usb yang digunakan pada model pintu gerbang ini hanya 1 meter.
5. Model Pintu grbang ini hanya digunakan untuk pintu masuk kendaraan saja

#### **1.4. Tujuan**

Tujuan dari skripsi ini adalah untuk membuat sebuah model pintu gerbang yang menggunakan sistem keamanan dengan memakai ID dan password ketika akan memasuki pabrik. Model pintu gerbang ini menggunakan IC Mikrokontroler AT89S51 sebagai pengendali utamanya.

#### **1.5. Metodologi Penulisan**

Metode yang digunakan dalam penulisan skripsi ini adalah

##### **1. Studi Pustaka**

Memperoleh data dengan cara membaca dan mempelajari buku literature yang berhubungan dengan penyusunan skripsi ini.

##### **2. Studi Lapangan**

Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.

Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

## **1.6 Sistematika Pembahasan**

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

### **BAB I   Pendahuluan**

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, sistematika pembahasan dari alat yang direncanakan.

### **BAB II   Landasan Teori**

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perencanaan dan pembuatan alat ini yang meliputi rangkaian dengan metode pengiriman data secara serial dan Mikrokontroler AT89S51.

### **BAB III   Perencanaan Dan Pembuatan Alat**

Pada bab ini dibahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*).

### **BAB IV   Pengujian Alat**

Pada bab ini dibahas tentang proses serta hasil dari pengujian alat, yang didasarkan oleh pengukuran-pengukuran.

### **BAB V   Penutup**

Pada bab ini akan disampaikan kesimpulan dari perencanaan dan pembuatan sistem ini.

## BAB II

### LANDASAN TEORI

#### 2.1. Mikrokontroller AT89S51

##### 2.1.1. Pendahuluan

Mikrokontroller bisa dipandang sebagai sebuah mini komputer yang terintegrasi dalam sebuah chip. Didalam satu chip mikrokontroller 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 mikroprocessor, mikrokontroller adalah sebuah perangkat serbaguna, yang fungsi kerjanya dapat ditentukan melalui sebuah perangkat lunak yang mendeskripsikan sebuah sistem yang diinginkan.

Mikrokontroller AT89S51 adalah mikrokontroler ATMEL yang kompatibel penuh dengan mikrokontroler keluarga MCS – 51, membutuhkan daya rendah, memiliki performance yang tinggi dan merupakan *microcomputer* 8 bit yang dilengkapi 4Kbyte EEPROM (*Electrical Erasable and Programmable Read Only Memory*) dan 128 Byte RAM *internal*. Program memori yang dapat diprogram ulang dalam sistem atau menggunakan programmer *Nonvolatile* memori konvensional. Dalam sistem mikrokontroler terdapat dua hal yang mendasar, yaitu: perangkat lunak dan perangkat keras yang keduanya saling terkait dan mendukung.

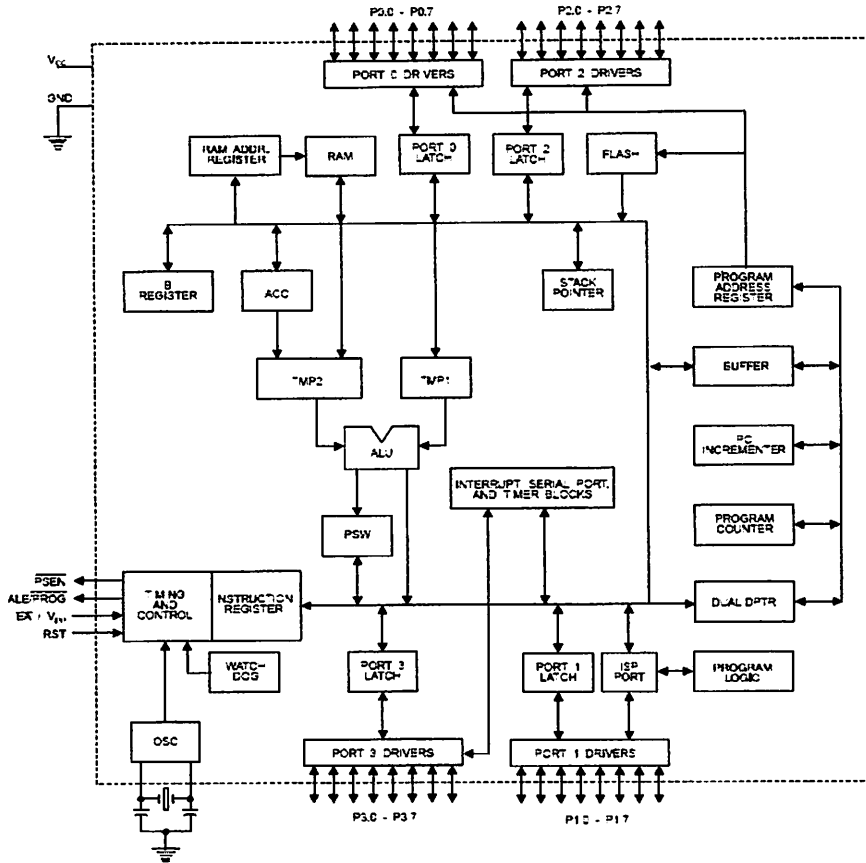
### 2.1.2. Perangkat keras mikrokontroler AT89S51

Secara umum Mikrokontroler AT89S51 memiliki :

- ❖ CPU 8 bit termasuk keluarga MCS-51
- ❖ 4 Kb *Flash* memory
- ❖ 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*.
- ❖ 32 buah Port I/O, tersusun atas P0 – P3, masing – masing 8 bit.
- ❖ 2 *Timer/ counter* 16 bit
- ❖ 2 *Serial Port Full Duplex*
- ❖ Kecepatan pelaksanaan intruksi per siklus 1 us pada frekuensi clock 12 Mhz
- ❖ 2 DPTR (*Data Pointer*)
- ❖ *Watchdog Timer*
- ❖ Fleksibel ISP Programming

Dengan keistimewaan Di atas pembuatan alat menggunakan AT89S51 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak.

Adapun blok diagram dari Mikrokontroler AT89S51 adalah sebagai berikut :

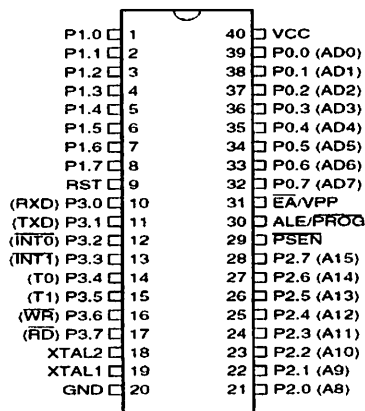


**Gambar 2.1** Diagram Blok Mikrokontroler AT89S51

(Sumber : Data Sheet Atmel AT89S51)

**2.1.3. Konfigurasi Pin Mikrokontroler AT89S51**

Mikrokontroler AT89S51 terdiri dari 40 pin dengan konfigurasi sebagai berikut :



**Gambar 2.2** Konfigurasi Pin-Pin AT89S51

(Sumber : Data Sheet Atmel AT89S51)

Fungsi tiap pin-nya adalah sebagai berikut :

### 1. Pin 1 sampai 8, Port 1

Merupakan 8 bit I/O Bi-directional yang dilengkapi dengan internal Pull - Up. Ketika diberikan logika '1' pin ini akan di *Pull-Up* secara *internal* sehingga dapat digunakan sebagai *input*. Sebagai masukan jika pin – pin ini dihubungkan ke ground maka masing – masing pin ini dapat menghantarkan arus karena di *Pull-High* secara internal. Port 1 juga menerima *Low Order Address Bytes* selama melakukan verifikasi program.

Pada *port 1* di AT89S51 pin ini mempunyai alternatif seperti pada tabel berikut ini:

**Tabel 2.1** Fungsi – fungsi Alternative Port 1

Port Pin	Alternative Functions
P1.5	MOSI (Master Output Slave Input)
P1.6	MISO ((Master Input Slave Output)
P1.7	SCK (Serial Clock)

(Sumber : Data Sheet Atmel AT89S51)

### 2. Pin 9, RST (*Reset*)

Merupakan pin yang aktif tinggi (*high*), pin ini aktif tinggi selama dua siklus mesin yang akan membuat mikrokontroler AT 89S51 menjalankan rutin *reset*.

### 3. Pin 10 sampai 17, Port 3

Port 3 sebagai 8 bit I/O Bi-directional yang dilengkapi dengan *Pull-Up Internal*. Penyangga keluaran port 3 dapat memberikan atau menyerap

arus empat masukan TTL (sekitar 1,6 mA). Jika diberikan logika '1' pada pin - pin port 3, maka masing – masing pin akan di *Pull High* oleh *Pull-Up internal* sehingga dapat digunakan sebagai *input-an*. Sebagai inputan, jika pin – pin port 3 dihubungkan ke *ground*, maka masing – masing kaki akan memberikan arus karena di *Pull High* secara internal, dimana Port 3 juga mempunyai fungsi-fungsi khusus yang dimiliki oleh keluarga MCS-51. Fungsi tersebut dapat dilihat dalam berikut ini :

**Tabel 2.2 Fungsi Khusus Pada Port 3**

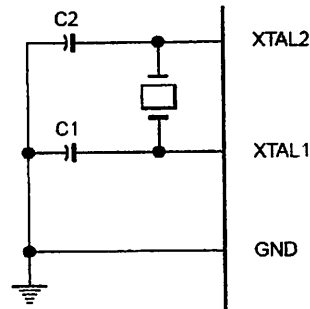
Nama Penyemat	Fungsi Khusus
<i>Port 3.0</i>	RxD (port masukan serial)
<i>Port 3.1</i>	TxD (port keluaran serial)
<i>Port 3.2</i>	/INT0 (masukan interupsi eksternal 0)
<i>Port 3.3</i>	/INT1 (masukan interupsi eksternal 1)
<i>Port 3.4</i>	T0 (masukan pewaktu eksternal 0)
<i>Port 3.5</i>	T1 (masukan pewaktu eksternal 1)
<i>Port 3.6</i>	/WR (sinyal tulis memori data eksternal)
<i>Port 3.7</i>	/RD (sinyal baca memori data eksternal)

(Sumber : Data Sheet Atmel AT89S51)

#### 4. Pin 18 sampai 19, *X-TAL 1* dan *X-TAL 2*

X-TAL 1 merupakan masukan ke rangkaian osilator *internal* sedangkan X-TAL 2 keluaran dari rangkaian osilator *internal*. Untuk keperluan ini diperlukan kapasitor penstabil sebesar 30pF. Dan nilai dari X-TAL tersebut antara 3 – 33 Mhz. Untuk lebih jelasnya dapat dilihat gambar pemasangan X-TAL serta kapasitor yang digunakannya.





**Gambar 2-3** Osilator Eksternal AT89S51

(Sumber : Data Sheet Atmel AT89S51)

**5. Pin 20, GND (*Ground*)**

Dihubungkan dengan *ground* rangkaian.

**6. Pin 21 sampai 28, Port 2**

Port 2 berfungsi sebagai 8 bit I/O Bi-directional yang dilengkapi dengan internal *Pull-Up* Penyangga keluaran port 2 dapat memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA) Jika diberikan logika '1' pada pin – pin port 2, maka masing – masing pin akan di *Pull High* secara *internal* sehingga dapat digunakan sebagai *input-an*. Sebagai *input-an* jika pin – pin port 2 dihubungkan ke *ground* (di *Pull-Low*), maka, masing – masing pin dapat menghantarkan arus karena di *Pull High* secara *internal*. Port 2 mengeluarkan alamat bagian tinggi (A8-A15), selama pengambilan instruksi dari memori program eksternal dan selama pengaksesan memori data eksternal yang menggunakan perintah dengan alamat 16-bit (dengan perintah “MOVX @DPTR”).

**7. Pin 29,  $\overline{\text{PSEN}}$  (*Program Store Enable*)**

Pin ini aktif rendah yang merupakan *strobe* pembacaan ke program memori eksternal.

**8. Pin 30, ALE (*Address Latch Enable*) /  $\overline{PROG}$**

Keluaran ALE menghasilkan pulsa – pulsa untuk menahan alamat rendah (A0-A7) pada port 0, selama dilakukan proses baca atau tulis memori *external*. Pin ini juga berfungsi sebagai masukan pulsa program ( $\overline{PROG}$ ) selama pemrograman EEPROM *external*. Pada operasi normal, ALE akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan atau pendetakan (*clocking*).

**9. Pin 31,  $\overline{EA}$  / VPP (*External Access*)**

Dapat diberikan logika rendah (*ground*) atau logika tinggi (+5V). Jika diberikan logika tinggi maka mikrokontroller akan mengakses program dari ROM *internal* (EEPROM/*Flash Memori*), dan jika diberikan logika rendah maka mikrokontroller akan mengakses program dari memori *external* yang berlokasi 0000h sampai FFFFh.

**10. Pin 32 sampai 39, Port 0**

*Port 0* terdiri dari 8 saluran *input* atau *output* dua arah, tanpa internal *pull-up*. Port 0 merupakan bus alamat rendah (A<sub>0</sub> – A<sub>7</sub>), yang dimultipleks dengan saluran bus data (D0-D7), yang digunakan pada saat mengakses memori data *external* dan memori program *external*.

**11. Pin 40, VCC**

Merupakan masukan catu daya 5 volt dengan toleransi kurang lebih 10%.

#### 2.1.4. Organisasi Memori

Organisasi yang dimiliki oleh AT89S51 yang terdiri atas :

##### 1. RAM Internal

Memori sebesar 128 *byte* yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara. RAM *internal* terdiri atas :

###### ➤ **Register Banks**

AT89S51 mempunyai delapan buah register yang terdiri atas R0 hingga R7. Kedelapan register ini selalu terletak pada alamat 00H hingga 07H pada setiap kali sistem direset. Namun posisi R0 hingga R7 dapat dipindah ke bank 1 (08 H hingga 0FH), bank 2 (10H hingga 17H) dan bank 3 (18H hingga 1FH), dengan mengatur bit RS0 dan RS1.

###### ➤ **Bit Addressable RAM**

RAM pada alamat 20H hingga 2FH dapat diakses secara pengalamatan *bit* (*bit addressable*) sehingga hanya dengan sebuah instruksi saja setiap *bit* dalam area ini dapat *diset, clear, AND, OR*.

###### ➤ **RAM keperluan umum**

RAM keperluan umum dimulai dari alamat 30H hingga 7FH dan dapat diakses dengan pengalamatan langsung maupun tak langsung. Pengalamatan langsung dilakukan ketika salah satu *operand* merupakan bilangan yang menunjukkan lokasi yang dialamati.

##### 2. Special Function Register

Register fungsi khusus (*Special Function Register*) terletak pada 128 *byte* bagian atas memori data internal dan berisi *register-register*

untuk pelayanan *latch port, timer, program status words, control peripheral* dan sebagainya. Alamat register fungsi khusus ditunjukkan pada Tabel 2-3.

*Register-register* ini hanya dapat diakses dengan pengalamatan langsung. Enam belas alamat pada register fungsi khusus dapat dialamati *perbit* maupun *per-byte* dan terletak pada alamat  $80_H$ - $FF_H$ . Secara perangkat keras, register fungsi khusus ini dibedakan dengan memori data internal.

**Tabel 2.3** Special Function Register

Simbol	Nama Register	Alamat
ACC	Accumulator	$E0_H$
B	Register B	$F0_H$
PSW	Program Status Word	$D0_H$
SP	Stack Pointer	$81_H$
DPTR	Data Pointer 2 Byte	
DPL	Bit rendah	$82_H$
DPH	Bit Tinggi	$83_H$
P0	Port 0	$80_H$
P1	Port 1	$90_H$
P2	Port 2	$A0_H$
P3	Port 3	$B0_H$
IP	Interrupt Priority Control	$D8_H$
IE	Interrupt Enable Control	$A8_H$
TMOD	Timer/Counter Mode Control	$89_H$
TCON	Timer/Counter Control	$88_H$
TH0	Timer/Counter 0 High Control	$8C_H$
TL0	Timer/Counter 0 Low Control	$8A_H$
TH1	Timer/Counter 1 High Control	$8D_H$
TL1	Timer/Counter 1 Low Control	$8B_H$
SCON	Serial Control	$98_H$
SBUF	Serial Data Buffer	$99_H$
PCON	Power Control	$87_H$

(Sumber : Hafindo Elektronik & Education)

Beberapa macam register fungsi khusus yang sering digunakan adalah sebagai berikut ini :

- *Accumulator* (ACC) merupakan register untuk penambahan dan pengurangan. Perintah *mnemonic* untuk mengakses akumulator disederhanakan sebagai A.
- Register B merupakan register khusus yang berfungsi melayani operasi perkalian dan pembagian.
- *Program Status Word* (PSW) yang terletak pada alamat D0H terdiri dari beberapa *bit* status yang menggambarkan kejadian di akumulator sebelumnya. Yaitu *carry bit*, *auxiliary carry*, dua *bit* pemilih bank, bendera *overflow*, *parity bit*, dan dua bendera yang dapat didefinisikan sendiri oleh pemakai. Keterangannya sebagai berikut :

- Flag Carry

Flag Carry (terletak pada alamat D7H) mempunyai fungsi sebagai pendeteksi terjadinya kelebihan pada operasi penjumlahan atau terjadi pinjam (*borrow*) pada operasi pengurangan. Misalnya jika data pada accumulator adalah FFH dan dijumlahkan dengan bilangan satu atau lebih, akan terjadi kelebihan dan membuat carry menjadi Set, sedangkan jika data pada accumulator adalah 00H dan dikurangkan dengan bilangan satu atau lebih, akan terjadi peminjaman dan membuat carry juga menjadi set.

- Flag Auxiliary Carry

Flag Auxiliary Carry akan selalu Set pada saat proses penjumlahan terjadi carry dari bit ketiga hingga bit keempat.

- Flag 0

Flag 0 digunakan untuk tujuan umum bergantung pada kebutuhan pemakai.

- Bit Pemilih Register Bank

Register Bank Select Bits (RS0 dan RS1) atau Bit Pemilih Register Bank digunakan untuk menentukan lokasi dari Register Bank (R0 hingga R7) pada memori. RS0 dan RS1 selalu bernilai nol setiap kali system direset sehingga lokasi dari R0 hingga R7 akan berada di alamat 00H hingga 07H.

- Flag Overflow

Flag Overflow akan diset jika pada operasi aritmatik menghasilkan bilangan yang lebih besar dari pada 128 atau lebih kecil dari - 128.

- Bit Pariti

Bit Pariti akan diset jika jumlah bit 1 dalam accumulator adalah ganjil dan akan clear jika jumlah bit 1 dalam accumulator genap. Jika data dalam accumulator adalah 10101110b atau AEH pariti akan diset. Data AEH mempunyai lima bit yang berkondisi 1 atau dapat disebut mempunyai bit 1 dalam jumlah yang ganjil. Bit pariti ini digunakan untuk proses yang berhubungan dengan serial port yaitu sebagai *Check sum*.

- *Stack Pointer (SP)* merupakan register 8 bit yang dapat diletakkan di alamat manapun pada RAM *internal*. Isi *register* ini ditambah sebelum data disimpan, selama instruksi PUSH dan CALL. Pada saat *reset*, *register* SP diinisialisasi pada alamat 07<sub>H</sub>, sehingga *stack* akan dimulai pada lokasi 08<sub>H</sub>.

- *Data Pointer (DPTR)* terdiri dari dua register, yaitu untuk *byte* tinggi (*Data Pointer High, DPH*) dan *byte* rendah (*Data Pointer Low, DPL*) yang berfungsi untuk pengalamatan alamat 16 bit.
- Port 0 sampai Port 3 merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada port 0, 1, 2, 3. Masing-masing register ini dapat dialamati *per-byte* maupun *per-bit*.
- *Serial data buffer (SBUF)* merupakan dua *register* yang terpisah, *register buffer* pengirim dan sebuah *register buffer* penerima. Meletakkan data pada SBUF berarti meletakkan pada *buffer* pengirim yang akan mengirimkan data melalui transmisi serial. Membaca data SBUF berarti menerima data dari *buffer* penerima.
- *Control Register* terdiri dari register yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu *register IP (Interrupt Priority)* dan *register IE (Interrupt Enable)*. Untuk mengontrol pelayanan *timer/counter* terdapat *register* khusus, yaitu register TCON (*timer/counter control*) serta pelayanan port serial menggunakan register SCON (*Serial Port Control*).
- **Register Timer**  
AT89S51 mempunyai dua buah 16 bit Timer/Counter, yaitu Timer 0 dan Timer 1. Timer 0 terletak pada alamat 8AH untuk TL0 dan 8CH untuk TH0 dan Timer 1 terletak pada alamat 8BH untuk TL1 dan 8DH untuk TH1.

### ➤ Register Interupt

AT89S51 mempunyai lima buah interupsi dengan dua level prioritas interupsi. Interupsi akan selalu nonaktif setiap kali sistem direset. Register – register yang berhubungan dengan interrupt adalah *Interrupt Enable Register (IE)* atau Register Pengaktif Interupsi pada alamat A8H untuk mengatur keaktifan tiap – tiap interrupt dan *Interrupt Priority Register (IP)* atau Register Prioritas Interupsi pada alamat B8H.

### ➤ Register Port Serial

AT89S51 mempunyai sebuah *on chip serial port* (serial port dalam keping) yang dapat digunakan untuk berkomunikasi dengan peralatan lain yang menggunakan serial port juga seperti modem, shift register dan lain – lain.

Buffer (Penyangga) untuk proses pengiriman maupun pengambilan data terletak pada register SBUF, yaitu pada alamat 99H. Sedangkan untuk mengatur mode serial dapat dilakukan dengan mengubah isi dari SCON yang terletak pada alamat 98H.

## 3. Flash PEROM

AT89S51 memiliki 4Kb *Flash PEROM (Programmable and Erassable Read Only Memori)*, yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah perangkat programmer hingga 1000 kali. Program yang ada pada *Flash PEROM* akan dijalankan jika pada saat sistem di-*reset*, pin EA/VP berlogika satu sehingga mikrokontroler aktif berdasarkan program yang ada pada *flash PEROM*nya. Namun, jika



EA/VP berlogika nol, mikrokontroller aktif berdasarkan program yang berada pada memori *external*.

#### 2.1.5. Mode Pengalamatan.

Mode pengalamatan yang digunakan pada AT89S51 adalah sebagai berikut:

- a) Mode pengalamatan segera (*immediate addressing mode*).

Cara ini menggunakan konstanta, misalnya: **MOV A, #20H**. Data konstanta merupakan data yang menyatu dengan instruksi, contoh instruksi tersebut diatas mempunyai arti bahwa data konstantanya yaitu 20H, (sebagai data konstanta harus diawali dengan '#') disalin ke akumulator A.

- b) Mode pengalamatan langsung (*direct addressing mode*).

Cara ini dipakai untuk menunjuk data yang berada di suatu lokasi memori dengan cara menyebut lokasi (alamat) memori tempat data tersebut berada, misalnya: **MOV A, 30H**. Instruksi ini mempunyai arti bahwa data yang berada di dalam memori dengan lokasi 30h disalin ke akumulator. Bedanya dengan pengalamatan segera yaitu jika pada pengalamatan segera menggunakan tanda '#' yang menandai 20H sebagai data konstan, sedangkan pada instruksi ini tidak menggunakan '#' sehingga 30H diartikan sebagai suatu lokasi memori.

- c) Mode pengalamatan tidak langsung (*indirect addressing mode*).

Cara ini dipakai untuk mengakses data yang berada di dalam memori, tetapi lokasi memori tidak disebut secara langsung tapi di-'titip'-kan

ke register lain, misalnya: **MOV A, @R0**. R0 adalah register serba guna yang dipakai untuk menyimpan lokasi memori, sehingga instruksi ini mempunyai arti memori yang alamat lokasinya tersimpan dalam R0 isinya disalin ke akumulator A. Tanda '@' dipakai untuk menandai lokasi memori yang tersimpan di dalam R0. *Register* serba guna R0 berfungsi sebagai register penyimpanan alamat (*indirect address*), selain R0 register serba guna lainnya, R1 juga bisa dipakai sebagai register penampung alamat.

- d) Mode pengalamatan register (*register addressing mode*).

Misalnya: **MOV A, R5**, instruksi ini mempunyai arti bahwa data dalam register serba guna R5 disalin ke akumulator A. Instruksi ini menjadikan register serba guna R0 sampai R7 sebagai tempat penyimpanan data yang praktis dan kerjanya sangat cepat.

- e) Mode pengalamatan kode tidak langsung (*code indirect addressing mode*).

MCS51 mempunyai cara penyebutan data dalam memori program yang dilakukan secara tak langsung, misalnya: **MOVC A, @A+DPTR**. Instruksi MOV diganti dengan MOVC, tambahan huruf C tersebut dimaksud untuk membedakan bahwa instruksi ini digunakan untuk memori program (MOV tanpa huruf C artinya digunakan untuk memori data). Tanda '@' digunakan untuk menandai A+DPTR yang berfungsi untuk menyatakan lokasi memori yang isinya disalin ke Akumulator A, dalam hal ini nilai yang tersimpan dalam DPTR (*Data Pointer Register* – 2 byte) ditambah dengan nilai yang

tersimpan dalam akumulator A (1 byte) sama dengan lokasi memori program yang diakses.

