

# TUGAS AKHIR

**PERENCANAAN DAN PEMBUATAN PENCATAT WAKTU  
MENGUNAKAN AT89S51 UNTUK MENINGKATKAN KINERJA  
SATPAM**



Disusun oleh :

**NAMA : MAULANA MALIK**

**NIM : 04.52.214**



**KONSENTRASI TEKNIK ELEKTRONIKA  
PROGRAM STUDI TEKNIK ELEKTRONIKA D-III  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
SEPTEMBER 2008**

FINAN. RASUT

REKAPITULASI PERALIHAN KELOMPOK PERUSAHAAN  
DARI PERUSAHAAN KE PERUSAHAAN  
MELIPUTI

1. 01. 01. 01. 01

PERUSAHAAN : PT. ABC

PERUSAHAAN : PT. DEF

REKAPITULASI PERALIHAN KELOMPOK PERUSAHAAN  
DARI PERUSAHAAN KE PERUSAHAAN  
MELIPUTI PERUSAHAAN KELOMPOK PERUSAHAAN  
DARI PERUSAHAAN KE PERUSAHAAN  
MELIPUTI PERUSAHAAN KELOMPOK PERUSAHAAN

**LEMBAR PERSETUJUAN  
TUGAS AKHIR**

**PERENCANAAN DAN PEMBUATAN PENCATA WAKTU  
MENGUNAKAN AT89S51 UNTUK MENINGKATKAN KINERJA  
SATPAM**

**Disusun oleh :**

**Nama : Maulana Malik**

**Nim : 04.52.214**

**Mengetahui**

**Ketua Program Studi**

**Teknik Elektro D-III**



**Ir. Choirul Saleh, MT**  
**NIP.Y. 1018 800 190**

**Diperiksa dan Disetujui**

**Dosen Pembimbing**

**Joseph Dedy Irawan ST,MT**  
**NIP. 132/315.178**

**PROGRAM STUDI TEKNIK ELEKTRO D-III  
KONSENTRASI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**September 2008**

## ABSTRAK

### **Perencanaan dan Pembuatan Pencatat Waktu Menggunakan AT89S51 Untuk Meningkatkan Kinerja Satpam**

(Maulana Malik, 0452214)

(Dosen Pembimbing : Joseph Dedy Irawan,ST,MT)

Seringnya terjadi pencurian pada gudang barang di malam hari (meskipun sudah ada satpam yang bertugas jaga malam) merupakan salah satu akibat dari kerja satpam yang tidak maksimal. Laporan dengan judul Alat Pencatat Waktu untuk Meningkatkan Kinerja Satpam Menggunakan AT89S51 ini berguna untuk memberikan informasi kepada kita sekaligus memantau kerja satpam di malam hari, apakah satpam tersebut sudah melakukan tugas jaga keliling atau tidak.

Alat ini terdiri dari dua bagian, salah satunya berupa alat portabel untuk dibawa satpam pada saat bertugas keliling, sedang alat lainnya berupa alat permanen yang menempel pada dinding di setiap daerah yang diawasi. Pada alat portabel terdapat RTC ( Real Time Clock ) sebagai pemberi informasi, LCD ( Liquid Cristal Display ) untuk display, EEPROM yang berguna sebagai memori serta mikrokontroller untuk mengontrol keseluruhan proses pengiriman maupun penerimaan data. Sedangkan pada bagian permanen yang menempel di dinding hanya terdapat mikrokontroller untuk mengatur konfigurasi tempat atau lokasi yang harus dicek oleh petugas satpam, serta DB9 female sebagai konektor antara alat portabel ke alat permanen yang nantinya akan mendapat masukan supply tegangan dari alat portabel yaitu berupa hexa desimal yang merupakan kode dari masing-masing lokasi tempat boks permanen di letakan.

Berdasarkan hasil pengujian, alat telah bekerja dengan baik. Untuk mengeset jam dan tanggal agar sesuai dengan waktu itu, dilakukan di PC. Tampilan waktu, tanggal dan lokasi ditampilkan oleh LCD. Setelah proses pengecekan, untuk dapat melihat keseluruhan data yang tersimpan di EEPROM, alat dapat dihubungkan ke PC (Personal Komputer ) untuk selanjutnya dapat di cetak atau disimpan, alat ini digunakan untuk perseorangan atau kelompok satpam dan dapat mengontrol 2 sektor sekaligus.

**KeyWord** : *Mikrokontroller AT89S51, RTC, EEPROM, Pencatat waktu personal komputer.*



# بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## Lembar Persembahan

Terima kasih yang sebesar besarnya kepada ﷻ milik-Nya semua yang ada di alam semesta, kekayaan, keabadian, kemuliaan, hikmah dan kemampuan. Dari-Nya

semua kasih sayang, pertolongan, cinta dan kebaikan. Untuk Keluargaku tercinta, IBU dan BAPAK, Terimakasih atas doa, moril dan materilnya yang tak ternilai dengan dengan apa pun.

Semoga anak kalian ini menjadi orang yang berbakti dan sukses, serta menjadi kebanggaan kalian. Buat kakak-kakaku Teh Eli, Ani, Ida, Aa Yayan, terima kasih atas dukungannya selama ini dan buat adikku Rima, calon guru he....., semoga cita-citamu tercapai dan bisa meneruskan cita-cita bapak, senior kita he.....

Dan untuk semua keluarga ku di Malang. Buat ibu, mas Oong, mba Sandra terima kasih atas doa dan dukungannya untuk teman2ku yang cool abiz CV. Sahabat : Fajar, Olan, Dedy.. Sukses jon..!?! Anak2 yang mendahului kita ..hez.. yang duluan lulus maksudnya, Iis, Rendi, Olan, Fajar, Dedi Beni, Indra maksih semangatnya. Anak2 seperjuangan menghadapi TA

Rudra, Nita, Rofik, Hadi terima kasih atas bantuannya ya..... Buat anak2 yang masih berjuang dengan TA, SEMANGAT.....

Buat semua Sahabat-sahabatku yang ada di Tasikmalaya Bokenk, Erik, Agel, Akang Solohin hatur nuhun ka sadayana kanggo doa sareng dukunganna. Ga lupa buat Cevi, kuliah yang bener..... N belajar yang rajin man.....

## KATA PENGANTAR

Dengan mengucapkan puji syukur kepada Allah SWT, atas limpahan rahmat dan hidayah-Nya,serta perlindungan,pertolongan dan ridho-Nya kepada kami, sehingga kami dapat menyelesaikan Tugas Akhir ini tepat pada waktunya,dan juga iringan salam dan sholawat bagi junjungan kami Nabi Muhammad SAW.

Dengan pengerahan segenap usaha, akhirnya kami dapat menyelesaikan tugas akhir kami yang berjudul :

**“ PERENCANAAN DAN PEMBUATAN PECHAT WAKTU MENGGUNAKAN AT89S51 UNTUK MENINGKATKAN KINERJA SATPAM”**

Pada kesempatan ini penyusun mengucapkan terima kasih pada :

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Choirul Saleh, MT selaku Ketua Jurusan Teknik Elektro D-III Institut Teknologi Nasional Malang
3. Bapak Joseph Dedy Irawan,ST,MT selaku dosen pembimbing Tugas Akhir.
4. Kepada semua pihak yang telah membantu penyelesaian Tugas Akhir ini.

Tugas Akhir ini adalah kewajiban bagi setiap mahasiswa Institut Teknologi Nasional Malang dengan tujuan untuk memenuhi persyaratan untuk memperoleh kelulusan pada program Diploma III Jurusan Teknik Elektronika D-III di Institut Teknologi Nasional Malang.

Dengan terselesaikannya buku laporan tugas akhir ini, kami berharap semoga buku ini dapat membawa manfaat bagi pembaca umumnya dan juga bagi kami pada khususnya serta semua pihak yang berkepentingan. Kami menyadari bahwa kami adalah manusia biasa yang tidak luput dari kesalahan dan kekurangan. Untuk itu, kritik dan saran yang bersifat membangun kami harapkan untuk perbaikan selanjutnya.

Malang, September 2008

Penulis

## DAFTAR ISI

	Hal
HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN .....	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xii
<b>BAB I : PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat .....	2
1.5 Metodologi .....	3
1.6 Sistematika Penulisan.....	4
<b>BAB II : LANDASAN TEORI.....</b>	<b>5</b>
2.1 Mikrokontroler AT89S51 .....	5
2.2 12C BUS .....	11
2.2.1 Sinyal Dasar 12C.....	12
2.2.2 Transmisi Data .....	14

2.3 Serial RTC (Real Time Clock) .....	18
2.3.1 Fungsi kaki-kaki (pin) DS1307 (ibid) .....	18
2.3.2 Cara Kerja RTC Serial DS1307 .....	19
2.3.3 Peta Alamat RAM dan RTC .....	19
2.3.4 Waktu dan Kalender.....	20
2.3.5 Register Kontrol.....	21
2.3.6 Serial Data Bus.....	22
2.3.7 Transfer Data Pada Serial Bus.....	24
2.4 Serial EEPROM .....	27
2.4.1 Pin-pin IC AT24Cxx .....	28
2.4.2 Proses Pengisian EEPROM AT24Cxx.....	29
2.4.3 Proses Pembacaan AT24Cxx.....	31
2.5 Liquid Crystal Display (LCD) .....	33
2.5.1 Konfigurasi LCD .....	33
2.5.2 Instruksi Operasi Dasar .....	35
2.5.2.1 Register.....	35
2.5.2.2 Busy flag .....	35
2.5.2.3 Address counter.....	36
2.5.2.4 Display Data RAM (DD RAM) .....	36
2.5.2.5 Character Generator ROM (CG ROM) .....	36
2.5.2.6 Character Generator RAM (CG RAM) .....	36
2.6 Port Serial DB-9 dan RS-232.....	37
2.6.1 MAX 232 .....	42
2.7 Borland Delphi.....	44

2.7.1 Tipe Data .....	44
2.7.2 Data Base Dasktop.....	46
2.7.3 Qick Report.....	46
<b>BAB III : PERENCANAAN DAN PEMBUATAN ALAT .....</b>	<b>47</b>
3.1 Perencanaan Elektronik .....	49
3.1.1 Rangkaian MCU Pada Alat Portabel (Satpam) .....	49
3.1.2 Rangkaian Real Time Clock (RTC) .....	52
3.1.3 Rangkaian Liquid Crystal Display (LCD) .....	53
3.1.4 Rangkaian Serial EEPROM .....	55
3.1.5 Rangkaian Serial Interface RS-232 .....	56
3.1.6 Rangkaian MCU Pada Alat Permanen (Pos 1 dan Pos 2) .....	58
3.1.7 Rangkaian Indikator Batteray Low .....	58
3.2 Perencanaan Perangkat Lunak (Software) .....	60
<b>BAB IV : PENGUJIAN ALAT .....</b>	<b>62</b>
4.1 Pengujian Hardware.....	62
4.1.1 Pengujian RTC .....	62
4.1.2 Pengujian EEPROM.....	65
4.1.3 Pengujian RS-232.....	66
4.1.4 Pengujian Perangkat I2C .....	68
4.2 Pengujian Software.....	70
4.2.1 Tahap Pertama.....	70
4.2.2 Tahap Kedua.....	71
4.2.3 Tahap Ketiga .....	73
4.3 Spesifikasi Alat .....	76

<b>BAB V : PENUTUP .....</b>	<b>78</b>
5.1 Kesimpulan Alat.....	78
5.2 Saran .....	79

**DAFTAR PUSTAKA**

**LAMPIRAN**

## DAFTAR GAMBAR

	Hal
Gambar 2.1 Blok Diagram AT89S51 .....	7
Gambar 2.2 Bentuk Fisik AT89S51 .....	8
Gambar 2.3 Rangkaian Osilator .....	11
Gambar 2.4 Sinyal-sinyal Dasar I2C .....	13
Gambar 2.5 Transmisi Data Pada I2C Bus .....	14
Gambar 2.6 Mengalami IC Secara Sederhana .....	16
Gambar 2.7 Mengalami Metode 11 Byte .....	17
Gambar 2.8 Mengalami Metode 16 Byte .....	17
Gambar 2.9 Kaki-Kaki DS1307 .....	19
Gambar 2.10 Peta Alamat DC1307 .....	20
Gambar 2.11 Register Waktu RTC .....	21
Gambar 2.12 Transfer Data Pada I2C Bus .....	24
Gambar 2.13 Mode Tulis .....	26
Gambar 2.14 Mode Baca .....	27
Gambar 2.15 Konfigurasi IC AT24Cxx .....	28
Gambar 2.16 Metode 11 Bit Addressing .....	30
Gambar 2.17 Metode 16 Bit Addressing .....	30
Gambar 2.18 Proses Pembacaan EEPROM .....	32
Gambar 2.19 Proses Pembacaan Satu EEPROM .....	32
Gambar 2.20 Diagram Blok LCD .....	33
Gambar 2.21 Prinsip Dasar Port Serial .....	38



Gambar 2.22 DTE Device Connector .....	40
Gambar 2.23 DCE Device Connector .....	40
Gambar 2.24 Transfer Data AntaraDTE dan DCE.....	41
Gambar 2.25 Konfigurasi Pin dan Blok Diagram RS232 .....	43
Gambar 2.26 Level Converter TTL ke RS232 .....	44
Gambar 3.1 Blok Diagram Keseluruhan Alat .....	47
Gambar 3.2 Perencanaan Rangkaian AT89S51 .....	49
Gambar 3.3 Perancangan Rangkaian RTC.....	53
Gambar 3.4 Perancangan Rangkaian LCD .....	54
Gambar 3.5 Perancangan Rangkaian EEPROM .....	56
Gambar 3.6 Serial Interface RS-232 .....	57
Gambar 3.7 Perencanaan Rangkaian AT89S51 Pada Alat Permanen .....	58
Gambar 3.8 Rangkaian Indikator Batteray.....	59
Gambar 3.9 Flowchart Program Pada Alat Portabel ( Satpam ) .....	60
Gambar 3.10 Flowchart Program Pada Alat Permanen (Pos 1 dan Pos 2 ) .....	60
Gambar 4.1 Blok Diagram Pengujian RTC .....	63
Gambar 4.2 Hasil Tampilan Pada LCD .....	64
Gambar 4.3 Tampilan Pada Form Untuk Merbah Waktu dan Tanggal .....	64
Gambar 4.4 Diagram Blok Pengujian EEPROM .....	65
Gambar 4.5 Hasil Tampilan Pembacaan data Pada LCD.....	66
Gambar 4.6 Input Tegangan Pada RS232 .....	67
Gambar 4.7 Sinyal Input RS232 .....	67
Gambar 4.8 Sinyal Output RS232 ke PC .....	67
Gambar 4.8 Diagram Blok Pengujian Perangkat I2C .....	68

Gambar 4.9 Sinyal SDA.....	70
Gambar 4.10 Sinyal SCL .....	70
Gambar 4.11 Tampilan Icon.....	71
Gambar 4.12 Form Untuk Merubah Jam dan Tanggal.....	71
Gambar 4.13 Tampilan Form untuk menampilkan tampilan utama .....	72
Gambar 4.14 Form tampilan utama.....	72
Gambar 4.15 Tampilan pada LCD untuk pengabsenan pos 1 .....	73
Gambar 4.16 Tampilan pada LCD untuk pengabsenan pos 2.....	73
Gambar 4.17 Tampilan Form ketika mau menampilkan data.....	74
Gambar 4.18 Tampilan Form Data Pengecekan Satpam .....	74
Gambar 4.19 Bentuk Tampilan Print Out Laporan Hasil Pengecekan Satpam.....	74
Gambar 4.20 Alat Keseluruhan.....	75

## DAFTAR TABEL

	Hal
Tabel 2.1 Register kontrol DS1307.....	21
Tabel 2.2 Frekwensi Output Gelombang Kotak .....	22
Tabel 2.3 Konfigurasi pin-pin IC AT24Cxx .....	28
Tabel 2.4 Konfigurasi pin-pin.....	34
Tabel 2.5 Register Seleksi .....	35
Tabel 2.6 Fungsi Terminal pada LCD.....	37
Tabel 2.7 Deskripsi Pin DB-9 (male) dan Fungsinya .....	41



# BAB I

## LATAR BELAKANG

### 1.1 Latar Belakang

Perkembangan teknologi yang semakin maju dalam berbagai bidang, membuat segala sesuatu menjadi lebih mudah, cepat, praktis dan efisien, sehingga membuat setiap orang menginginkan segala sesuatu dapat dilakukan dengan mudah. Contohnya dalam pengaturan sistem keamanan suatu perusahaan, pabrik, atau gudang penyimpanan barang.

Sampai saat ini masih banyak dijumpai perusahaan, pabrik, maupun gudang tempat penyimpanan barang yang masih menggunakan jasa dari seorang satpam untuk mengawasi ataupun mengontrol keamanan pabriknya pada siang hari atau malam hari. Para satpam tersebut bekerja bergiliran setiap jamnya mengontrol keadaan disekitar pabrik dengan cara mengitarinya. Tapi meskipun demikian masih saja terjadi pencurian ataupun kehilangan. Hal ini sering disebabkan oleh kelalaian dari satpam yang sedang bertugas karena tidak melakukan giliran berjaga keliling. Dengan demikian perlu adanya suatu sistem pengontrolan tersendiri terhadap kinerja dari para satpam tersebut.

Berdasarkan hal itu saya mencoba merencanakan serta membuat alat yang bisa digunakan untuk mengontrol kinerja dari satpam dengan cara mencatat waktu pengecekan serta tempat yang telah dilalui oleh satpam. Sehingga kita dapat

mengetahui apakah satpam tersebut benar-benar melakukan tugasnya. Dengan cara melihat catatan waktu pengecekan serta lokasi yang disimpan oleh alat ini.

## 1.2 Rumusan masalah

*Kasih Intro*

1. Bagaimana merancang sistem keamanan pada suatu bangunan.
2. Bagaimana merancang sistem perangkat keras alat.
3. Bagaimana merancang sistem perangkat lunak alat.

## 2.2 Batasan Masalah

*Kasih Intro*

1. Komunikasi serial antara MCU, Serial EEPROM, serta RTC dengan sistem I2C bus
2. Komunikasi antara MCU dengan computer secara serial
3. Alat permanen yg terdapat di setiap lokasi pengecekan keliling sebanyak 2 buah

## 1.4 Tujuan

*Kasih Intro*

1. Memantau kinerja satpam agar lebih optimal
2. Memudahkan pengecekan terhadap kinerja satpam pada malam hari
3. Meningkatkan keamanan.

## 1.5 Metodologi

Untuk mencapai tujuan yang direncanakan, maka pada tugas akhir ini menggunakan metodologi sebagai berikut :

a. Studi literatur

Metode ini digunakan untuk mempelajari, mengkaji, dan mengumpulkan data-data yang berkaitan dengan perancangan dan pembuatan alat pada tugas akhir ini.

b. Perencanaan hardware dan software alat

Mengadakan pemilihan komponen perangkat keras dan pemrograman yang digunakan untuk pembuatan alat dan penyusunan urutan kerja sistem.

c. Pembuatan alat

Merangkai komponen-komponen yang telah terpilih beserta perangkat keras pendukung lainnya.

d. Pengujian alat

Melakukan pengujian permodul untuk memastikan bahwa masing-masing modul telah bekerja dengan baik.



## **1.6 Sistematika Penulisan**

Laporan akhir ini berisi tentang pembahasan mengenai proses perancangan serta pembuatan alat pencatat waktu menggunakan AT89S51 untuk meningkatkan kinerja satpam adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi serta sistematika penulisan laporan.

### **BAB II LANDASAN TEORI**

Membahas tentang landasan teori yang mendasari proses perencanaan dan pembuatan alat

### **BAB III PERENCANAAN DAN PEMBUATAN ALAT**

Membahas tentang proses perencanaan dan pembuatan alat itu sendiri

### **BAB IV PENGUKURAN DAN PENGUJIAN ALAT**

Pada bab ini berisi tentang analisis data yang diperoleh dari hasil percobaan

### **BAB V PENUTUP**

Pada bab ini berisi tentang kesimpulan dan saran dari keseluruhan proses perencanaan dan pembuatan alat



## BAB II

### LANDASAN TEORI

Landasan teori bertujuan untuk membantu agar dapat memahami suatu sistem, dimana landasan teori dapat dijadikan sebagai bahan acuan dalam perencanaan pembuatan alat. Dengan pertimbangan tersebut, maka landasan teori merupakan bagian penting yang harus dipahami sebagai acuan untuk pembahasan selanjutnya.