## 2.2. Modul FT232BM

Perkembangan teknologi komputer dimana seiring dengan perkembangan ilmu pengetahuan yang semakin cepat pada PC atau laptop untuk keluaran terbaru jumlah port serial RS232 semakin lama semakin berkurang, jika pada PC lama biasanya terdapat dua buah konektor RS232 maka sekarang hanya terdapat satu buah konektor saja sehingga keberadaan port serial RS232 sekarang telah digantikan oleh port USB yang mempunyai banyak kelebihan dibandingkan port serial RS232. Adapun contoh pengiriman informasi secara serial melalui sebuah mikrokontroler yang dikirimkan ke PC melalui port USB seperti Modul FT232BM yaitu modul interface yang digunakan untuk aplikasi dari mikrokontroler ke USB, hubungan ini dilakukan secara serial dengan kata lain sebuah modul yang dapat mengkonversikan data serial yang berasal dari mikrokontroler. Secara garis besar modul ini berfungsi untuk mengubah data USB yang berasal dari port USB menjadi data serial dengan level tegangan TTL sehingga pengguna dapat melakukan komunikasi data serial melalui port USB. Keunggulan digunakannya modul ini adalah kemampuannya untuk mengirim data lebih cepat dibandingkan dengan komunikasi serial dengan menggunakan Port RS232 serta untuk kecepatan transfer data serial yang dapat di pakai oleh IC ini yaitu sebesar 300bps sampai 9600 bps.

Modul FT232BM mempunyai spesifikasi sebagai berikut:

1. Memiliki tegangan kerja 4,4- 5,25 volt DC.

2. Tersedia 2 LED untuk indicator Tx dan Rx data pada komunikasi serial.
3. Memiliki baudrate 3Mbps (TTL), 1Mbps (RS-232), 3Mbps (RS-422/RS-485).
4. Pin sinyal kontrol (arah) untuk komunikasi RS-485 yang bekerja secara otomatis.
5. Kompatibel dengan USB 1.1 dan USB 2.0.
6. Memiliki output dengan level TTL 5 volt .
7. Memiliki EEPROM eksternal untuk menyimpan data PID, VID, nomor serial dan deskripsi produk pengisian datanya melalui USB.
8. Virtual COM port driver (VCP) dan D2xx (USB *Direct Drivers* + DLL *S/W Interface*) untuk Windows 98, 98SE, ME, 2000 dan XP.
9. Mendukung format UART dengan 7/8 bit data , 1 / 2 stop bit dan *Odd/Even /Mark/Space/No Parity*.

Adapun gambar fisik dari modul FT232BM dapat digambarkan sebagai berikut:

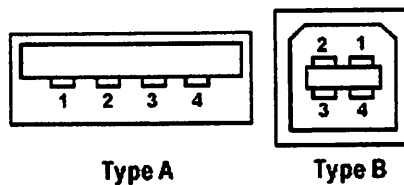


**Gambar 2.4** Modul FT232BM  
(Sumber : Data Sheet FT232BM)

Gambar diatas merupakan modul dari FT232BM secara fisik yang terdiri dari konektor USB TIPE B dan chip FT232 sebagai komponen utama dari modul COM to USB tersebut serta komponen-komponen pendukung lainnya.

### 2.2.1. Kabel USB

Untuk menghubungkan modul FT232BM dengan komputer maka diperlukan kabel konektor USB dimana hanya terdapat 2 macam konektor yang digunakan dalam menghubungkan modul tersebut, yaitu konektor type A dan konektor type B seperti terlihat dalam gambar 2.5. Konektor type A dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada bagian computer sedangkan konektor type B dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada modul FT232BM sedangkan untuk peralatan USB yang sederhana, misalnya mouse, biasanya tidak pakai konektor B melainkan konektor tipe A hal ini dilakukan untuk menghemat biaya sehingga kabel langsung dihubungkan ke bagian dalam mouse.

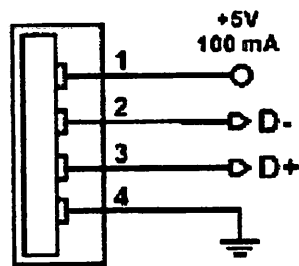


**Gambar 2.5** Konektor USB

(Sumber : I/O Bus & Motherboard)

Kabel USB terdiri dari 4 kabel ditambah konduktor pembungkus kabel seperti pelindung yang biasanya dijumpai dalam kabel audio. Kabel nomor 1 dipakai untuk menyalurkan sumber daya dengan tegangan 5 Volt, jika diperlukan peralatan USB boleh mengambil daya dari saluran ini tidak lebih dari 100 mA. Komputer yang dilengkapi dengan antarmuka USB, wajib menyediakan daya sebesar 500 mA untuk keperluan ini. Peralatan USB yang memerlukan daya lebih dari ketentuan tersebut di atas, harus menyediakan sendiri sumber daya untuk keperluan kerja peralatan tersebut.

Kabel nomor 4 adalah ground sebagai saluran balik sumber tegangan 5 Volt. Kabel nomor 2 dan nomor 3 dipakai untuk pengiriman sinyal. Kabel nomor 2 bernama D- dan kabel nomor 3 bernama D+, tegangan pada dua saluran ini berubah antara 0 Volt dan 3,3 Volt seperti terlihat pada gambar 2.6 berikut:



**Gambar 2.6** Konektor Pin USB Tipe A  
(Sumber : innovative\_electronics)

### 2.2.2. Sinyal USB ( Universal Serial Bus )

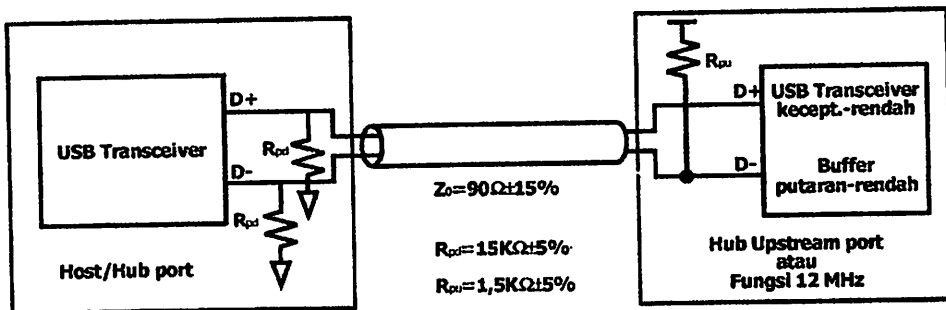
Secara umum USB terdiri dari 4 buah pin yaitu pada pin 1 merupakan sumber tegangan 5 Volt , pin nomor 2 merupakan pin D+ serta pin nomor 3 merupakan pin D- dimana pin nomor 2 dan pin nomor 3 merupakan jalur pengiriman dan penerima data yang berasal dari mikrokontroler sedangkan pin nomor 4 merupakan ground.

Sinyal digital yang dikirim melalui dua saluran tersebut dikatakan sebagai 'difference signal', artinya sinyal digital '0' atau '1' tidak dinyatakan dengan besarnya tegangan pada saluran tersebut terhadap ground, seperti halnya sinyal digital yang dipakai dalam IC TTL (transistor Transistor Logic) atau dalam saluran RS232.

Sinyal digital dinyatakan dengan perbedaan tegangan antara dua kabel tersebut. Jika tegangan pada saluran D+ lebih tinggi dari tegangan pada saluran D-

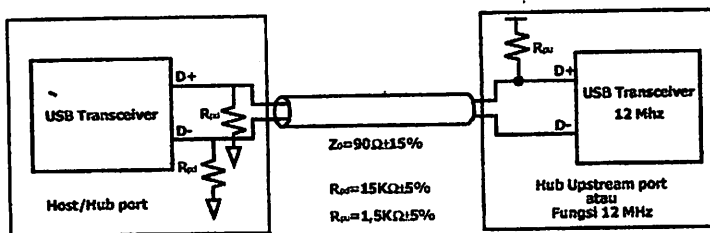
, maka informasi yang dikirimkan adalah sinyal digital '1', sebaliknya sinyal digital '0' dinyatakan dengan tegangan pada D+ lebih kecil dari D-.

Rentang tegangan USB adalah 0,3 volt hingga 3,6 volt (pada beban 1,5 k $\Omega$ ). Logika *high* didapat jika tegangan sudah melebihi 2,8 volt terhadap ground dan logika *low* didapat jika tegangan dibawah 0,3 volt terhadap ground.



**Gambar 2.7a** Perkabelan USB Kecepatan Rendah.

(Sumber : I/O Bus & Motherboard)



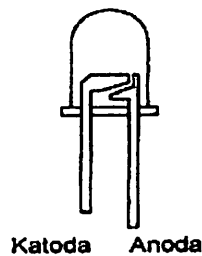
**Gambar 2.7b** Perkabelan USB kecepatan penuh.

(Sumber : I/O Bus & Motherboard)

### 2.3. LED (*Light Emitting Diode*) Infra Merah.

LED infra merah digunakan untuk menghasilkan sinar infra merah. Prinsip kerjanya adalah, pada waktu LED infra merah dibias forward, elektron dari pita konduksi melewati junction dan jatuh kedalam hole pita valensi, sehingga elektron-elektron tersebut memancarkan energi. Pada diode penyearah biasa, energi ini dikeluarkan sebagai energi panas, tetapi pada LED, energi ini dipancarkan sebagai cahaya. Sedangkan pada LED infra merah akan

memancarkan cahaya yang tidak kelihatan, hal ini dapat dibenarkan dalam sistem tanda bahaya pencuri, penyampaian informasi rahasia dan ruang lingkup yang lain yang membutuhkan pancaran yang tidak tampak. Simbol LED infra merah yang sering digunakan ditunjukkan dalam gambar 2.8.



**Gambar 2.8** Struktur Led Infrared

(Sumber : [www.en.wikipedia.org](http://www.en.wikipedia.org))

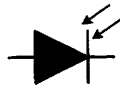
Led infra merah merupakan PN junction yang memancarkan radiasi infra merah yang tidak tampak oleh mata kita. Apabila pada anoda diberi tegangan positif dan katoda ke ground, maka LED menjadi “ON” dan arus akan mengalir dari anoda ke katoda. Pada reaksi semikonduktor suatu dioda akan terjadi perpindahan elektron dari type N menuju type P dan berpindahnya hole dari type P ke type N. Proses rekombinasi antara elektron dengan hole, mengakibatkan pelepasan energi berupa cahaya.

#### **2.4. Photodioda**

Photodioda yang peka terhadap cahaya infra merah mempunyai bentuk seperti dioda LED dan ada yang sudah terintegrasi bersama dengan satu komponen IC yaitu modul penerima infra merah (*infra receiver module*) yang keluarannya sudah dalam bentuk sinyal digital. Pada prinsip dasar kerja photodioda yaitu pada keadaan normal photodioda berlaku sebagai dioda biasa yang dapat



mengantarkan arus listrik dari anoda ke katoda, namun mempunyai tahanan balik yang besar. Bila cahaya luar mengenai junction photodiode maka tahanan balik akan mengecil dan menimbulkan arus balik sehingga photodiode berlaku sebagai photodiode yang dipasang balik atau dibias reverse. Berikut adalah symbol photodiode:



**Gambar 2.9** Simbol Photodiode

(Sumber : [www.en.wikipedia.org](http://www.en.wikipedia.org))

## 2.5. Motor DC

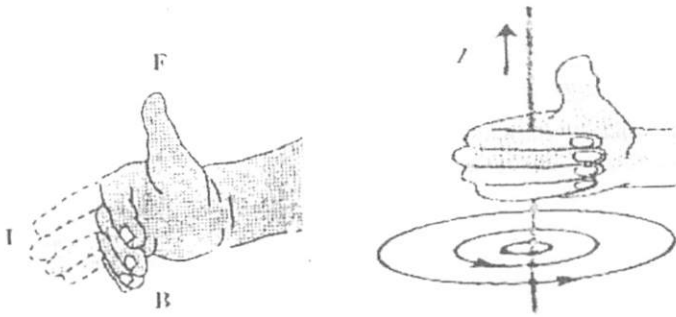
Motor dc adalah suatu motor penggerak yang dikemudikan dengan arus searah (DC), dan merupakan salah satu bentuk mesin arus searah yang akan berfungsi bila :

- Kumpan medan, untuk menghasilkan medan magnet.
- Kumpan jangkar, untuk mengimbaskan ggl pada konduktor-konduktor yang terletak pada alur-alur jangkar.
- Celah udara, yang memungkinkan berputarnya jangkar dalam medan magnet.

### 2.5.1. Prinsip Kerja Motor DC

Apabila kumpan jangkar dari mesin arus searah dialiri arus dan kumpan medan diberi penguatan, maka akan timbul gaya *lorenz* pada setiap sisi kumpan jangkar. Arah medan magnet dapat ditentukan dengan kaidah tangan kanan dengan ibu jari menunjukkan arah gaya putar dari arus listrik yang mengalir dalam sebuah kumpan jangkar yang berada dalam medan magnet, jari tengah

menunjukkan arah arus listrik yang mengalir pada konduktor, dan jari telunjuk menunjukkan arah medan magnet.



**Gambar 2.10** Kaidah Tangan kanan

(Sumber: Pengantar Teknik Tenaga Listrik. Ir. Hamzah Berahim)

Adapun besarnya *gaya lorentz* yang bekerja pada kumparan jangkar tersebut adalah :

$$F = B \cdot I \cdot L$$

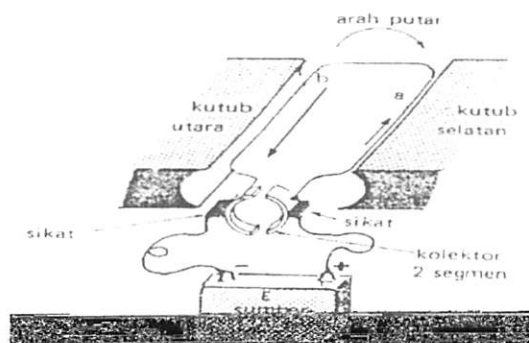
Dimana :

F = Gaya *lorentz*

B = Kerapatan fluks magnet (weber)

L = Panjang konduktor (meter)

I = Arus listrik (ampere)



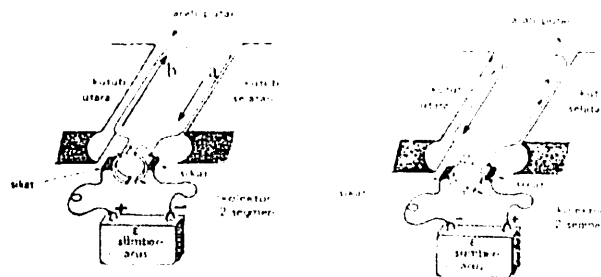
**Gambar 2.11** Cara Kerja Motor DC

(Sumber: Pengantar Teknik Tenaga Listrik. Ir. Hamzah Berahim)

Perputaran kumparan jangkar dalam medan magnet dan konduktor jangkar yang dialiri arus menimbulkan kopel yang memotong medan magnet tersebut sehingga pada konduktor jangkar akan timbul tegangan induksi (ggl). Ggl ini arahnya berlawanan dengan tegangan jepit jangkar.

### 2.5.2. Pengendali Arah Putaran Motor DC

Pada gambar berikut ini memperlihatkan cara kerja motor dc dalam mengendalikan dua arah putar yaitu dengan membalikan output catu daya pada input motor dc.



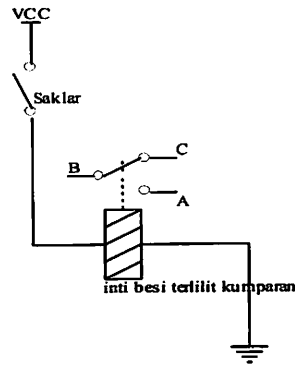
**Gambar 2.12** Pengendali Arah Putaran Motor DC

(Sumber: Pengantar Teknik Tenaga Listrik. Ir. Hamzah Berahim)

Pada gambar 2.12 diatas, agar arah putar motor dc berubah, maka polaritas tegangan pada catu daya harus dibalik.

### 2.6. Relay

Relay adalah komponen elektronika yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada sebuah besi lunak. Jika kumparan dialiri arus listrik maka inti besi akan menjadi magnet dan menarik pegas sehingga kotak AB terhubung dan BC terputus.

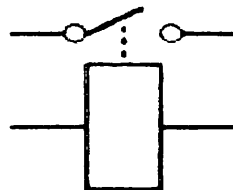


**Gambar 2.13** Cara Kerja Relay  
(Sumber: Relay, [www.national.com](http://www.national.com))

Relay merupakan suatu alat untuk menghubungkan atau memerlukan kontak antara komponen yang satu dengan yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang ditimbulkan dari adanya medan magnet listrik yang dihasilkan oleh kumparan yang melilit pada besi lunak.

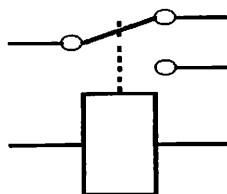
Ada beberapa macam relay, antara lain:

- SPST (*Single Pin Single Terminal*)



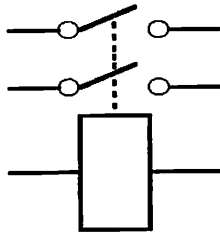
**Gambar 2.14** Relay SPST  
(Sumber: Relay, [www.national.com](http://www.national.com))

- SPDT (*Single Pin Dual Terminal*)



**Gambar 2.15** Relay SPDT  
(Sumber: Relay, [www.national.com](http://www.national.com))

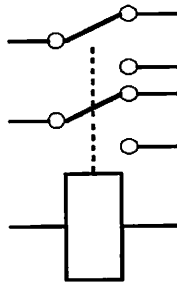
- DPST (*Dual Pin Single Terminal*)



**Gambar 2.16** Relay DPST

(Sumber: Relay, [www.national.com](http://www.national.com))

- DPDT (*Dual Pin Dual Terminal*)



**Gambar 2.17** Relay DPDT

(Sumber: Relay, [www.national.com](http://www.national.com))

## 2.7. Keypad

Keypad merupakan salah satu sarana untuk memasukkan suatu data ke komputer atau minimum sistem. Untuk rangkaian keypad dalam tugas akhir ini digunakan keypad matrik 4 x 4.

Keypad matrik 4 x 4 merupakan susunan 16 tombol membentuk keypad sebagai sarana masukan ke mikrokontroller, meskipun jumlah tombol ada 16 tapi hanya memerlukan 7 jalur port paralel, seperti terlihat dalam Gambar 2.18.



The text in this section is extremely faint and illegible. It appears to be a series of lines of text, possibly a list or a set of instructions, but the characters are too light to be read accurately. The text is distributed across the middle and lower portions of the page.

The text at the bottom of the page is also very faint and illegible. It may represent a signature, a date, or a footer, but the details are completely obscured by the low contrast of the scan.



**Gambar 2.18** Keypad Matrik 4 x 4  
(Sumber: Data Sheet Keypad)

## 2.8. LCD M1632

LCD M1632 adalah sebuah modul LCD dot matrix dengan kebutuhan daya kecil, dilengkapi dengan panel LCD bertingkat kontras cukup tinggi dan pengontrol LCD internal. Pengontrol mempunyai pembangkit karakter ROM/RAM dan penampil data RAM. Semua fungsi penampil diatur oleh instruksi-instruksi dan modul dapat dengan mudah diantar-mukakan dengan unit mikroprosesor. Keistimewaan modul LCD M1632 adalah :

- Dua baris tampilan LCD yang terdiri dari 5X7 dot matrix ditambah kursor, masing-masing baris 16 karakter.
- ROM membangkitkan 192 tipe karakter.
- RAM untuk data penampil sebanyak 80X8 bit (maksimum 80 karakter).
- Osilator internal.
- Catu daya +5 Volt.
- Secara otomatis akan reset saat catu daya dinyalakan.

LCD mempunyai 16 kaki. Fungsi tiap-tiap kaki ditunjukkan dalam Tabel 2.4.

**Tabel 2.4.** Fungsi Kaki-kaki M1632

No	Nama Penyemat	Fungsi
1	Vss	0 V (GND)
2	Vcc	5V $\pm$ 10%
3	Vee	Drive LCD
4	RS	Pemilih Register 0 = Instruksi register (tulis) Bussy flag dan address counter (baca) 1 = Register data (baca dan tulis)
5	R/W	Sinyal pemilih baca dan tulis 0 = tulis                      1 = baca
6	Enable	Sinyal untuk mengawali operasi
7-14	Data Bus	Saluran data
15	V + BL	Pengendali kecerahan latar belakang LCD, 4-4,42 V dan 50-200 mA
16	V – BL	Pengendali kecerahan latar belakang LCD, 0 V

(Sumber : Seiko Instruments Inc.)

## 2.9. Limit Switch

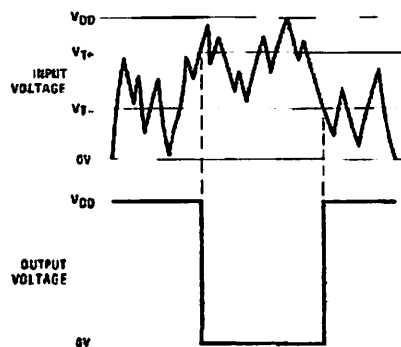
*Limit switch* ini adalah suatu komponen detektor manual atau panic switch. Komponen ini merupakan suatu komponen detector manual atau panic switch. Komponen ini merupakan yang paling sederhana sekali dimana dalam memperoleh respon dari luar sangat mudah dan hanya mempunyai dua posisi yaitu NO (normaly open) jika dalam posisi normal artinya tidak ada respon dari luar dia akan mempunyai kondisi terbuka (off), dan jika NC (normaly close) adalah dalam posisi normal dia akan berkondisi (on).

*Limit switch* merupakan peralatan mekanis yang dapat dipergunakan untuk menentukan posisi fisik dari peralatan. Jenis saklar ini akan aktif bila suatu level melebihi batasan yang telah di tetapkan atau keluar dari jangkauan yang ditentukan.



## 2.10. Schmitt Trigger

Bentuk gelombang yang buruk dan mempunyai waktu naik turun yang sangat lambat seperti dalam gambar 2.19 dapat mengakibatkan operasi yang tidak dapat diandalkan, apabila dihubungkan langsung ke penghitung, gerbang, atau rangkaian lainnya. *Schmitt trigger* digunakan untuk mempersegikan sinyal input dan membuat sinyal menjadi lebih baik.

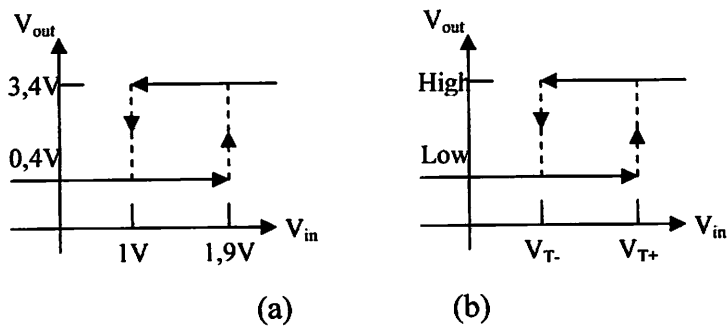


**Gambar 2.19** *Schmitt Trigger* untuk pembentukan gelombang  
(Sumber : Datasheet *Inverter Schmitt Trigger 74LS14*)

Gambar 2.20 (a) menunjukkan cara kerja sebuah *schmitt trigger*. Tegangan keluaran antara logika 0 dan 1 ialah 0,4V atau 3,4V. Bila  $V_{out}$  berada pada keadaan rendah, maka diperlukan untuk menaikkan  $V_{in}$  sedikit diatas 1,9V guna menghasilkan suatu perpindahan. Setelah berada pada keadaan tinggi,  $V_{out}$  tetap pada 3,4V dan  $V_{in}$  turun sedikit dibawah 1V. Pada saat ini, keluaran akan kembali ke keadaan rendah, yaitu 0,4V. Perpindahan yang cepat ini ditunjukkan oleh garis titik-titik.

Gambar 2.20 (b) menunjukkan grafik bagi setiap *schmitt trigger*. Nilai  $V_m$  yang mengakibatkan keluaran meloncat dari keadaan rendah ke tinggi disebut tegangan ambang menuju positif, yang dilambangkan sebagai  $V_{T+}$ . Demikian

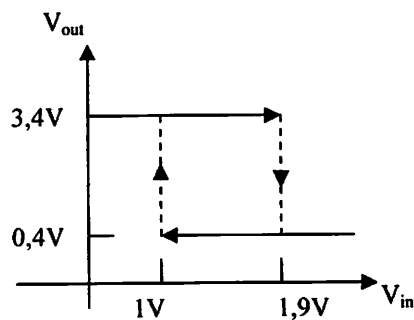
pula,  $V_{in}$  yang mengakibatkan keluaran berpindah dari keadaan tinggi ke keadaan rendah disebut tegangan ambang menuju negatif, yang dilambangkan sebagai  $V_{T-}$ .



**Gambar 2.20** Karakteristik *schmitt trigger*

(Sumber : Datasheet *Inverter Schmitt Trigger 74LS14*)

Sebagai contoh *inverter schmitt trigger* adalah IC TTL 74LS14 yang berisi 6 buah *inverter schmitt trigger*. Untuk  $V_{cc}$  sebesar 5V, tegangan ambang secara umum adalah  $V_T$  sebesar umum adalah  $V_{OL}$  sebesar 0,4V dan  $V_{OH}$  sebesar 3,4V. Grafik masukan dan keluaran IC TTL 74LS14 ditunjukkan pada Gambar 2.21.

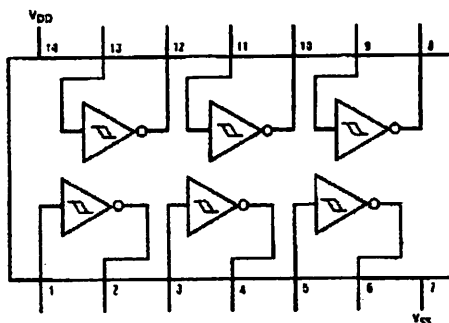


**Gambar 2.21** Karakteristik *Inverter Schmitt Trigger*

(Sumber : Datasheet *Inverter Schmitt Trigger 74LS14*)

Akibat adanya inversi, maka keluaran IC TTL 74LS14 menjadi terbalik. Bila masukan melampaui 1,9V, keluaran berpindah ke keadaan rendah. Bila masukan kurang dari 1V, keluaran berpindah kembali ke keadaan tinggi. Keluaran dari inverter schmitt trigger ialah kebalikan dari masukannya. *Schmitt*

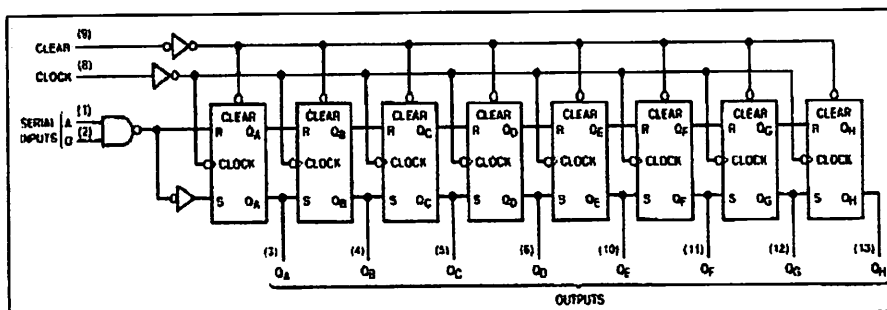
*trigger* dengan gerbang not yang dikemas dalam bentuk IC TTL 74LS14 diperlihatkan dalam gambar 2.22.



**Gambar 2.22 Inverter Schmitt Trigger IC 74LS14**  
(Sumber : Datasheet Inverter Schmitt Trigger 74LS14)

### 2.11. IC 74LS164

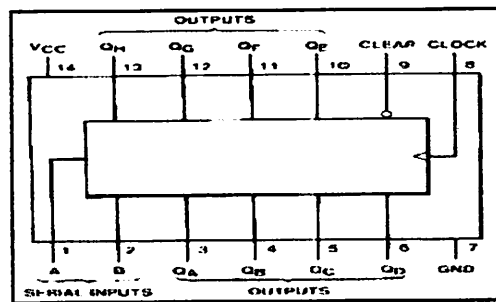
Bagian ini akan merinci salah satu dari sekian banyak register geser, yaitu register yang masukannya seri dan keluarannya paralel 8 bit. IC yang digunakan adalah IC 74LS164 yang merupakan register 8 bit yang terpicu sentuh dengan masukan serial. Semua keluaran paralel tersedia atas setiap *flip-flop D internal*. Diagram rinci tersedia dalam gambar 2.23 memperlihatkan penggunaan delapan *flip-flop D internal*, masing-masing dengan keluaran data paralelnya (QA sampai QB).



**Gambar 2.23 Diagram Detail Register Geser 8 Bit 74LS164**

(Sumber : Fairchild Semikonduktor Datasheet)

IC 74LS164 terlihat memiliki sebuah masukan serial. Data dimasukkan secara serial melalui salah satu dari dua masukan (A dan B) di NAND-kan. Kedua masukan ini bisa digabung menjadi satu masukan atau bisa juga salah satu ditentukan sebagai logika tinggi, sedangkan yang satu lagi untuk pemasukan data. Masukan pengatur ulang utama *Clear* untuk IC 74LS164 merupakan input aktif rendah. Dalam tabel kebenaran (Tabel 2.5) memperlihatkan bahwa bila diaktifkan, masukan *Clear* akan mematikan semua input yang lainnya, dan mengembalikan semua *flip-flop* ke 0. IC 74LS164 menggeser data suatu tempat kekanan dalam setiap peralihan rendah ke tinggi dalam masukan *Clock*.



**Gambar 2.24** Konfigurasi Kaki-Kaki 74LS164

(Sumber : Fairchild Semikonduktor Datasheet)

Untuk tabel kebenaran dari IC jenis dapat dilihat pada tabel berikut :

**Tabel 2.5.** Tabel Kebenaran IC 74LS164

Inputs				Outputs			
Clear	Clock	A	B	QA	QB	...	QH
L	X	X	X	L	L	...	L
H	L	X	X	QA0	QB0	...	QH0
H	↑	H	H	H	QAn	...	QGn
H	↑	L	X	L	QAn	...	QGn
H	↑	X	L	L	QAn	...	QGn

(Sumber : Fairchild Semikonduktor Datasheet)

H = Logika Tinggi

L = Logika Rendah

X = Sembarang masukan termasuk transisi

↑ = Transisi dari rendah ke tinggi

QA0, QB, 0, QH0 = masing-masing logika pada QA, QB atau QH sebelum terjadi kondisi-kondisi jalan masuk stasioner yang ditunjukkan.

QAn, QGn = logika QA atau QG sebelum transisi ↑ pada *clock* yang paling akhir, menunjukkan penggeseran satu bit.

## BAB III

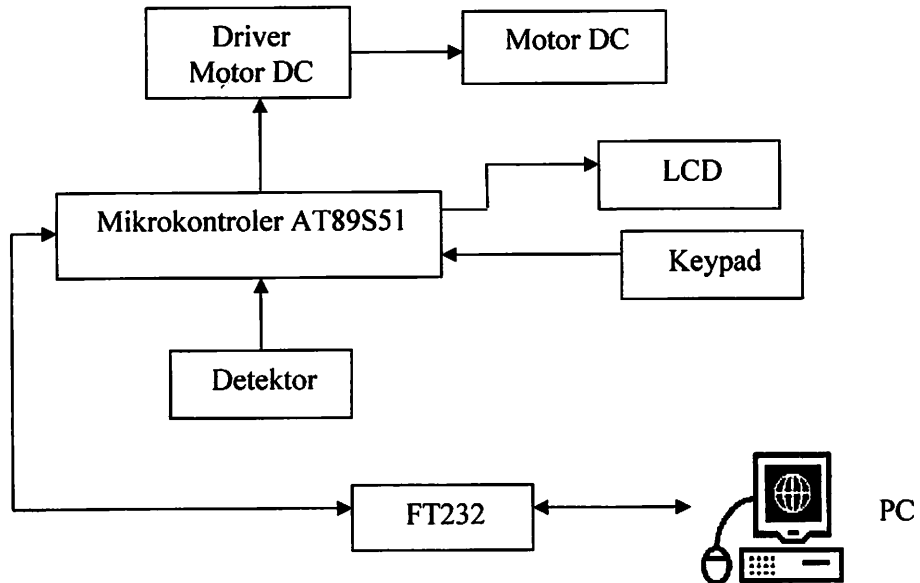
### PERENCANAAN DAN PEMBUATAN ALAT

#### 3.1. Pendahuluan

Pada dasarnya perencanaan dan pembuatan alat ini secara garis besar dibagi atas dua bagian, yaitu:

1. Perencanaan perangkat keras
2. Perencanaan perangkat lunak

Komponen yang dipakai dalam perencanaan ini antara lain mikrokontroler AT89S51 sebagai kontrol utama, dengan komponen pendukung meliputi *Keypad*, LCD, detektor yang terdiri dari *infrared* dan *photodiode*, *relay*, motor DC dan modul FT232BM sebagai antarmuka dengan PC. Untuk mendeskripsikan keseluruhan sistem dari alat ini dapat dilihat dari diagram blok di bawah ini.



**Gambar 3.1** Blok Diagram Sistem

Fungsi dari masing – masing blok diagram adalah sebagai berikut :

### 1. Keypad

Keypad yang digunakan berupa keypad matrik 4 x 4 yang berfungsi sebagai inputan id dan password dari setiap kendaraan yang akan melewati pintu gerbang.

### 2. LCD

Berfungsi untuk menampilkan inputan dari keypad.

### 3. Mikrokontroler AT89S51

Memproses data dari inputan keypad untuk verifikasi id dan password. Pada proses verifikasi ini mikrokontroler berkomunikasi dengan PC.

### 4. Driver Motor DC

Driver motor DC berfungsi sebagai pembalik polaritas motor DC, sehingga motor dapat berputar kekiri dan kekanan untuk menutup dan membuka pintu gerbang.

### 5. Motor DC

Motor DC disini berfungsi untuk membuka dan menutup pintu gerbang.

### 6. Detektor

Detektor yang digunakan adalah *infrared*, *photodiode*, dan *Limit switch*. *Infrared* dan *photodiode* berfungsi sebagai saklar yang memberi perintah ke mikrokontroler untuk menutup gerbang. Sedangkan *Limit switch* berfungsi sebagai saklar yang memberi perintah ke mikrokontroler untuk mematikan motor ketika pintu telah terbuka atau tertutup.

### 7. Modul FT232BM

Mengkonversi pengiriman data serial dari mikrokontroler menjadi USB sebagai *interface* pada PC.

## 8. *Personal Computer* (PC)

Digunakan sebagai *database* yang berisi data-data sopir yang diperbolehkan masuk pabrik.

- **Prinsip Kerja Sistem**

Adapun prinsip kerja dari sistem ini adalah sebagai berikut :

Tombol-tombol *keypad* ditekan, menghasilkan data *input* ke mikrokontroler. Data *input* dari *keypad*, diolah dan ditampilkan pada LCD. Setelah itu, data dikirim secara serial melalui USB ke PC menggunakan modul FT232BM. Data *input* dibandingkan dengan data pada database. Jika data *input* sesuai dengan data pada database, maka pintu akan terbuka. Setelah kendaraan melewati detektor, detektor *receiver* akan memberikan sinyal ke rangkaian kendali dan pintu akan menutup. Mekanisme kerja seperti ini berlaku pada pintu keluar pabrik dan pintu masuk pabrik.

## 3.2. Perencanaan Perangkat Keras

### 3.2.1. Mikrokontroler AT89S51

Penggunaan Mikrokontroler AT89S51 harus didukung oleh beberapa rangkaian penunjang agar dapat melakukan fungsinya, antara lain rangkaian *clock* dan *reset*. Selain itu juga harus ditentukan penggunaan *port – port*-nya dan sinyal – sinyal yang digunakan untuk mendukung proses yang ingin dilakukan.

#### 3.2.1.1. Rangkaian Clock

Kecepatan proses pengolahan data pada mikrokontroler ditentukan oleh *Clock* (pewaktu) yang dikendalikan oleh mikrokontroler tersebut. Pada



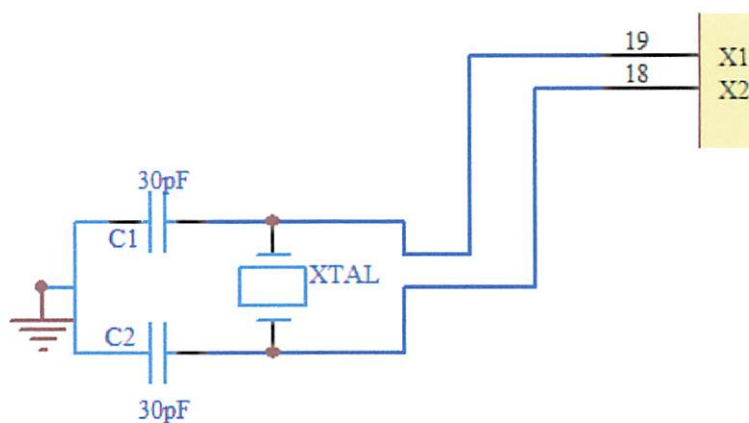
mikrokontroler AT89S51 terdapat *internal clock* generator yang berfungsi sebagai sumber *clock*, tapi masih memerlukan rangkaian tambahan untuk membangkitkan *clock* yang diperlukan.

Rangkaian *clock* ini terdiri atas dua buah kapasitor dan sebuah kristal yang terangkai sedemikian rupa dan kemudian dihubungkan dengan chip yang tersedia pada AT89S51.

Dalam perencanaan ini menggunakan :

- ✓  $C = 30 \text{ pF}$ . Penentuan besarnya kapasitansi disesuaikan dengan spesifikasi lembar data AT89S51
- ✓ X-Tal = 11,0592 MHz.

Adapun gambar rangkaian *clock* dapat dilihat pada gambar dibawah ini :



Dengan menggunakan nilai kristal diatas maka dapat dihitung waktu yang diperlukan untuk satu siklus mesin.

Diketahui :  $F = 11,0592 \text{ Mhz}$

$$T = \frac{1}{F}$$

Sehingga :

$$T = \frac{1}{11,0592 \text{ MHz}} = \frac{1}{11,0592} \mu\text{s}$$

Maka untuk satu siklus mesin dari mikrokontroller AT89S51 adalah sebesar :

$$T_{me} = 12 \times T$$

$$T_{me} = 12 \times \frac{1}{11,0592} \mu\text{s} = 1,09 \mu\text{s}$$

$$T_{me} = 1,09 \mu\text{s}$$

### 3.2.1.2. Rangkaian Reset

Untuk *mereset* mikrokontroler AT89S51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*.

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11,0592 \text{ MHz}} \text{ s} = 9,042 \times 10^{-8} \text{ s}$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk *mereset* mikrokontroler adalah :

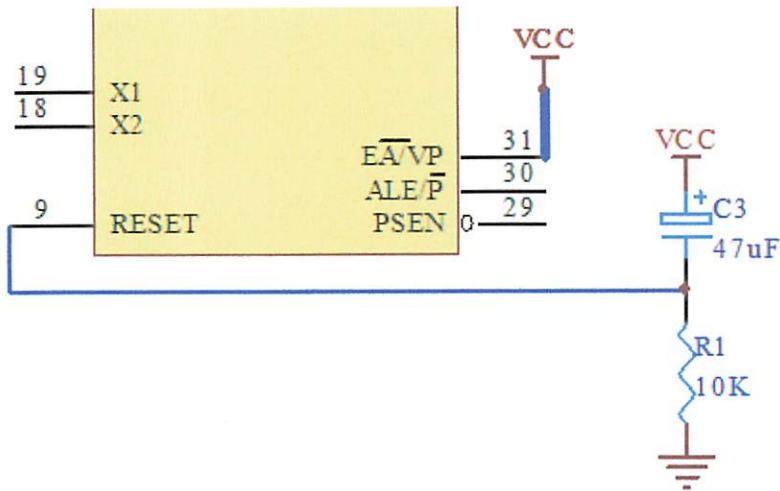
$$\begin{aligned} \text{reset}(\text{min}) &= T \times \text{periode yang dibutuhkan} \\ &= 9,042 \times 10^{-8} \times 24 = 2,18 \mu\text{s} \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2,18  $\mu\text{s}$  untuk *mereset*. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dengan menentukan nilai R = 10k $\Omega$  dan C = 47  $\mu\text{F}$ , maka :

$$t = 0,357 R.C = 0,357 \times 10. 10^3 \Omega \times 47.10^{-6} = 167,8 \text{ ms}$$

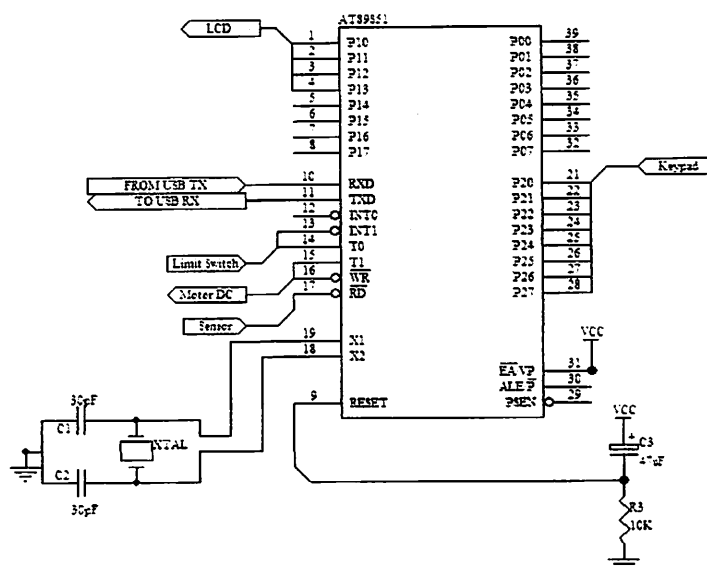
Jadi dengan nilai komponen  $R = 10 \text{ k}\Omega$  dan  $C = 47\mu\text{F}$  dapat memenuhi syarat untuk waktu yang dibutuhkan oleh mikrokontroler.

Maka dengan demikian nilai 't' jauh lebih besar dibandingkan dengan nilai minimalnya.



**Gambar 3.3** Rangkaian Reset

### 3.2.1.3. Konfigurasi Kaki Mikrokontroler AT89S51



**Gambar 3.4** Konfigurasi Kaki-Kaki AT89S51

Dari 40 pin di atas, yang kita pergunakan adalah :

1. Port 1
  - ✓ P1.0 – P1.3 digunakan sebagai input data bus LCD M1632.
2. Port 2
  - ✓ P2.0 – P2.7 digunakan untuk input Keypad Matrik 4 X 4.
3. Port 3
  - ✓ P3.0 (RXD) digunakan untuk menerima transfer data dari luar lewat USB.
  - ✓ P3.1 (TXD) digunakan untuk mengirim data keluar lewat USB.
  - ✓ P3.3 – P3.4 digunakan untuk Limit Switch
  - ✓ P3.5 – P3.6 digunakan untuk Driver Motor yang mengontrol arah putaran motor DC.

#### 4. X-TAL1 dan XTAL2

✓ Digunakan untuk input dari *clock* tambahan

#### 5. Reset

✓ Input Reset

#### 6. GND

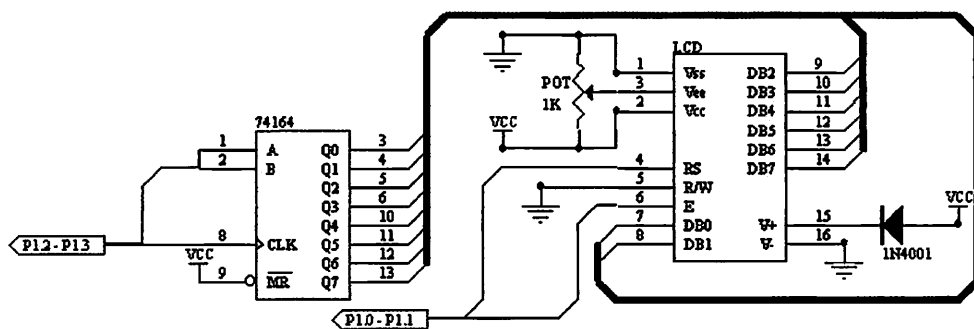
✓ Ground

#### 7. VCC

✓ VCC pada Mikrokontroller diberikan tegangan sebesar 5 Volt sesuai dengan operasi chip tunggal yang diperbolehkan pada Data Sheet.

### 3.2.2. Rangkaian LCD M1632 ( *Liquid Crystal Display* )

Dalam aplikasi ini menggunakan sebuah layar LCD (*Liquid Crystal Display*) yaitu jenis *Seiko Instrument M1632* yang merupakan LCD dua baris dengan setiap barisnya terdiri 16 karakter dan menggunakan IC 74LS164 yang merupakan register geser 8 bit yang memiliki jalan masuk deret tergerbang. Gambar hubungan antara LCD, IC74LS164 dan mikrokontroler dapat dilihat dalam Gambar 3.5.



**Gambar 3.5 Rangkaian LCD**

LCD dot matrik ini membutuhkan sepuluh buah pin masukan/keluaran dari mikrokontroler dan IC 74164. Adapun dua buah pin yaitu port 1.0 pada penyemat RS yang digunakan sebagai sinyal pemilih register dan port 1.1 pada penyemat Enable digunakan sebagai sinyal operasi awal, sinyal enable ini mengaktifkan data tulis atau baca oleh mikrokontroler, penyemat DB0-DB7 yang dihubungkan ke pin data IC74164 digunakan untuk menampilkan karakter yang dikehendaki oleh mikrokontroler. Ketika terdapat data pada jalur data, data tersebut akan ditahan dengan memberikan *clock* pin E pada LCD. Pin RS menentukan apakah data yang ditahan akan digunakan sebagai instruksi untuk mengatur *setting* tampilan pada LCD atau sebagai kode karakter yang diperlukan LCD untuk menampilkan suatu karakter. Sedangkan untuk pin R/W pada LCD dihubungkan ke *ground* karena dalam hal ini LCD hanya melakukan operasi write atau operasi menampilkan karakter.

Untuk pin Vcc pada LCD dihubungkan ke supply +Vcc dan Vss dihubungkan ke *ground*. Pin V<sub>EE</sub> beserta pin Vcc dan Vss dihubungkan ke *trimmer potensio* atau kadang disebut dengan *trimpot*. *Trimpot* ini digunakan untuk mengatur kontras dari tampilan LCD dengan cara mengubah tegangan pada pin V<sub>EE</sub>.

Pada lembaran datasheet modul LCD M1632 Seiko Instrument Inc. disebutkan bahwa :

*Power supply LCD meliputi :*

V<sub>ss</sub> = 0 V

V<sub>cc</sub> = 5 V ± 5% (2mA)

*Power supply back light :*

$$V + BL = 4 - 4,2 \text{ V (50 sampai 200 mA)}$$

$$V - BL = 0 \text{ V (GND)}$$

Pada input  $V + BL$  dipasang sebuah dioda 1N4001 (bahan silicon dengan  $V_d = 0,65 \text{ V}$  sampai  $0,7 \text{ V}$ ). Tujuannya adalah didapatkan tegangan  $V + BL$  sebesar  $4 - 4,2 \text{ V}$  dengan perhitungan sebagai berikut :

$$V_{cc} = V_d + (V + BL)$$

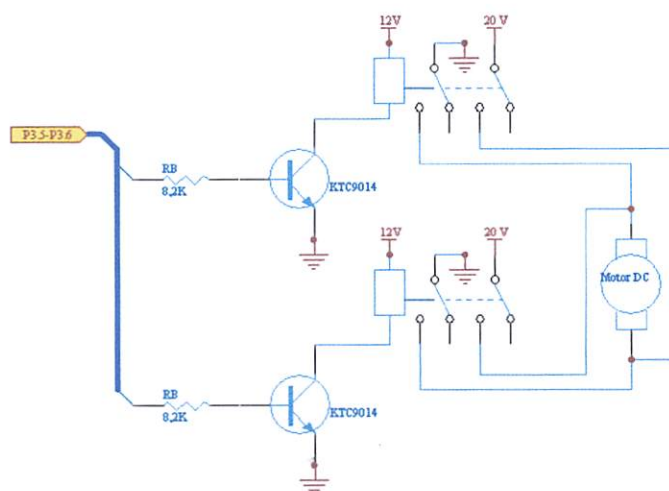
$$5 = 0,7 + (V + BL)$$

$$(V + BL) = 5 - 0,7 = 4,3 \text{ Volt}$$

Dipilih dioda 1N4001 karena arus maksimum yang bisa dilewatkan oleh dioda ini sebesar  $1 \text{ A}$ .

### 3.2.3. Rangkaian Driver Motor DC

Rangkaian driver yang digunakan untuk menggerakkan relai yang terhubung dengan motor DC terdiri dari transistor NPN KTC9014 yang difungsikan sebagai transistor *switching*.



**Gambar 3.6** Rangkaian Driver Motor

Dengan resistansi relay sebesar  $800 \Omega$ , tegangan catu sebesar  $12 \text{ V}$ , dan  $V_{ce}$  saturasi sebesar  $0,25 \text{ V}$  maka besar arus ( $I_{relay}$ ) adalah :

$$I_{relay} = \frac{V_{cc} - V_{ce}}{R_{relai}}$$

$$I_{relay} = \frac{12\text{v} - 0,25\text{v}}{800\Omega}$$

$$= 14.68 \text{ mA}$$

Data transistor KTC9014 yang diperoleh dari *datasheet* adalah :

- $V_{ce}$  saturasi =  $0,25 \text{ volt}$ .
- $H_{fe}$  minimum =  $60$ .
- $H_{fe}$  maximum =  $1000$ .
- $V_{be} = 0,7 \text{ volt}$ .