#### 2.1 Mikrokontroler AT89S51

Mikrokontroler AT89S51 adalah anggota keluarga MCS-51. Port serial pada AT89S51 bersifat dupleks-penuh atau full duplex, artinya port serial bisa menerima dan mengirim secara bersamaan. Selain itu juga memiliki penyangga penerima, artinya port serial mulai bisa menerima byte yang kedua sebelum byte pertama dibaca oleh register penerima (jika sampai byte yang kedua selesai diterima sedangkan byte pertama belum juga dibaca, maka salah satu byte akan hilang). Penerimaan dan pengiriman data port serial melalui register SBUF. Penulisan ke SBUF berarti mengisi register pengiriman SBUF sedangkan pembacaan dari SBUF berarti membaca register penerimaan. Port serial pada AT89S51 bisa digunakan dalam 4 mode kerja yang berbeda. Dari 4 mode tersebut, 1 mode diantaranya bekerja secara sinkron dan 3 lainnya bekerja secara asinkron. Keempat mode kerja tersebut adalah :

- Mode 0 : Mode ini bekerja secara sinkron, data serial dikirim dan diterima melalui kaki P3.0 (Rx D), sedangkan kaki P3.1 (Tx D) dipakai untuk

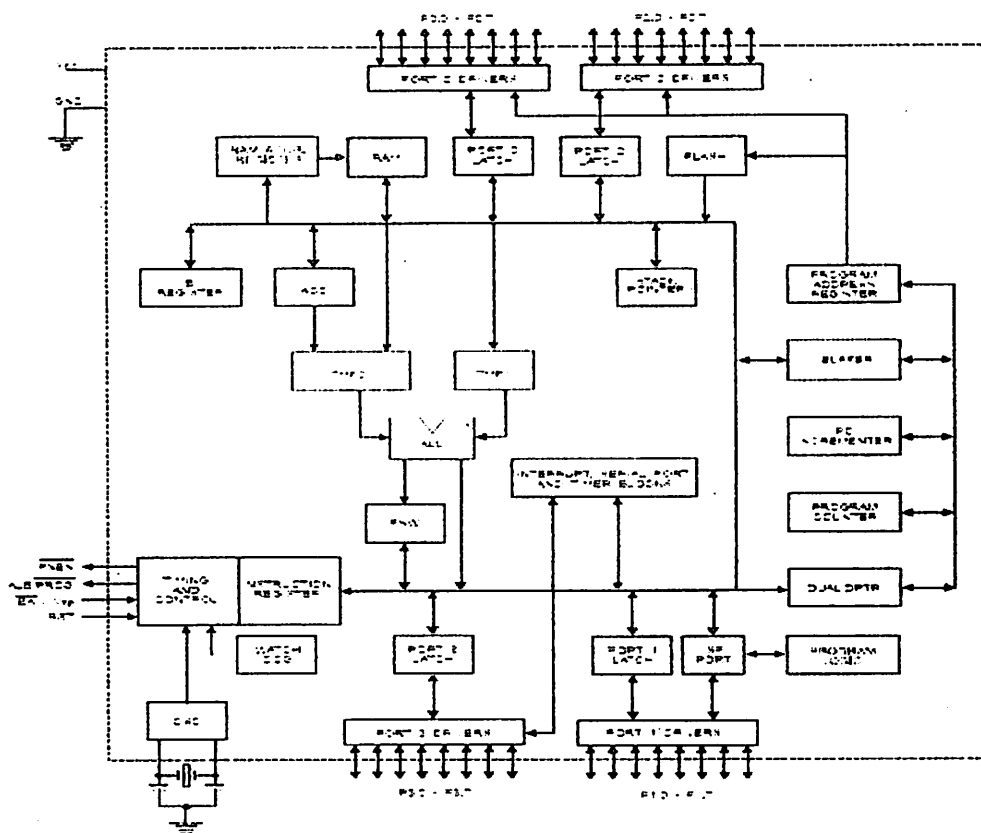
menyalurkan detak pendorong data serial yang dibangkitkan AT89S51. Data dikirim/diterima 8 bit sekaligus, dimulai dari bit yang bobotnya paling kecil atau LSB (bit 0) dan diakhiri dengan bit yang bobotnya paling besar atau MSB (bit 7). Kecepatan pengiriman data (baudrate) adalah 1/12 frekuensi kristal yang digunakan.

- Mode 1 : Pada mode ini tetap yaitu, data dikirim dan diterima melalui kaki P3.0 (RXD), secara asinkron (juga mode 2 dan 3). Pada Mode 1 data atau diterima 10 bit sekaligus, diawali dengan 1 bit start, disusul dengan 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), diakhiri dengan 1 bit stop. Pada AT89S51 yang berfungsi sebagai penerima bit stop adalah RB8 dalam register SCON. Kecepatan pengiriman data (baud rate ) bisa diatur sesuai dengan keperluan. Mode inilah (mode 2 dan 3) yang umum dikenal sebagai UART (Universal Asynchronous Receiver/Transmitter).
- Mode 2 : Data dikirim/diterima 11 bit sekaligus, diawali dengan 1 bit start, disusul 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), kemudian bit ke 9 yang bias diatur lebih lanjut, diakhiri dengan 1 bit stop. Pada AT89S51 yang berfungsi sebagai penerima, bit 9 ditampung pada bit RB8 dalam register SCON, sedangkan bit stop diabaikan tidak ditampung. Kecepatan pengiriman data (baud rate) bisa dipilih antara 1/32 atau 1/64 frekuensi kristal yang digunakan.
- Mode 3 : mode ini sama dengan mode 2, hanya saja kecepatan pengiriman data (baud rate) bisa diatur sesuai dengan keperluan, seperti halnya Mode

1. Pada mode asinkron (mode 1, mode 2, mode 3), port AT89S51 bekerja secara full duplex.

Struktur memori yang dimiliki oleh AT89S51 adalah sebagai berikut :

- RAM Internal, memori sebesar 128 byte yang biasanya digunakan untuk menyimpan variable atau data yang bersifat sementara.
- Special Function Register (Register Fungsi Khusus), memori yang berisi register-register yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut, seperti timer, serial dan lain-lain.
- Flash PEROM, memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51.



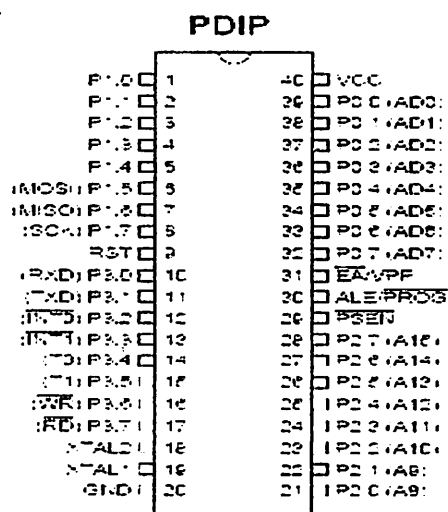
Gambar 2.1 Blok Diagram AT89S51  
Sumber :www. atmel.com

Secara umum keistimewaan yang dimiliki oleh mikrokontroler AT89S51 adalah sebagai berikut:

- Sebuah CPU 8 bit yang termasuk ke dalam keluarga MCS-51.
- 4 Kbyte Reprogrammable Flash memory.
- Kapasitas RAM internal sebesar 128 byte.
- Empat buah programmable port I/O, masing-masing terdiri atas 8 buah jalur I/O.
- Dua buah pencacah (counter) atau pewaktu (timer) 16-bit.
- Kecepatan operasi hingga 24 MHz.
- Enam buah jalur selaan (interrupt).
- Sebuah port serial full duplex.

Berikut ini adalah bentuk fisik dan konfigurasi dari kaki-kaki pada MCU

AT89S51:



Gambar 2.2 Bentuk Fisik AT89S51  
Sumber: [www.atmel.com](http://www.atmel.com)



Fungsi dari tiap-tiap kaki adalah sebagai berikut:

1. GND

Dihubungkan dengan ground rangkaian.

2. VCC

Dihubungkan dengan sumber tegangan +5 Volt.

3. Port 0 (P0.0 – P0.7)

Port merupakan I/O 8 bit dua arah. Port ini dihubungkan sebagai multipleks bus alamat rendah dan bus data selama pengaksesan ke memory luar.

4. Port 1 (P1.0 – P1.7)

Port 1 dapat berfungsi sebagai input ataupun output dan bekerja baik untuk operasi bit maupun byte, tergantung dari pengaturan software.

5. Port 2 (P2.0 – P2.7)

Port dapat digunakan sebagai alamat bus byte tinggi selama adanya akses ke memory program luar atau memory data luar.

6. Port 3 (P3.0 – P3.7)

Port ini selain memiliki fungsi I/O juga mempunyai fungsi khusus sebagai berikut:

RD (P3.7) Sinyal pembaca memory data luar.

WR (P3.6) Sinyal penulisan memory data luar.

T1 (P3.5) Masukkan dari pewaktu atau pencacah 1.

T0 (P3.4) Masukkan dari pewaktu atau pencacah 0.

INT1 (P3.3) Masukkan interupt 1.

INT0 (P3.2) Masukkan interupt 0.



TXD (P3.1) Keluaran pengiriman data untuk serial port.

(Asynchronous) atau sebagai keluaran clock (Synchronous)

RXD (P3.0) Masukan penerimaan data serial (Synchronous)

#### 7. RST / VPD

Merupakan pin input yang aktif tinggi, jika pin ini aktif tinggi selama 2 siklus mesin ketika oscilator bekerja akan mereset peralatan.

#### 8. ALE (Address Lacth Enable)

Pin ALE (aktif tinggi) mengeluarkan pulsa output untuk melacth suatu byte alamat rendah selama mengakses ke memory eksternal. ALE dapat mengendalikan TTL. Pin ini juga merupakan input pulsa program yang aktif rendah selama pemrograman EPROM. Pada operasi normal, ALE dikeluarkan pada suatu kecepatan yang konstan yaitu 1/6 dari frekwensi oscilator, dan digunakan suatu timing eksternal atau untuk tujuan pengeclockan.

#### 9. PSEN (Program Strobe Enable)

Pin ini aktif rendah yang merupakan Strobe pembacaan ke program Memory Eksternal.

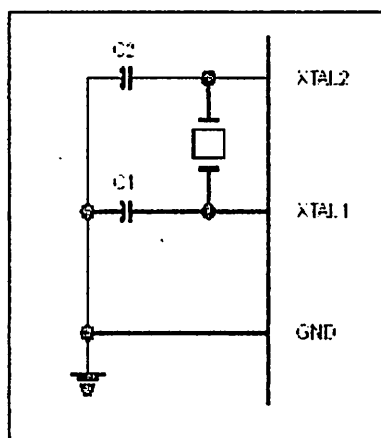
#### 10. XTAL 1 dan XTAL 2

Pin XTAL 1 merupakan pin input ke penguat oscilator pembalik dan pin XTAL 2 merupakan pin output dari penguat oscilator pembalik.

#### 11. EA/VPP (External Access/prpgraming supply Voltage)

Pin EA harus dihold rendah secara external atau dihubungkan ke VCC agar 8951 dapat mengakses kode mesin dari program memory external dengan lokasi \$0000 sampai \$ 0FFF.

Mikrokontroler AT89S51 memiliki osilator internal (on chip oscillator) yang dapat digunakan sebagai sumber pewaktuan (clock) bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal atau resonator keramik antara pin XTAL1 dan pin XTAL2 dan sebuah kapasitor ke ground. Untuk kristalnya dapat digunakan kristal dengan frekuensi sampai dengan 24 MHz. Gambar berikut menunjukkan rangkaian osilator yang digunakan.



Gambar 2.3. Rangkaian Osilator  
Sumber: [www.atmel.com](http://www.atmel.com)

## 2.2 12C BUS

12C bus merupakan suatu teknik transfer data antar IC yang melibatkan dua saluran, yaitu saluran data secara seri (SDA) dan saluran clock (SCL). Kedua saluran inilah yang dinamakan 12C bus. Teknik transfer data semacam ini dapat menghemat serta mempermudah dalam pembuatan layout PCB karena dalam menghubungkan antar IC satu dengan yang lain hanya membutuhkan dua jalur SDA dan SCL saja. IC-IC 12C itu dibedakan menjadi induk (*master*) dan anak buah (*slave*), yang dimaksud dengan induk adalah peralatan 12C yang memulai transfer data dan yang membangkitkan clock. Dalam hal ini yang bertindak sebagai *master* adalah Mikrokontroler yang bertugas mengendalikan 12C Bus.

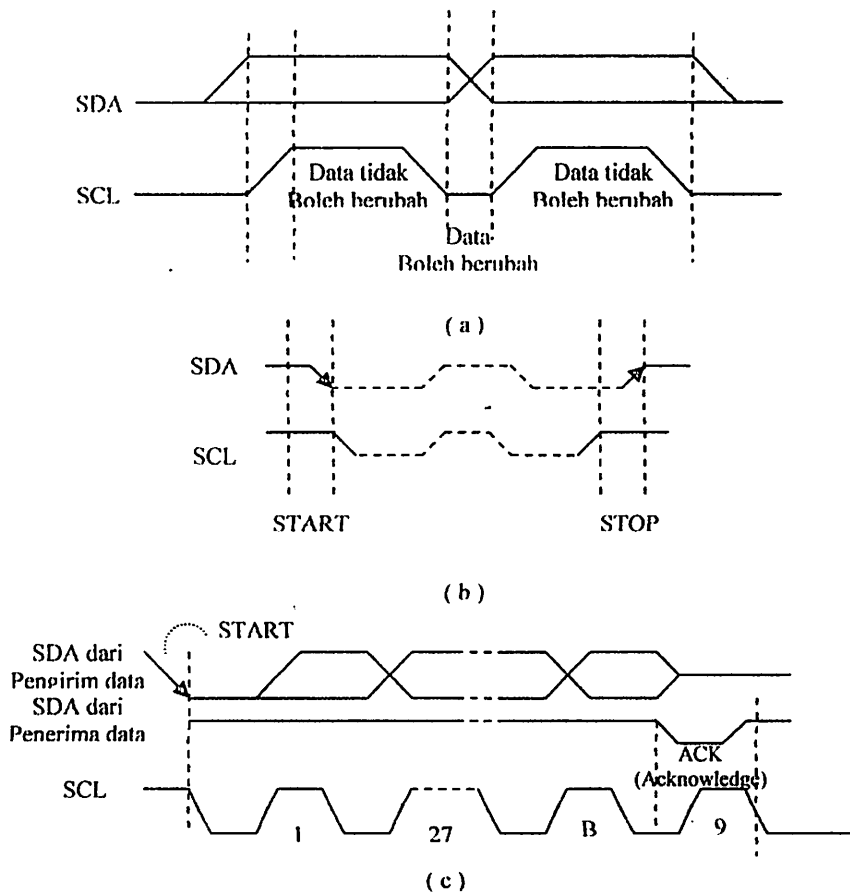
Agar data dari master bisa didistribusikan ke semua slave dengan tepat, menurut konsep I2C semua jenis IC I2C diproduksi dengan nomor group tersendiri.

### 2.2.1. Sinyal Dasar I2C

Mengingat hanya 2 saluran saja yang dipakai I2C Bus, padahal I2C Bus diharapkan bisa dipakai membentuk jaringan kecil dengan banyak peralatan I2C, maka dalam konsep I2C ditentukan sinyal dan tatacara dasar untuk memperlancar komunikasi antara peralatan I2C tersebut. Sinyal dasar I2C meliputi sinyal START, STOP dan ACK sebagai berikut :

SCK merupakan sinyal clock untuk 'mendorong' data di SDA, dalam keadaan tidak ada transfer data SDA dan SCK harus dalam keadaan '1'. Data di SDA boleh berubah hanya pada saat SCK = '0'. Isi SDA diambil peralatan I2C pada saat SCL berubah dari '1' menjadi '0'. Jika terjadi perubahan SDA pada saat SCL = '1', perubahan ini diartikan sebagai sinyal START atau STOP.

Sinyal START menandakan master akan mulai mengirim data, sinyal ini terlihat pada Gambar 2.4 b berupa perubahan tegangan SDA dari '1' menjadi '0' pada saat SCK = '1'. Sinyal STOP menandakan master akan mengakhiri komunikasi data, sinyal ini terlihat di Gambar 2.4 berupa perubahan tegangan SDA dari '0' menjadi '1' pada saat SCK = '1'



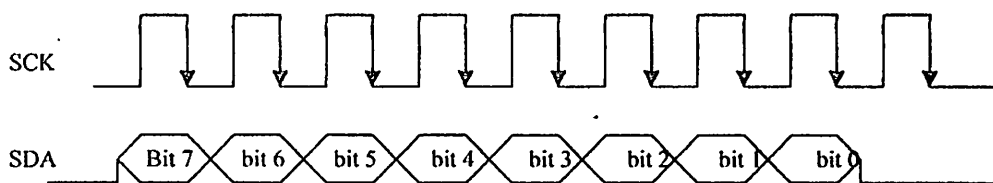
Gambar 2.4. Sinyal-Sinyal Dasar I2C  
Sumber:www.atmel.com

Ada beberapa hal yang perlu diingat apabila melakukan teknik transfer data dengan metode ini. Sinyal START dan STOP muncul pada saat awal dan akhir pengiriman 1 blok data bukan sinyal yang muncul pada awal dan akhir pengiriman 1 byte data. Apabila data sudah diterima dengan baik maka penerima data akan mengirimkan sinyal ACK yang menandakan bahwa pengiriman data telah sukses. Sinyal ACK ini akan muncul pada *clock* yang ke -9. Sinyal ACK diberikan pada setiap penerimaan 1 byte data.

### 2.2.2. Transmisi Data

Teknik transmisi data dalam 12C bus pada gambar 2.5 dapat dijelaskan sebagai berikut :

1. Clock SCK dibangkitkan oleh *master*, SDA boleh berubah pada saat SCK = '0'.
2. Data di SDA bisa dibangkitkan oleh *master* maupun *slave*, tergantung pada arah transmisi data.
3. Pengiriman data dilakukan per 1 byte, digeser serempak dengan SCK bit demi bit yang dimulai dari bit 7 sampai bit 0.
4. Pada hitungan SCK yang ke sembilan, pengirim data harus membuat SDA = '1' dengan maksud agar penerima data bisa mengirimkan sinyal ACK (= '0') sebagai tanda terima kiriman data.



Gambar 2.5. Transmisi data pada 12C bus  
Sumber: [www.atmel.com](http://www.atmel.com)

### 2.2.3. Metode Pengalamatan 12C

Karena IC 12C hanya dikendalikan lewat kaki SDA dan SCK saja, tidak ada sarana lainnya dari Mikrokontroler yang bisa dipakai untuk mengendalikan 12C,

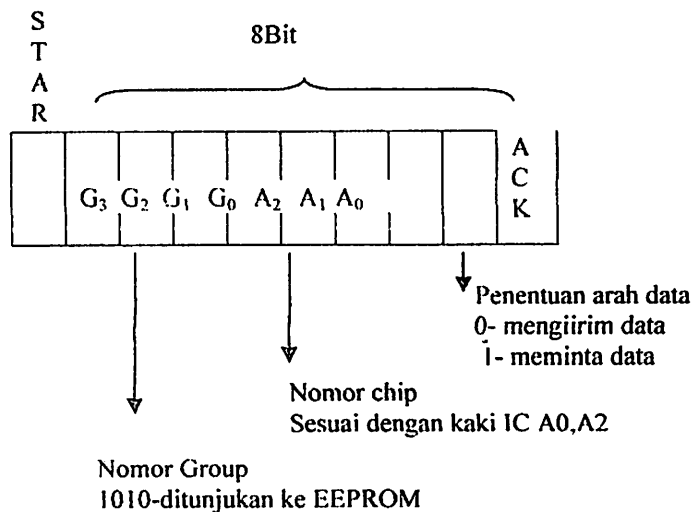
maka alamat yang dipakai untuk memilih isi IC 12C dikirimkan secara serial pula, persis seperti halnya pengiriman data. Pengalamatan dasar IC 12 C dilakukan dengan Nomor Group dan Nomor Chip. Nomor Group adalah nomor yang diberikan oleh pabrikan yang memproduksi IC 12C.

Sedangkan untuk Nomor Chip adalah nomor yang diberikan pada masing-masing chip pada kaki A0, A1, dan A2 dari masing-masing IC. Dalam IC 12C tertentu A0, A1, dan A2 tidak dihubungkan ke kaki IC tapi dipakai didalam IC untuk menomeri register/memori di dalam IC yang bersangkutan.

Setelah master 12C mengirimkan sinyal START byte pertama yang dikirim berisi nomor Group, nomor Chip dan 1 bit lagi sebagai Penentu Arah Data. Mekanisme kerja byte pertama tersebut bisa dijelaskan sebagai berikut :

1. IC-2 pada 12C Bus yang mempunyai Nomor Group sama dengan Nomor Group dalam byte pertama tersebut akan terpanggil.
2. Berikutnya IC-IC dengan Nomor Group sama tersebut akan membandingkan Nomor Chip dalam byte pertama, dalam hal ini ada 2 kemungkinan :
  - Bagi IC yang mempunyai kaki A0..A2, nomor Chip dalam byte pertama tersebut dibandingkan dengan level tegangan kaki A0...A2, bila ternyata sama maka IC bersangkutan akan meneruskan komunikasi melalui 12 Bus.
  - Bagi IC yang tidak mempunyai kaki A0...A2, nomor Chip dalam byte pertama dipakai untuk menomeri register/memori didalam IC bersangkutan.

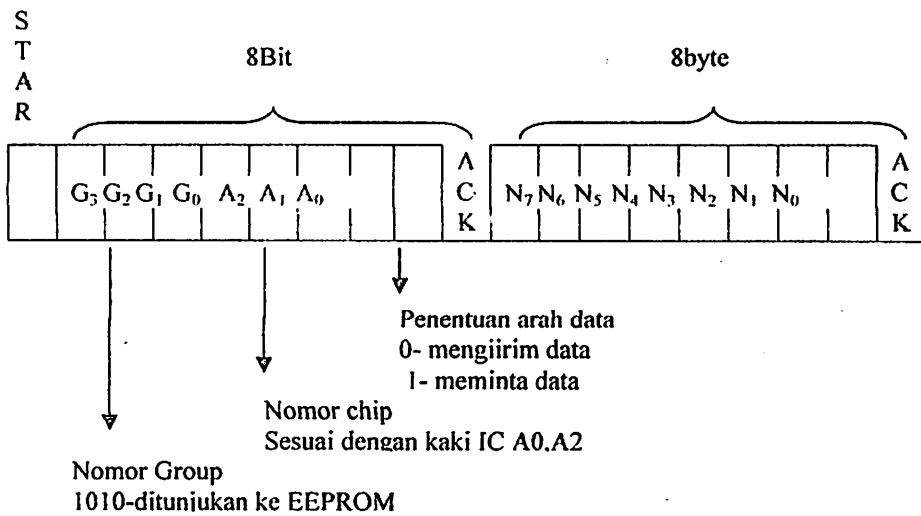
Bit Penentu Arah Data dipakai untuk memberi tahu IC 12C arah data yang dikehendaki, apakah *master* akan mengirim data atau master menghendaki kiriman data.



Gambar 2.6. Mengalami IC Secara Sederhana  
Sumber: [www.atmel.com](http://www.atmel.com)

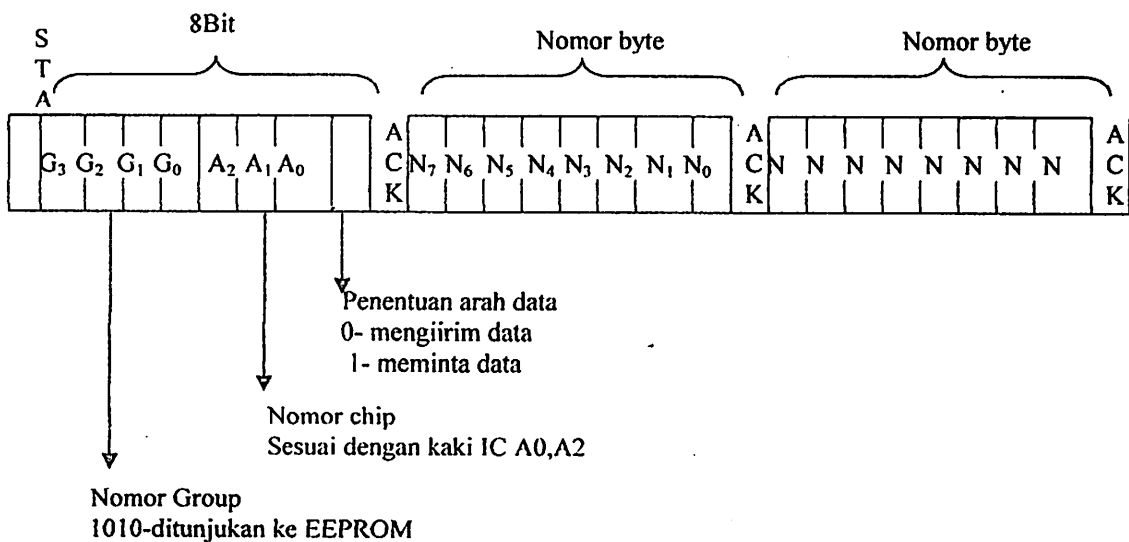
Cara pengalamatan di atas hanya bisa dipakai untuk mengalami sampai 8 byte, untuk kapasitas yang lebih besar dipakai metode pengalamatan 11 bit (8 bit N0..N7 dan 3 bit A0...A2). Pada kiriman data byte kedua berisi 8 bit alamat tambahan yang disebut sebagai Nomor Byte. Metode ini bisa dipakai untuk mengalami sampai kapasitas 2048 byte, dan dipakai pada AT24C01 sampai AT24C16.





Gambar 2.7. Mengalami Metode 11 Byte  
 Sumber: www.atmel.com

Untuk kapasitas yang lebih besar dari 2048 dipakai Metode Pengalamatan 16 bit dalam hal ini A0...A2 sepenuhnya dipakai untuk Nomor Chip, tidak dipinjam pakai untuk mengalami memori. Yang memakai cara ini adalah AT24C164, AT24C32 dan AT24C64.



Gambar 2.8. Mengalami Metode 16 Byte  
 Sumber: www.atmel.com

### 2.3 Serial RTC (Real Time Clock)

*Real Time Clock* (RTC) merupakan alat pencatat sekaligus pemberi informasi waktu. RTC yang kami gunakan disini merupakan RTC serial tipe DS1307 yang diproduksi oleh Dallas Semiconductor. Tujuan dari penggunaan serial RTC tipe DS1307 ini adalah untuk memudahkan transfer data dan juga untuk menghemat penggunaan kaki-kaki atau pin-pin pada mikrokontroler AT89S51, karena disini hanya dibutuhkan dua jalur data saja yang digunakan sebagai jalur transfer dan juga penerimaan data. DS1307 sendiri merupakan serial RTC yang mempunyai daya rendah dengan 56 *byte* NV SRAM dan pewaktu dalam format BCD (*Binary Code Decimal*). *Address* dan data di transfer secara serial melalui dua kabel *bit-directional bus*. Pewaktu terdiri dari detik, menit, jam, hari, bulan, tahun. DS1307 memiliki *built-in power sense* yang akan otomatis terhubung ke supply baterai apabila terjadi *power failure*.

#### 2.3.1. Fungsi Kaki-kaki (pin) DS1307 (ibid):

##### 1. SCL (*Serial Clock Input*):

SCL digunakan untuk sinkronisasi pertukaran data pada *serial interface*.

##### 2. SDA (*Serial Data Input/Output*)

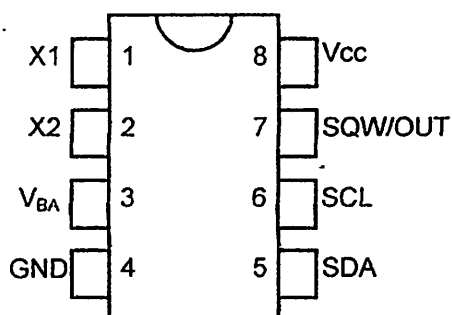
SDA merupakan pin input – output untuk dua kabel *serial interface*.

SDA ini membutuhkan resistor *pull-up eksternal*.

##### 3. SQW/OUT (*Square Wave/Output Driver*)

Ketika *enable*, SQWE di-set ke “1”. SQW/OUT akan berfungsi ketika diberi tegangan Vcc atau Vbat. SQW/OUT ini juga membutuhkan

resistor *pull-up eksternal*. Berikut ini adalah gambar kaki-kaki RTC serial DS1307 (ibid):



Gambar 2.9. Kaki-Kaki DS1307  
Sumber: [www.dalsemi.com](http://www.dalsemi.com)

### 2.3.2. Cara Kerja RTC Serial DS1307

DS1307 bekerja sebagai *slave* pada serial bus. Akses terjadi pada saat ada kondisi START dan tersedia kode identifikasi yang ada pada *register* alamat. Selanjutnya register – register dapat diakses secara berurutan sampai suatu kondisi STOP terjadi. Saat  $V_{CC}$  lebih kecil dari  $1,25 \times V_{bat}$ , DS1307 akan mengakhiri suatu akses yang sedang terjadi dan mereset *address counter*. Input yang masuk tidak akan diakses untuk mencegah kesalahan penulisan data. Saat  $V_{CC} < V_{bat}$ , DS1307 akan switch ke mode baterai arus rendah. Ketika *power-up* DS1307 akan *switch* dari baterai ke  $V_{CC}$  saat  $V_{CC} > V_{bat} + 0,2V$  dan akan mengakses input yang masuk ketika  $V_{CC} > 1,25 \times V_{bat}$ .

### 2.3.3. Peta Alamat RAM dan RTC

Peta alamat untuk RTC dan register RAM dari DS1307 ditunjukkan oleh gambar 2.10. Register-register RTC berada pada alamat 00h-07h. Sedangkan

register-register RAM berada pada alamay 08h- 3Fh. Selama aksese *multi-byte*, saat *adres pointer* mencapai alamat 3Fh, alamat akhir RAM, maka akan *reset* kembali ke 00h.

00H	SECOND
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
	56x8
3FH	

Gambar 2.10. Peta alamat DS1307  
Sumber: [www.dalsemi.com](http://www.dalsemi.com)

#### 2.3.4. Waktu dan Kalender

Waktu dan kalender dapat diakses dengan membaca *byte register* yang tepat. Register –register RTC ditunjukkan pada gambar 2.8. Waktu dan kalender dapat di-set atau diinisialisasi dengan menulis byte register yang tepat. isi dari register waktu dan kalender dalam format BCD. Bit ke 7 dari register 0 adalah bit CH. Bit ini akan *disable* osilator apabila diberi logika ‘1’. Dan akan *enable* kan osilator apabila diberi logika ‘0’. DS1307 dapat dijalankan dalam mode 12 jam atau 24 jam. Bit ke 6 dari register jam sebagai bit mode 12 atau 24 jam. Saat diberi logika *high* ‘1’, maka mode 12 jam yang dipilih. Pada mode 12 jam , bit ke 5 adalah bit AM/PM yang pada saat high menunjukkan PM. Pada mode 24 jam , bit ke 5 adalah bit 10 jam yang kedua (20-23 jam).

Pada 12C bus, saat START, waktu ditransfer ke register kedua. Informasi waktu dapat dibaca dari sini, sementara *clock* terus berjalan. Ini akan mempermudah proses pembacaan *register* pada saat pembaharuan dari register utama sedang dibaca.

BIT 7				BIT 0				
CH	10 SECONDS			SECONDS				00-59
X	10 MINUTES			MINUTES				00-59
X	12 24	10 HR AMP	10 MIN	HOURS				01-12 00-59
X	X	X	X	X	DAY			1-7
X	X	10 DATE		DATE				01-31 01-31 01-31
X	X	X	10 MONTH	MONTH				01-12
10 YEAR				YEAR				00-99
OUT	X	X	SQWE	X	X	MS1	MS0	

Gambar 2.11. Register waktu RTC  
Sumber: [www.dalsemi.com](http://www.dalsemi.com)

### 2.3.5. Register Kontrol

Register kontrol DS1307 digunakan untuk mengontrol kerja dari pin SQW/OUT.

Tabel 2.1. Register kontrol DS1307

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Sumber: [www.dalsemi.com](http://www.dalsemi.com)

**OUT (Output Kontrol):** bit mengontrol level output pin SQW/OUT saat output gelombang kotak tidak aktif (disable). Bila SQWE = 0 logika pada pin SQW/OUT = 1 jika OUT = 1 dan SQW/OUT = 0 jika OUT = 0.

**SQWE (Square Wave Enable):** saat bit ini di-set '1' maka akan mengaktifkan output osilator. Frekwensi dari output gelombang kotak tergantung dari nilai bit-bit RS0 dan RS1. Bersamaan dengan di-set nya output gelombang kotak ke 1 Hz, register clock akan memulai proses pembaharuan pada saat transisi turun dari gelombang kotak.

**RS (Rate Select):** bit ini mengontrol frekwensi dari output gelombang kotak saat output gelombang kotak aktif. Tabel berikut berisi sejumlah frekwensi yang dapat dipilih menggunakan bit RS.

Tabel 2.2. Frekwensi Output Gelombang Kotak

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096 kHz
1	0	8.192kHz
1	1	32.768 kHz

Sumber: [www.dalsemi.com](http://www.dalsemi.com)

### 2.3.6. Serial Data Bus

DS1307 merupakan alat yang menggunakan system *bit-directional*. Perangkat yang bertindak sebagai pengirim data ke system bus disebut sebagai *transmitter* sedangkan yang bertindak sebagai penerima disebut *receiver*. Piranti yang mengontrol pesan disebut *master*. Sedangkan yang dikontrol oleh *master* disebut *slave*. System bus harus dikontrol oleh sebuah master yang menimbulkan sinyal *clock* (SCL), mengontrol akses bus,serta menimbulkan kondisi START dan STOP. DS1307 bertindak sebagai slave pada system 12C bus.

Ada beberapa syarat yang harus dipenuhi agar data yang dikirim tidak akan hilang atau terkorupsi:

1. Data yang ditransfer hanya dapat diinisialisasi pada saat bus dalam keadaan tidak sibuk.
2. Selama proses transfer data, jalur data harus tetap stabil meskipun *clock* dalam keadaan HIGH. Perubahan pada jalur data saat jalur *clock high* akan dianggap sebagai sinyal kontrol.

Berdasarkan hal-hal di atas, kondisi dari bus dapat didefinisikan sebagai berikut:

**Bus tidak sibuk:** kedua jalur data dan clock dalam kondisi HIGH.

**Start data transfer:** perubahan kondisi pada jalur data dari HIGH ke LOW, saat clock HIGH didefinisikan sebagai kondisi START.

**Stop data transfer:** perubahan kondisi pada jalur data dari LOW ke HIGH, saat clock HIGH didefinisikan sebagai kondisi STOP.

**Data valid:** kondisi dari jalur data menunjukkan data *valid*, ketika setelah kondisi START jalur data tetap stabil selama durasi periode HIGH sinyal *clock*. Data pada jalur data harus dirubah pada periode LOW dari sinyal *clock*. Ada satu pulsa *clock* untuk tiap satu bit data.

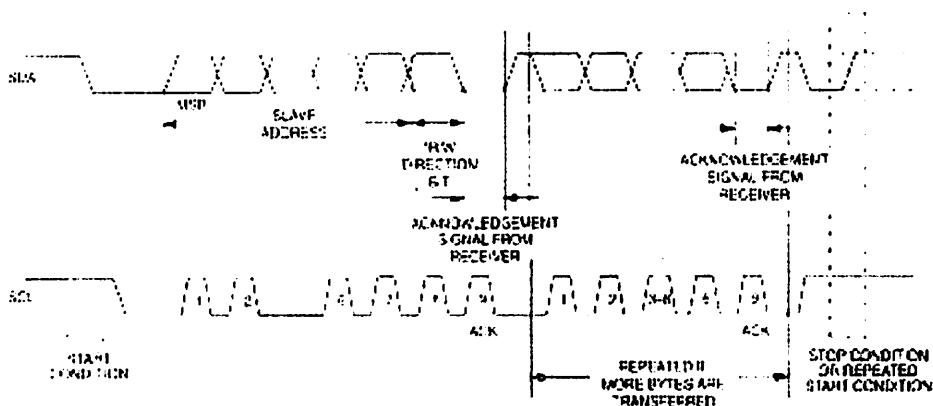
Setiap data yang ditransfer diinisialisasi dengan sebuah kondisi START dan diakhiri dengan sebuah kondisi STOP. Jumlah dari byte yang dikirim antara kondisi START dan STOP tidak terbatas jumlahnya, dan ditentukan oleh *master*.

Data ditransfer per-byte, bit *acknowledge* diterima pada bit ke -9. Dalam spesifikasi I2C bus bekerja dalam dua mode, yaitu mode *reguler* (100KHz) dan *fast mod* (400KHz). DS1307 beroperasi pada mode *reguler* (100KHz).

Yang dimaksud *acknowledge* di sini adalah, tiap penerima data yang sudah dialamatkan, wajib untuk membangkitkan sebuah *acknowledge* setiap penerimaan tiap satu byte data.

Penerima sinyal *acknowledge* harus memberikan kondisi low pada jalur SDA selama proses pulsa *clock acknowledge* seperti halnya jalur SDA yang terus stabil pada kondisi LOW selama periode HIGH dari pulsa *clock acknowledge*. Master harus memberikan sebuah sinyal untuk mengakhiri suatu pengiriman data tanpa membangkitkan bit *acknowledge* pada byte terakhir yang telah keluar dari *slave*. *Slave* harus meninggalkan jalur data HIGH untuk membuat *master* membangkitkan kondisi STOP.

### 2.3.7. Transfer Data Pada Serial Bus



Gambar 2.12. Transfer data pada I2C bus  
Sumber: [www.dalsemi.com](http://www.dalsemi.com)



Berdasarkan dari bit R/W ada dua tipe tranfer data,yaitu:

1. Pengirim data dari master transmitter ke slave register

Byte pertama yang dikirim oleh master adalah alamat slave. Diikuti oleh sejumlah byte data. Kemudian slave membalas dengan mengirim sebuah sinyal *acknowledge*. Setelah penerimaan satu byte data. Yang pertama dikirim adalah MSB dari data tersebut.

2. Pengirim data dari slave transmitter ke master receiver

Byte pertama yang dikirim oleh master adalah slave. Kemudian slave membalas dengan mengirim sebuah sinyal *acknowledge*. Dan diikuti dengan pengiriman sejumlah byte data oleh slave. lalu master membalas dengan mengirim sebuah sinyal *acknowledge* setelah penerimaan keseluruhan byte data. Pada akhir penerimaan byte terakhir sinyal *not acknowledge* dikirim.

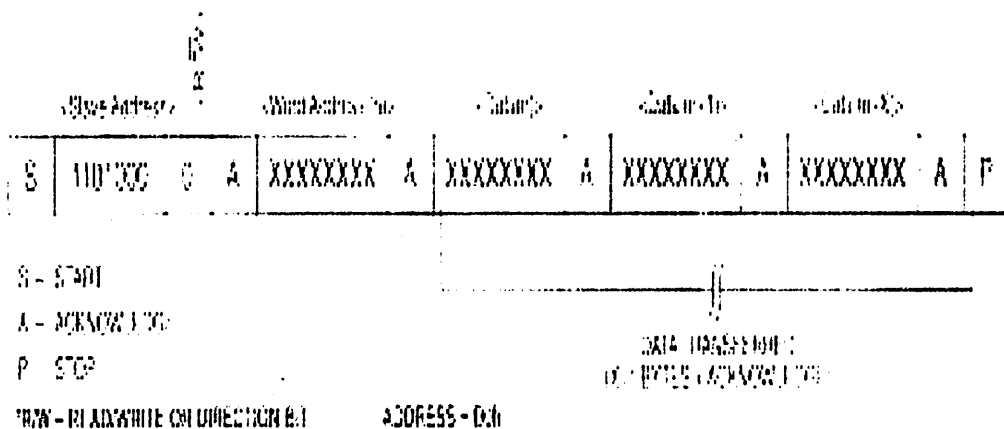
Master membangkitkan keseluruhan pulsa clock dan kondisi START dan STOP. Proses pengiriman data diakhiri dengan sebuah kondisi STOP suatu perulangan kondisi START.

DS1307 dapat beroperasi dalam dua mode :

1. Mode Tulis

Serial data dan clock diterima melalui SDA dan SCL. Satu bit *acknowledge* dikirim setiap akhir dari penerimaan satu byte data. Kondisi START dan STOP dianggap sebagai awal dan akhir dari serangkaian proses pengiriman

data. Proses pengalamatan dilakukan setelah penerimaan alamat *slave* dan bit tujuan. Byte pertama yang diterima setelah kondisi START adalah byte alamat yang ditimbulkan oleh *master*. Byte alamat berisi 7 bit alamat DS1307, 1101000, diikuti oleh bit R/W, untuk mode tulis R/W = 0. Setelah proses penerimaan dan *decoding* byte alamat, maka *master* mengeluarkan sinyal *acknowledge* pada jalur SDA. Setelah DS1307 menerima sinyal *acknowledge* dan bit tulis, *master* mengirim register alamat ke DS1307. Ini akan mengeset register pointer pada DS1307. Selanjutnya *master* akan memulai mengirim setiap byte data ke DS1307 dengan satu sinyal *acknowledge* dari DS1307 untuk tiap satu byte data yang telah diterima. *Master* akan membangkitkan kondisi STOP untuk mengakhiri pengiriman data.