Dalam perancangan ini digunakan  $hfe$   $60$  dengan pertimbangan transistor tetap mampu menggerakkan relay meskipun penguatan yang dipakai adalah minimum, maka:

$$I_B = \frac{I_c}{hfe}$$

$$I_B = \frac{14,68\text{mA}}{60}$$

$$= 0,245 \text{ mA}$$

Sehingga dalam perancangan ini digunakan transistor KTC9014.

Jika  $V_{bb}$  adalah tegangan keluaran dari mikrokontroler saat logika tinggi yaitu sebesar  $2,4\text{V}$  dan  $V_{be} = 0,7 \text{ V}$  maka besar resistansi  $R_b$  adalah :

$$R_B = \frac{V_{bb} - V_{be}}{I_B}$$



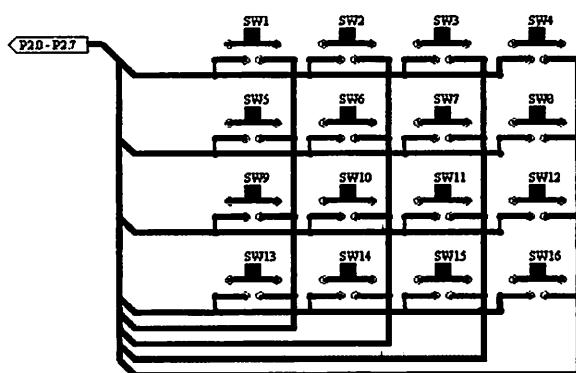
$$R_B = \frac{2,4v - 0,7v}{0,245mA}$$

$$= 6,94K \Omega \cong 8,2K \Omega$$

Dengan resistansi  $R_B$  sebesar  $8,2K \Omega$ , arus  $I_B$  menjadi  $0,21 \text{ mA}$  sehingga telah diperoleh arus  $I_B$  yang dibutuhkan.

### 3.2.4. Rangkaian Keypad

Keypad yang digunakan adalah keypad matriks  $4 \times 4$ . Port yang digunakan untuk sinyal port 2.0 – port 2.3 dari mikrokontroler masuk ke kelompok baris keypad, sedangkan kelompok kolom keypad dihubungkan ke port 2.4 – port 2.7 mikrokontroler. Untuk fungsi dari tombol-tombol keypad tergantung pada pemrogram. Berikut blok diagram dari penyambungan keypad ke mikrokontroler.

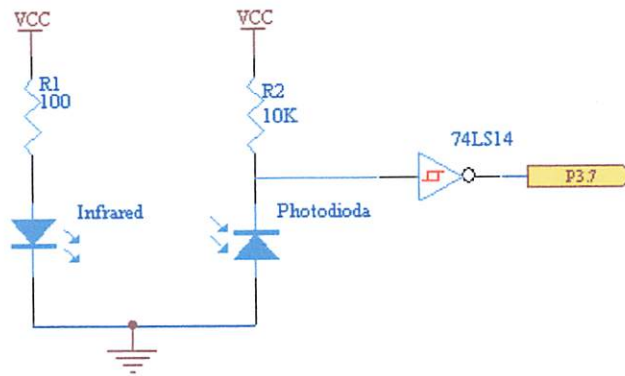


**Gambar 3.7** Blok Diagram Hubungan Keypad Dengan Mikrokontroler

### 3.2.5. Rangkaian Sensor.

Dalam perancangan alat ini digunakan *LED Infrared* sebagai pemancar dan *photodiode* sebagai penerima. Keduanya dipasang di belakang pintu gerbang yang saling berhadapan dan *horizontal* terhadap pintu gerbang. Ketika

kendaraan melewati pintu gerbang maka detektor akan memberikan sinyal ke mikrokontroler untuk menutup pintu gerbang. Untuk mendapatkan kuat cahaya *infrared*, maka diperlukan pula perhitungan yang tepat. Dengan adanya  $V_{cc}$  sebesar 5 Volt maka diperlukan pembatas tegangan dengan nilai resistor 'R'.



**Gambar 3.8** Rangkaian Detektor

Untuk nilai  $R_1$  dengan ketentuan  $V_f$  maksimal pada LED sebesar 1,65 V dan  $I_f$  sebesar 20 mA maka:

$$\begin{aligned} R_1 &= \frac{V_{cc} - V_f}{I_f} \\ &= \frac{5 - 1,65}{20 \times 10^{-3}} = 167,5 \Omega \approx 100 \Omega \end{aligned}$$

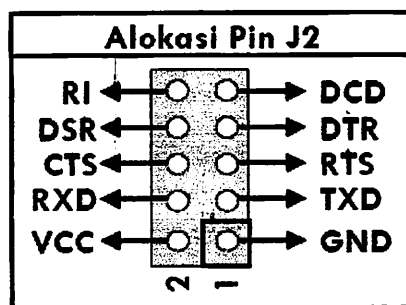
Untuk nilai  $R_2$  dengan  $I_{pd} = 0,5 \text{ mA}$  dan  $V_{cc} = 5 \text{ V}$  maka dapat dihitung untuk nilai  $R_2$  :

$$\begin{aligned} R_2 &= \frac{V_{cc}}{I_{pd}} \\ &= \frac{5 \text{ V}}{0,5 \text{ mA}} \\ &= 10 \text{ K}\Omega \end{aligned}$$

Keluaran tegangan dari photodiode akan dimasukkan ke rangkaian gerbang not smitch trigger IC 74LS14. IC ini akan membuat output berlogika “0” ketika masukan dibawah 1,8V dan output berlogika “1” ketika masukan diatas 2,2V. Dengan sifat itu jika kita memiliki dua kondisi keluaran dari rangkaian awal sensor yaitu 4,34 V (ketika terhalang, atau pada saat photodiode tidak bekerja ) dan 1,3V (ketika tidak terhalang, atau pada saat photodiode bekerja), maka keluaran dari rangkaian gerbang not smicth trigger adalah logika “0” ketika terhalang dan logika “1” ketika tidak terhalang.

### 3.2.6. Perancangan USB

Perancangan USB ini menggunakan modul FT232BM dengan penampang port sebagai berikut :



**Gambar 3.9** Pemakaian PIN pada port Modul FT232BM

Keterangan gambar :

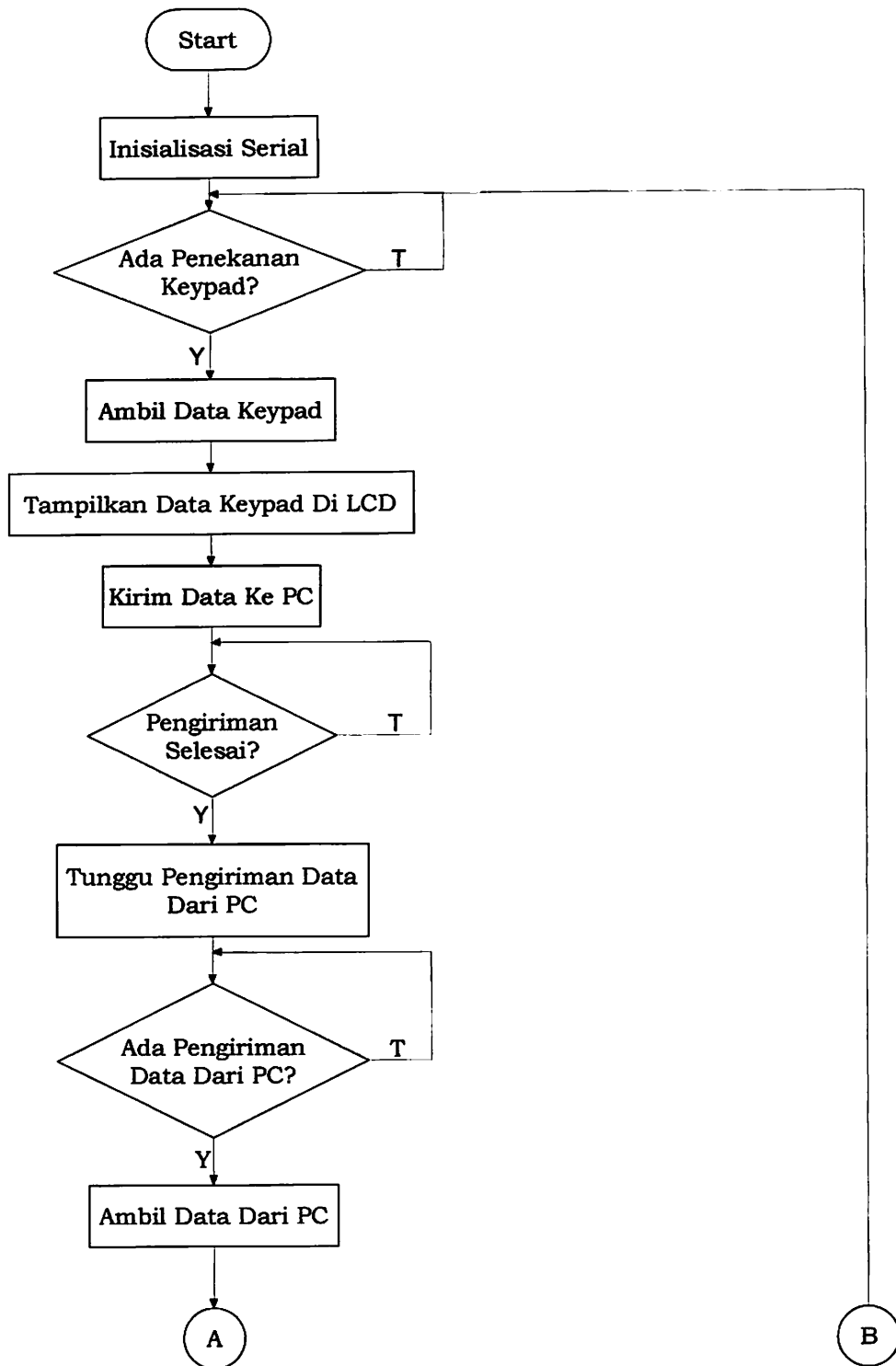
1. Port TXD dihubungkan pada port RXD mikrokontroler, dan port RXD converter dihubungkan pada port TXD mikrokontroler.
2. Port VCC dan GND dihubungkan pada catu daya DC 5 Volt.

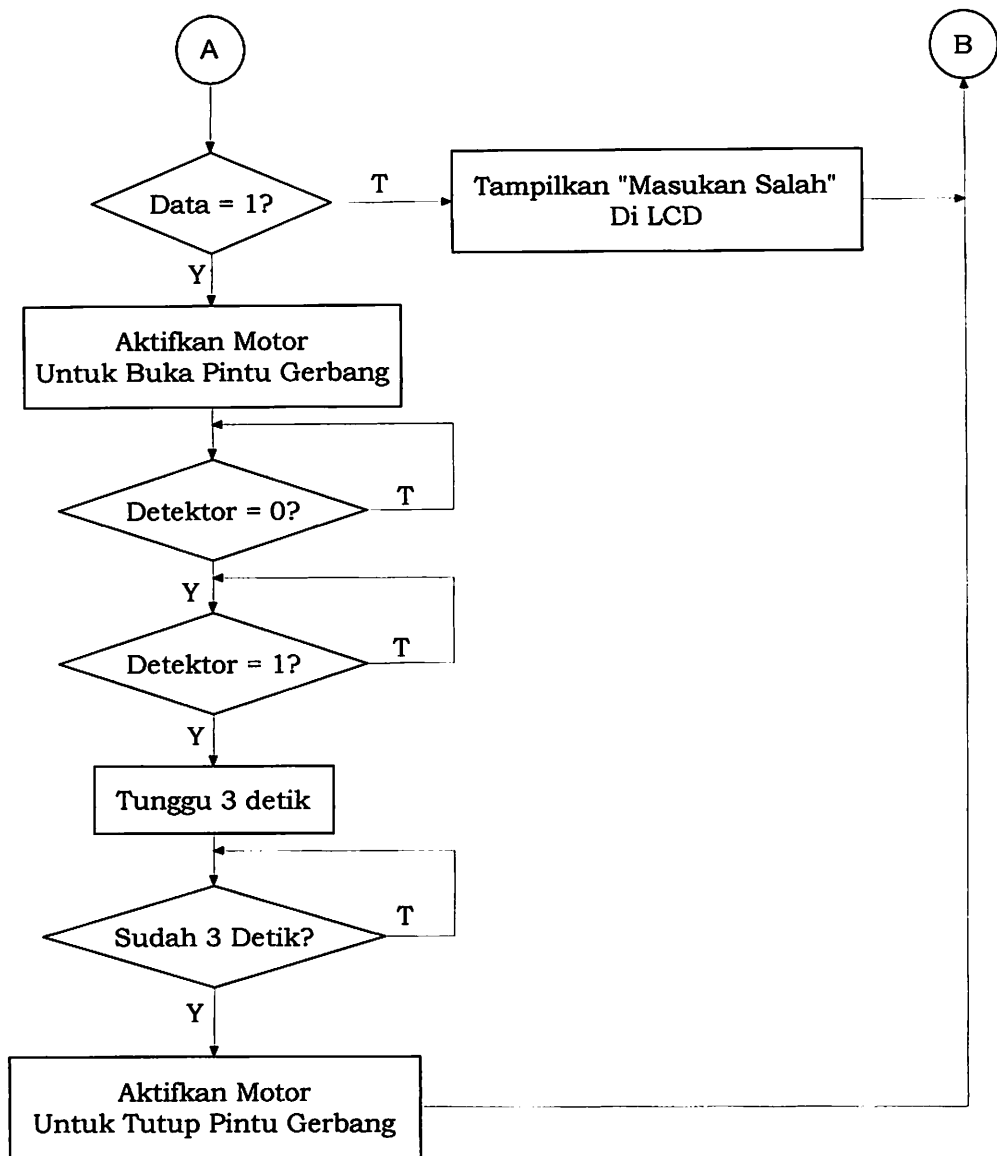
### 3.3. Perancangan Perangkat Lunak

Perancangan perangkat lunak diperlukan untuk menjalankan sistem sesuai yang kita harapkan. Untuk pemakaian mikrokontroler, perlu direncanakan perangkat lunak yang dapat mengatur sistem tersebut. Perangkat lunak disini adalah susunan perintah-perintah (program) di dalam memori yang harus dilaksanakan oleh mikrokontroler.

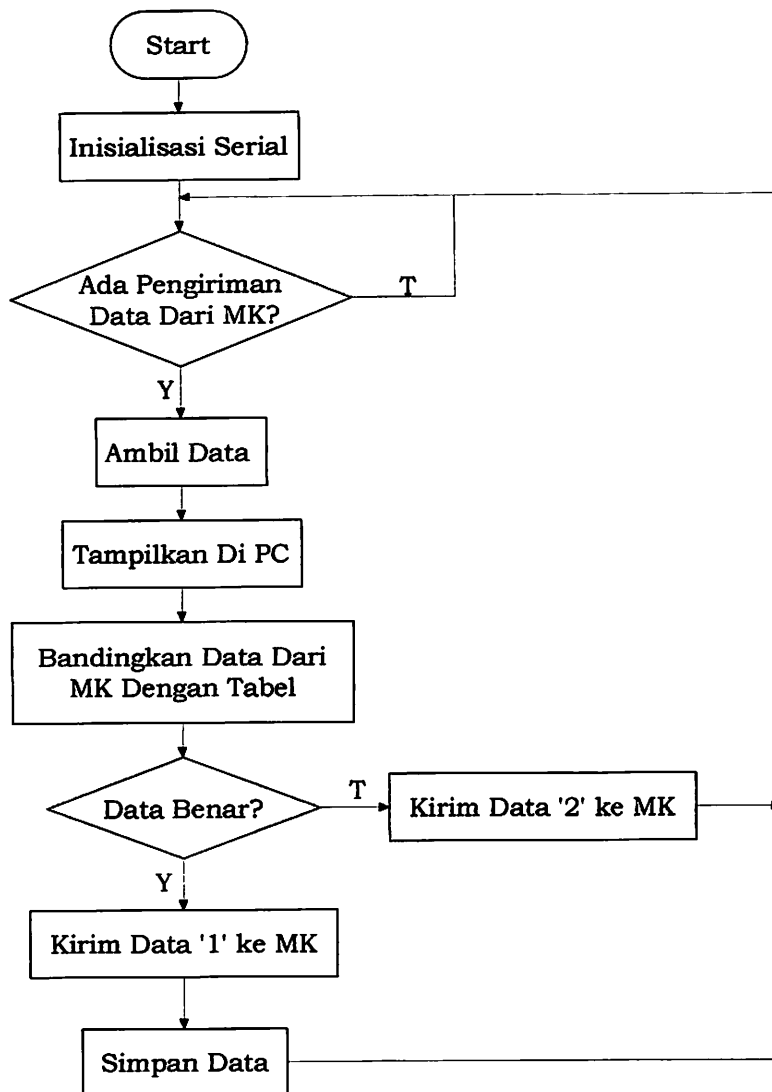
Memori merupakan fasilitas utama karena disinilah disimpan perintah-perintah yang harus dikerjakan. Memori di sini dapat dibedakan menurut fungsinya menjadi program dan memori data. Menurut letaknya memori dapat dibedakan menjadi memori program dan memori luar. Memori luar diberikan bila memori didalam mikrokontroller tidak mencukupi untuk menampung semua program dan data. Perancangan perangkat lunak (*software*) didasarkan perencanaan perangkat keras yang telah dibuat sebelumnya. Untuk mendapatkan sistem kerja yang diharapkan, *software* dari alat tersebut terdapat di bagian lampiran dan diagram alir (*flowchart*) tersebut adalah sebagai berikut :

### 3.3.1. Diagram Alir Software Pada Mikrokontroller





### 3.3.2. Diagram Alir Program Komputer



## **BAB IV**

### **PENGUJIAN ALAT**

#### **4.1. Tujuan**

Bab ini akan membahas tentang pengujian alat yang telah dirancang. Adapun tujuan dari pengujian ini adalah untuk mengetahui apakah *hardware* dan *software* dapat bekerja sesuai dengan kondisi yang diinginkan, maka dilakukan pengujian pada alat dan sistem kerja alat, yang mana prosedur pengujian meliputi:

1. Pengujian *Hardware*.
2. Pengujian sistem secara keseluruhan.

#### **4.2. Pengujian Perangkat Keras(*Hardware*)**

Dalam pengujian alat dibagi dalam beberapa sub sistem dari instrumen dan peralatan, diantaranya adalah :

1. Pengujian Detektor
2. Pengujian LCD
3. Pengujian Modul FT232BM
4. Pengujian Keseluruhan Alat

Pengujian perangkat keras ini mencakup pengujian rangkaian elektronika pada masing-masing blok maupun blok secara keseluruhan.

##### **4.2.1. Pengujian Detektor**

###### **4.2.1.1. Tujuan**

Untuk mengetahui apakah detektor tersebut dapat mendeteksi adanya penghalang atau tidak agar sistem dapat bekerja dengan baik.

###### **4.2.1.2. Alat yang digunakan**



... ..  
... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

... ..  
... ..

- Penghalang
- Voltmeter digital
- Catu Daya 5 Volt

#### 4.2.1.3. Prosedur pengujian

1. Memberikan catu daya 5 Volt pada rangkaian detektor.
2. Menghubungkan kaki photodiode dengan Voltmeter.
3. Memberi penghalang pada sinyal infrared yang mengenai photodiode.
4. Mengamati nilai tegangan pada voltmeter digital, saat diberi penghalang dan tidak diberi penghalang.

#### 4.2.1.4. Hasil pengujian



**Gambar 4.1** Tegangan Keluaran Photodiode Saat Tidak Terhalang



**Gambar 4.2** Tegangan Keluaran Photodiode Saat Terhalang

**Tabel 4.1** Hasil Pengukuran Tegangan Keluaran Photodioda

No	Vout Photodioda (V)	
	Saat Tidak Terhalang	Saat Terhalang
1	1,30	4,34
2	1,32	4,35
3	1,29	4,34
4	1,30	4,31
5	1,32	4,34
Rata-rata	1,306	4,336

Pada pengukuran photodioda saat terhalang didapatkan  $I_{pt}$  sebesar 0,04 mA.

Sehingga  $V_{out}$  perhitungan dapat dicari dengan rumus :

$$\begin{aligned}
 V_{cc} &= I_{pt} \cdot R + V_{out} \\
 5 &= (0,04 \cdot 10^{-3} \cdot 10 \cdot 10^3) + V_{out} \\
 V_{out} &= 5 - 0,4 \\
 &= 4,6 \text{ Volt}
 \end{aligned}$$

Pada pengukuran photodioda saat tidak terhalang didapatkan  $I_{pt}$  sebesar 0,66 mA.

Sehingga  $V_{out}$  perhitungan dapat dicari dengan rumus :

$$\begin{aligned}
 V_{cc} &= I_{pt} \cdot R + V_{out} \\
 5 &= (0,66 \cdot 10^{-3} \cdot 10 \cdot 10^3) + V_{out} \\
 V_{out} &= 5 - 6,6 \\
 &= 1,6 \text{ Volt}
 \end{aligned}$$

**Tabel 4.2** Hasil Perhitungan Tegangan Keluaran Photodioda

Vout Photodioda (V)	
Saat Tidak Terhalang	Saat Terhalang
1,6	4,6

Berdasarkan tabel pengukuran dan perhitungan di atas, maka dapat dicari %

kesalahan dengan menggunakan rumus :

$$\% \text{kesalahan} = \left| \frac{(\text{Rata - rata Pengukuran}) - \text{Perhitungan}}{\text{Perhitungan}} \right| \times 100\%$$

- Error tegangan keluaran photodioda saat tidak terhalang adalah:

$$\begin{aligned} \% \text{kesalahan} &= \left| \frac{(\text{Rata - rata Pengukuran}) - \text{Perhitungan}}{\text{Perhitungan}} \right| \times 100\% \\ &= \left| \frac{1,306 - 1,6}{1,6} \right| \times 100\% \\ &= 18,4\% \end{aligned}$$

- Error tegangan keluaran photodioda saat terhalang adalah:

$$\begin{aligned} \% \text{kesalahan} &= \left| \frac{(\text{Rata - rata Pengukuran}) - \text{Perhitungan}}{\text{Perhitungan}} \right| \times 100\% \\ &= \left| \frac{4,336 - 4,6}{4,6} \right| \times 100\% \\ &= 5,7\% \end{aligned}$$



**Gambar 4.3** Tegangan Keluaran Schmitt Trigger Saat Detektor Tidak Terhalang



**Gambar 4.4** Tegangan Keluaran Schmitt Trigger Saat Detektor Terhalang

**Tabel 4.3 Hasil Pengujian Detektor**

NO	Jarak	Hasil (Pada detektor/photodiode)
1	25 cm	Aktif
2	40 cm	Aktif
3	45 cm	Tidak Aktif

- **Analisa Hasil Pengujian**

Dalam pengujian detektor ini, photodiode hanya mampu menerima sinyal dari *infrared* maksimal sejauh 40 cm.

#### **4.2.2. Pengujian LCD**

##### **4.2.2.1. Tujuan**

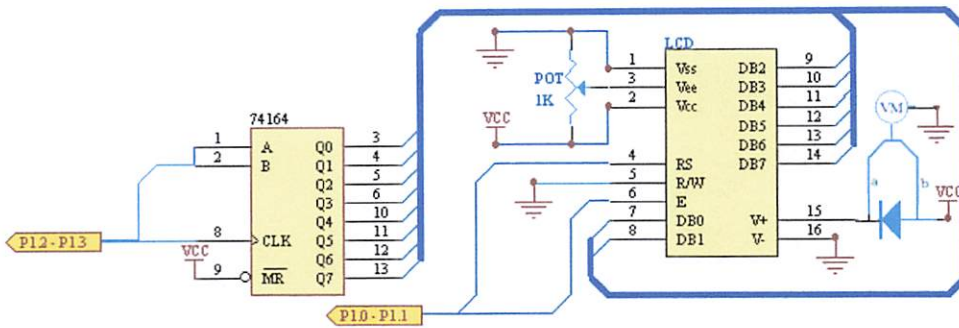
Adapun tujuan dari pengujian rangkaian ini untuk mengetahui kondisi keluaran LCD yaitu sebagai tampilan, juga mencatat nilai tegangan yang masuk pada LCD sebelum dan sesudah melewati diode.

##### **4.2.2.2. Alat-alat yang digunakan**

- LCD
- Rangkaian mikrokontroler AT 89S51.
- Catu daya.
- Voltmeter digital

##### **4.2.2.3. Prosedur pengujian**

1. Menyusun rangkaian pengujian seperti pada gambar 4.7.

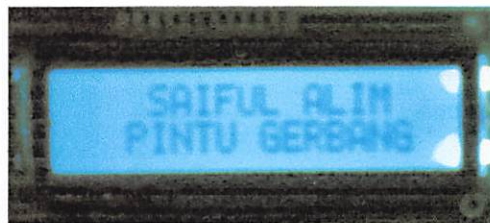


**Gambar 4.5** Rangkaian Pengujian LCD

1. Membuat *software* pengujian rangkaian LCD, program ini berisi inialisasi *mikrokontroler* dan LCD.
3. Mengaktifkan catu daya.
4. Mengoperasikan program dan hasil keluaran akan ditunjukkan pada layar penampil kristal cair.
5. Mengukur besarnya tegangan awal yang masuk pada LCD dan tegangan setelah lewat pada dioda.