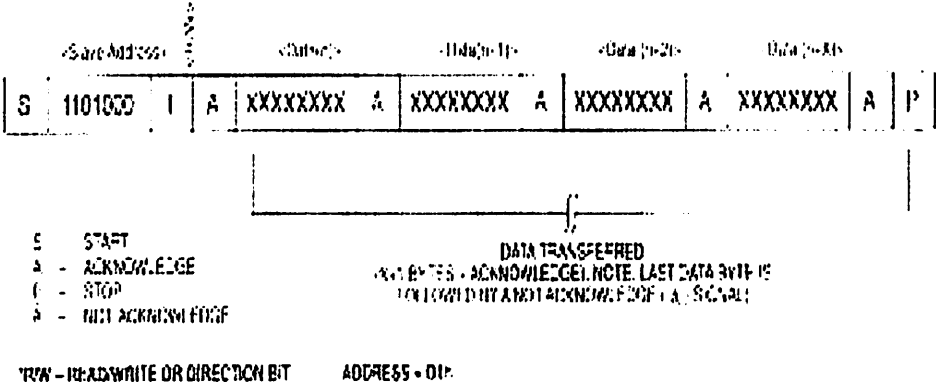


Gambar 2.13. Mode Tulis  
Sumber: [www.dalsemi.com](http://www.dalsemi.com)

### 1. Mode Baca

Byte pertama diterima dan ditangani sama seperti pada mode tulis. Hanya saja bit R/W di-set berkebalikan dengan mode tulis. Data dikirim oleh DS1307 melalui SDA saat SCL diberi input sinyal *clock*. Kondisi START dan STOP

merupakan awal dari akhir serangkaian pengiriman serial. Pertama diterima setelah kondisi START adalah byte alamat yang ditimbulkan oleh *master*. Byte alamat berisi 7 alamat DS1307, 1101000, diikuti oleh R/W dimana untuk mode baca R/W =1. Setelah proses penerimaan dan *decoding byte* alamat, DS1307 akan mengirim sinyal *acknowledge* ke jalur SDA. DS1307 kemudian mulai mengirim data dimulai dengan register alamat yang ditunjuk oleh *register pointer*. Jika register pointer belum ditulisi sebelumnya, sebelum proses inialisasi dari mode baca, maka alamat yang pertama dibaca adalah yang terakhir disimpan di *register pointer*. Untuk mengakhiri proses baca, DS1307 harus menerima sinyal “*not acknowledge*” (NACK) dari *master*.



Gambar 2.14. Mode Baca  
 Sumber: www.dalsemi.com

**2.4 Serial EEPROM**

Atmel memproduksi Serial EEPROM jenis 12C dengan kode AT24Cxx, AT merupakan kode pabrik Atmel. 24 menandakan bahwa IC tersebut adalah Serial EEPROM, sedangkan xx merupakan angka yang mengindikasikan

kapasitas Serial EEPROM itu dalam satuan KiloBit. Misalakan AT24C08, IC ini merupakan IC SEEPROM 12C 8 KiloBit (1 KiloByte).

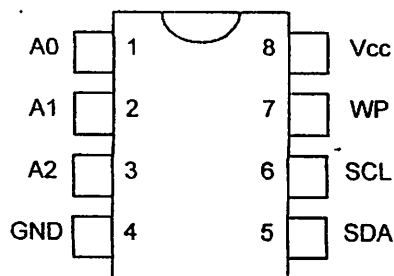
Keluarga AT24Cxx terdiri dari 9 jenis dan memiliki kapasitas memori yang berbeda-beda tetapi memiliki konfigurasi pin-pin yang sama.

Gambar 2.3. Konfigurasi pin-pin IC AT24Cxx

Type IC	Kapasitas Byte	Penomoran Chip	Max. Chip Per Bus
AT24C01	128	Tidak ada	1
AT24C01A	128	A0.A1.A2	8
AT24C02	256	A0.A1.A2	8
AT24C04	512	A1.A2	4
AT24C08	1024	A2	2
AT24C16	2048	Tidak ada	1
AT24C164	2048	A0.A1.A2	8
AT24C32	4096	A0.A1.A2	8
AT42C64	8192	A0.A1.A2	8

Sumber: [www.atmel.com](http://www.atmel.com)

#### 2.4.1 Pin-pin IC AT24Cxx



Gambar 2.15. Konfigurasi pin-pin IC AT24Cxx  
Sumber: [www.atmel.com](http://www.atmel.com)

1. Pin 1-3 (A0,A1,dan A2) merupakan fasilitas untuk penomoran chip,hal ini diperlukan kalau dalam satu rangkaian dipakai lebih dari satu IC SEEPROM sejenis.
2. Pin no.5 (Pin SDA) dan pin no.6 (pin SCL) kedua pin inilah yang membentuk 12C Bus.
3. Pin no.7 (WP-Write Protect) merupakan pin yang digunakan untuk melindungi isi yang disimpan dalam IC SEEPROM. Apabila pin WP diberi kondisi '1' maka isi IC SEEPROM tidak dapat diganti. Maka untuk dapat mengganti isi dari IC SEEPROM pin WP harus dalam kondisi '0'.
4. Pin no.8 (VCC) merupakan pin untuk jalur tegangan supply positif (5 volt dc).
5. Pin no.4 (GND) merupakan pin ground IC SEEPROM.

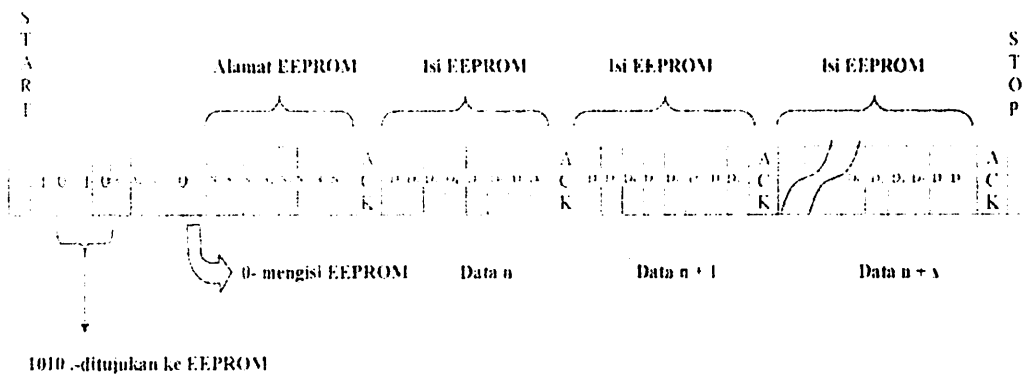
Meskipun demikian, A0,A1, dan A2 tidak selalu ada pada IC anggota AT24Cxx, akibatnya jumlah IC yang boleh dipasang pada 12C Bus tidak sama. Seperti yang telah ditunjukkan dalam tabel 2. Untuk IC AT24C01 sampai AT24C16 menggunakan metode pengalamatan 11 bit, sedangkan untuk IC AT24C164 sampai AT25C64 menggunakan metode pengalamatan 16 bit.

#### 2.4.2. Proses pengisian SEEPROM AT24Cxx

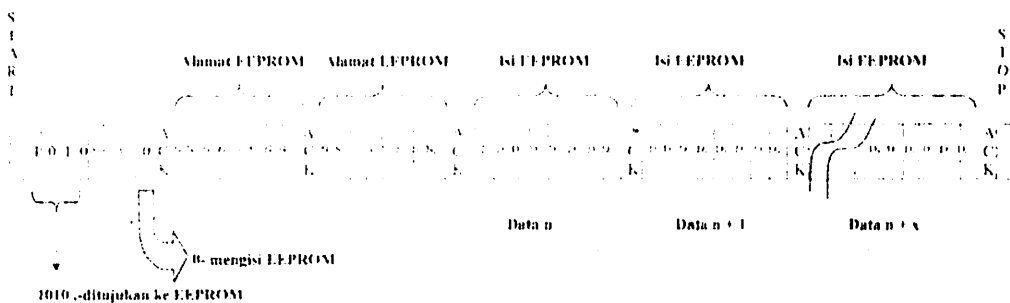
Dalam satu proses pengisian data SEEPROM sinyal START dan sinyal STOP masing-masing cukup dikirim satu kali saja. Yakni sinyal START dipakai

untuk mengawali proses dan sinyal STOP dipakai untuk mengakhiri proses. Kedua sinyal itu bukanlah awalan dan akhiran dari pengiriman 1 byte data.

Setelah mengirimkan alamat EEPROM yang akan diisi, mikrokontroler mengirim data yang diisikan ke AT24Cxx. Setiap kali selesai menyimpan data AT24Cxx dengan sendirinya menaikkan alamat EEPROM yang disimpannya, dengan demikian kiriman data selanjutnya akan disimpan ke memori berikutnya. Proses pengisian ini akan berhenti setelah mikrokontroler menutup komunikasi ini dengan sinyal STOP.



Gambar 2.16. Metode 11 bit Addressing  
Sumber : [www.atmel.com](http://www.atmel.com)



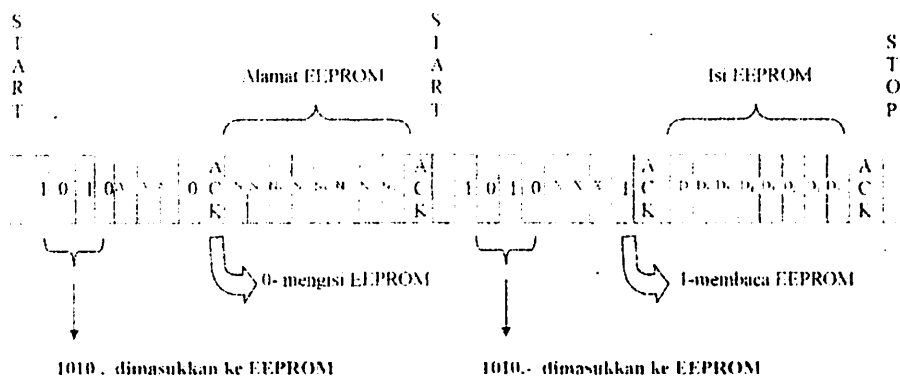
Gambar 2.17. Metode 16 bit addressing  
Sumber : [www.atmel.com](http://www.atmel.com)

Pada dasarnya kedua metode itu sama perbedaannya terletak pada:

1. Dalam metode pengalamatan 11 bit, alamat SEEPROM dikirim di A0..A2 dalam pengiriman byte pertama dan NO..N7 dalam pengiriman byte kedua
2. Dalam metode pengalamatan 16 bit, alamat SEEPROM dikirim di NO..N15 dalam pengiriman byte kedua dan NO..N7 dalam pengiriman byte ketiga.

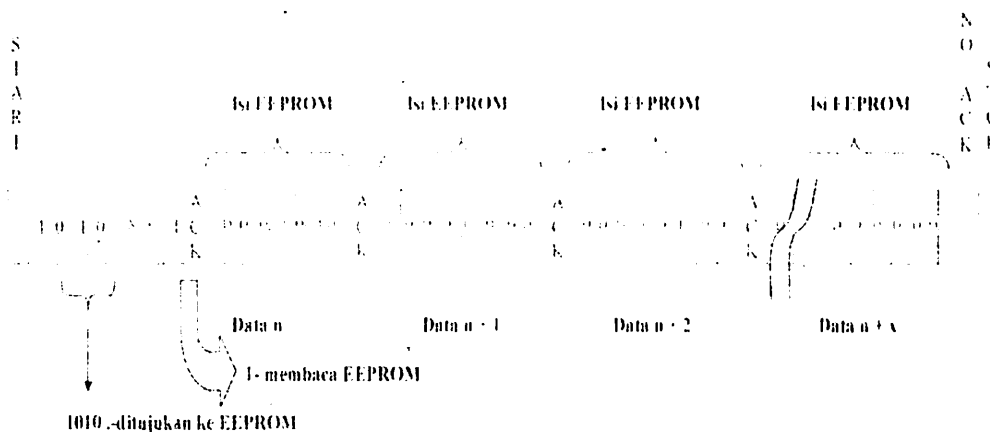
#### 2.4.3 Proses Pembacaan AT24Cxx

Pada proses pembacaan mula-mula Mikrokontroler mengirimkan alamat EEPROM yang akan dibaca isinya. Proses ini mirip dengan bagian awal pengisian EEPROM yang di bahas diatas. Setelah itu Mikrokontroler mengirim sinyal START sekali lagi, disusul dengan perintah dengan membaca isi EEPROM yang sesungguhnya. Selesai membaca isi SEEPROM Mikrokontroler menutup komunikasi dengan mengirimkan sinyal STOP.



Gambar 2.18. Proses Pembacaan EEPROM  
 Sumber : www.atmel.com

Pembacaan isi EEPROM bisa pula dilakukan dengan banyak byte sekaligus, setiap membaca 1 byte isi EEPROM, Mikrokontroler menjawab dengan sinyal ACK, pada saat pembacaan byte yang terakhir komunikasi ditutup dengan sinyal NACK dan STOP.



Gambar 2.19. Proses Pembacaan satu blok EEPROM  
 Sumber : www.atmel.com

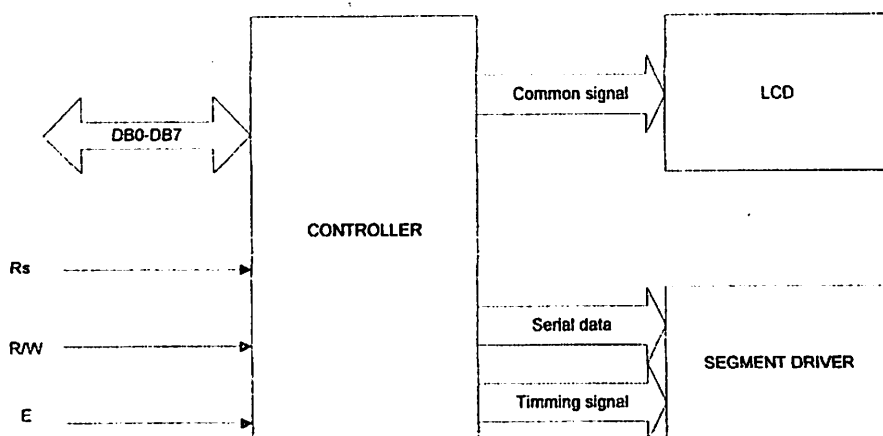


## 2.5 Liquid Crystal Display ( LCD )

### 2.5.1. Konfigurasi LCD

Liquid Crystal Display adalah modul tampilan berkonsumsi daya yang relatif rendah dan terdapat sebuah kontroller CMOS di dalamnya. Kontroller tersebut sebagai pembangkit karakter dari ROM/ RAM dan display data RAM. Semua fungsi tampilan dikontrol oleh suatu intruksi dan modul LCD dapat dengan mudah untuk diinterfacekan dengan mikroprocessor/ mikrokontroller. Input yang diperlukan untuk mengendalikan modul ini berupa bus data yang termultipleks dengan bus alamat dan 3 bit sinyal kontrol. Pengendali dot matrik LCD dilakukan secara internal oleh kontroller internal pada modul LCD sendiri.

LCD merupakan suatu bentuk kristal cair yang akan beremulsi apabila dikenakan tegangan padanya. Tampilannya ini berupa dot matrik 5 x LCD sehingga jenis huruf yang dapat ditampilkan akan lebih banyak dan lebih baik resolusinya jika dibandingkan dengan 7 segment.



Gambar 2.20. Diagram Blok LCD  
Sumber: [www.Delta-Elektronik.com](http://www.Delta-Elektronik.com)

LCD tipe M1632 memiliki ciri-ciri sebagai berikut :

- LCD ini terdiri dari 32 karakter dengan 2 baris masing-masing 16 karakter dengan display dot matrik 5 x 7
- Karakter generator ROM dengan 192 tipe karakter
- Karakter generator RAM dengan 8 tipe karakter
- 80 x 8 bit display data RAM
- Dapat diinterfacekan ke MPU 8 atau 4
- Dilengkapi fungsi tambahan : display clear, cursor home, display ON/OFF, cursor ON/OFF, display character blink, cursor shift, dan display shit
- Internal data
- Internal otomatis, reset pada saat power ON
- +5 volt PSU tunggal

Tabel 2.4. Konfigurasi pin-pin

NO	SYMBOL	LEVEL	FUNCTION
1	Vss	-	0 Grond
2	Vcc	-	5 V + 10%
3	Vec	-	LCD Drive
4	Rs	H/L	H : Data Input L : Instruksi Input
5	R/W	H/L	H : read L : write
6	E	H/L	Enable signal
7-14	DB0-DB7	H/L	Data Bus

15	Light LCD	-	Menyalakan lampu LCD max 200 mA
16	Light LCD	-	Ground

Sumber: [www.Delta-elektronik.com](http://www.Delta-elektronik.com)

## 2.5.2. Instruksi Operasi Dasar

### 2.5.2.1 Register

Kontroller dari LCD mempunyai dua buah register 8 bit yaitu register instruksi (IR) dan register data (RD). IR menyimpan instruksi seperti display clear, cursor shift dan display data (DD RAM) serta character generator (CG RAM). DR menyimpan data untuk ditulis di DD RAM atau CG RAM atau membaca data dari DD RAM atau CG RAM. Ketika data ditulis ke DD RAM atau CG RAM maka DR secara otomatis menulis data ke DD RAM atau CG RAM dan data pada DD RAM atau CG ram hendak dibaca maka alamat data ditulis pada IR sedangkan data alamat dimasukkan melalui DR dan mikroprosessor membaca data dari DR.

Tabel 2.5. Register Seleksi

Rs	R/W	OPERASI
0	0	Seleksi IR, IR Write Display Clear
0	1	Busy Flag (DB7) @counter (DB0-DB7)Reas
1	0	Seleksi DR, DR Write
1	1	Seleksi DR,DR Read

Sumber: [www.Delta-elektronik.com](http://www.Delta-elektronik.com)

### 2.5.2.2. Busy flag

Busy flag menunjukkan bahwa modul siap untuk menerima instruksi selanjutnya. Sebagaimana yang terlihat pada tabel register seleksi sinyal akan melalui DB7. Jika RS = 0 dan R/W = 1. Jika bernilai 1 maka

modul sedang melakukan kerja internal dan instruksi tidak dapat diterima. Sehingga status dari flag ini harus diperiksa sebelum melaksanakan instruksi selanjutnya.

#### 2.5.2.3. Address counter

AC menunjukkan lokasi memori dalam modul LCD. Pemilihan lokasi alamat itu diberikan lewat register instruksi (IR). Ketika data ada pada A, maka AC secara otomatis menaikkan atau menurunkan alamat tergantung dari Entry Mode Set.

#### 2.5.2.4. Display Data RAM (DD RAM)

Pada LCD masing-masing line mempunyai range alamat tersendiri. Alamat ini diekspresikan dengan bilangan Hexadesimal. Untuk itu 1 range alamat berkisar 00H-0FH sedangkan untuk line 2 range alamat berkisar antara 40H-4FH.

#### 2.5.2.5. Character Generator ROM (CG ROM)

CG ROM mempunyai tipe dot matrik 5 x 7 dan pada LCD telah tersedia ROM sebagai pembangkit character dalam kode ASCII.

#### 2.5.2.6. Character Generator RAM (CG RAM)

CG RAM digunakan untuk pembuatan karakter tersendiri melalui program.

Tabel 2.6. Fungsi Terminal pada LCD

Nama signal	Jml Term	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Sebagai lalulintas data dan instruksi ke atau dari MPU Low Byte
DB4-DB7	4	I/O	MPU	Sebagai lalulintas data atau instruksi 2 arah upper byte. DB7 sebagai busy flag
E	1	I	MPU	Sinyal start (read/write)
R/W	1	I	MPU	Seleksi sinyal 0 = write 1 = read
Rs	1	I	MPU	Seleksi register
VLC	1	-	PS	0 = instruksi reg (wr) Busy flag addr counter (rd) 1 = data reg (wr dan rd)
7	1	-	PS	Mengatur tampilan LCD
Vss	1	-	PS	+5volt

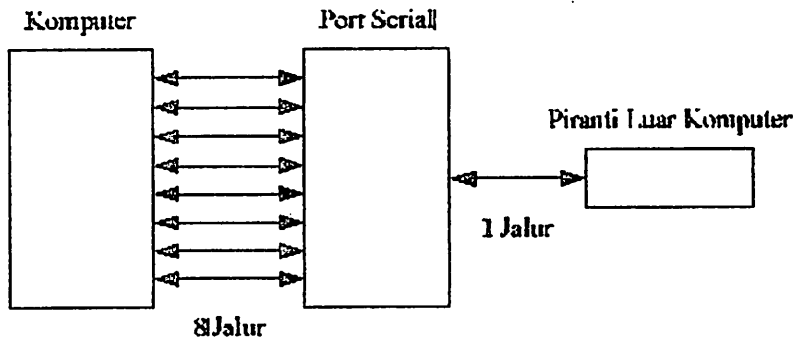
Sumber: [www.Delta-Elektronik.com](http://www.Delta-Elektronik.com)

## 2.6. Port Serial DB-9 dan RS-232

Salah satu perangkat penunjang proses komunikasi data serial pada terminal komputer dengan dunia luar adalah port serial. Seperti halnya port paralel yang sering digunakan untuk keperluan interfacing, maka port serial juga merupakan gerbang I/O dari sebuah komputer. Hanya saja di dalam port serial akan dilakukan perubahan bentuk data paralel menjadi data serial pada jalur keluaran dan sebaliknya data serial menjadi data paralel pada jalur masukan.

Port serial adalah piranti untuk mengubah informasi parallel (datang dari komputer), menjadi informasi serial (pergi ke modem, saluran telepon, maupun media yang lain, misalnya kontroller). Demikian juga sebaliknya, mengubah data

serial dari media luar komputer menjadi paralel data dalam komputer. Gambar dibawah menunjukkan Prinsip Dasar Port Serial.



Gambar 2.21 Prinsip Dasar Port Serial

Sumber: Rizqi Firmansyah, 2006, Robot Pengelasan ( Hardware ), Surabaya, Proyek Akhir Politeknik Elektronika Surabaya.

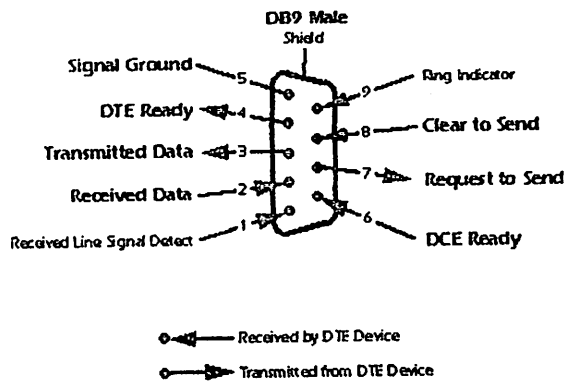
Jadi pada komunikasi serial hanya ada satu bit informasi yang ditransmisikan melalui port serial pada suatu saat. Dengan demikian secara otomatis hanya ada satu kabel yang diperlukan dalam proses transmisi data ini. Dikatakan pula bahwa komunikasi data secara serial memiliki kesederhanaan dalam hal pengkabelannya. Hal tersebut dapat terwujud tidak lain karena adanya port serial yang merupakan piranti dari sebuah komputer yang sanggup mengubah bentuk data paralel menjadi data serial dan juga sebaliknya dari data serial menjadi data paralel.

RS-232C merupakan salah satu standard interface dalam proses transfer data antar komputer terutama dalam bentuk serial transfer. RS- 232C merupakan kependekan dari Recommended Standard number 232 revision-C. Standard ini dibuat oleh Electronic Industry Association (EIA), untuk interface antara peralatan terminal data dan peralatan komunikasi data, dengan menggunakan data biner sebagai data yang ditransmisi. RS-232C adalah suatu serial data interface

standar yang dikeluarkan oleh EIA. Standarisasi meliputi konektor, fungsi serta posisi tiap pin timing (pewaktu) dan level tegangan atau arus. Standar ini juga berisikan karakteristik sinyal listrik, karakteristik mekanik dari cara operasional rangkaian fungsional.

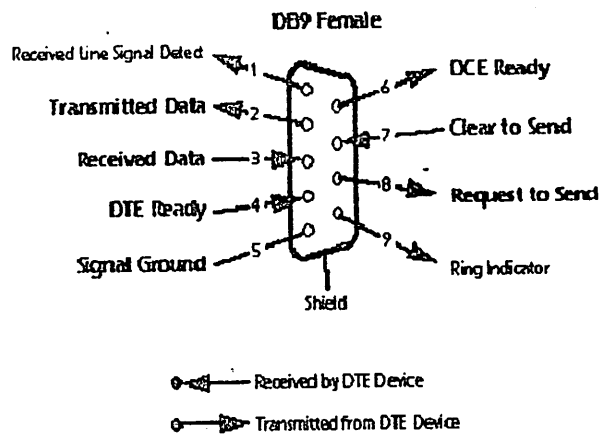
Dalam transmisi data, sinyal penanda adanya bit yang dikirim didefinisikan sebagai tegangan lebih dari -3 volt terhadap ground. Sebuah space didefinisikan sebagai tegangan yang lebih positif dari +3 volt terhadap ground. Sedangkan sinyal tidak didefinisikan bila diantara +3 volt dan -3 volt. Oleh karena itu, interfacing untuk level tegangan dari suatu sistem minimum yang mempunyai peralatan LSI komunikasi data serial menggunakan RS-232C sedikit mengalami kesulitan. Hal ini dikarenakan tegangan catu daya yang dipergunakan dalam sistem itu biasanya adalah single voltage +5 volt. Padahal keluaran RS-232 harus dapat mengeluarkan tegangan simetris (plus-minus). Sangat boros, apabila menggunakan tegangan catu simetris, tetapi hal ini dapat diatasi dengan menggunakan IC DC to DC Converter khusus seperti MAX 232. Konektor standard hubungan antara RS-232 ini menggunakan konektor DB-9

Proses transfer secara serial menggunakan RS-232 antara 2 buah terminal biasanya memerlukan sebuah DTE (*Data Terminal Equipment*) untuk masing-masing terminal, dan sebuah DCE (*Data Communicationn Equipment*).



Gambar 2.22 DTE Device Connector

Sumber: Wisma Wulandari, 2007, Perencanaan dan Pembuatan Alat Pengendali Kamera Menggunakan Aplikasi PC, Malang, Tugas Akhir Teknik Elektronika D3 ITN Malang.



Gambar 2.23 DCE Device Connector

Sumber: Wisma Wulandari, 2007, Perencanaan dan Pembuatan Alat Pengendali Kamera Menggunakan Aplikasi PC, Malang, Tugas Akhir Teknik Elektronika D3 ITN Malang.

Data yang ditransfer dari satu terminal akan diterima oleh terminal lainnya, dan demikian juga sebaliknya melalui perangkat peralatan di atas. Tabel berikut menjelaskan deskripsi pin out untuk DB-9 dan fungsi masing-masing pin.

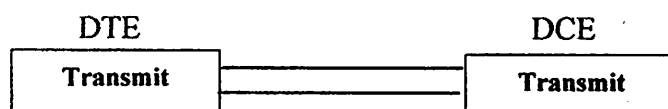


Tabel 2.7. Deskripsi Pin DB-9 (male) dan Fungsinya

Pin	Singkatan	Keterangan	Fungsi
1	CD	<i>Carrier Detect</i>	Saat modem mendeteksi suatu 'carrier' dari modem lain (di tempat lain) maka sinyal ini akan diaktifkan.
2	RD	<i>Receive Data</i>	Untuk penerimaan data serial (RXD)
3	TD	<i>Transmit Data</i>	Untuk pengiriman data serial (TXD)
4	DTR	<i>Data Terminal Ready</i>	Kebalikan dari DSR, memberitahukan bahwa UART siap melakukan komunikasi
5	SG	<i>Signal Ground</i>	Ground
6	DSR	<i>Data Set Ready</i>	Memberitahu UART bahwa modem siap melakukan hubungan komunikasi
7	RTS	<i>Request To Send</i>	Sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran data
8	CTS	<i>Clear To Send</i>	Digunakan untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data
9	RI	<i>Ring Indikator</i>	Akan aktif jika modem mendeteksi adanya saluran telepon

Sumber: Wisma Wulandari, 2007, Perencanaan dan Pembuatan Alat Pengendali Kamera Menggunakan Aplikasi PC, Malang, Tugas Akhir Teknik Elektronika D3 ITN Malang.

Jenis data yang akan ditransfer adalah dalam bentuk data biner (bit per bit transfer) dengan satuan baud untuk kecepatan proses transfernya (bit per detik).



Gambar 2.24 Tranfer Data Antara DTE dengan DCE

Sumber: Wisma Wulandari, 2007, Perencanaan dan Pembuatan Alat Pengendali Kamera Menggunakan Aplikasi PC, Malang, Tugas Akhir Teknik Elektronika D3 ITN Malang.

Konsep *interface* antara DTE dan DCE dilakukan dengan cara sebagai berikut

- Ketika DTE ingin mengirim data, sebuah protocol yaitu RTS dikirimkan untuk memberitahu DCE.
- Pada saat itu input RTS pada DCE menjadi aktif.
- Jika DCE mampu menerima data, maka ia akan membalasnya dengan mengirim CTS.
- Begitu DTE menerima balasan, input CTS-nya diaktifkan.
- Pengiriman data dilakukan melalui TxD.
- Penerimaan data dilakukan melalui RxD.
- Proses tersebut dilakukan secara berulang-ulang sampai semua data selesai ditransfer.

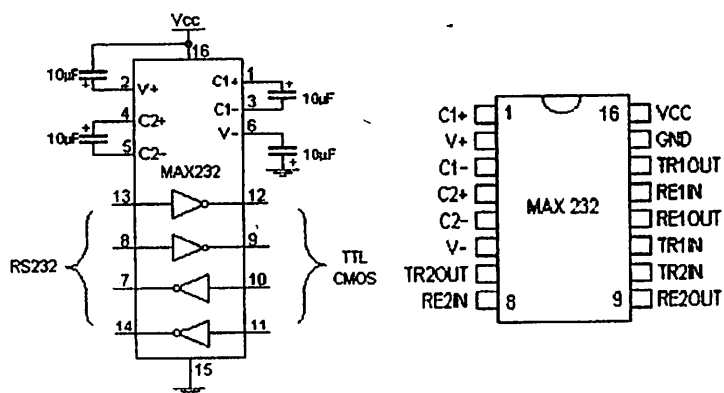
#### 2.6.1. MAX 232

Saluran data pada port seri PC menggunakan standar RS232, dimana logic 0 (low) dinyatakan sebagai tegangan antara +3 Volt sampai +15 Volt dan logic 1 (high) dinyatakan sebagai tegangan antara -3 Volt sampai -15 Volt. Level tegangan ini tidak sesuai dengan level tegangan yang dipakai pada port seri mikrokontroller yang menggunakan standard TTL (Transistor Transistor Logic), yaitu level tegangan baku dalam rangkaian – rangkaian digital.

Dalam standar TTL logic 0 (low) dinyatakan sebagai tegangan antara 0 Volt sampai 0.8 Volt, dan logic 1 (high) dinyatakan sebagai tegangan antara 3,5 Volt sampai 5 Volt. Karena perbedaan tegangan tersebut, maka agar port seri PC tidak merusak port seri Mikrokontroller antara keduanya dipasangkan IC

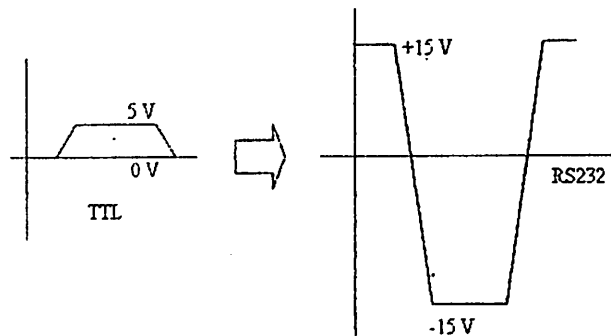
MAX232. Pada dasarnya IC ini hanya digunakan sebagai pengubah tegangan ke level TTL, tidak berfungsi sebagai pengkodean sinyal yang melewati RS232, dan juga tidak mengkonversi data serial ke parallel.

Konverter RS-232 level akan menganggap tegangan antara +3 hingga +15 Volt sebagai tegangan ' 0 ' sedangkan tegangan -3 hingga -15 Volt dianggap



Gambar 2.25 Konfigurasi Pin dan Blok diagram RS 232  
 Sumber: <http://www.alldatasheet.com/ic/1232.pdf>

tegangan ' 1 ', level antara -3 hingga +3 tidak didefinisikan, sebab di daerah ini kemungkinan adalah noise. Level TTL diatas 2 Volt yang dianggap sebagai level ' 1 ' akan dikonversikan ke level RS232 yaitu sebesar -15 Volt, sedangkan level ' 0 ' TTL, yaitu tegangan di bawah 0.8 Volt, akan dikonversikan ke +15Volt, demikian juga pada konversi sebaliknya, level +3 hingga +15 Volt akan dikonversikan ke level TTL 5 Volt dan -3 hingga -15 Volt akan dikonversikan ke 0 Volt.



Gambar 2.26 Level Converter TTL Ke RS-232  
 Sumber: <http://www.alldatasheet.com/ic/1232.pdf>

## 2.7. Borland Delphi

Borland Delphi merupakan bahasa pemrograman berbasis Windows. Bahasa pemrograman Delphi merupakan pengembangan dari bahasa Pascal. Dalam delphi ada beberapa perangkat yang dapat digunakan untuk membuat program aplikasi, diantaranya adalah (Modul Hardware programming with Delphi ver:01):

1. Component Pallete
2. Form Designer
3. Code Editor
4. Object Inspector

### 2.7.1. Tipe Data

Ada berbagai tipe macam data yang dapat digunakan pada saat membuat program. Pemilihan tipe data yang tepat diperlukan dalam membuat program yang

efisien karena berguna dalam menghemat memori, kecepatan serta ketelitian penghitungan.

1. Tipe Integer

Tipe data integer digunakan untuk menyatakan bilangan yang tidak memiliki angka decimal.

2. Tipe Real

Tipe data real merupakan floating-point (ditulis dalam  $a \times 10^b$ ) yang digunakan untuk menyatakan bilangan yang mempunyai angka decimal.

3. Tipe Boolean

Tipe data boolean dipakai untuk menyatakan karakter satu huruf.

4. Tipe Character

Tipe data character digunakan untuk menyatakan karakter satu huruf.

5. Tipe String

Tipe data string dipakai untuk menyatakan sederetan karakter yang membentuk satu kesatuan misalnya nama, alamat kode barang dan lain-lain. Tipe data string bisa juga diisi karakter kosong atau satu karakter.

6. Tipe Array

Array adalah variabel tunggal yang dapat dipakai untuk menyimpan sekumpulan data sejenis. Untuk membedakan antara yang satu dengan yang lainnya array menggunakan nomor elemen di sebelah kanan nama array.

### 2.7.2 Database Desktop

Database desktop merupakan utilitas yang disediakan Delphi untuk membuat tabel, indeks memanipulasi tabel. Dalam membuat file database desktop programmer akan dihadapkan pada kotak dialog yang harus diisi untuk membuat field. Dalam membuat field ada empat data yang harus diisikan:

Field name: menentukan nama field

Type : menentukan type field

Size : menentukan ukuran field

Key : field yang diberi tanda \* pada kolom key harus berupa urutan field yang paling atas.

### 2.7.3 Qick Report

Quick report merupakan salah satu komponen Delphi yang berfungsi untuk membuat laporan sederhana. Dengan Qreport programmer juga dapat mencetak hasil dari laporan yang dibuat. Dalam membuat Qreport harus disertakan pula souch atau sumber dari tabel tada base yang akan dibuat.

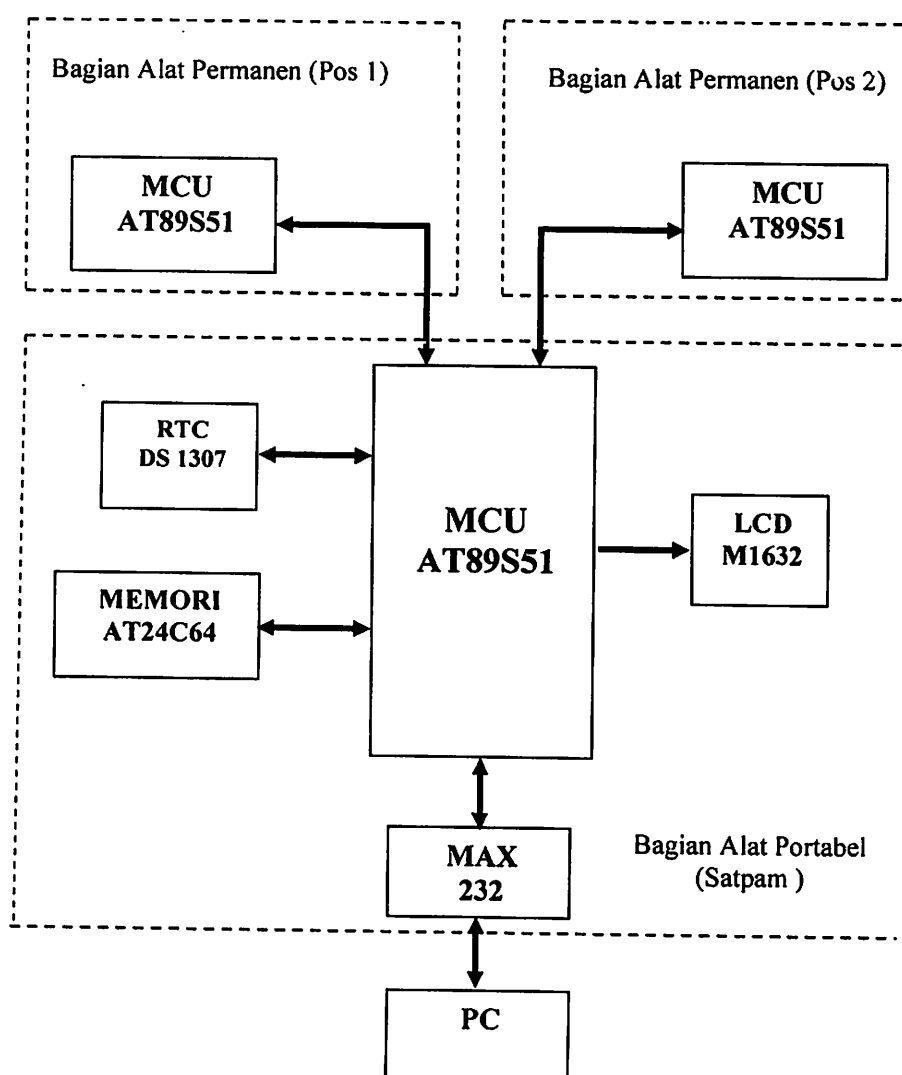




### BAB III

## PERENCANAAN DAN PEMBUATAN ALAT

Pada Bab III ini akan dibahas perencanaan dan perancangan alat, dari *hardware* sistem sampai *softwarena*, untuk lebih detailnya *hardware* sistem dapat dilihat pada blok diagram di bawah ini :



Gambar 3.1. Blok Diagram Keseluruhan Alat



### Prinsip kerja:

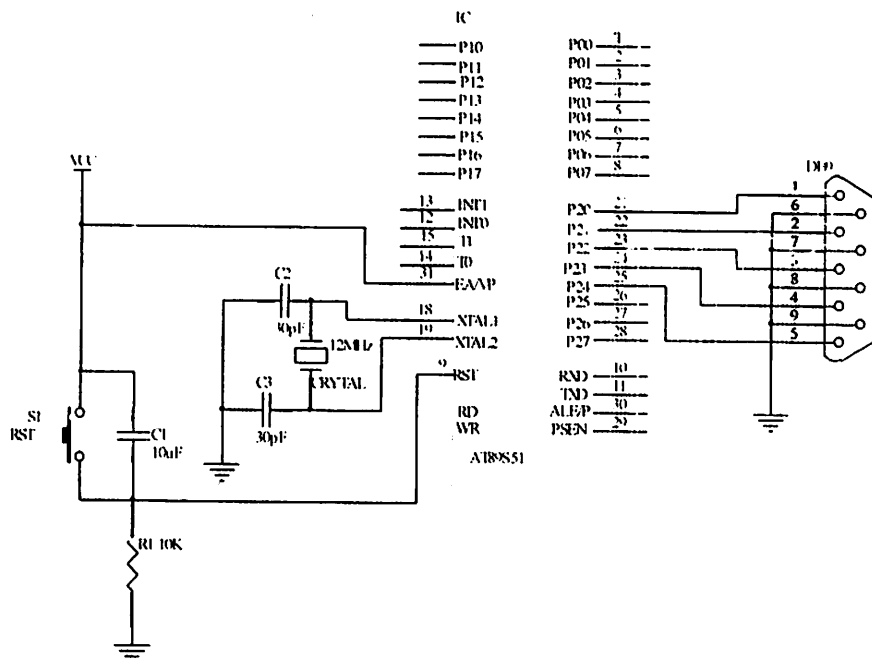
Disini alat terdiri dari dua bagian yang terpisah. Bagian pertama berupa alat portabel yang harus di bawa oleh satpam pada saat melakukan tugas pengecekan keliling. Alat ini berupa kotak yang terdapat konektor DB-9 Male pada ujung kotak. Pada alat ini terdapat RTC sebagai pemberi informasi waktu, LCD sebagai display untuk menampilkan jam dan tanggal. MEMORI untuk menyimpan data pengecekan yang telah dilakukan yaitu berupa jam, tanggal dan lokasi. Serta mikrontroller sebagai pusat kontrol pengiriman serta penerimaan data. Alat ini akan menyimpan data jam, tanggal saat itu, serta lokasi pada waktu konektor DB-9 Male dimasukan ke konektor DB-9 Female. Dan apabila konektor DB-9 Male tidak dimasukan ke konektor DB-9 Female, maka display hanya akan memberikan tampilan waktu yang berasal dari RCT. Untuk sumber daya yang digunakan untuk menyuplai alat ini saya menggunakan batere re-charge 12 volt.

Bagian kedua dari alat ini yaitu berupa alat permanen yang berada pada tiap lokasi yang harus dikontrol oleh satpam.. Alat ini berupa kotak yang terdapat konektor DB-9 Female pada ujung kotak. Dan terdapat mikrokontroller untuk mengatur konfigurasi setiap posisi. Kemudian untuk mengetahui kinerja alat pada malam hari alat harus dihubungkan ke PC yang kemudian data disimpan di PC sebagai arsip.