#### 4.2.2.4. Hasil pengujian

Dari hasil pengujian maka didapatkan tampilan seperti yang terlihat pada gambar berikut ini :



**Gambar 4.6** Hasil Pengujian LCD

**Tabel 4.5** Hasil Pengujian Rangkaian LCD

No	Tegangan Awal LCD (Volt)	Tegangan Setelah Melewati Dioda (Volt)
1	4,87	4,25

**Gambar 4.7** Pengukuran Tegangan Awal LCD**Gambar 4.8** Pengukuran Tegangan Setelah Melewati Dioda.

### 4.2.3 Pengujian Modul FT232BM

#### 4.2.3.1. Tujuan

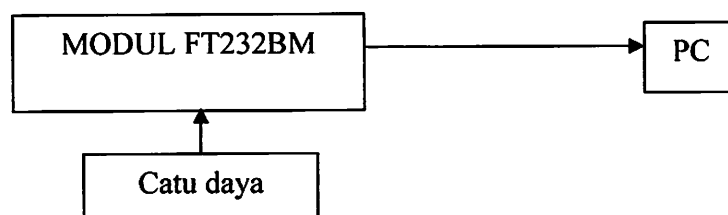
Tujuan dari pengujian ini adalah untuk mengetahui apakah modul USB bisa digunakan untuk mengirim dan menerima data atau tidak.

#### 4.2.3.2 Alat Yang Digunakan

- Modul FT232BM
- Catu daya
- PC
- Kabel USB
- Jumper

#### 4.2.3.3. Prosedur Pengujian

Adapun prosedur pengujianya dilakukan dengan cara memilih *COM* yang dideteksi oleh PC kemudian mencoba melakukan pengiriman data melalui program sederhana yang telah dibuat ,data akan dikirim melalui kabel USB menuju Modul USB dengan menggunakan indikator apabila data di kirim dari PC ke USB maka led indikator pada modul (Rx) akan menyala dan sebaliknya pada saat data dikirim kembali dari USB ke PC maka indikator pada modul (Tx) akan menyala,dengan prosedur pengujian seperti dibawah ini.



**Gambar 4.9** Blok Pengujian Modul FT232BM

Menghubungkan rangkaian Modul USB ke PC menggunakan kabel USB.

- Menginstall driver modul FT232BM.
- Membuka program pada delphi



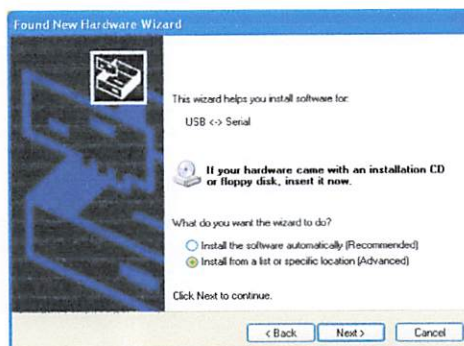
1. Untuk melakukan proses penginstallan driver USB maka kita menghubungkan modul dengan PC menggunakan kabel USB seperti gambar 4.12 yaitu kabel konektor USB tipe A terhubung ke PC dan konektor tipe B terhubung pada Modul USB :
2. Sebelum melakukan percobaan, terlebih dahulu kita menginstal driver Modul USB tersebut yaitu :

- Pada saat kabel konektor USB dihubungkan pada PC maka PC akan mendeteksi adanya perangkat baru seperti pada gambar 4.10:



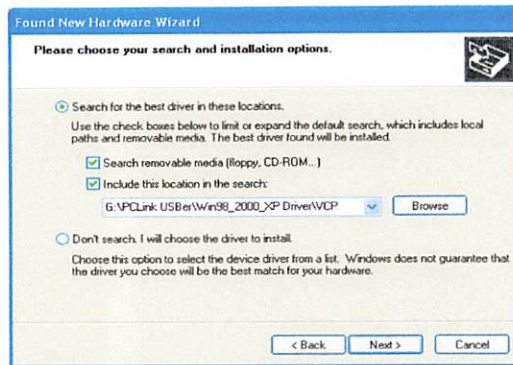
**Gambar 4.10** PC Mendeteksi *Hardware* baru

- Setelah terdeteksi maka perses selanjutnya menekan Button *Next*,setelah button ditekan maka akan muncul seperti gambar 4.11 Perintah tersebut merupakan pemberitahuan apakah penginstallan dilakukan secara otomatis atau manual,tetapi pada saat menginstall dipilih yang manual:



**Gambar 4.11** Perintah Untuk Memilih Cara Peng-*install*-an

- Proses selanjutnya saat *button Next* ditekan maka PC akan menampilkan perintah untuk memberitahu dimana lokasi driver USB tersebut seperti gambar 4.12 dan gambar 4.13 :



**Gambar 4.12** Perintah Memberitahu Lokasi *Driver* Modul USB



**Gambar 4.13** Lokasi *Driver*

- Apabila proses diatas telah selesai maka selanjutnya adalah proses peng-*install-an* dengan menekan *button Next* yang selanjutnya PC akan memproses *driver* tersebut seperti gambar 4.14 dan setelah proses *transfer* selesai maka akan tampil bahwa *driver* baru telah selesai di *install* seperti gambar 4.15.



**Gambar 4.14** Proses *Transfer Driver* USB



**Gambar 4.15** Proses Peng-*install-an* Selesai

4. Setelah Proses peng-*install-an* selesai maka untuk melakukan pengujian terhadap modul USB, digunakan sebuah program Delphi sederhana yang dibuat agar dapat digunakan untuk menguji modul USB tersebut.

Program Uji Komunikasi Modul USB menggunakan Delphi :

```

procedure TForm1.Button1Click(Sender: TObject);
begin
  ComPort1.WriteStr('gerbang')
end;

```

```

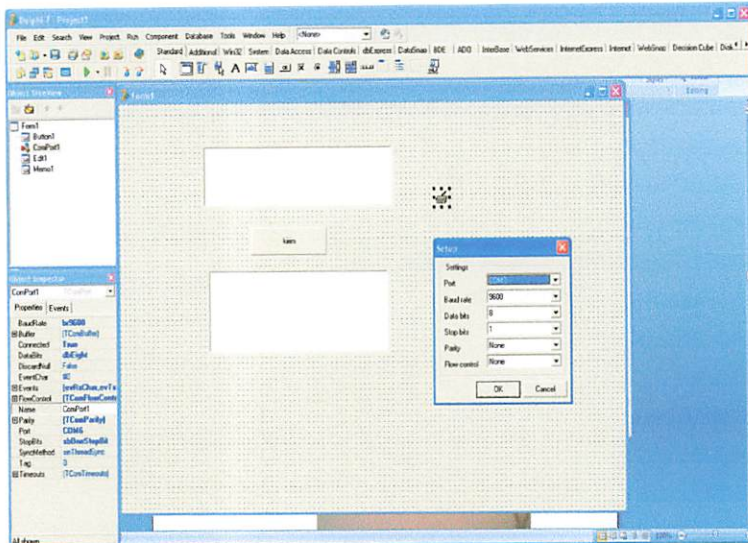
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data:string;
begin
ComPort1.ReadStr(data,count);
Memo1.Text:=Memo1.Text+data;
end;
end.

```

Program diatas sebagai percobaan mengirim data melalui PC ke modul USB di mana komponen yang digunakan pada delphi adalah

- *ComPort* : komponen ini digunakan sebagai koneksi antara program Delphi dengan modul USB dan pada percobaan ini PC mendeteksi COM 3 untuk melakukan pengiriman
- Edit : Komponen ini digunakan sebagai input pada text memo.
- Memo : Komponen ini berfungsi untuk menampilkan data.
- Button : Komponen ini merupakan tombol untuk mengirim data

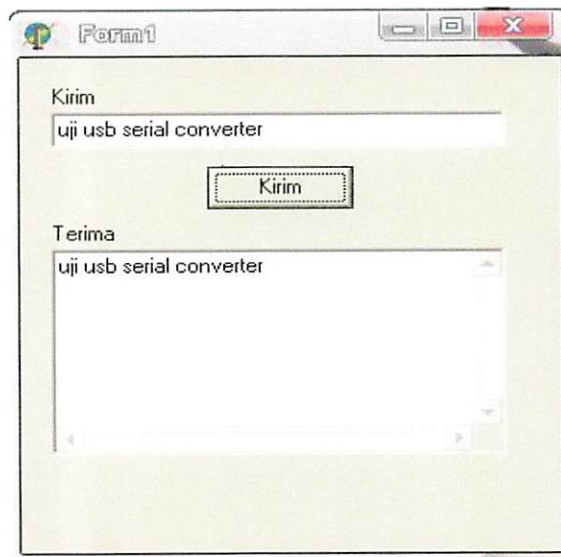
Sebelum program dijalankan terlebih dahulu *setting* COM yang dideteksi oleh PC seperti gambar 4.16 setelah itu program di-*run* maka tampilan pada saat program dijalankan seperti gambar 4.17 dan nyala indikator TX dan RX pada modul FT232BM saat data dikirim seperti pada Gambar 4.18.



**Gambar 4.16** Tampilan Delphi Saat *Setting* Komponen *Com Port*.

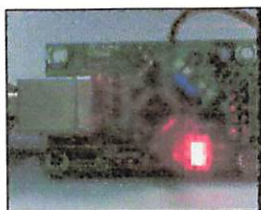
#### 4.2.3.4 Hasil Pengujian

- Data yang dikirim PC berupa tulisan “uji usb serial converter” maka data yang diterima oleh PC pun sama “uji usb serial converter”. Terlihat seperti pada gambar 4.17



**Gambar 4.17** Tampilan Saat Program Dijalankan.

- Pada saat terjadi pengiriman dan penerimaan data oleh PC maka led indikator pada modul FT232BM akan menyala. Terlihat seperti gambar 4.18



**Gambar 4.18** Nyala Led Indikator pada Modul FT232BM

#### **4.2.4. Pengujian Keseluruhan Alat**

##### **4.2.4.1. Tujuan**

Adapun tujuan dari pengujian seseluruhan alat adalah untuk mengetahui apakah alat yang di buat sudah bekerja atau belum.

##### **4.2.4.2. Prosedur pengujian**

1. Mengaktifkan alat dengan menekan tombol ON/OFF
2. Menghidupkan PC sebagai tempat database
3. Menekan tombol keypad untuk proses masukan ID dan Password
4. Memberi halangan pada sinar *infrared* untuk mengetahui bahwa detektor sudah bekerja
5. Mengamati proses yang terjadi

##### **4.2.4.3. Hasil pengujian**

Ketika masukan ID dan password diterima, maka didapatkan tampilan pada PC seperti yang terlihat pada gambar berikut ini :



**Gambar 4.19** Tampilan PC Saat ID dan Password Diterima



**Gambar 4.20** Model Pintu Gerbang

Tabel 4.5 Spesifikasi Model Pintu Gerbang

Spesifikasi Model Pintu Gerbang	
Berat	4,5 kg
Panjang	73 cm
Lebar	30 cm
Tinggi	36 cm
Tinggi Pintu	35 cm
Lebar Pintu	20 cm
Panjang Pintu	45 cm
Detektor	<i>Infrared, Photodiode, Limit Switch</i>
Penggerak Gerbang	Motor DC
Input Data	Keypad 4x4
Tampilan	LCD
Pengolah Data	Mikrokontroler AT89S51



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian dan evaluasi dari perancangan dan pembuatan model sistem keamanan pintu gerbang pada pabrik berbasis mikrokontroler AT89S51 dikomunikasikan ke pc melalui usb ini maka dapat diambil kesimpulan sebagai berikut:

1. Pada saat pengujian modul FT232BM maka pada LED indikator akan menyala (berkedip) ketika terjadi proses pengiriman data.
2. Dari hasil pengujian, pada saat konektor kabel USB tipe A dihubungkan ke PC dengan *port* USB yang berbeda maka PC akan mendeteksi COM yang berbeda.
3. Dalam pengujian detektor, photodiode hanya mampu menerima sinyal dari *infrared* maksimal sejauh 40 cm.
4. Nilai tegangan keluaran photodiode saat tidak terhalang adalah sebesar 1,3 V dan ketika terhalang sebesar 4,34 V.
5. Error tegangan keluaran photodiode saat tidak terhalang adalah 18,4% dan error tegangan keluaran photodiode saat terhalang adalah 5,7%.

#### **5.2. Saran**

Semoga pada perancangan dan pembuatan model sistem keamanan pintu gerbang pada pabrik berbasis mikrokontroler AT89S51 dikomunikasikan ke pc melalui usb ini nantinya dapat dikembangkan lagi menjadi sebuah sistem dengan

tingkat keamanan yang lebih tinggi untuk mengatasi ketidakamanan pada proses keluar masuknya kendaraan dalam sebuah pabrik.

## DAFTAR PUSTAKA

- [1] [www.Innovative\\_electronics.com](http://www.Innovative_electronics.com).
- [2] Sutawi, Dwi, "I/O BUS & Motherboard" Andi, Yogyakarta, 2004.
- [3] Data Sheet AT89S51
- [4] Data Sheet LCD M1632
- [5] Data Sheet IR LED HIR4254C
- [6] Hafindo, Pelatihan Microcontroller MCS-51 Programming and Interfacing, Hafindo Electronic & Education, Malang, 2001.
- [7] Putra.E.A,2002,"*Belajar Mikrokontroller AT89C51/52/53*",Yogyakarta,Gava Media
- [8] [www.en.wikipedia.org](http://www.en.wikipedia.org)
- [9] Usb Complete - Everything You Need To Develop Custom Usb Peripherals (3. edition) by Jan Axelson (2005)
- [10] Data Sheet 74LS14
- [11] Data Sheet FT232BM

# LAMPIRAN



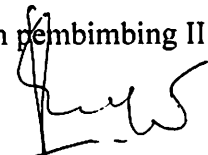
## FORMULIR BIMBINGAN SKRIPSI

Nama : Saiful Alim  
Nim : 03.17.032  
Masa Bimbingan : 15 Desember 2007 s/d 15 Juni 2008  
Judul Skripsi : Perencanaan dan Pembuatan Prototipe Sistem Keamanan Pintu Gerbang Pada Pabrik Berbasis Mikrokontroler AT89S51 Dikomunikasikan Ke PC Melalui USB

No	Tanggal	Uraian	Paraf Pembimbing
1	10/1/2008	Revisi Bab I : latar belakang, Bab II : no urut gambar ,	
2		sumber yang jelas, perbaiki kata-kata.	
3	2/2/2008	Bab III : Diagram blok rangkaian, vcc pada rangkaian LCD	
4		kasih dioda, pelajari teori penunjang, flowchart.	
5	11/2/2008	Pengujian sensor kasih gambar, gambar alat, hasil pengujian	
6		LCD , spesifikasi alat	
7	16/2/2008	Kesimpulan : hasil pengujian	
8	18/2/2008	Pelajari semua teori penunjang yang dipakai	
9	20/2/2008	ACC Seminar Hasil	
10	4/3/2008	ACC Ujian Kompre	

Malang.

Dosen pembimbing II

  
**Ir. Eko Nurcahyo**  
NIP.Y.1028700172

Form S-4b



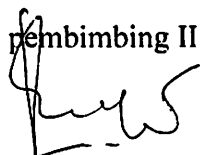
**FORMULIR BIMBINGAN SKRIPSI**

Nama : Saiful Alim  
Nim : 03.17.032  
Masa Bimbingan : 15 Desember 2007 s/d 15 Juni 2008  
Judul Skripsi : Perencanaan dan Pembuatan Prototipe Sistem Keamanan Pintu Gerbang Pada Pabrik Berbasis Mikrokontroler AT89S51 Dikomunikasikan Ke PC Melalui USB

No	Tanggal	Uraian	Paraf Pembimbing
1	10/1/2008	Revisi Bab I : latar belakang, Bab II : no urut gambar ,	
2		sumber yang jelas, perbaiki kata-kata.	
3	2/2/2008	Bab III : Diagram blok rangkaian, vcc pada rangkaian LCD	
4		kasih dioda, pelajari teori penunjang, flowchart.	
5	11/2/2008	Pengujian sensor kasih gambar, gambar alat, hasil pengujian	
6		LCD , spesifikasi alat	
7	16/2/2008	Kesimpulan : hasil pengujian	
8	18/2/2008	Pelajari semua teori penunjang yang dipakai	
9	20/2/2008	ACC Seminar Hasil	
10	4/3/2008	ACC Ujian Kompre	

Malang.

Dosen pembimbing II

  
**Ir. Eko Nurcahyo**  
NIP.Y.1028700172

Form S-4b



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

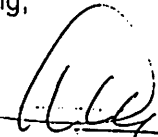
Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : SAIFUL ALINI  
NIM : 0317032  
Perbaikan meliputi :

# judul di perbaiki dan perancangan  
(Pantolan → kawat)

# Hal 47, Rangkaian Driver.

Malang,

 15/3/2018  
(Saiful Alini)



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

## Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : SAIFUL ALIM  
NIM : 0317032  
Perbaikan meliputi :

~. Abstrak ?

~. Tujuan ?


~. Pengujian & Error ?

~. Kesimpulan ?

~. Kata<sup>2</sup> Tugas Akhir hal 3.

~. Perhit. Rangk. Reset &.

Matang,

(  )





FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Saiful Alim  
NIM : 03.17.032  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Masa Bimbingan : 15 Desember 2007 s/d 15 Juni 2008  
Judul Skripsi : Perencanaan Dan Pembuatan Model Sistem Keamanan Pintu Gerbang Pada Pabrik Berbasis Mikrokontroler AT89S51 Dikomunikasikan Ke PC Melalui USB.

Tanggal	Uraian	Paraf
Penguji I 15 Maret 2008	❖ Judul Disesuaikan Dengan Perancangan (Prototipe → Model)	
	❖ Hal 47, Rangkaian Driver	
Penguji II 15 Maret 2008	❖ Abstrak	 31/03/08
	❖ Tujuan	
	❖ Pengujian & Error	
	❖ Kesimpulan	
	❖ Kata-kata Tugas Akhir, Hal 3	
	❖ Perhitungan Rangkaian Reset	

Mengetahui,

Dosen Pembimbing I

(Joseph Dedy Irawan, ST, MT)  
NIP.132315178

Dosen Pembimbing II

(Ir. Eko Nurcahyo)  
NIP.Y.1028700172

Dosen Penguji,

Penguji I

(Ir. Sidik Noertjahjono, MT)  
NIP.Y. 10287700167

Penguji II

(Ir. Teguh Herbasuki, MT)  
NIP.Y. 1038900209



```
#include <at89x51.h>
#include "lcdku.c"
#include "myser1.c"
```

```
#define putus P3_7
#define putar P3_6
#define lstutup P3_4
#define lsbuka P3_5
#define sorang P3_3
```

```
unsigned char i,k[20],a;
char mn,jm,dt;
```

```
#define MatrikX1 P2_4
#define MatrikX2 P2_5
#define MatrikX3 P2_6
#define MatrikX4 P2_7
#define MatrikY1 P2_0
#define MatrikY2 P2_1
#define MatrikY3 P2_2
#define MatrikY4 P2_3
```

```
char Tombolnya()
{
```

```
    MatrikX1 = 0 ;
    MatrikX2 = 1 ;
    MatrikX3 = 1 ;
    MatrikX4 = 1 ;
```

```
    switch ( P2 & 0x0F )
```

```
    {
    case 0x0E:
        while(MatrikY1==0){;}
        return '1';
        break;
```

```
    case 0x0D:
        while(MatrikY2==0){;}
        return '4';
        break;
```

```
    case 0x0B:
        while(MatrikY3==0){;}
        return '7';
        break;
```

```
    case 0x07:
        while(MatrikY4==0){;}
        return 'N';
        break;
```

```
    }
    MatrikX1 = 1 ;
    MatrikX2 = 0 ;
    MatrikX3 = 1 ;
    MatrikX4 = 1 ;
```

```
    switch ( P2 & 0x0F )
```

```
    {
    case 0x0E:
        while(MatrikY1==0){;}
        return '2';
        break;
```

```
    case 0x0D:
        while(MatrikY2==0){;}
        return '5';
        break;
```

```
    case 0x0B:
        while(MatrikY3==0){;}
        return '8';
        break;
```

```
    case 0x07:
        while(MatrikY4==0){;}
        return '0';
        break;
```

```
    }
```

```

MatrikX1      = 1      ;
MatrikX2      = 1      ;
MatrikX3      = 0      ;
MatrikX4      = 1      ;
switch ( P2 & 0x0F )
{
case 0x0E:
    while(MatrikY1==0){;}
    return '3';
    break;
case 0x0D:
    while(MatrikY2==0){;}
    return '6';
    break;
case 0x0B:
    while(MatrikY3==0){;}
    return '9';
    break;
case 0x07:
    while(MatrikY4==0){;}
    return 'E';
    break;
}
MatrikX1      = 1      ;
MatrikX2      = 1      ;
MatrikX3      = 1      ;
MatrikX4      = 0      ;
switch ( P2 & 0x0F )
{
case 0x0E:
    while(MatrikY1==0){;}
    return 'R';
    break;
case 0x0D:
    while(MatrikY2==0){;}
    return 'M';
    break;
case 0x0B:
    while(MatrikY3==0){;}
    return 'U';
    break;
case 0x07:
    while(MatrikY4==0){;}
//return 'D';
return '.';
    break;
default:
    return 'z';
    break;
}
}

void key()
{
pintu=Tombolnya();
while (pintu=='Z') pintu=Tombolnya();
}
char rkey()
{
pintu=Tombolnya();
while (pintu=='Z') pintu=Tombolnya();
return pintu;
}
unsigned char kon(unsigned char n)
{
if (n <= 0x3a) return(n-0x30);
else return(n-0x37);
/*else if(n=='A') return(0xa);
else if(n=='B') return(0xb);
*/
}

```

```

    else if(n=='c') return(0xc);
    else if(n=='d') return(0xd);
    else if(n=='e') return(0xe);
    else if(n=='f') return(0xf);
    else return(0);*/
}

void cetakid()
{
    a=0; i=0;
    while(i<6)
    {
        key();
        a=pintu;
        k[i]=pintu;i++;

        dataout();
    }
}

void cetakpass()
{
    a=0;
    while((a!='E'))
    {
        key();
        a=pintu;
        k[i]=pintu;i++;
        if(i==16) i=15;
        else
        {
            pintu='*';
            dataout();
        }
    }
}

void kirimkom()
{
    i=0;pintu=0;
    while(pintu!='E')
    {
        pintu=k[i];
        PutChar(pintu);
        i++;
    }
}

void buka()
{
    putar=0;delay(500);putus=1;
    while(!sbuka==1);
    putus=0;
}

void tutup()
{
    putar=1;delay(500);putus=1;
    while(!stutup==1);
    putus=0;
}

//-----
// Program Utama
// -----
void main ()
{
    /* Begin of Main
    //TMOD=0x10;
    initser(0xfd);
    initlcd();
    delay(3000);
    EA=0;
    Send_Text("coba");
    putus=0;

```

```
tutup();
    while(1)/* Pengulangan Loop tanpa henti
        {/* Begin of while

        hapus();
            cetak(1,1,"ID  :");
            cetakid();
        cetak(2,1,"Pass:");
        cetakpass();
        kirimkom();
        hapus();
        a=GetChar();
        if(a=='s') {cetak(1,1,"Salah Password");
                    }
        else if(a=='b') {cetak(1,1,"Silahkan Masuk");
                        buka();
                        while(sorang==1);
                        delay(500);
                        while(sorang==0);
                        delay(3000);
                        tutup();
                    }

        delay(2000);

        }/* End of while
}/* End of Main
```

```

#ifndef lcdku_C
#define lcdku_C

#define e                P1_1
#define rs                P1_0
#define con_lcd          P1_2
#define data_lcd         P1_3

unsigned char pintu;

void tunda1ms()
{
    _asm

Loop:   djnz    R7,Loop
        djnz    R6,Liip
_endasm;
}
void time1ms()
{
    int i;
    for (i=0; i<150; i++)
        {;}
}

void delay(int n)
{
    int i;
    for (i=0; i< n ; i++)
        tunda1ms();//time1ms();
}

void delayx(int n)
{
    int i;
    for (i=0; i< n ; i++)
        ;
}
void geser()
{
    _asm
        mov a,_pintu
        clr p1.2
a1:    mov     b,#8
        rrc a
        mov p1.3,c
        nop
        nop
        nop
        nop
        nop
        nop
        clr p1.2
        djnz b,a1
_endasm;
}

void callout()
{
    rs=0;
    e=1;
    geser();

    e=0;
}
void dataout()

```

```

{
    rs=1;
    e=1;
    geser();

    e=0;
}

void initlcd()
{
    pintu=0x1;
    callout();
    pintu=0x38;
    callout();
    pintu=0x6;
    callout();
    pintu=0xc;
    callout();
}

void pos(int i,int n)
{
    if(i==1)
        pintu=0x80+n-1;
    else
        pintu=0xc0+n-1;
    callout();
}

void cetak(int i,int n,unsigned char *text)
{
    //a=0;
    //n=0;
    pos(i,n);
    while(*text)
    {
        pintu>(*text++);
        dataout();
        //n=n+1;
        //a=a+1;
    }
}

void hapus()
{
    cetak(1,1,"");
    cetak(2,1,"");
    pos(1,1);
}

#endif

```



```

#ifndef MYSER1_C
#define MYSER1_C

void Init_Serial()
{
    SCON    = 0x50 ;
    PCON    = 0x80 ;
    TMOD    = TMOD | 0x20 ;
    /* TL1 = -3; TH1 = -3; TR1 = 1;    /* 19200bps with 11.059MHz crystal
*/
    ES = 1;
    EA=1;
}
void initser(char baud)
{
    EA=1;

    TMOD=0x20;    //timer 1 mode 2 (autoreload)
    PCON=0x0 ;    // SMOD =0
    SM0=0;
    SM1=1;        //mode 1
    REN=1;        //terima dibolehkan
    TH1=baud;     //kecepatan 9600 bps (nilai reload)
    TL1=baud;     //nilai awal timer
    TR1=1;        // timer dijalankan
}
void PutChar(unsigned char Data_Kirim)
{
    ES    = 0 ;

    while(!TI){;}
    TI    = Lo;
    ES    = 1 ;
}

unsigned char GetChar(void)
{
    while(!RI){;}
    RI    = Lo ;
    //c++;
    return SBUF ;
}

void Send_Text(unsigned char *text)
{
    while( *text )    // while not end of text
    {
        PutChar(*text++);    // write character and increment
    }
    position
}
return;
}

void kirimgps(unsigned char *text)
{
    while( *text!=',' )    // while not end of text
    {
        PutChar(*text++);    // write character and increment
    }
    position
}
return;
}

void kirimasli(unsigned char *text)
{
    while( *text!='?' )    // while not end of text
    {

```

```
        putchar(*text++);        // write character and increment
position    }
return;
}

void enter()
{
    //putchar(0x0D);
    //putchar(0x0A);
    putchar(0x0D);
    putchar(0x0A);
}
#endif
```

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, ExtCtrls, DBCtrls, Menus, Buttons, Mask,  
ExtDlgs, jpeg, CPort, Grids, DBGrids, MPlayer;

type

TForm1 = class(TForm)  
  TbID: TTable;  
  TbMasuk: TTable;  
  Notebook1: TNotebook;  
  Label1: TLabel;  
  Label2: TLabel;  
  Label3: TLabel;  
  Label4: TLabel;  
  TbSlave: TTable;  
  DSID: TDataSource;  
  DSMasuk: TDataSource;  
  DSSlave: TDataSource;  
  MainMenu1: TMainMenu;  
  Karyawan1: TMenuItem;  
  MasukanBaru1: TMenuItem;  
  PerubahanData1: TMenuItem;  
  LihatData1: TMenuItem;  
  DBText2: TDBText;  
  DBText3: TDBText;  
  DBText4: TDBText;  
  Image1: TImage;  
  Label6: TLabel;  
  Label7: TLabel;  
  Label8: TLabel;  
  Label9: TLabel;  
  Label10: TLabel;  
  Label11: TLabel;  
  Label12: TLabel;  
  Label13: TLabel;  
  Label14: TLabel;  
  Label15: TLabel;  
  BitBtn1: TBitBtn;  
  BitBtn2: TBitBtn;  
  Image2: TImage;  
  BitBtn3: TBitBtn;  
  DBEdit1: TDBEdit;  
  DBEdit2: TDBEdit;  
  DBEdit3: TDBEdit;

```
DBEdit4: TDBEdit;
DBEdit5: TDBEdit;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
Image3: TImage;
DBLookupComboBox1: TDBLookupComboBox;
BitBtn4: TBitBtn;
BitBtn5: TBitBtn;
BitBtn6: TBitBtn;
SavePictureDialog1: TSavePictureDialog;
OpenPictureDialog1: TOpenPictureDialog;
DBGrid1: TDBGrid;
ComPort1: TComPort;
BitBtn7: TBitBtn;
DBEdit9: TDBEdit;
DBLookupComboBox2: TDBLookupComboBox;
MPlayer: TMediaPlayer;
GroupBox1: TGroupBox;
Label16: TLabel;
LJAM: TLabel;
LTGL: TLabel;
Timer1: TTimer;
BitBtn8: TBitBtn;
Image4: TImage;
Image5: TImage;
LihatDataUmum1: TMenuItem;
DBGrid2: TDBGrid;
procedure MasukanBaru1Click(Sender: TObject);
procedure PerubahanData1Click(Sender: TObject);
procedure LihatData1Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn5Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn6Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure DBLookupComboBox1Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure FormCreate(Sender: TObject);
procedure BitBtn7Click(Sender: TObject);
procedure DBLookupComboBox2Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure BitBtn8Click(Sender: TObject);
procedure LihatDataUmum1Click(Sender: TObject);
private
  { Private declarations }
public
```

1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025

1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025

1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025

1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025

```

    { Public declarations }
end;

var
    Form1: TForm1;
    gambar,data:string;
    status:integer;
implementation

{$R *.dfm}

procedure TForm1.MasukanBaru1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    TbID.Append;
    Image2.Visible:=false;
end;

procedure TForm1.PerubahanData1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='ubah';
    IF TbID.FieldName('foto').IsNull then
        begin
            ShowMessage('maaf tidak ada foto');
            Image3.Visible:=false;
        end
    else begin
        Image3.Visible:=true;

        Image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'
        +TbID.fieldbyname('nip').AsString+'.jpg');
        end;
    end;

procedure TForm1.LihatData1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='lihat';
    IF TbID.FieldName('foto').IsNull then
        begin
            ShowMessage('maaf tidak ada foto');
            Image3.Visible:=false;
        end
    else begin
        Image3.Visible:=true;

        Image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'
        +TbID.fieldbyname('nip').AsString+'.jpg');
        end;
    end;

```

end;

```
procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  TbID.Edit;
  Image3.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+TbID.fieldbyname('nip').AsString+'.jpg');
  Application.ProcessMessages;
  //ShowMessage(ExtractFilePath(Application.ExeName)+'\gambar\'+TbID.fieldbyname('nip').AsString+'.jpg');
  if gambar<>" then TbID.FieldName('foto').AsString:='ada';
  TbID.Post;
  Notebook1.ActivePage:='lihat';
  gambar:="";
  Image3.Visible:=false;
end;
```

```
procedure TForm1.BitBtn5Click(Sender: TObject);
begin
  TbID.Cancel;
  Notebook1.ActivePage:='lihat';
  gambar:="";
  Image3.Visible:=false;
end;
```

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  Image2.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+TbID.fieldbyname('nip').AsString+'.jpg');
  //ShowMessage(ExtractFilePath(Application.ExeName)+'\gambar\'+TbID.fieldbyname('nip').AsString+'.jpg');
  Application.ProcessMessages;
  if gambar<>" then TbID.FieldName('foto').AsString:='ada';
  TbID.Post;
  Notebook1.ActivePage:='lihat';
  gambar:="";
end;
```

```
procedure TForm1.BitBtn6Click(Sender: TObject);
begin
  TbID.Cancel;
  Notebook1.ActivePage:='lihat';
  gambar:="";
end;
```

```
procedure TForm1.BitBtn3Click(Sender: TObject);
begin
  if OpenPictureDialog1.Execute
```

```

    then begin
        Image2.Visible:=true;
        Image2.Picture.LoadFromFile (OpenPictureDialog1.FileName);
        gambar:='ada';
    end;
end;

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    Image1.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar'+TbID.fieldbyname('nip').AsString+'.jpg');
    if gambar<>" then TbID.FieldName('foto').AsString:='ada';
    TbID.Post;
    Notebook1.ActivePage:='lihat';
    gambar:="";
    TbID.Append;
end;

procedure TForm1.DBLookupComboBox1Click(Sender: TObject);
begin
    IF TbID.FieldName('foto').IsNull then
        begin
            ShowMessage('maaf tidak ada foto');
            Image3.Visible:=false;
        end
    else begin
        Image3.Visible:=true;

        Image3.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar'+TbID.fieldbyname('nip').AsString+'.jpg');
        end;
    end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data1:string;
begin
    ComPort1.ReadStr(data1,count);
    data:=data+data1;
    //ShowMessage(data);
    //ShowMessage(Copy(data,length(data),1));
    //ShowMessage(copy(data,1,6));
    //ShowMessage(copy(data,7,5));
    if Copy(data,length(data),1)='E' then
        begin
            if TbID.FindKey([copy(data,1,6)]) and
            (TbID.FieldName('password').AsString=copy(data,7,5)) and (length(data)=12)
            then
                begin

```



```

IF TbID.FieldName('foto').IsNull then
  begin
    ShowMessage('maaf tidak ada foto');
    Image1.Visible:=false;
  end
else begin
  Image1.Visible:=true;

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'
+TbID.fieldbyname('nip').AsString+'.jpg');
  end;
  Notebook1.ActivePage:='lihat';
// ShowMessage('benar');
  TbMasuk.Append;
  TbMasuk.FieldName('no').AsInteger:=TbMasuk.RecordCount+1;
  TbMasuk.FieldName('nip').AsString:=TbID.FieldName('nip').AsString;

TbMasuk.FieldName('nama').AsString:=TbID.FieldName('nama').AsString;
  TbMasuk.FieldName('jam').AsDateTime:=time;
  TbMasuk.FieldName('tgl').AsDateTime:=date;
  TbMasuk.Post;
  Application.ProcessMessages;
  TbSlave.Refresh;
  Application.ProcessMessages;
  ComPort1.WriteStr('b');
  status:=0;
end
else
  begin
// ShowMessage(IntToStr(status));
  status:=status+1;
// ShowMessage('salah');
  ComPort1.WriteStr('s');
  if status>=3 then
    begin
MPlayer.Close;

MPlayer.FileName:=ExtractFilePath(Application.ExeName)+'TES.MP3';//Concat
(txtFolder.Caption, mp3List.Items.Strings[mp3List.ItemIndex]);
  MPlayer.Open;
  MPlayer.Play;
// ShowMessage('salah >= 3');
  end;
  end;
  data:=";
  end
else if length(data)>12 then data:="
else if data='coba' then data:=";

```

```

end;

procedure TForm1.FormCreate(Sender: TObject);
begin
ComPort1.ShowSetupDialog;
ComPort1.Open;
LJAM.Caption:=TimeToStr(TIME);
LTGL.Caption:=DateToStr(date);
data:="";
gambar:="";
Notebook1.ActivePage:='lihat';
status:=0;
  IF TbID.FieldByName('foto').IsNull then
    begin
      ShowMessage('maaf tidak ada foto');
      Image1.Visible:=false;
    end
  else begin
    Image1.Visible:=true;

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'
+TbID.fieldbyname('nip').AsString+'.jpg');
    end;
end;

procedure TForm1.BitBtn7Click(Sender: TObject);
begin
TbID.Delete;
end;

procedure TForm1.DBLookupComboBox2Click(Sender: TObject);
begin
IF TbID.FieldByName('foto').IsNull then
  begin
    ShowMessage('maaf tidak ada foto');
    Image1.Visible:=false;
  end
else begin
    Image1.Visible:=true;

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'
+TbID.fieldbyname('nip').AsString+'.jpg');
    end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
LJAM.Caption:=TimeToStr(TIME);

```

```
L.TGL.Caption:=DateToStr(date);  
end;
```

```
procedure TForm1.BitBtn8Click(Sender: TObject);  
begin  
  if OpenPictureDialog1.Execute  
  then begin  
    Image3.Visible:=true;  
    Image3.Picture.LoadFromFile (OpenPictureDialog1.FileName);  
    gambar:='ada';  
  end;  
end;
```

```
procedure TForm1.LihatDataUmum1Click(Sender: TObject);  
begin  
  Notebook1.ActivePage:='lihat umum';  
end;  
  