### 3.1. Perencanaan Elektronik

#### 3.1.1. Rangkaian MCU Pada Alat Portabel ( Satpam )

AT89S51 digunakan untuk memenuhi kebutuhan transfer data dengan RCT (serial DS1307), dan dengan LCD M1632. Pada perencanaan digunakan AT89S51 dikarenakan mikrokontroller jenis ini banyak terdapat di pasaran dan murah harganya. Selain itu AT89S51 digunakan pada alat ini nantinya proses pengiriman serta penerimaan datanya secara serial.



Gambar 3.2. Perencanaan Rangkaian AT89S51

#### a. Cristal untuk AT89S51

Sesuai dengan rumus  $T$  intruksi maka dipilih crystal dengan frekuensi 12MHz. Hal ini sesuai dengan frekuensi crystal yang diperbolehkan pada keluarga MCS-51 yaitu berkisar antara 1MHz – 16MHz (Agus Pracoyo, 2002:13 ). Dan hal

ini untuk mempermudah perhitungan waktu untuk 1 instruksi (Agfianto Eko Putra,2002:22).

$$\text{Jadi } F_{\text{instruksi}} = 1/12 * 12\text{MHz} = 1\text{MHz}$$

$$\text{Maka } T_{\text{instruksi}} = 1/1 \text{ MHz} = 1\mu\text{S}$$

b. Reset

Untuk membangkitkan sinyal reset pada awal catu daya dihidupkan suatu reset otomatis akan dilakukan oleh pin RST ke rangkaian power-On Reset. Ketika catu daya dinyalakan, rangkaian akan menahan pin RST selang beberapa saat tergantung dari nilai kapasitor dan kecepatan pengisian muatannya. Tanpa adanya sinyal reset pada mikrocontroller saat awal pemberian catu daya yang akan menyebabkan MCU melalui eksekusi intruksinya dari lokasi yang tidak tentu atau random karena program counter yang tidak terinisialisasi.

Waktu yang digunakan untuk melakukan Power-On Reset dapat ditentukan dengan rumus RC, yaitu kapasitor akan terisi penuh setelah 5 RC ( Vademekum Elektronika, Hlm:88-89 ). Sehingga dengan waktu 100 mS pin reset diberi logika tinggi maka MCU akan reset. Dapat ditentukan:

$$5RC = 500 \text{ mS}$$

$$RC = 100 \text{ mS} = 0,1 \text{ S}$$

$$\text{Dengan nilai } R = 10\text{K}$$

Maka:

$$C = \frac{0,1}{10K} = 10\mu F$$

### c. Baud-Rate

Clock atau pewaktu digunakan untuk menentukan kecepatan yang dilakukan oleh mikrokontroller. Di dalam chip AT89S51 sudah tersedia osilator internal, cukup menghubungkan crystal pada pin XTAL1 dan XTAL2 serta kapasitor untuk menentukan frekuensi osilatornya, frekuensi crystal disesuaikan dengan pemilihan kecepatan yang ditentukan untuk menstrasfer data melalui pin serial interface dari AT89S51. Sistem ini dirancang untuk memiliki kemampuan boudrate sebesar 2400bps sehingga sesuai dengan crystal yang dipilih yaitu 12 MHz.

Pada perencanaan alat ini, MCU berfungsi sebagai otak dari pengaturan kerja alat secara perangkat lunak ( software ). Penggunaan crystal dengan nilai 12 MHz maka:

$$\text{Boud rate} = \frac{2^{SMOD}}{32} \times \frac{\text{Freq.Xtal}}{12 \times \{256 - (TH1)\}}$$

$$2400 = \frac{2^0}{32} \times \frac{11.0592MHz}{12 \times \{256 - (TH1)\}}$$

$$2400 = \frac{11.0592MHz}{384 \times \{256 - (TH1)\}}$$

$$TH1 = 244 \text{ (desimal)}$$

$$TH1 = 0x1\cdot4$$

Sedangkan untuk kapasitor pada krystal dipilih sesuai dengan data sheet yaitu 30pF dengan kerja MCU untuk satu cycle machine adalah:

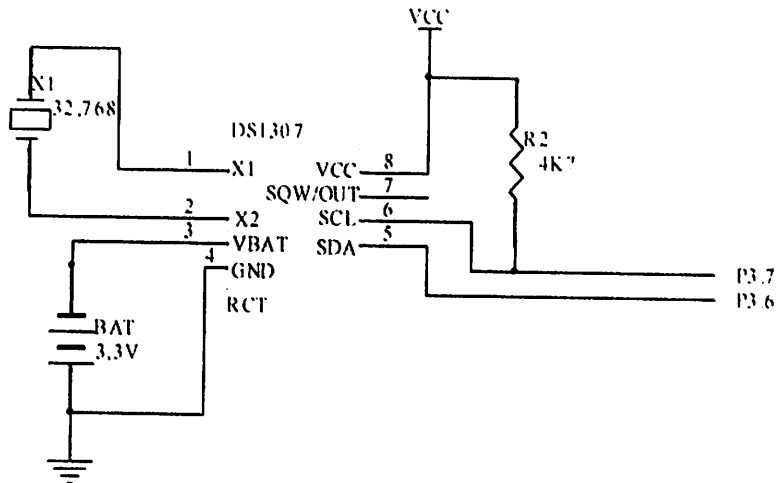
$$F = (12:4) \text{ MHz} = 3 \text{ MHz}$$

$$T = (1/f) = 0,30\mu\text{s}$$

### 3.1.2. Real Time Clock (RTC)

Pada proses perencanaan RTC digunakan sebagai pemberi informasi waktu. Untuk itu dipilih RTC tipe DS1307 yang merupakan RTC serial dengan kapasitas RAM sebesar 56 byte. Proses pengiriman dan penerimaan data antara RTC dengan mikrokontroller maupun dengan komponen lainnya menggunakan system 12C bus yang merupakan system transfer data menggunakan dua buah kabel serial yaitu SDA dan SCL.

Pada pin SCL dan SDA dihubungkan dengan pin SCL dan pin SDA pada rangkaian mikrokontroller AT89S51 yang berada di P3.7 dan P3.6 inilah yang disebut dengan system 12C bus. Kedua jalur tersebut digunakan baik oleh mikrokontroller maupun RTC untuk saling mengirim atau menerima data. Pin X1 dan X2 dihubungkan ke kristal 32,768KHz sebagai pembangkit frekuensi osilasi. Pin V<sub>BAT</sub> dihubungkan ke batere eksternal 3,3V yang berfungsi sebagai catu daya dari RTC.



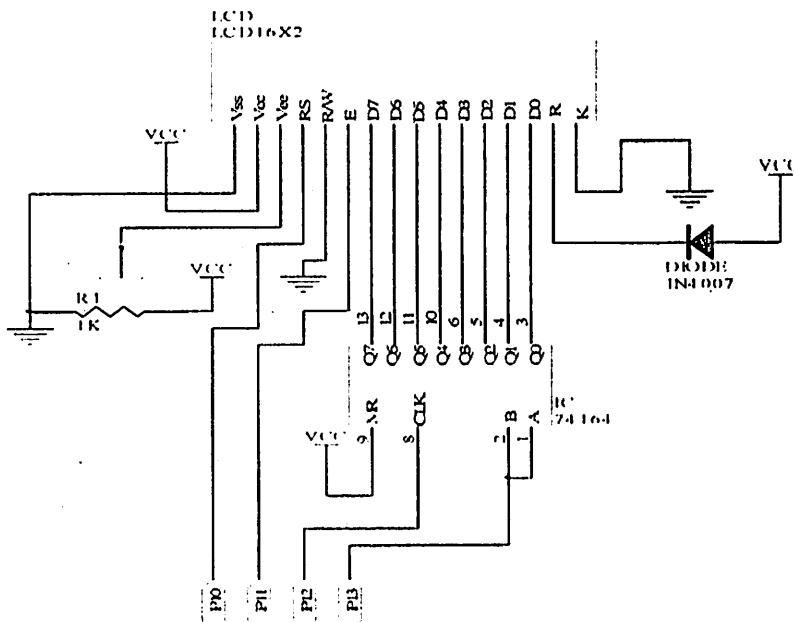
Gambar 3.3 Perancangan Rangkaian RTC

### 3.1.3. Perancangan *Liquid Crystal Display* (LCD)

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.

LCD dot matrik ini membutuhkan sepuluh buah pin masukan/keluaran dari mikrokontroler dan IC 74164. Dua buah pin yakni P1.0 pada penyemat RS yang digunakan sebagai sinyal pemilih register dan P1.1 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.



Gambar 3.4 Perancangan Rangkaian LCD

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>. Daftar tabel fungsi penyemat pada LCD dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Fungsi Penyemat LCD

Penyemat	Fungsi
DB0 – DB7	Merupakan saluran data, berisi perintah dan data yang

	akan ditampilkan di LCD.
<b>Enable</b>	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca.
<b>R/W</b>	Sinyal seleksi tulis atau baca  0: tulis  1: baca
<b>RS</b>	Sinyal pemilih <i>register</i>  0: masukan data  1: masukan instruksi

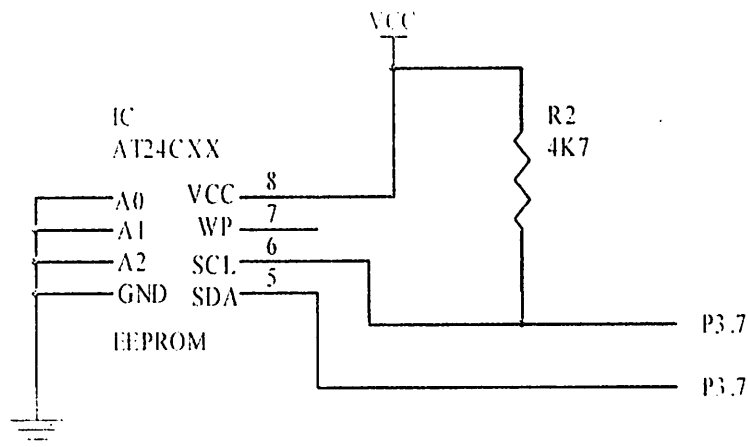
Seperti telah disebutkan sebelumnya bahwa data yang terdapat pada jalur data selain dianggap sebagai kode karakter dapat digunakan sebagai suatu perintah instruksi untuk mengatur setting dari tampilan LCD. Cara pemakaian data antara sebagai instruksi dengan kode karakter berbeda. Perbedaan hanyalah keadaan pin RS ketika data yang ada di jalur data ditahan oleh LCD dengan memberikan *clock* pada pin E.

#### 3.1.4. Serial EEPROM

Memori disini digunakan untuk menyimpan data pengecekan yang telah dilakukan oleh satpam. Untuk EEPROM dipilih tipe AT24C64 yang merupakan



EEPROM serial dengan kapasitas 64Kbit sehingga mampu menyimpan data sebesar 8Kbyte. Tipe ini digunakan karena system komunikasinya menggunakan I2C bus.



Gambar 3.5 Perancangan Rangkaian EEPROM

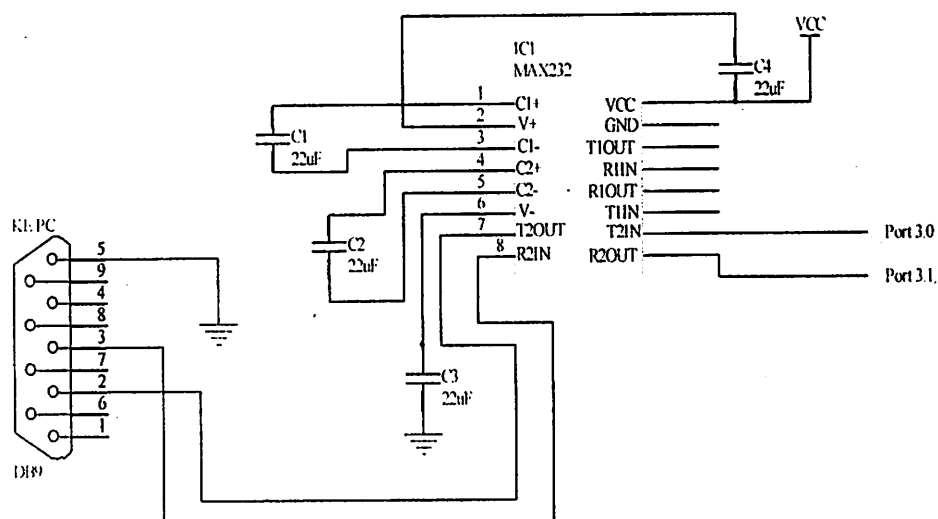
Pin SCL dan SDA dihubungkan ke P3.7 dan P3.6 pada mikrokontroller yang merupakan kaki SDA dan SCL.

#### 3.1.4. Perencanaan Serial Interface RS-232

Level tegangan TTL berbeda dengan level tegangan RS-232. Untuk menyesuaikan level tegangannya maka digunakan IC MAX-232 yang mempunyai kemampuan untuk merubah level tegangan TTL ke RS-232 dan sebaliknya. IC MAX-232 ini, mempunyai dua *charge pump internal* untuk mengubah tegangan +5 V ke +10 V yang sesuai untuk operasi RS-232. Converter pertama (*voltage doubler*) menggunakan kapasitor C1 untuk menggandakan tegangan +5 V menjadi +10 V pada C3 dikeluarkan V+. Converter kedua merupakan suatu inverter dengan menggunakan kapasitor C2 untuk membalik +10 V menjadi -10

V pada C4 keluaran V-. Gambar 3.6 menunjukkan rangkaian serial interface RS-232.

Masukan interface yang tidak digunakan dapat dibiarkan tidak terhubung karena telah tersedia resistor 400 k $\Omega$  dalam chip yang merupakan resistor *pullup* ke Vcc, sehingga keluaran dari interface yang tidak digunakan akan rendah karena seluruh rangkaian dalam chip ini diinvers. Untuk menghubungkan dengan komputer, maka dari keluaran MAX-232 menuju port serial dihubungkan ke konektor tipe DB-9 yang sesuai dengan standart komunikasi serial RS-232 pada saluran COM2. Pengaturan komunikasi datanya diatur melalui program.

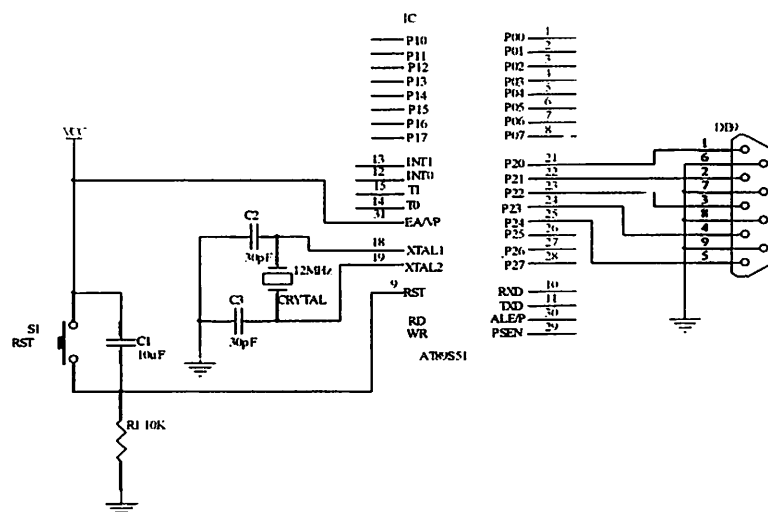


Gambar 3.6 Serial Interface RS-232

Pin T2OUT dan pin R2IN dari MAX232 dihubungkan dengan DB9, sedangkan pin R2OUT dihubungkan ke P3.0 dan pin T2IN dihubungkan dengan P3.1 Pada MCU. Pin-pin MAX232 yang berdasarkan *datasheet*

### 3.1.6. Rangkaian MCU Pada Alat Permanen ( Pos 1 dan pos 2 )

Alat permanen ini merupakan bagian alat yang terdapat disetiap lokasi yang harus dicek. Nantinya alat permanen ini akan mendapat masukan tegangan yang berasal dari alat bagian portabel. Pada alat permanen ini terdapat konektor DB-9 Female gunanya hanya sebagai konektor antara alat portabel dan alat permanen serta mikrokontroller yang digunakan untuk konfigurasi data yang fungsinya untuk membedakan tiap lokasi yang akan di cek.

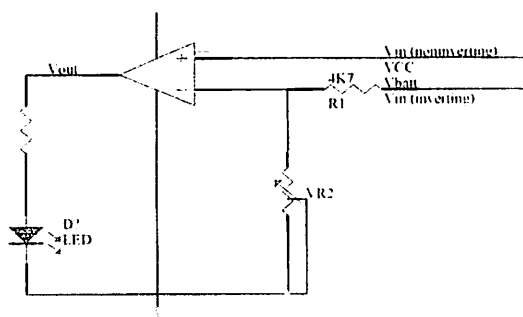


Gambar 3.7 Perencanaan Rangkaian AT89S51 Pada Alat Permanen

### 3.1.7. Indikator Battery Low

Indikator battery low ini terdapat pada alat yang portabel. Rangkaian ini berfungsi untuk memberitahukan bahwa battery pada alat itu sudah mencapai level yang rendah. Apabila battery masih belum lemah dayanya maka indikator yang berupa LED tidak menyala. Sedangkan apabila battery sudah lemah dayanya maka indikator yang berupa LED akan menyala. Untuk membuat rangkaian ini saya menggunakan op-amp yang difungsikan sebagai komparator (pembanding). Pada rangkaian ini yang dibandingkan adalah tegangan masukan

pada input inverting dan pada masukan input noninverting. Adapun rangkaiannya adalah sebagai berikut:



Gambar 3.8 Rangkaian Indikator Battery

Dalam pembuatan rangkaian indikator ini  $V_{in}$  (noninverting) adalah tegangan masukan yang tetap, yang berasal dari output IC regulator 7805, sedangkan pada  $V_{in}$  (inverting) adalah tegangan masukan yang berasal dari battery jadi tegangan ini akan selalu berubah. Dan direncanakan pada saat tegangan battery turun mencapai 6V maka indikator akan menyala.

Dengan memasang  $R_1$  sebesar 4K7 agar indikator menyala pada saat tegangan mencapai 6V,  $R_2$  adalah:

$$V_{in}(\text{inverting}) < V_{in}(\text{noninverting}) \quad V_{in}(\text{inverting}) < 5.$$

$$\text{Ditentukan } V_{in}(\text{inverting}) = 4,8 \text{ maka: } V_{in}(\text{inverting}) = \frac{R_2}{R_1 + R_2} \times V_{batt}$$

$$\text{Sehingga didapatkan: } 4,8 = \frac{6R_2}{4700 + R_2}$$

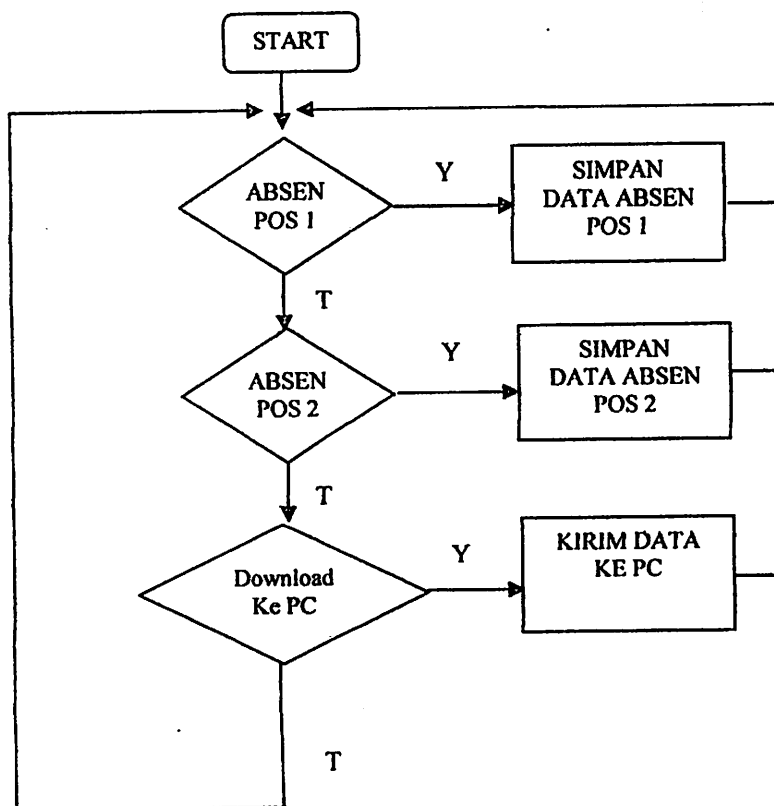
$$22560 + 4,8R_2 = 6R_2$$

$$R_2 = \frac{22560}{1,2}$$

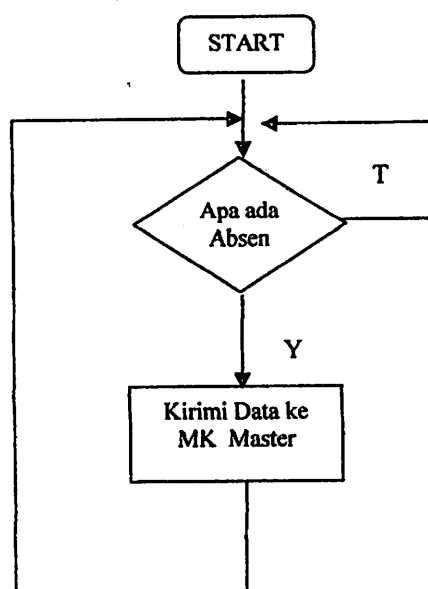
$$R_2 = 18800$$

Dengan didapat nilai  $R_2 = 18,8\text{Kohm}$  maka untuk dapat nilai yang sesuai maka di pasang VR sebesar 50Kohm. Sehingga nilai  $R_2$  dapat di atur.