end.
```

## Features

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

## Description

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

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



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

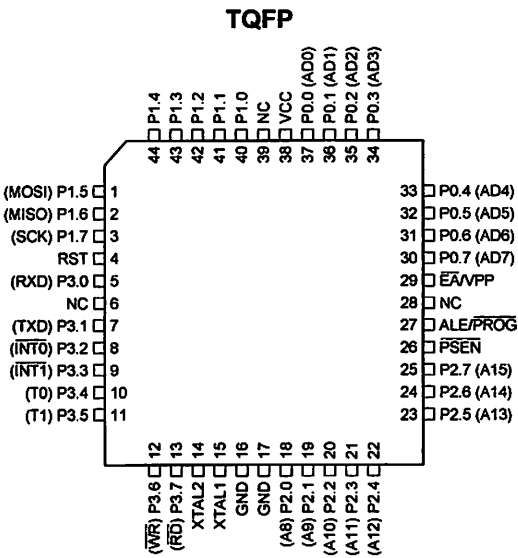
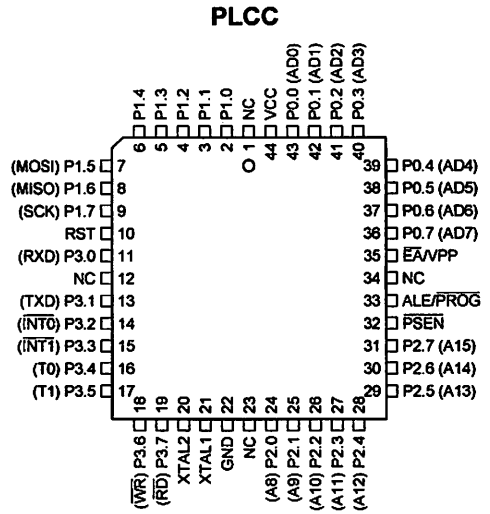
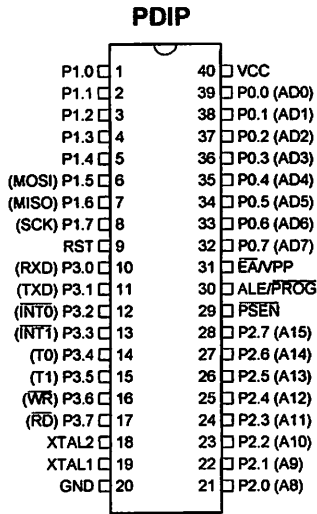
## AT89S51

Rev. 2487A-10/01

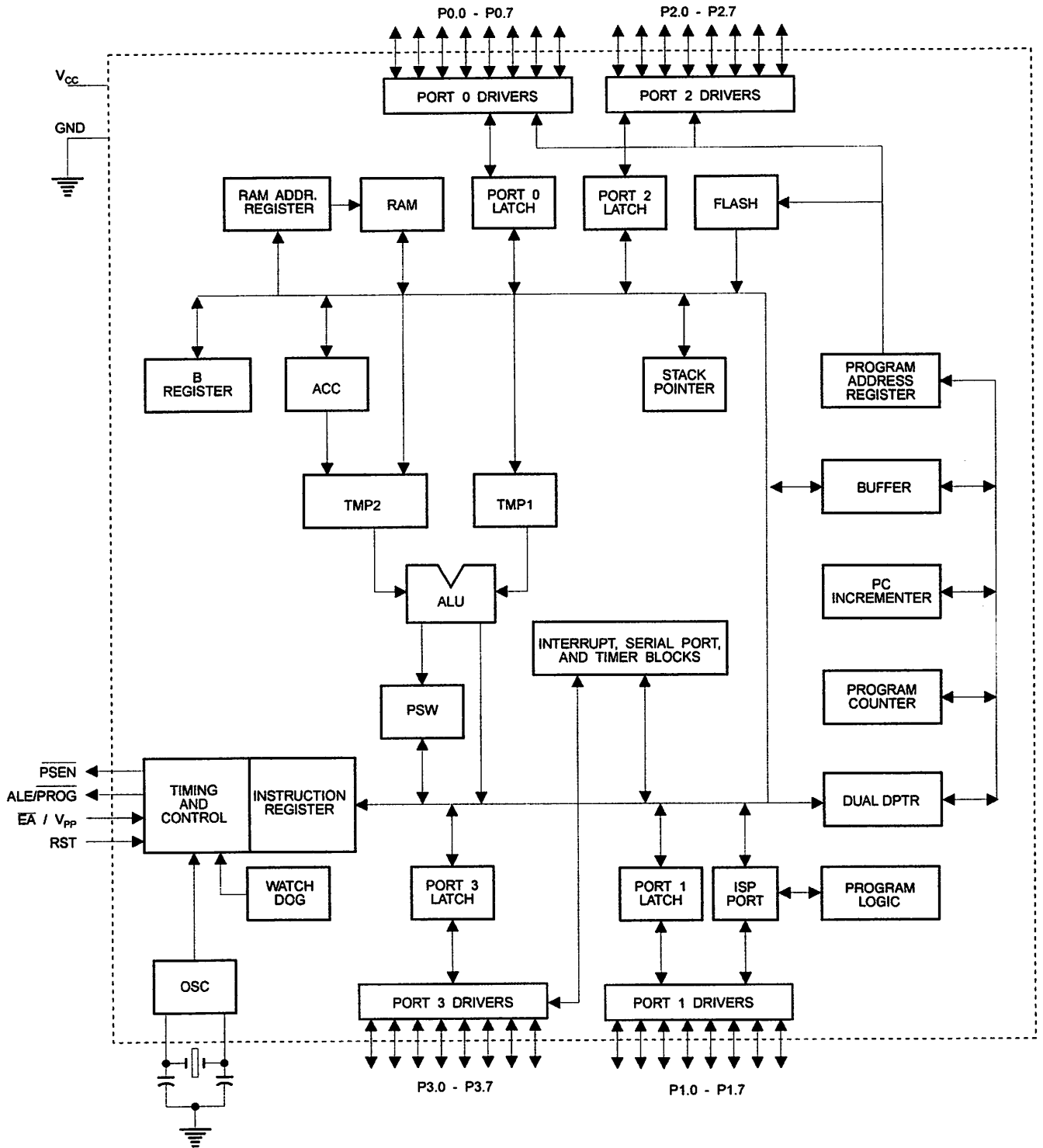




# Pin Configurations



Block Diagram





## Pin Description

**VCC** Supply voltage.

**GND** Ground.

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

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

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

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

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

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

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

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

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

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

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

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

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (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)

**RST**

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

**ALE/PROG**

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

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

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

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

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

**XTAL1**

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

**XTAL2**

Output from the inverting oscillator amplifier







# Special Function Registers

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

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

**Table 1. AT89S51 SFR Map and Reset Values**

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDTRST XXXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

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

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

**Table 2. AUXR: Auxiliary Register**

AUXR		Address = 8EH					Reset Value = XXX00XX0B		
Not Bit Addressable									
		-	-	-	WDIDLE	DISRTO	-	-	DISALE
Bit		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 = XXXXXX0B								
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	DPS
	-	-	-	-	-	-	-	0
								1
-	Reserved for future expansion							
DPS	Data Pointer Register Select							
	DPS							
	0	Selects DPTR Registers DP0L, DP0H						
	1	Selects DPTR Registers DP1L, DP1H						

## Memory Organization

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

## Program Memory

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

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

## Data Memory

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

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

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

## Using the WDT

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

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

## WDT During Power-down and Idle

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

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

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

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

## UART

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

## Timer 0 and 1

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

## Interrupts

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

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

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

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



**Table 4. Interrupt Enable (IE) Register**

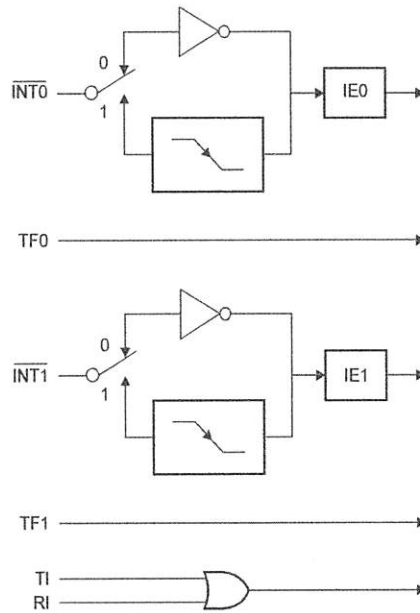
(MSB)				(LSB)			
EA	-	-	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.  
 Enable Bit = 0 disables the interrupt.

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

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

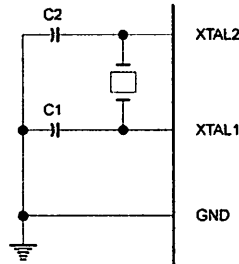
**Figure 1. Interrupt Sources**



**Oscillator Characteristics**

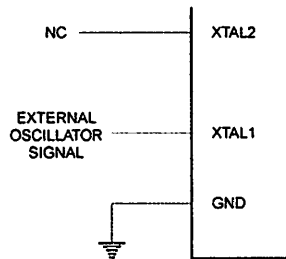
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected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

**Figure 2. Oscillator Connections**



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

**Figure 3. External Clock Drive Configuration**



**Idle Mode**

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

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

**Power-down Mode**

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





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

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

## Program Memory Lock Bits

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

Table 6. Lock Bit Protection Modes

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

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

## Programming the Flash – Parallel Mode

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

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

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

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise  $\overline{EA}/V_{pp}$  to 12V.
5. Pulse ALE/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  $\mu$ 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 RDY/BSY output signal. P3.0 is pulled low after ALE goes high during programming to indicate BUSY. P3.0 is pulled high again when programming is done to indicate READY.

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

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

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

**Chip Erase:** In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/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.

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.

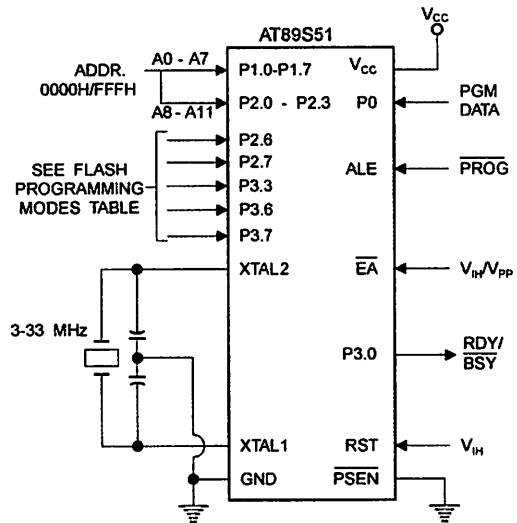
To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

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.

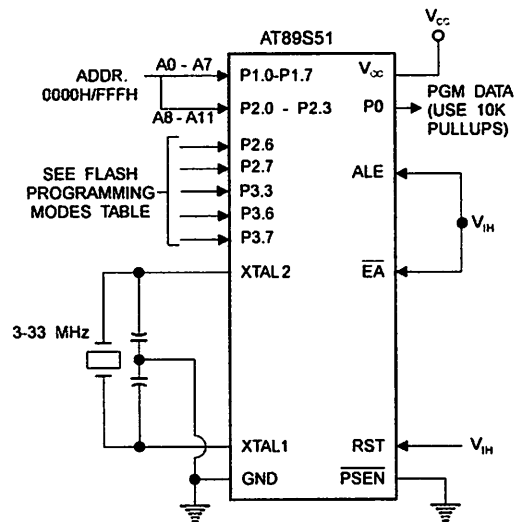




**Figure 4. Programming the Flash Memory (Parallel Mode)**



**Figure 5. Verifying the Flash Memory (Parallel Mode)**



## Flash Programming and Verification Characteristics (Parallel Mode)

$T_A = 20^\circ\text{C}$  to  $30^\circ\text{C}$ ,  $V_{CC} = 4.5$  to  $5.5\text{V}$

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

Figure 6. Flash Programming and Verification Waveforms - Parallel Mode

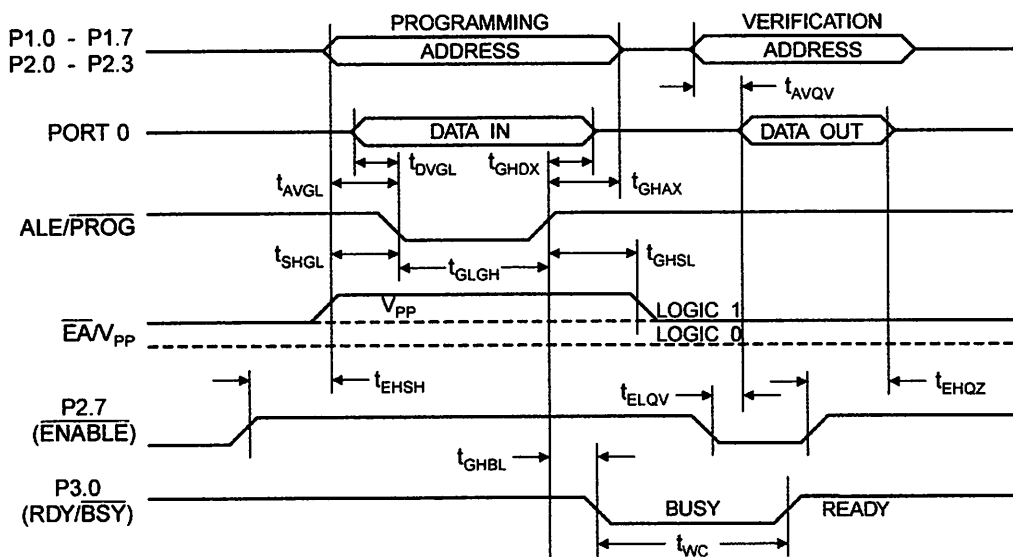
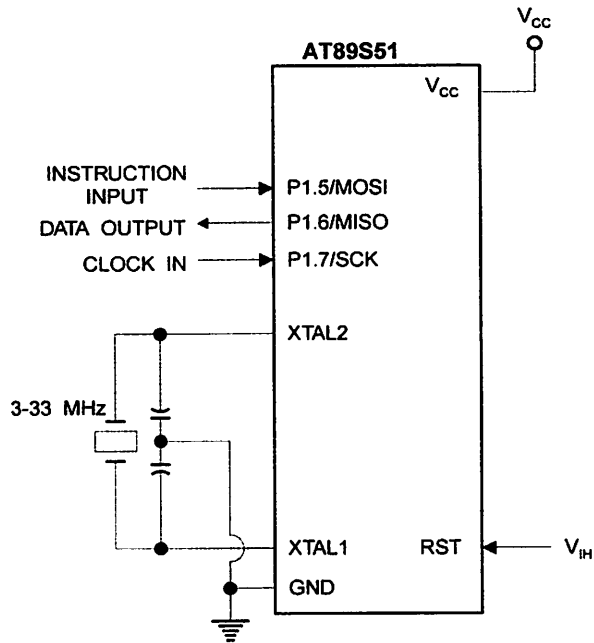
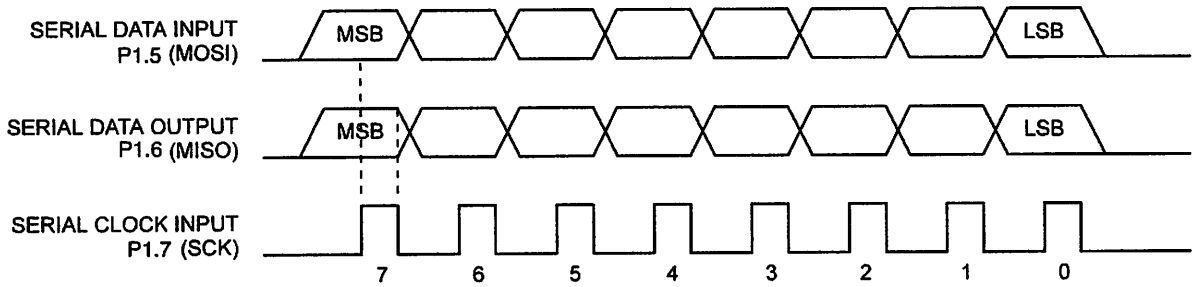


Figure 7. Flash Memory Serial Downloading



### Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms



**Table 8. Serial Programming Instruction Set**

Instruction	Instruction Format				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
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	7 DDDD DDDD DDDD DDDD	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	7 DDDD DDDD DDDD DDDD	Write data to Program memory in the byte mode
Write Lock Bits <sup>(2)</sup>	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LB LB xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes <sup>(1)</sup>	0010 1000	xxx A2 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

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

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

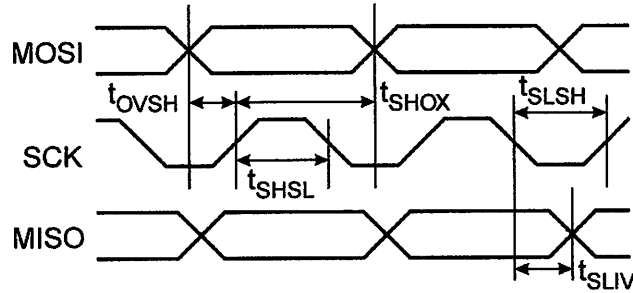


Table 9. Serial Programming Characteristics,  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 4.0 - 5.5\text{V}$  (Unless Otherwise Noted)

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





## Absolute Maximum Ratings\*

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

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

## DC Characteristics

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

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

Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

Maximum  $I_{OL}$  per port pin: 10 mA

Maximum  $I_{OL}$  per 8-bit port:

Port 0: 26 mA      Ports 1, 2, 3: 15 mA

Maximum total  $I_{OL}$  for all output pins: 71 mA

If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

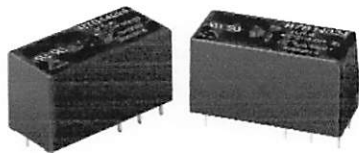
2. Minimum  $V_{CC}$  for Power-down is 2V.

## AC Characteristics

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

### External Program and Data Memory Characteristics

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



# RT series (DC Coil)

## 16 Amp PC Board Miniature Relay

File E22575  
File LR15734  
NR 6106

Users should thoroughly review the technical data before selecting a product part number. It is recommended that users also seek out the pertinent approvals files of the agencies/laboratories and review them to ensure the product meets the requirements for a given application.

### Features

- SPST through DPDT contact arrangements.
- Immersion cleanable and flux tight versions available.
- VDE 10mm spacing, 5kV dielectric, coil to contacts.
- UL Class F (155°C) coil insulation system.
- Conforms to UL 508, 1873, 353 and 1950.
- Low profile; 15.7mm height.
- Sensitive coil; 400mW.
- Withstand surge voltage of 10,000V.
- Potter & Brumfield or Schrack brand.

### Contact Data

**Arrangements:** 1 Form A (SPST-NO) Wiring Diagram Code 1, 2, 3.  
2 Form A (DPST-NO) Wiring Diagram Code 5.  
1 Form C (SPDT) Wiring Diagram Code 1, 2, 3.  
2 Form C (DPDT) Wiring Diagram Code 5.

**Material:** Silver-nickel 90/10.

**Minimum Load:** 12V/100mA.

**Expected Mechanical Life:** 10 million operations.

**Initial Contact Resistance:** 100 milliohms max @ 1A 12VDC.

Designed to meet UL/CSA/VDE ratings with relay properly vented. Remove vent nib after soldering and cleaning.

### UL/CSA/VDE Ratings @ 25°C

Code	NO/NC Load	Type	Operations
1	10A/10A @ 277VAC	Resistive/GP	100K
	10A/10A @ 30VDC	Resistive	100K
	12A/12A @ 250VAC	Resistive/GP	30K
	12A/12A @ 30VDC	Resistive	30K
	3/4 HP @ 480VAC*	Motor	6K
	1/2 HP @ 240VAC*	Motor	6K
	1/3 HP @ 120VAC*	Motor	6K
	48 LRA/10 FLA @ 240VAC*	Motor	30K
	TV-3 @ 120VAC*	Tungsten	25K
	A300, 720VA @ 240VAC*	Pilot Duty	30K
3	16A/16A @ 250VAC	Resistive/GP	50K
	20A/20A @ 277VAC	Resistive/GP	30K
	20A/20A @ 24VDC	Resistive	30K
	16A/16A @ 30VDC	Resistive	30K
	1 HP @ 480VAC*	Motor	6K
	1 HP @ 240VAC*	Motor	6K
	1/2 HP @ 120VAC*	Motor	6K
	60 LRA/10 FLA @ 250VAC*	Motor	30K
	TV-5 @ 120VAC*	Tungsten	25K
	A300, 720VA @ 240VAC*	Pilot Duty	30K
B300, 360VA @ 240VAC**	Pilot Duty	30K	
5	8A/8A @ 277VAC	Resistive/GP	100K
	8A/8A @ 30VDC	Resistive	100K
	10A/10A @ 250VAC	Resistive/GP	30K
	10A/10A @ 30VDC	Resistive	30K
	1/2 HP @ 240VAC*	Motor	6K
	1/4 HP @ 120VAC*	Motor	6K
	34.8 LRA/6 FLA @ 120VAC*	Motor	30K
	17.4 LRA/5 FLA @ 240VAC*	Motor	30K
	B300, 360VA @ 240VAC*	Pilot Duty	30K
	TV-3 @ 120VAC*	Tungsten	25K

\* Form A only

\*\* Form B only

### Coil Data @ 25°C

**Voltage:** 5 to 110VDC.

**Nominal Power @ 25°C:** 400mW.

**Duty Cycle:** Continuous.

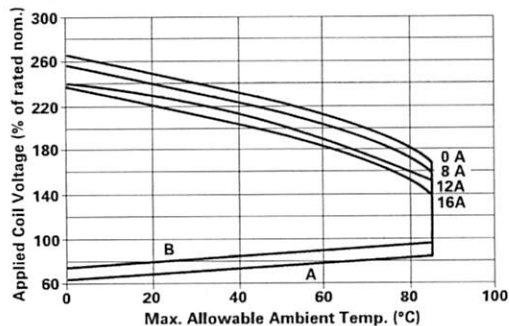
**Initial Insulation Resistance:** 10,000 megohms, min., at 25°C, 500VDC and 50% rel. humidity.

**Coil Construction:** UL Class F (155°C).

### Coil Data @ 25°C

Nominal Voltage VDC	DC Resistance in Ohms ±10%	Must Operate Voltage VDC	Nominal Coil Current (mA) - 50/60Hz.
005	62	3.5	80
006	90	4.2	66.7
009	202	6.3	44.4
012	360	8.4	33.3
018	810	12.6	22.2
024	1,440	16.8	16.7
048	5,760	33.6	8.3
060	9,000	42.0	8.0
110	30,250	77.0	4.3

### Max. Ambient Temp. vs. Coil Voltage



A: Coil temperature = Ambient temperature.

B: 110% of nominal coil voltage at rated contact load.

### Operate Data @ 25°C

**Must Operate Voltage(DC):** 70% of nominal.

**Must Release Voltage(DC):** 10% of nominal.

**Operate Time (Excluding Bounce):**

7 ms, typ., 15ms max. at nom. voltage.

**Release Time (Excluding Bounce):**

3 ms, typ., 6ms max. at nom. voltage.

### Environmental Data

**Temperature Range:**

**Storage:** -40°C to +105°C.

**Operating:** -40°C to +85°C at rated current.

**Vibration, Operational**

N.O.: 0.065" (1.65mm) max. excursions from 10 - 55 Hz:

N.C.: 0.032" (0.82mm) max. excursions from 10 - 55 Hz:

with no contact opening >10µs.

### Mechanical Data

**Termination:** Printed circuit terminals.

**Enclosures:** RT 1, 2, 3, 4: Flux-tight, top vented, plastic case.

RT B, C, D, E: Immersion cleanable, plastic case.

**Weight:** 0.35 oz. (10g) approximately.

Dimensions are shown for reference purposes only.

Dimensions are in inches over (millimeters) unless otherwise specified.

Specifications and availability subject to change.

www.tycoelectronics.com  
Technical support:  
Refer to inside back cover.



# 1N4001 - 1N4007

## Features

- Low forward voltage drop.
- High surge current capability.



DO-41

COLOR BAND DENOTES CATHODE

## General Purpose Rectifiers

### Absolute Maximum Ratings\* T<sub>A</sub> = 25°C unless otherwise noted

Symbol	Parameter	Value							Units
		4001	4002	4003	4004	4005	4006	4007	
V <sub>RRM</sub>	Peak Repetitive Reverse Voltage	50	100	200	400	600	800	1000	V
I <sub>F(AV)</sub>	Average Rectified Forward Current, .375 " lead length @ T <sub>A</sub> = 75°C	1.0							A
I <sub>FSM</sub>	Non-repetitive Peak Forward Surge Current 8.3 ms Single Half-Sine-Wave	30							A
T <sub>stg</sub>	Storage Temperature Range	-55 to +175							°C
T <sub>J</sub>	Operating Junction Temperature	-55 to +175							°C

\*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

### Thermal Characteristics

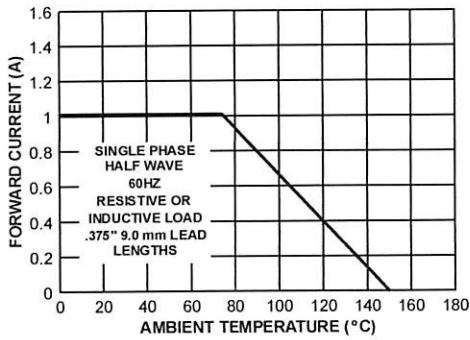
Symbol	Parameter	Value	Units
P <sub>D</sub>	Power Dissipation	3.0	W
R <sub>θJA</sub>	Thermal Resistance, Junction to Ambient	50	°C/W

### Electrical Characteristics T<sub>A</sub> = 25°C unless otherwise noted

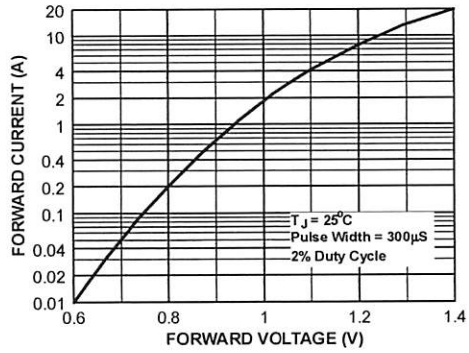
Symbol	Parameter	Device							Units
		4001	4002	4003	4004	4005	4006	4007	
V <sub>F</sub>	Forward Voltage @ 1.0 A	1.1							V
I <sub>rr</sub>	Maximum Full Load Reverse Current, Full Cycle T <sub>A</sub> = 75°C	30							μA
I <sub>R</sub>	Reverse Current @ rated V <sub>R</sub> T <sub>A</sub> = 25°C T <sub>A</sub> = 100°C	5.0 500							μA μA
C <sub>T</sub>	Total Capacitance V <sub>R</sub> = 4.0 V, f = 1.0 MHz	15							pF

Typical Characteristics

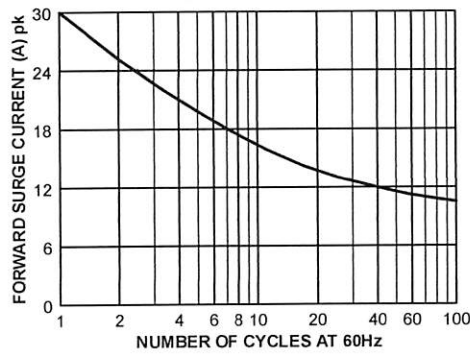
Forward Current Derating Curve



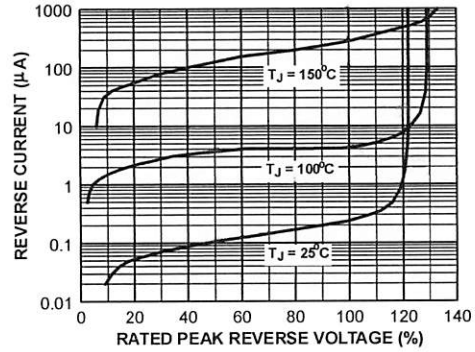
Forward Characteristics



Non-Repetitive Surge Current



Reverse Characteristics



## TRADEMARKS

The following are registered and unregistered trademarks Fairchild Semiconductor owns or is authorized to use and is not intended to be an exhaustive list of all such trademarks.

ACEx™	FACT™	ImpliedDisconnect™	PACMAN™	SPM™
ActiveArray™	FACT Quiet Series™	ISOPLANAR™	POP™	Stealth™
Bottomless™	FAST®	LittleFET™	Power247™	SuperSOT™_3
CoolFET™	FASTR™	MicroFET™	PowerTrench®	SuperSOT™_6
CROSSVOLT™	FRFET™	MicroPak™	QFET™	SuperSOT™_8
DOMETM	GlobalOptoisolator™	MICROWIRE™	QS™	SyncFET™
EcoSPARK™	GTO™	MSX™	QT Optoelectronics™	TinyLogic®
E <sup>2</sup> CMOSTM	HiSeC™	MSXPro™	Quiet Series™	TruTranslation™
EnSigna™	I <sup>2</sup> C™	OCX™	RapidConfigure™	UHC™
Across the board. Around the world.™		OCXPro™	RapidConnect™	UltraFET®
The Power Franchise™		OPTOLOGIC®	SILENT SWITCHER®	VCX™
Programmable Active Droop™		OPTOPLANAR™	SMART START™	

## DISCLAIMER

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

## LIFE SUPPORT POLICY

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

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, or (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

## PRODUCT STATUS DEFINITIONS

### Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative or In Design	This datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	This datasheet contains preliminary data, and supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
No Identification Needed	Full Production	This datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice in order to improve design.
Obsolete	Not In Production	This datasheet contains specifications on a product that has been discontinued by Fairchild semiconductor. The datasheet is printed for reference information only.



# EVERLIGHT ELECTRONICS CO., LTD.

DEVICE NUMBER : DIH-425-032

REV : 1.0

ECN : \_\_\_\_\_

PAGE : 1/8

## 5mm Infrared LED

MODEL NO : HIR4254C

### ■ Features :

- High radiant intensity
- Peak wavelength  $\lambda_p=850\text{nm}$
- View angle  $60^\circ$
- High reliability

### ■ Description :

- EVERLIGHT's Infrared Emitting Diode (HIR4254C) is a high intensity diode, molded in a water clear plastic package.

The device is spectrally matched with phototransistor, photodiode and infrared receiver module.

### ■ Applications :

- Free air transmission system
- Optoelectronic switch
- Infrared remote control units with high power requirement
- Floppy disk drive

PART NO.	CHIP	LENS COLOR
	MATERIAL	
HIR	GaAIAs	Water clear



# EVERLIGHT ELECTRONICS CO., LTD.

DEVICE NUMBER : DIH-425-032

REV : 1.0

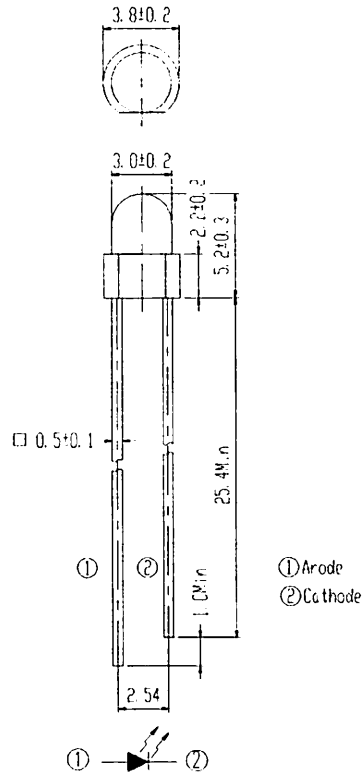
ECN : \_\_\_\_\_

PAGE : 2/8

## 5mm Infrared LED

MODEL NO : HIR4254C

### Package Dimensions :



### Notes :

1. All dimensions are in millimeter.
2. Protruded resin under flange 1.5 mm Max.