### 3.2. Perencanaan Perangkat Lunak (software)



Gambar 3.9 Flowchart Program Pada Alat Portabel ( Satpam )



Gambar 3.10 Flowchart Program Pada Alat Permanen (Pos 1 dan Pos 2 )

Sebagai tampilan pada proses pembacaan data ke PC digunakan software Borland Delphi yang memungkinkan untuk membuat tampilan semenarik mungkin dan juga dapat menyimpan data secara format data base sehingga nantinya dapat diakses maupun di cetak. Database digunakan sebagai tempat pembacaan data sementara (saat di akses) sekaligus sebagai tempat penyimpanan data yang akan dicetak.

Borland Delphi merupakan software yang menggunakan bahasa pascal sebagai bahasa pemograman. Dengan delphi diharapkan nantinya akan memudahkan bagi orang yang ingin membaca maupun mencetak hasil dari keseluruhan data yang telah masuk ke PC.





## **BAB IV**

### **PENGUJIAN ALAT**

Tahapan yang paling penting pada perancangan alat ini adalah pengujian. Dari pengujian dapat diketahui apakah alat yang dibuat berkerja dengan baik atau masih kurang sempurna.

Pada bab ini akan dibahas tentang pengujian alat yang telah dirancang dengan tujuan supaya antara perancangan dan pembuatan dengan hasil keperluan yang ada, dipandang dari segi perancangan *hardware* maupun *software* dapat sesuai dengan kondisi yang diinginkan. Dengan diadakan pengujian alat, maka pada tugas akhir ini dapat membuktikan bahwa alat yang dibuat dapat berjalan sesuai dengan kondisi masukkan yang ada, sehingga memberikan keluaran yang sesuai pula.

#### **4.1 Pengujian Hardware**

##### **4.1.1 Pengujian RTC**

Tujuan dalam pengujian RTC untuk mengetahui apakah komponen yang digunakan dapat berfungsi dengan baik atau tidak. Untuk menguji RTC, sistem MCU dan LCD di hubungkan dengan RTC. Kemudian pada Eprom Emulator diberikan listing program dalam bahasa C untuk mengeset waktu dan tanggal yang terdapat di RTC dan menampilkan hasil keluaran



waktu dan tanggal tersebut pada tampilan LCD dan tampilkan pada form.

Diagram blok pengujian RTC seperti pada gambar di bawah in:



Gambar 4.1 Blok Diagram Pengujian RTC

- Untuk pengesetan waktu dan tanggal digunakan listing program seperti berikut:

```

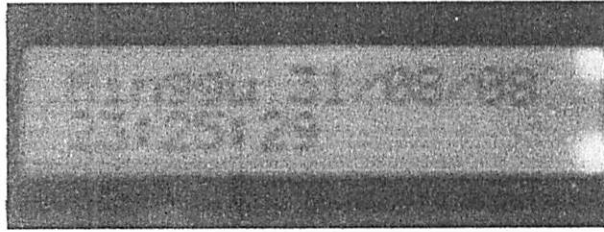
// -----
// Program Utama
// -----
void main ()
{/** Begin of Main

while(1)** Pengulangan Loop tanpa henti
  {/** Begin of While

a=read_rtc(3);hari(a);asa=' ';dataout();
a=read_rtc(4);asa=a/10+48;dataout();asa=a%10+48;dataout();
asa='/';dataout();
a=read_rtc(5);asa=a/10+48;dataout();asa=a%10+48;dataout();
asa='.';dataout();
a=read_rtc(6);asa=a/10+48;dataout();asa=a%10+48;dataout();
pos(2,1);
a=read_rtc(2);asa=a/10+48;dataout();asa=a%10+48;dataout();
asa=':';dataout();
a=read_rtc(1);asa=a/10+48;dataout();asa=a%10+48;dataout();
asa=':';dataout();
a=read_rtc(0);asa=a/10+48;dataout();asa=a%10+48;dataout();

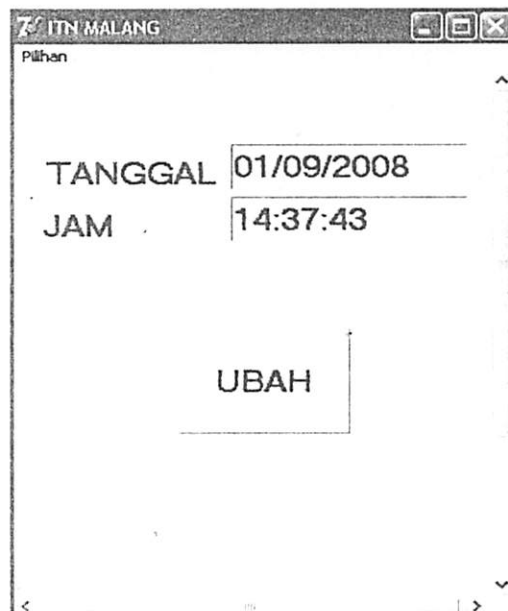
  }/** End of While
}/** End of Main
  
```

Dari hasil pengujian maka di dapat tampilan seperti di bawah ini:



Gambar 4.2 Hasil Tampilan Pada LCD

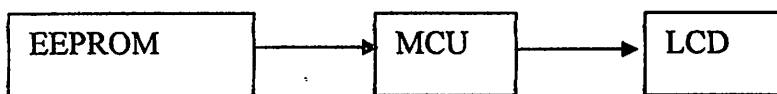
- Setelah pengujian RTC dilakukan,yaitu dengan menjalankan program pengesetan waktu dan tanggal seperti terlihat pada program di atas, dengan hasil keluaran yaitu tampilan pada LCD terlihat pada gambar 4.2. Setelah itu program Delphi di aktifkan dan tampilan form terlihat seperti pada gambar di bawah ini:

A screenshot of a Delphi form titled "ITN MALANG". The form has a "Pilihan" label at the top left. It contains two text boxes: "TANGGAL" with the value "01/09/2008" and "JAM" with the value "14:37:43". Below these boxes is a button labeled "UBAH". The form has a standard Windows-style title bar with minimize, maximize, and close buttons.

Gambar 4.3 Tampilan pada Form untuk merubah Waktu dan tanggal

### 4.1.2 Pengujian EEPROM

Prosedur pengujiannya yaitu membuat perangkat lunak pada pengujian EEPROM. Program ini akan melakukan proses penulisan data di alamat EEPROM dengan mencoba menyimpan karakter 'A'. Dan melakukan proses pembacaan. Diagram blok pengujian EEPROM seperti pada gambar di bawah ini:



Gambar 4.4 Diagram Blok Pengujian EEPROM

- Contoh listing Program penulisan data:

```

void tulis(unsigned char alamat, unsigned char nilai)
{
    i2c_start();
    i2c_write(EEPROM_ADDR);
    i2c_write(alamat);
    i2c_write(nilai);
    i2c_stop();
    delay_ms(10);
}
main ( )
{
    tulis (0,'A');
    while (1);
}
  
```

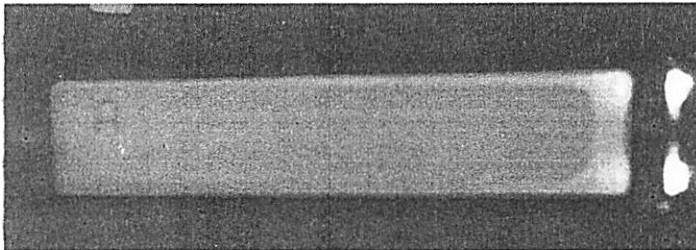
Setelah melakukan penulisan data, selanjutnya EEPROM akan melakukan pembacaan data. Dan hasil pembacaan data EEPROM akan di tampilkan di LCD.

- Contoh listing program pembacaan data:

```

unsigned char baca(unsigned char alamat)
{
    unsigned char datax;
    i2c_start();
    i2c_write(EEPROM_ADDR);
    i2c_write(alamat);
    i2c_stop();
    i2c_start();
    i2c_write(EEPROM_ADDR | 1);
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}
main ( )
{
    baca (0); data out ( );
    while (1);
}

```

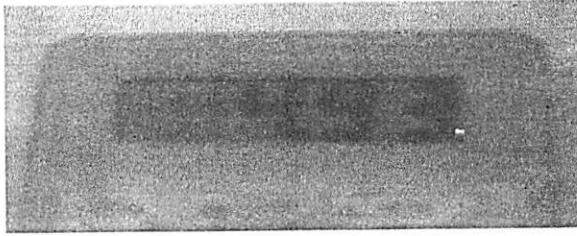


Gambar 4.5 Hasil Tampilan pembacaan data pada LCD

#### 4.1.3 Pengujian RS 232

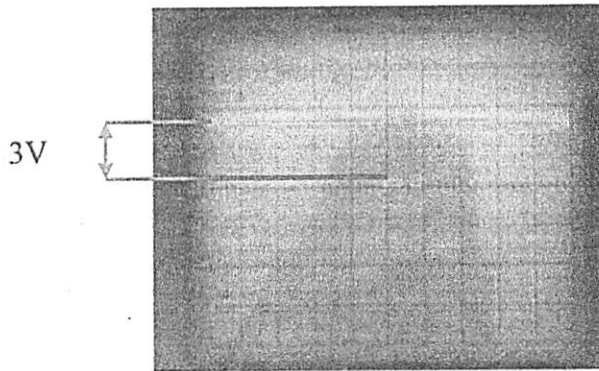
Pengujian RS 232 ini bertujuan untuk melihat adanya sinyal pada *transmitter* RS 232. Untuk melihat bentuk gelombang ini dilakukan dengan menggunakan osiloskop dan Multimeter Digital.

- Gambar Input Tegangan yang dibutuhkan oleh RS 232



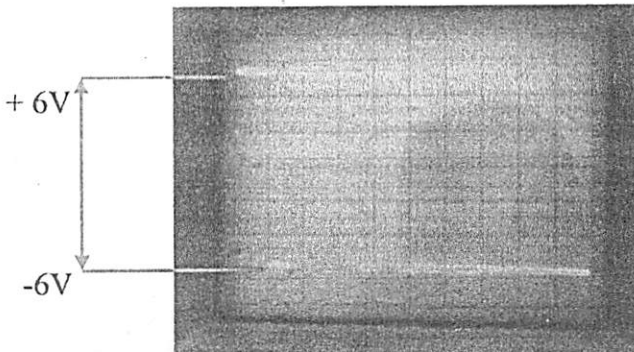
Gambar 4.6 *Input* tegangan pada RS 232

- Gambar Sinyal RS 232 dari MCU ke PC Level TTL  
(Volt/div = 2V, Time/div = 0.5ms)



Gambar 4.7 Sinyal *input* RS 232

- Gambar *output* dari RS 232 ke PC Level RS232  
(Volt/div = 2V, Time/div = 0.5ms)



Gambar 4.7 Sinyal *output* RS 232 ke PC

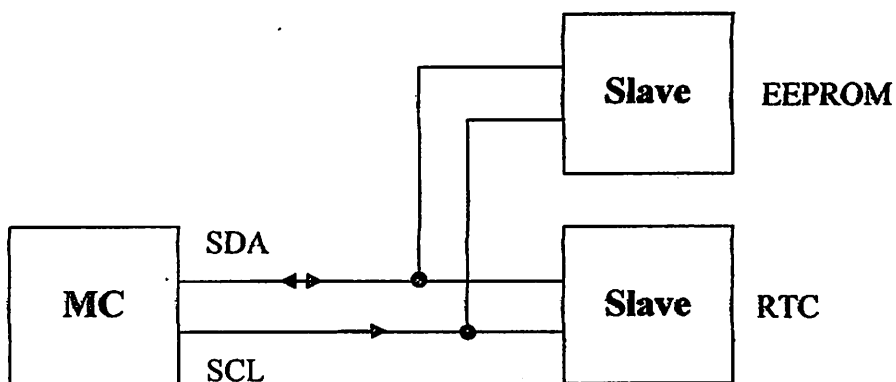
- **Analisis Hasil Pengujian**

Dari hasil pengujian pada gambar 4.10 dan 4.11 didapatkan tegangan sebesar 4.93V dan sebagai inputan ke mikrokontroler level tegangan tersebut telah memenuhi standart logika TTL dan pada gambar 4.12 disisi RS 232 level tegangan sebesar 12V telah memenuhi syarat level tegangan RS 232, yaitu antara +15 hingga +3 V untuk logika "0" dan -15 dan -3 V untuk logika "1".

#### 4.1.4 Pengujian Perangkat I2C

- Pengujian perangkat i2C ini bertujuan untuk mengetahui komunikasi dalam i2C. Alat yang di gunakan dalam pengujian ini berupa osiloskop untuk melihat sinyal SDA dan SCL.

Diagram blok pengujian perangkat I2C dapat dilihat di bawah ini:



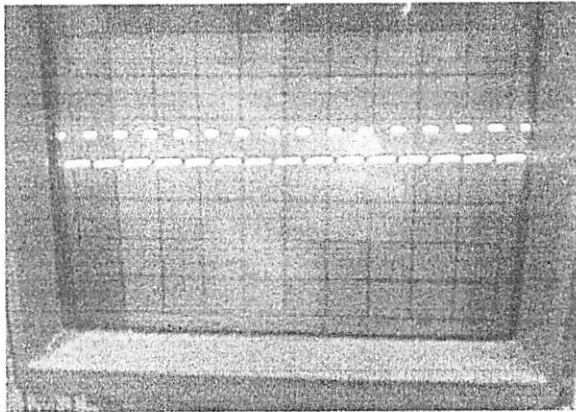
Gambar 4.8 Diagram Blok Pengujian Perangkat I2C

- **Penyimpanan dan Pembacaan Data**

Pada sistem terdapat dua media hardware yang bersistem I2C, yaitu *serial* EEPROM AT24Cxx dan *serial* RTC DS1307, kedua media ini memiliki karakteristik yang berbeda dalam hal penyimpanan data. *Serial* EEPROM 24Cxx memiliki keterbatasan penulisan ulang, tetapi dapat menyimpan data pada saat power dimatikan. *Serial* RTC DS1307 tidak memiliki keterbatasan dalam penulisan ulang, tetapi DS1307 membutuhkan baterai untuk menyimpan data dan menjalankan jam. Apabila baterai tidak dipasang, semua data yang ada pada DS1307 akan hilang.

Untuk mengetahui proses penyimpanan data yang dilakukan sudah benar, maka dilakukan pengujian dari data data yang disimpan. Cara menguji penyimpanan data dilakukan dengan cara mengisi data pada *serial* EEPROM dan *serial real time clock* dengan menggunakan menu yang telah dibuat, dan untuk mengecek kebenarannya dilakukan dengan cara menampilkan data-data tersebut dengan menu yang ada. Selain menggunakan menu yang ada pada sistem, data pengisian juga dapat dicek melalui fasilitas download yang ada pada program komputer.

- Gambar Hasil pengujian Sinyal SDA



Gambar 4.9 Sinyal SDA

- Gambar Hasil pengujian Sinyal SCL



Gambar 4.10 Sinyal SCL

## 4.2 Pengujian Software

pengujian perangkat lunak bertujuan untuk mengetahui apakah fungsi dari program secara keseluruhan berjalan dengan baik atau tidak.

### 4.2.1 Tahap pertama

Di bawah ini adalah tampilan icon untuk mengeksekusi program





Gambar 4.11 Tampilan Icon

Pada tahap awal pengeksekusian program, akan muncul form untuk pengesetan jam dan tanggal dengan cara meng-klik "UBAH" agar sesuai dengan waktu pada saat itu. Seperti pada gambar 4.12 di bawah ini:

Gambar 4.12 Form untuk merubah jam dan tanggal

#### 4.2.2 Tahap kedua

Setelah mengeset waktu dan tanggal kemudian klik "PILIHAN" yang ada di sisi kiri atas form pada gambar 4.13. Dan akan muncul berupa menu pilihan yaitu refresh, ambil data, tampilan utama, ubah RTC dan print. Dan untuk menampilkan tampilan utama kita tinggal klik "Tampilan Utama" dan akan muncul form seperti pada gambar 4.14 di bawah ini:

The screenshot shows a web browser window with the title 'ITN MALANG'. The menu includes 'Refresh', 'Ambil Dota', 'Tampilan Utama', 'Ubah RTC', and 'Print'. The 'Tampilan Utama' option is selected. Below the menu, there are two input fields: one for a date containing '01/09/2008' and another for a time containing '14:42:20'. A 'UBAH' button is located below the time field.

Gambar 4.13 Tampilan Form untuk menampilkan tampilan utama

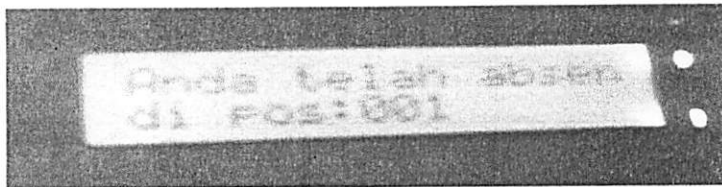
The screenshot shows a web browser window with the title 'ITN MALANG'. It displays a table with the following columns: 'NO', 'PDS', 'TANGGAL', and 'JAM'. The table is currently empty. Below the table, there is a 'HAPUS' button.

Gambar 4.14 Form tampilan utama

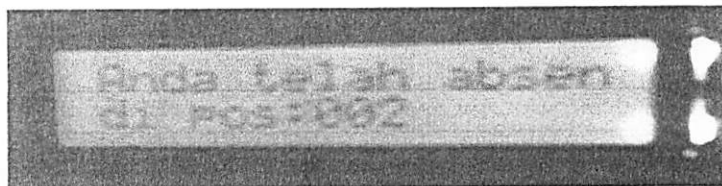
### 4.2.3 Tahap ketiga

Tahap ketiga adalah tahap untuk menampilkan data setelah petugas satpam melakukan Pegecekan dan pengabsenan pada setiap lokasi yang harus di cek oleh satpam yaitu di pos 1 dan pos 2 .

Di bawah ini adalah hasil tampilan pada LCD ketika satpam sudah selesai melakukan tugasnya dan melakukan pengabsenan. Tampilan pengabsenan pada pos 1 dan pos 2 terlihat seperti pada gambar di bawah ini:

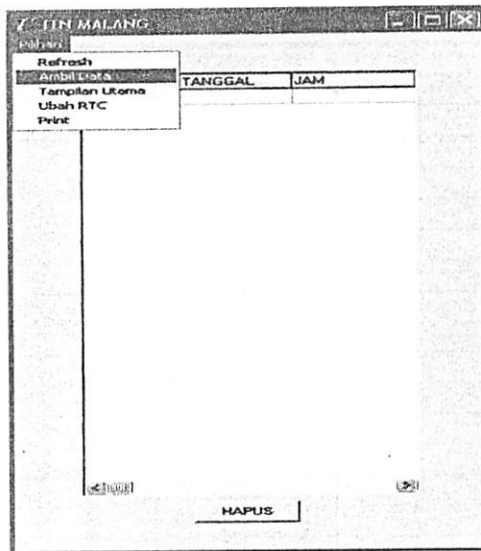


Gambar 4.15 Tampilan pada LCD untuk Pengabsenan pos 1

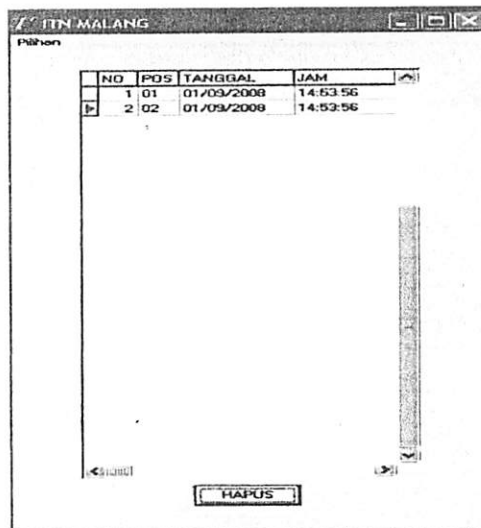


Gambar 4.16 Tampilan pada LCD untuk pengabsenan pos 2

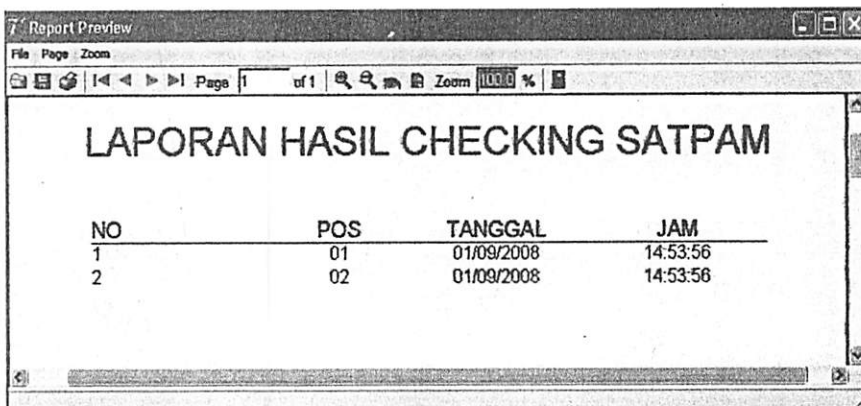
Dan untuk melihat data pada form,kita tinggal klik "PILIHAN" yang ada di kiri atas Form,kemudian klik "ambil data" seperti pada gambar 4.17. Data hasil pengecekan yaitu berupa jam,tanggal serta lokasi (pos 1 dan pos 2) akan segera tampil di form. Hasil tampilan pada form terlihat pada gambar 4.18. Dan untuk mencetak data yang sudah tersimpan pada form kita tinggal klik "print" dan hasil tampilan seperti pada gambar 4.19.