3. Lead spacing is measured where the lead emerge from the package.
4. Lens color : Water clear.
5. Above specification may be changed without notice. EVERLIGHT will reserve authority on material change for above specification.
6. These specification sheets include materials protected under copyright of EVERLIGHT corporation . Please don't reproduce or cause anyone to reproduce them without EVERLIGHT's consent.
7. When using this product , please observe the absolute maximum ratings and the instructions for use outlined in these specification sheets. EVERIGHT assumes no responsibility for any damage resulting from use of the product which does not comply with the absolute maximum ratings and the instructions included in these specification sheets.



# EVERLIGHT ELECTRONICS CO., LTD.

DEVICE NUMBER : DIH-425-032REV : 1.0

ECN : \_\_\_\_\_

PAGE : 3/8

## 5mm Infrared LED

MODEL NO : HIR4254C

### Absolute Maximum Ratings at $T_A = 25^\circ\text{C}$

Parameter	Symbol	Rating	Unit	Notice
Continuous Forward Current	$I_F$	50	mA	
Peak Forward Current Pulse width=100 $\mu\text{s}$ , Duty cycle=1%	$I_{FP}$	1.0	A	
Reverse Voltage	$V_R$	5	V	
Operating Temperature	$T_{opr}$	-40 ~ +85	$^\circ\text{C}$	
Storage Temperature	$T_{stg}$	-40 ~ +85	$^\circ\text{C}$	
Soldering Temperature	$T_{sol}$	260	$^\circ\text{C}$	4mm from mold body less than 5 seconds
Power Dissipation at(or below) 25 $^\circ\text{C}$ Free Air Temperature	$P_d$	100	mW	

### Electronic Optical Characteristics :

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Radiant Intensity	$E_e$	2.8	4.5	---	mW/sr	$I_F=20\text{mA}$
		---	20	---		$I_F=100\text{mA}, t_p=100\mu\text{s}, t_p/T=0.01$
		---	250	---		$I_F=1\text{A}, t_p=100\mu\text{s}, t_p/T=0.01$
Peak Wavelength	$\lambda_p$	---	850	---	nm	$I_F=20\text{mA}$
Spectral Bandwidth	$\Delta\lambda$	---	45	---	nm	$I_F=20\text{mA}$
Forward Voltage	$V_F$	---	1.45	1.65	V	$I_F=20\text{mA}$
		---	1.80	2.40		$I_F=100\text{mA}, t_p=100\mu\text{s}, t_p/T=0.01$
		---	4.10	5.25		$I_F=1\text{A}, t_p=100\mu\text{s}, t_p/T=0.01$
Reverse Current	$I_R$	---	---	10	$\mu\text{A}$	$V_R=5\text{V}$
View Angle	$2\theta_{1/2}$	---	60	---	deg	$I_F=20\text{mA}$

LIQUID CRYSTAL DISPLAY MODULE

M 1 6 3 2

USER MANUAL

Seiko Instruments Inc.

**PREFACE**

This manual describes technical informations on functions and instructions of M1632 from Seiko Instruments Inc. Please read this instruction manual carefully to understand all the module functions and make the best use of them. Description details may be changed without notice.

**Revision Record**

<u>Edition</u>	<u>Revision</u>	<u>Date</u>
1	Original	April 1985
2	Completely revised	Jan. 1987

© Seiko Instruments Inc. 1987

Printed in Japan



## 1. GENERAL

### 1.1 General

The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view TN LCD panel and a CMOS LCD drive controller built in. The controller has a built-in character generator ROM/RAM, and display data RAM. All the display functions are controlled by instructions and the module can easily be interfaced with an MPU. This makes the module applicable to a wide range of purposes including terminal display units for microcomputers and display units for measuring gages.

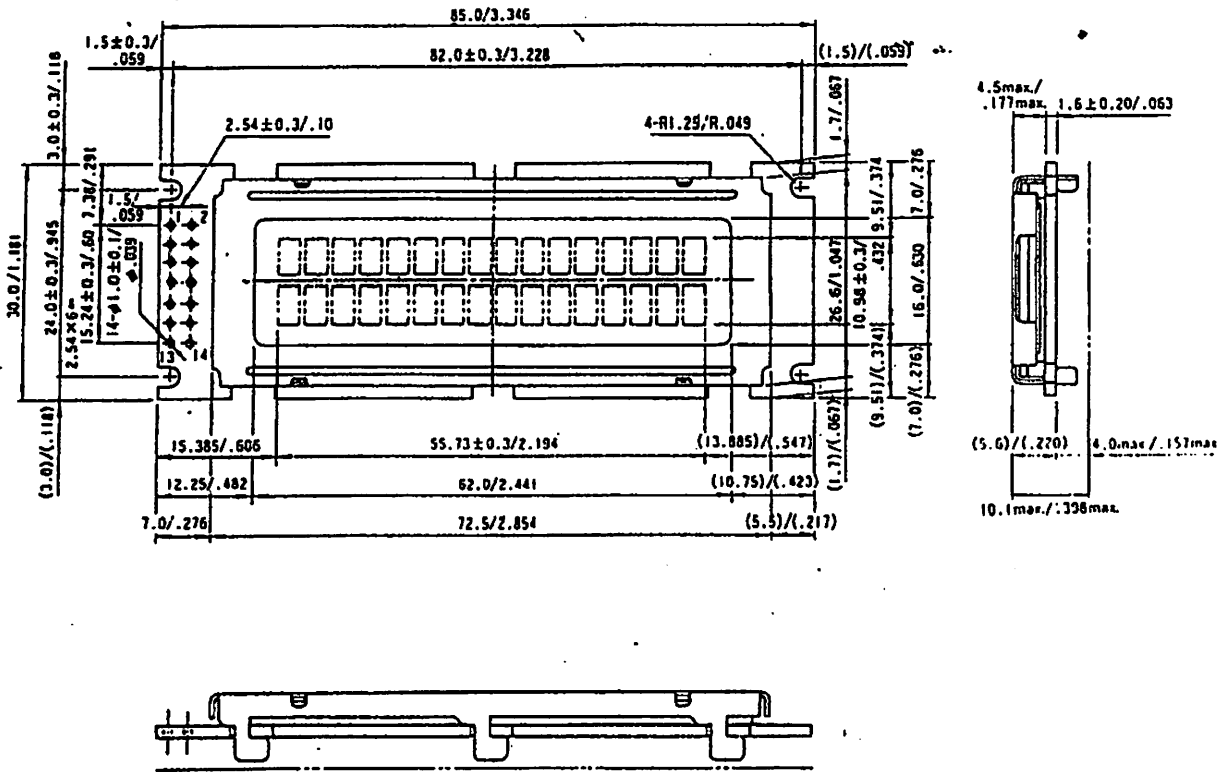
### 1.2 Features

- 16-character, two-line TN liquid crystal display of 5 x 7 dot matrix + cursor
- Duty ratio: 1/16
- Character generator ROM for 192 character types.  
(character font: 5 x 7 dot matrix)
- Character generator RAM for eight character types (program write)  
(character font: 5 x 7 dot matrix)
- 80 x 8 bit display data RAM (80 characters maximum)
- Interface with four-bit and eight-bit MPUs possible
- Display data RAM and character generator RAM readable from MPU
- Many instruction functions

Display Clear, Cursor Home, Display ON/OFF, Cursor ON/OFF, Display Character Blink, Cursor Shift, and Display Shift

- Built-in oscillator circuit
- +5 V single power supply
- Built-in automatic reset circuit at power-on
- CMOS process
- Operating temperature range: 0°C to 50°C

### 1.3 Dimensions Diagram



Unit : mm/inch  
General tolerance : ± 0.5 mm

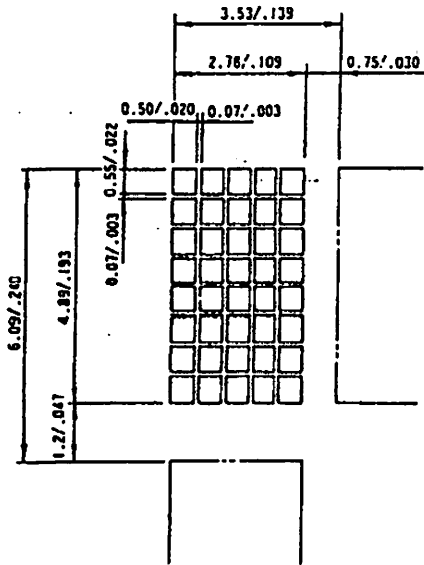
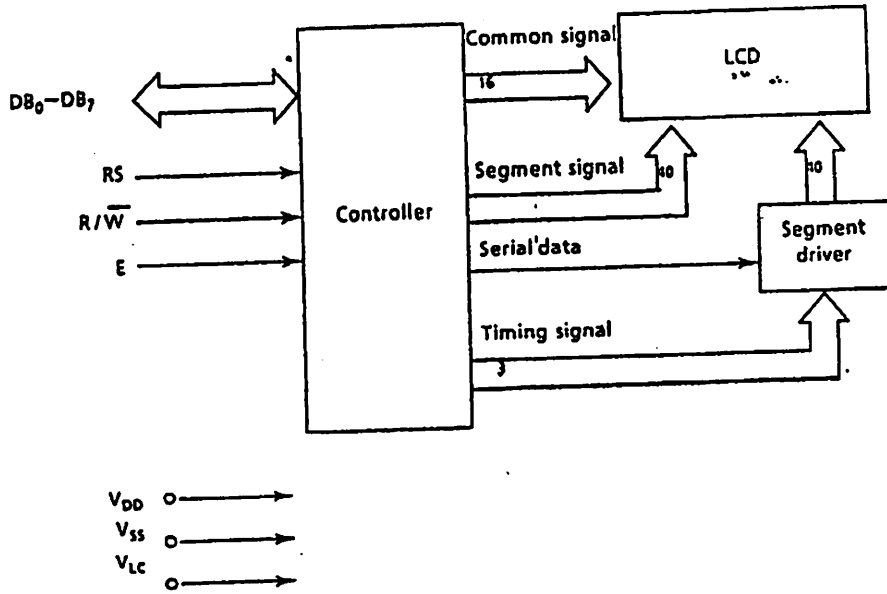


Figure 1 Dimensions diagram

No.	Symbol	Level	Function	
1	Vss	-	Power Supply	0V (GND)
2	Vcc	-		5V ±10%
3	Vcc	-		for LCD Drive
4	RS	H/L	H: Data Input L: Instruction Input	
5	R/W	H/L	H:READ L:WRITE	
6	E	H, $\downarrow$	Enable Signal	
7	DB0	H/L	Data Bus	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L		
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	V+ BL	-	Back Light Supply	4 - 4.2V 50-200mA
16	V- BL	-		0V (GND)

### 1.4 Block Diagram



## 1.5 Absolute Maximum Ratings

$V_{SS} = 0V$

Item	Symbol	Standard	Unit	Remarks
Power supply voltage	$V_{DD}$	-0.3 to +7.0	V	
	$V_{LC}$	$V_{DD} - 13.5$ to $V_{DD} + 0.3$	V	
Input voltage	$V_{in}$	-0.3 to $V_{DD} + 0.3$	V	
Operating temperature	$T_{opr}$	0 to +50	°C	
Storage temperature	$T_{stg}$	-20 to +60	°C	At 50% RH

## 1.6 Electrical Characteristics

$V_{DD} = 5V \pm 5\%$ ,  $V_{SS} = 0V$ ,  $T_A = 0^\circ C$  to  $50^\circ C$

Item		Symbol	Conditions	Standard			Unit
				Min.	Typ.	Max.	
Input voltage	High	$V_{IH1}$		2.2	-	$V_{DD}$	V
	Low	$V_{IL1}$		0	-	0.6	V
Output voltage (TTL)	High	$V_{OH1}$	$-I_{OH} = 0.205\text{ mA}$	2.4	-	-	V
	Low	$V_{OL1}$	$I_{OL} = 1.2\text{ mA}$	-	-	0.4	V
Output voltage (CMOS)	High	$V_{OH2}$	$-I_{OH} = 0.04\text{ mA}$	$0.9V_{DD}$	-	-	V
	Low	$V_{OL2}$	$I_{OL} = 0.04\text{ mA}$	-	-	$0.1V_{DD}$	V
Power supply voltage		$V_{DD}$		4.75	5.00	5.25	V
		$-V_{LC}$	$V_{DD} = 5V, T_A = 25^\circ C$	-	0.25	-	V
Current consumption		$I_{DD}$		-	2.0	3.0	mA
		$I_{LC}$	$V_{LC} = 0.25V$	-	-	1.0	mA
Clock oscillation freq.		$f_{osc}$	Resistance oscillation	190	270	350	kHz

# 1.7 Optical Characteristics

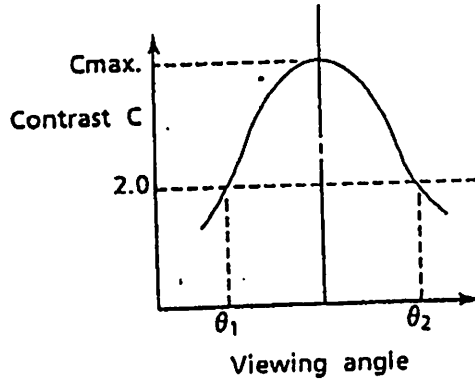
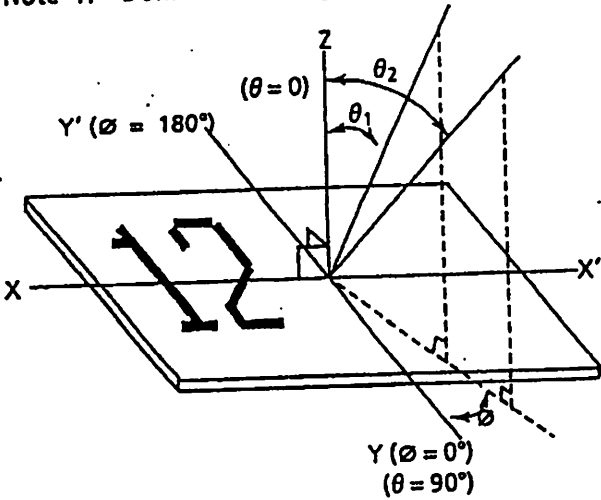
## 1.7.1 Optical characteristics

Maximum viewing angle: 6 o'clock ( $\varnothing = 0^\circ$ )  
 $T_A = 25^\circ\text{C}$ ,  $V_{opr} = 4.75\text{ V}$

Item	Symbol	Conditions	Min.	Typ.	Max.	Remarks
Viewing angle	$\theta_2 - \theta_1$	$C \geq 2.0$ , $\varnothing = 0^\circ$	35	-	-	See Notes 1 and 2.
Contrast	C	$\theta = 25^\circ$ , $\varnothing = 0^\circ$	5	8	-	See Note 3.
Rise time	$t_{on}$	$\theta = 25^\circ$ , $\varnothing = 0^\circ$	-	60 ms	70 ms	See Note 4.
Fall time	$t_{off}$	$\theta = 25^\circ$ , $\varnothing = 0^\circ$	-	150 ms	170 ms	See Note 4.

Note 1: Definition of angles  $\varnothing$  and  $\theta$

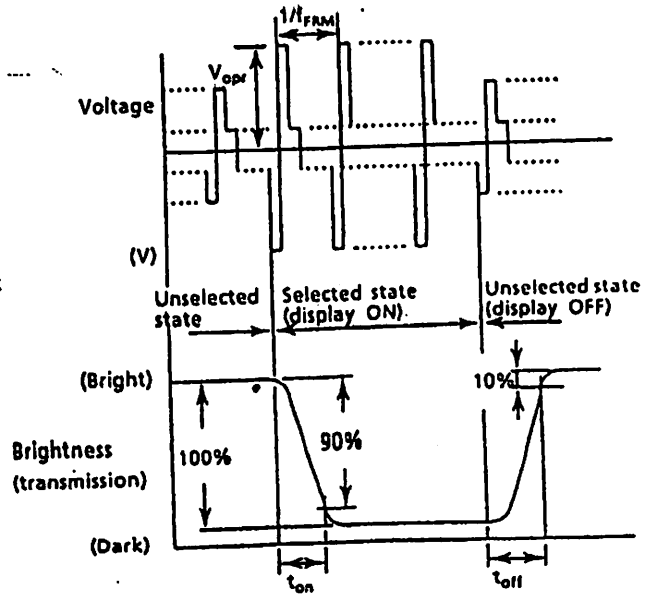
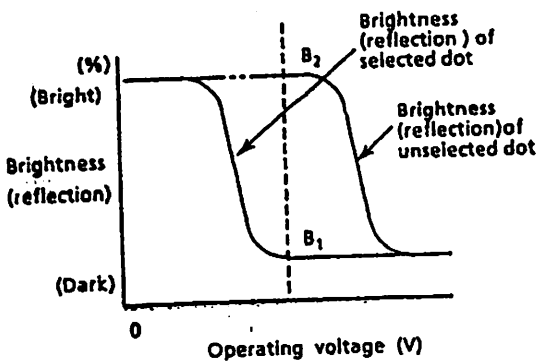
Note 2: Definition of viewing angles  $\theta_1$  and  $\theta_2$



Note 3: Definition of contrast C

Note 4: Definition of response time

$$C = \frac{\text{Brightness (reflection) of unselected dot (B2)}}{\text{Brightness (reflection) of selected dot (B1)}}$$



$V_{opr}$ : Operating voltage (V)  
 $f_{FRAM}$ : Frame frequency (Hz)  
 $t_{on}$ : Response time (rise)(ms)  
 $t_{off}$ : Response time (fall)(ms)

### 1.7.2 Recommended operating voltage

The viewing angle and screen contrast of the LCD panel can be varied by changing the liquid crystal operating voltage ( $V_{opr}$ ), that is  $V_{LC}$ .

The optical characteristics is influenced by an ambient temperature. The recommended value of  $V_{opr}$  for an ambient temperatures are shown below.

Temperature (°C)	0	10	25	40	50
Voltage $V_{opr}$ (V)	5.00	4.90	4.75	4.60	4.50

$$V_{opr} = V_{DD} - V_{LC}$$

## 2. OPERATING INSTRUCTIONS

### 2.1 Terminal Functions

Table 1 Terminal functions

Signal name	No. of terminals	I/O	Destination	Function
DB <sub>0</sub> to DB <sub>3</sub>	4	I/O	MPU	Tristate bidirectional lower four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. If the interface data is 4 bits, the signals are not used.
DB <sub>4</sub> to DB <sub>7</sub>	4	I/O	MPU	Tristate bidirectional upper four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. DB <sub>7</sub> is also used as a busy flag.
E	1	Input	MPU	Operation start signal: The signal activates data write or read.
$\overline{R/W}$	1	Input	MPU	Read (R) and Write (W) selection signals 0: Write 1: Read
RS	1	Input	MPU	Register selection signals 0: Instruction register (Write) Busy flag and address counter (Read) 1: Data register (Write and Read)
V <sub>LC</sub>	1	-	Power supply	Power supply terminal for driving liquid crystal display: The screen contrast can be varied by changing V <sub>LC</sub> .
V <sub>DD</sub>	1	-	Power supply	+5 V
V <sub>SS</sub>	1	-	Power supply	Ground terminal: 0 V

## 2.2 Basic Operations

### 2.2.1 Registers

The controller has two kinds of eight-bit registers: the instruction register (IR) and the data register (DR). They are selected by the register select (RS) signal as shown in Table 2.

The IR stores instruction codes such as Display Clear and Cursor Shift, and the address information of display data RAM (DD RAM) and character generator RAM (CG RAM). They can be written from the MPU, but cannot be read to the MPU.

The DR temporarily stores data to be written into DD RAM or CG RAM, or read from DD RAM or CG RAM. When data is written into DD RAM or CG RAM from the MPU, the data in the DR is automatically written into DD RAM or CG RAM by internal operation. However, when data is read from DD RAM or CG RAM, the necessary data address is written into the IR. The specified data is read out to the DR and then the MPU reads it from the DR. After the read operation, the next address is set and DD RAM or CG RAM data at the address is read into the DR for the next read operation.

Table 2 Register selection

RS	$\overline{R/W}$	Operation
0	0	IR selection, IR write. Internal operation: Display clear
0	1	Busy flag (DB <sub>7</sub> ) and address counter (DB <sub>0</sub> to DB <sub>6</sub> ) read
1	0	DR selection, DR write. Internal operation: DR to DD RAM or CG RAM
1	1	DR selection, DR read. Internal operation: DD RAM or CG RAM to DR

### 2.2.2 Busy flag (BF)

The flag indicates whether the module is ready to accept the next instruction. As shown in Table 2, the signal is output to DB<sub>7</sub> if RS = 0 and  $\overline{R/W}$  = 1. If the value is 1, the module is working internally and the instruction cannot be accepted. If the value is 0, the next instruction can be written. Therefore, the flag status needs to be checked before executing an instruction. If an instruction is executed without checking the flag status, wait for more than the execution time shown by 2.4 Instruction Outline.



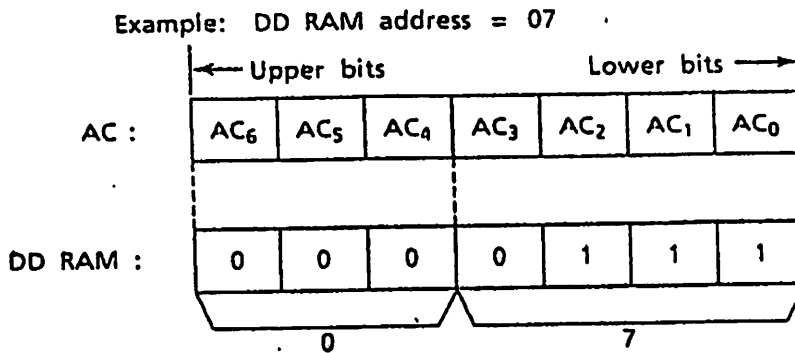
### 2.2.3 Address counter (AC)

The counter specifies an address when data is written into DD RAM or CG RAM and the data stored in DD RAM or CG RAM is read out. If an Address Set instruction (for DD RAM or CG RAM) is written in the IR, the address information is transferred from the IR to the AC. When display data is written into or read from DD RAM or CG RAM, the AC is automatically incremented or decremented by one according to the Entry Mode Set. The contents of the AC are output to DB<sub>0</sub> to DB<sub>6</sub> as shown in Table 2 if RS = 0 and  $\overline{R/W} = 1$ .

### 2.2.4 Display data RAM (DD RAM)

DD RAM has a capacity of up to  $80 \times 8$  bits and stores display data of 80 eight-bit character codes. Some storage areas of DD RAM which are not used for display can be used as general data RAM.

A DD RAM address to be set in the AC is expressed in hexadecimal form as follows.



00H to 0FH of the DD RAM address is set in the line 1, and 40H to 4FH in the line 2.

Note : The addresses in the digit 16 of line 1 and the digit 1 of line 2 are not consecutive.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DD RAM address
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

If the display is shifted, DD RAM address 00H to 27H are displayed in line 1 and 40H to 67H in line 2. The following figures are examples of display shifts.

**\*Left shift**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	DD RAM address
Line 2	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	

**\*Right shift**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	DD RAM address
Line 2	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	

## 2.2.5 Character generator ROM (CG ROM)

Character generator ROM generates 192 types of 5 x 7 dot-matrix character patterns from eight-bit character codes.

Table 3 shows the correspondence between the CG ROM character codes and character patterns.

## 2.2.6 Character generator RAM (CG RAM)

CG RAM is used to create character patterns freely by programming. Eight types of character patterns can be written.

Table 4 shows the character patterns created from CG RAM addresses and data. To display a created character pattern, the character code in the left column of the table is written into DD RAM corresponding to the display position (digit). The areas not used for display are available as general data RAM.

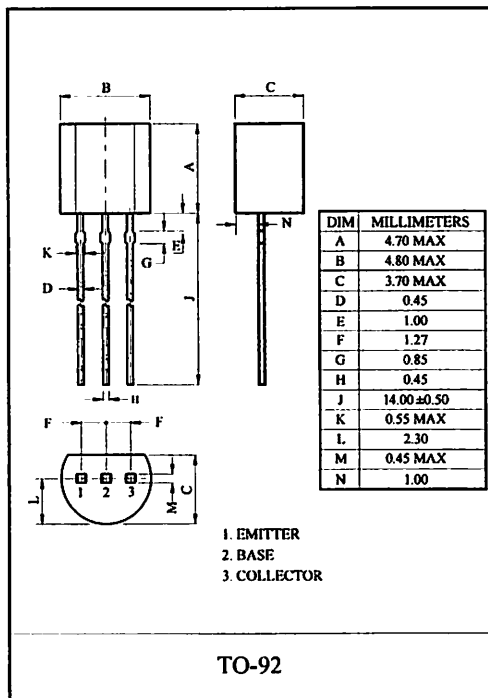
GENERAL PURPOSE APPLICATION.  
SWITCHING APPLICATION.

#### FEATURES

- Excellent  $h_{FE}$  Linearity  
:  $h_{FE}(I_C=0.1mA)/h_{FE}(I_C=2mA)=0.95$ (Typ.)
- Low Noise :NF=1dB(Typ.) at f=1kHz.
- Complementary to KTC9015.

#### MAXIMUM RATING (Ta=25°C)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Collector-Base Voltage	$V_{CBO}$	60	V
Collector-Emitter Voltage	$V_{CEO}$	50	V
Emitter-Base Voltage	$V_{EBO}$	5	V
Collector Current	$I_C$	150	mA
Emitter Current	$I_E$	-150	mA
Collector Power Dissipation	$P_C$	625	mW
Junction Temperature	$T_j$	150	°C
Storage Temperature Range	$T_{sg}$	-55 ~ 150	°C



#### ELECTRICAL CHARACTERISTICS (Ta=25°C)

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Collector Cut-off Current	$I_{CBO}$	$V_{CB}=50V, I_E=0$	-	-	50	nA
Emitter Cut-off Current	$I_{EBO}$	$V_{EB}=5V, I_C=0$	-	-	100	nA
DC Current Gain	$h_{FE}$ (Note)	$V_{CE}=5V, I_C=1mA$	60	-	1000	
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	$I_C=100mA, I_B=10mA$	-	0.1	0.25	V
Transition Frequency	$f_T$	$V_{CE}=10V, I_C=1mA, f=100MHz$	60	-	-	MHz
Collector Output Capacitance	$C_{ob}$	$V_{CB}=10V, I_E=0, f=1MHz$	-	2.0	3.5	pF
Noise Figure	NF	$V_{CE}=6V, I_C=0.1mA, R_g=10k\Omega, f=1kHz$	-	1.0	10	dB

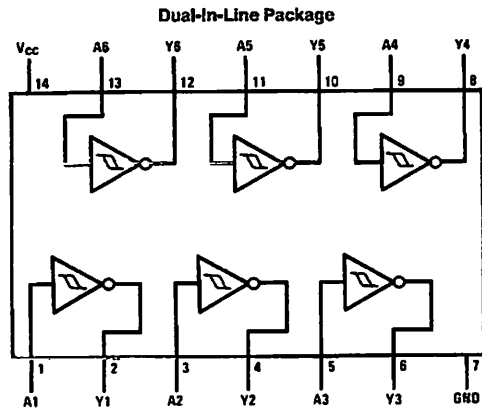
Note :  $h_{FE}$  Classification A:60 ~ 150, B:100 ~ 300, C:200 ~ 600, C1:200 ~ 320, C2:280 ~ 420, C3:380 ~ 600, D:400 ~ 1000

## 54LS14/DM74LS14 Hex Inverters with Schmitt Trigger Inputs

### General Description

This device contains six independent gates each of which performs the logic INVERT function. Each input has hysteresis which increases the noise immunity and transforms a slowly changing input signal to a fast changing, jitter free output.

### Connection Diagram



TL/F/6353-1

### Function Table

$$Y = \bar{A}$$

Input	Output
A	Y
L	H
H	L

H = High Logic Level

L = Low Logic Level

### Absolute Maximum Ratings (Note)

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

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
54LS	-55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

### Recommended Operating Conditions

Symbol	Parameter	54LS14			DM74LS14			Units
		Min	Nom	Max	Min	Nom	Max	
$V_{CC}$	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
$V_{T+}$	Positive-Going Input Threshold Voltage (Note 1)	1.5	1.6	2.0	1.4	1.6	1.9	V
$V_{T-}$	Negative-Going Input Threshold Voltage (Note 1)	0.0	0.0	1.1	0.0	0.0	1	V
HYS	Input Hysteresis (Note 1)	0.4	0.8		0.4	0.8		V
$I_{OH}$	High Level Output Current			-0.4			-0.4	mA
$I_{OL}$	Low Level Output Current			4			8	mA
$T_A$	Free Air Operating Temperature	-55		125	0		70	°C

### Electrical Characteristics (Recommended operating free air temperature range (unless otherwise noted))

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
$V_I$	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$			-1.5	V
$V_{OH}$	High Level Output Voltage	$V_{CC} = \text{Min}, I_{OH} = \text{Max}$ $V_{IL} = \text{Max}$	54LS 2.5	3.4		V
$V_{OL}$	Low Level Output Voltage	$V_{CC} = \text{Min}, I_{OL} = \text{Max}$ $V_{IH} = \text{Min}$	54LS 0.25	0.25	0.4	V
		$V_{CC} = \text{Min}, I_{OL} = 4 \text{ mA}$	DM74 0.25	0.25	0.4	
$I_{T+}$	Input Current at Positive-Going Threshold	$V_{CC} = 5V, V_I = V_{T+}$	DM74	-0.14		mA
$I_{T-}$	Input Current at Negative-Going Threshold	$V_{CC} = 5V, V_I = V_{T-}$	DM74	-0.18		mA
$I_I$	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}, V_I = 7V$ $V_{CC} = \text{Max}, V_I = 10.0V$	DM74 54LS		0.1	mA
$I_{IH}$	High Level Input Current	$V_{CC} = \text{Max}, V_I = 2.7V$			20	$\mu\text{A}$
$I_{IL}$	Low Level Input Current	$V_{CC} = \text{Max}, V_I = 0.4V$			-0.4	mA
$I_{OS}$	Short Circuit Output Current	$V_{CC} = \text{max}$ (Note 3)	54LS	-20	-100	mA
			DM74	-20	-100	
$I_{OCH}$	Supply Current with Outputs High	$V_{CC} = \text{Max}$		8.6	16	mA
$I_{OCL}$	Supply Current with Outputs Low	$V_{CC} = \text{Max}$		12	21	mA

Note 1:  $V_{CC} = 5V$ .

Note 2: All typicals are at  $V_{CC} = 5V, T_A = 25^\circ\text{C}$ .

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

**Switching Characteristics** at  $V_{CC} = 5V$  and  $T_A = 25^\circ C$  (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	$R_L = 2\text{ k}\Omega$				Units
		$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
		Min	Max	Min	Max	
$t_{PLH}$	Propagation Delay Time Low to High Level Output	5	22	8	25	ns
$t_{PHL}$	Propagation Delay Time High to Low Level Output	5	22	10	33	ns