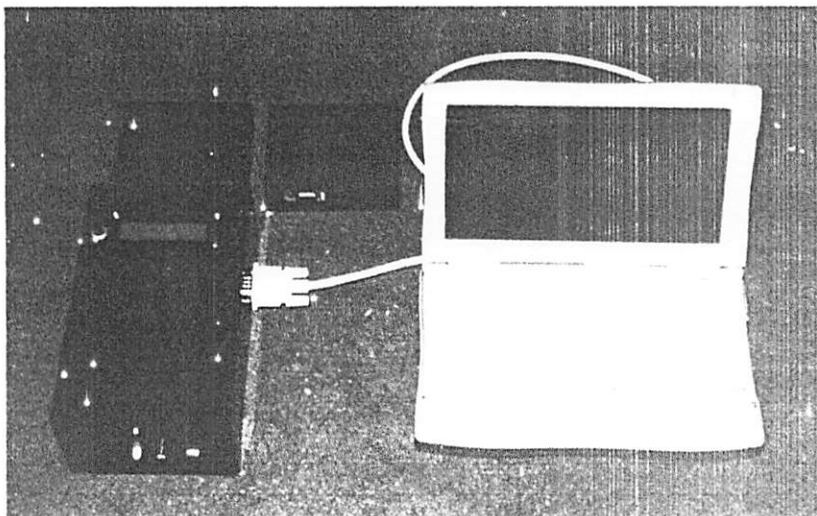
Gambar 4.17 Tampilan Form ketika mau menampilkan data



Gambar 4.18 Tampilan Form Data Pengecekan Satpam



Gambar 4.19 Bentuk Tampilan Print Out Laporan Hasil Pengecekan Satpam



Gambar 4.20 Alat Keseluruhan

### 4.3 Spesifikasi alat

Alat pencatat waktu dengan menggunakan AT89S51 untuk meningkatkan kinerja satpam ini mempunyai spesifikasi alat sebagai berikut:

1. Alat ini terdiri dari dua bagian yang terpisah. Yang pertama adalah box portabel yang di dalamnya terdapat MCU,RTC,LCD dan EEPROM. Yang kedua adalah bagian permanen yang menempel di dinding, di dalamnya terdapat MCU AT89S51.
2. Alat ini dapat digunakan untuk sekelompok kerja satpam dan dapat mengecek dua tempat sekaligus. Dan dapat digunakan untuk berkali-kali checking.
3. Alat ini menggunakan AT89S51 sebagai pusat kendali dari seluruh system.
4. Menggunakan Serial RTC DS1307 yang menggunakan sytem komunikasi I2C Bus. Untuk memberikan informasi waktu.
5. Menggunakan Serial EEPROM AT24C64 yang juga menggunakan sytem komunikasi I2C Bus dengan kapasitas 64 kilobit (8 kilo byte).
6. Untuk pengesetan jam dan tanggal dilakukan di PC
7. Menggunakan LCD untuk menampilkan jam,tanggal dan posisi terakhir yang telah di cek.
8. Menggunakan batteray 12 volt yang di hubungkan pada IC regulator 7805 untuk kebutuhan supplytegangan pada rangkaian.dan membutuhkan daya sebesar 0,6 W.

9. Menggunakan IC MAX 232 untuk mengkonversi tegangan level TTL menjadi RS232 level untuk komunikasi antara mikrokontroller dengan PC.
10. Mikrokontroller AT89S51 yang terdapat di alat permanen digunakan untuk mengatur konfigurasi setiap tempat yang berbeda-beda.
11. Box portabel berukuran  $T = 8 \text{ cm}$ ,  $P = 20 \text{ cm}$ ,  $L = 12 \text{ cm}$  sedangkan box permanen berukuran  $T = 3 \text{ cm}$ ,  $P = 10 \text{ cm}$   $L = 8 \text{ cm}$ .





INSTITUT TEKNOLOGI NASIONAL  
MALANG

MALANG



## **BAB V**

### **PENUTUP**

#### **5.1 KESIMPULAN**

Dari proses perencanaan hingga sampai dengan proses pengujian pada sistem ini, dapat diambil beberapa kesimpulan yang diantaranya :

1. Input tegangan sebesar 4.93V ke mikrokontroler level tegangan telah memenuhi standart logika TTL dan disisi RS 232 level tegangan sebesar 12V telah memenuhi syarat level tegangan RS 232, yaitu antara +15 hingga +3 V untuk logika "0" dan -15 dan -3 V untuk logika "1".
2. Untuk menghubungkan antar IC digunakan system 12C bus yang merupakan sytem komunikasi serial dengan menggunakan jalur yaitu SDA (Serial Data Input-Output ) dan SCL ( Serial Clock Input )
3. Dengan system 12C bus port-port yang digunakan lebih sedikit dan memudahkan dalam pembuatan lay out pcb.
4. Komunikasi MCU dengan PC menggunakan RS 232 level harus menggunakan baud-rate yang sama agar terjadi sinkronisasi pada data yang akan masuk ke PC.

## 5.2 SARAN

1. Untuk tampilan pada PC dengan menggunakan Borlan Delphi dapat di buat lebih menarik lagi dengan menambahkan komponen-komponen lainnya.
2. Agar lebih efektif dalam menggunakannya, alat ini dapat dikembangkan dengan menambahkan program didalamnya untuk menyimpan identitas satpam.
3. Bentuk fisik yang dibuat lebih kacil lagi dengan memaksimalkan pengaturan tata letak pcb nya.
4. Alat ini dapat dikembangkan dengan menambahkan Buzzer sebagai alarm untuk memberitahukan pada petugas satpam, jam berapa dia harus melakukan pengecekan keliling dengan waktu yang telah ditentukan.



## DAFTAR PUSTAKA

<http://www.atmel.com>

<http://www.dalsemi.com>

<http://www.Delta-Elektronik.com>

<http://www.alldatasheet.com/icl232.pdf>

Rizqi Firmansyah, 2006, *Robot Pengelasan ( Hardware )*, Surabaya, Proyek Akhir Politeknik Elektronika Surabaya.

Wisma Wulandari, 2007, *Perencanaan dan Pembuatan Alat Pengendali Kamera Menggunakan Aplikasi PC*, Malang, Tugas Akhir Teknik Elektronika D3 ITN Malang.





Lampiran : 1(satu) berkas Proposal  
Perihal : Permohonan Kesiediaan Dosen Pembimbing

Kepada : Yth. Bapak/Ibu. Joseph Dedy Irawan,ST,MT  
Institut Teknologi Nasional Malang  
di  
M a l a n g

Dengan hormat,

Kami yang bertanda tangan dibawah ini :

N a m a : Maulana Malik  
No. Mahasiswa : 0452214  
Program Studi : Teknik Elektro D-III

Dengan ini mengajukan permohonan, sekiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing Utama/Pendamping\*), untuk penyusunan Tugas Akhir .

Judul Tugas Akhir : ..Perancangan dan Pembuatan Pencatat Waktu Menggunakan.....  
..AT89S51..untuk.Meningkatkan.Kinerja.SATPAM.....

Adapun tugas tersebut adalah salah satu syarat untuk menempuh Ujian Tugas Akhir Program D-III.

Demikian permohonan kami dan atas kesediaan Bapak/Ibu kami ucapkan terima kasih.



II. Choirul Saleh,MT  
NIP.Y. 1018800190

Malang, 19 Mei 2008

Maulana Malik

**KESEDIAAN PEMBIMBING  
TUGAS AKHIR**

Sesuai permohonan dari mahasiswa :

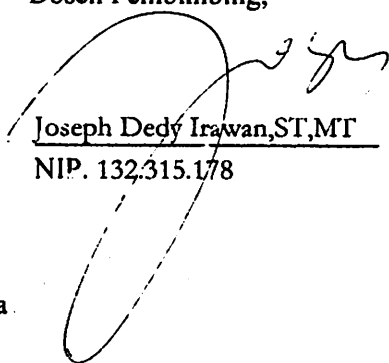
N a m a : Maulana Malik.....  
No. Mahasiswa : 0452214.....  
Program Studi : Teknik Elektro D-III.....  
Judul Tugas Akhir : Perancangan dan Pembuatan Pencatat Waktu Menggunakan.....  
AT89S51 untuk Meningkatkan Kinerja SATPAM.....  
.....  
.....

Bahwa kami bersedia membimbing Tugas Akhir dari mahasiswa tersebut.

Jangka waktu penyelesaian Tugas Akhir selama 4 (empat) bulan mulai tanggal  
19/05/2008 s/d 19/09/2008 dan apabila dalam jangka waktu tersebut belum selesai maka tugas akhir  
tersebut dinyatakan GUGUR

Malang, 19 Mei 2008

Dosen Pembimbing;

  
Joseph Dedy Irawan, ST, MT  
NIP. 132.315.178

Nb :

Setelah disetujui agar formulir ini diserahkan Mahasiswa  
yang bersangkutan kepada sekretaris Program Studi  
Teknik Elektro D-III



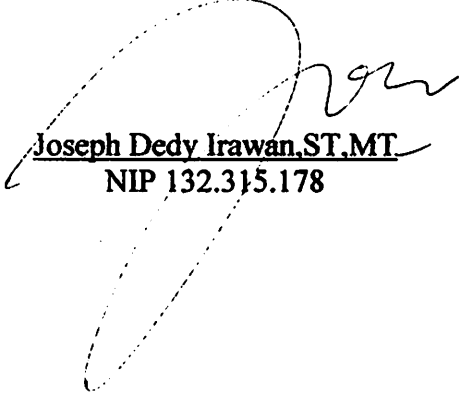
Institut Teknologi Nasional Malang  
Fakultas Teknologi Industri  
Jurusan Teknik Elektro D-III

Lembar Asistensi Bimbingan Tugas Akhir

Nama : MAULANA MALIK  
NIM : 04.52.214  
Jurusan : Teknik Elektronika  
Judul : Perencanaan dan Pembuatan Pencatat Waktu Menggunakan  
AT89S51 Untuk Meningkatkan Kinerja Satpam  
Dosen Pembimbing : Joseph Dedy Irawan,ST,MT

No	Tanggal	Asistensi	Paraf
1	23/08 /7	Revisi perubahan pada Alat Perukuran	
2	25/08 /7	Revisi Pengesetan Jam dan tanggal di latutan di PC	
3	19/08 /8	Revisi flow chart dan Blok diagram	
4	11/08 /9	Revisi BAB II dan BAB IV	
5	15/08 /9	MC KOMPONEN	
6			

Malang, September 2008

  
Joseph Dedy Irawan,ST,MT  
NIP 132.315.178








INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
PROGRAM STUDI TEKNIK ELEKTRO D-III  
MALANG


LEMBAR PERBAIKAN

Nama : Maulana Malik  
NIM : 04.52.214  
Program Studi : Teknik Elektro D-III  
Konsentrasi : Teknik Elektronika  
Judul Tugas Akhir : Perencanaan Dan Pembuatan Pencatat Waktu  
Menggunakan AT89S51 Untuk Meningkatkan Kinerja  
Satpam


No	Revisi	TTD
1.	Blok Diagram	
2	Saran di Tambahkan	

Telah diperiksa / disetujui

  
(Ir. Eko Nurcahyo)  
Dosen Penguji I

  
(Irmalia Suryani Faradisa, ST. MT)  
Dosen Penguji II

Mengetahui

  
(Joseph Dedy Irawan, ST. MT)  
Dosen Pembimbing



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO D-III  
PROGRAM STUDI TEKNIK ELEKTRONIKA  
MALANG

BERITA ACARA UJIAN TUGAS AKHIR  
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Maulana Malik  
NIM : 04.52.214  
Jurusan : Teknik Elektro D-III  
Program Studi : Teknik Elektronika  
Judul Tugas Akhir : Perancangan dan Pembuatan Pencatat Waktu Menggunakan  
AT89S51 Untuk Meningkatkan Kinerja Satpam

Dipertahankan dihadapan Tim Penguji Tugas Akhir jenjang Diploma Tiga (D-III) :

Pada Hari : Rabu

Tanggal : 24 September 2008

Dengan Nilai : A



(Ir. Mochtar Asroni, MSME)  
Ketua Majelis Penguji

Panitia ujian Tugas Akhir



(Ir. H. Choirul Saleh, MT)  
Sekretaris Majelis Penguji

Anggota Penguji



(Ir. Eko Nurcahyo)  
Anggota Penguji I



(Irmalia Suryani Faradisa, ST. MT)  
Anggota Penguji II

# LAMPIRAN



m0.c

```
clude <at89x51.h>
clude "Pending.c"
clude "myser1.c"
clude "lcdku.c"
clude "adda.c"
```

```
fine tom          P3_5
fine j10          P2_0
fine j11          P2_1
fine j12          P2_2
fine j13          P2_3
```

```
igned char a,d[20];
igned int i,j;
t,tjam;
g time;
```

```
d koni(unsigned int t)
```

```
su=t/1000+48;dataout();
=(t/100%10)+48;dataout();
=(t/10%10)+48;dataout();
=(t%10)+48;dataout();
```

```
d inittimer()
```

```
TH0=(-1000/256)-1;
TL0=(-1000%256)-1;
TMOD=0x01; //Timer 0 sebagai 16 bit counter
ET0=1; //Aktifkan interupsi Timer 0
EA=1; //Aktifkan Sistem interupsi MCS51
TR0=1; //Jalankan Timer 0
```

```
d TimerInterrupt (void) interrupt 1 using 1
```

```
0=(-1000/256)-1;
TL0=(-1000%256)-1;
e++;
```

```
igned char loopadc(char kanal,unsigned char loop)
```

```
igned int suhu,suhu1,j;
j=0;
or(j=0;j<loop;j++)
```

```
suhu1=read_adc(kanal);
suhu=suhu+suhu1;
```

```
suhu=suhu/loop;
urn suhu;
```

```
d hari(unsigned char ha)
```

```
ha==1) cetak(1,1,"Senin");
e if(ha==2) cetak(1,1,"Selasa");
e if(ha==3) cetak(1,1,"Rabu");
e if(ha==4) cetak(1,1,"Kamis");
e if(ha==5) cetak(1,1,"Jumat");
e if(ha==6) cetak(1,1,"Sabtu");
e if(ha==7) cetak(1,1,"Minggu");
```

```
d SerialInterrupt (void) interrupt 4 using 1
```

```
while(!RI){}
```

m0.c

```
RI = Lo ;  
if(SBUF=='a') t=1;  
else if(SBUF=='j') {tjam=1;j=0;}  
else if(tjam==1){d[j]=SBUF;j++;}  
if(j==12){tjam=0;}
```

d masukdata()

```
igned char ma,su,kd;  
baca(0);  
kd*10;  
read_rtc(2);  
is(ma+1,su);  
read_rtc(1);  
is(ma+2,su);  
read_rtc(0);  
is(ma+3,su);  
read_rtc(4);  
is(ma+4,su);  
read_rtc(5);  
is(ma+5,su);  
read_rtc(6);  
is(ma+6,su);  
is(ma+7,a);  
is(0,kd+1);
```

d kirimas(unsigned char ir)

```
igned char im;  
ir/10+48;PutChar(im);  
ir%10+48;PutChar(im);
```

d kirimdata()

```
igned char ki,ri;  
Char('a');  
baca(0);  
(i=0;i<ki;i++)
```

```
i=baca(i*10+1);kirimas(ri);  
i=baca(i*10+2);kirimas(ri);  
i=baca(i*10+3);kirimas(ri);  
i=baca(i*10+4);kirimas(ri);  
i=baca(i*10+5);kirimas(ri);  
i=baca(i*10+6);kirimas(ri);  
i=baca(i*10+7);kirimas(ri);
```

```
is(0,0);  
;
```

d ubahjam()

```
igned char ub;  
ek();  
ak(1,1,"ubah jam");  
le(tjam==1);  
d[0]-48;ub=ub*10+d[1]-48;  
te_rtc(2,ub);  
d[2]-48;ub=ub*10+d[3]-48;  
te_rtc(1,ub);  
d[4]-48;ub=ub*10+d[5]-48;  
te_rtc(4,ub);  
d[6]-48;ub=ub*10+d[7]-48;  
te_rtc(5,ub);  
d[8]-48;ub=ub*10+d[9]-48;  
te_rtc(6,ub);  
d[10]-48;ub=ub*10+d[11]-48;  
te_rtc(3,ub);
```

```

busek();
-----
Program Utama
-----
int main ()
/* Begin of Main
initser(230);
initlcd();delay(200);t=0;tjam=0;
cetak(1,2,"ABSEN SATPAM");
/*
write_rtc(0,0);
write_rtc(1,14);
write_rtc(2,11);
write_rtc(3,3);

write_rtc(4,13);
write_rtc(5,8);
write_rtc(6,8);
*/
// tulis(0,0);
busek();
koni(a);
while(1)/* Pengulangan Loop tanpa henti
{ /* Begin of while
if(tom==0)
{
busek();
i=j13*8+j12*4+j11*2+j10;
if(a>=15)cetak(1,1,"Masukan salah");
else
{
cetak(1,1,"Anda telah absen");
cetak(2,1,"di pos:");
koni(a);
masukdata();
}
delay(2000);
busek();

f(t==1)kirimdata();
f(tjam==1)ubahjam();
=read_rtc(3);hari(a);asu=' ';dataout();
=read_rtc(4);asu=a/10+48;dataout();asu=a%10+48;dataout();
su='/';dataout();
=read_rtc(5);asu=a/10+48;dataout();asu=a%10+48;dataout();
su='/';dataout();
=read_rtc(6);asu=a/10+48;dataout();asu=a%10+48;dataout();
os(2,1);
=read_rtc(2);asu=a/10+48;dataout();asu=a%10+48;dataout();
su=': ';dataout();
=read_rtc(1);asu=a/10+48;dataout();asu=a%10+48;dataout();
su=': ';dataout();
=read_rtc(0);asu=a/10+48;dataout();asu=a%10+48;dataout();

} /* End of while
* End of Main

```

ADDA.C

```
#ifndef adda_C
#define adda_C

#define ADDA_ADDR          0x90
#define EEPROM_ADDR       0xA0
#define RTC_ADDR          0xD0
#define RTC_Detik         0
#define RTC_Menit         1
#define RTC_Jam           2
#define RTC_Hari          3
#define RTC_Tanggal       4
#define RTC_Bulan         5
#define RTC_Tahun         6

#define SCL                P2_1
#define SDA                P2_0
#define SCL                P3_6
#define SDA                P3_7
```

```
char code *ptrTextHari[7] =
```

```
    "Sen",
    "Sel",
    "Rab",
    "Kam",
    "Jum",
    "Sab",
    "Mgg"
```

```
void i2c_start(void)
```

```
    SCL = 0;
    SDA = 1;
    SCL = 1;
    _asm
    nop
    nop
    nop
    nop
    _endasm;
    SDA = 0;
    _asm
    nop
    nop
    nop
    nop
    _endasm;
    SCL = 0;
```

```
void i2c_stop(void)
```

```
    SCL = 0;
    SDA = 0;
    SCL = 1;
    _asm
    nop
    nop
    nop
    nop
    _endasm;
    SDA = 1;
```

```
void i2c_write(unsigned char angka)
```



ADDA.C

```
unsigned char i;
```

```
for (i=8; i!=0; i--)
{
    SDA = (angka & 0x80)==0x80 ? 1 : 0;
    SCL = 1;
    _asm
    nop
    nop
    nop
    nop
    _endasm;
    SCL = 0;
    angka = angka << 1;
}
```

```
SDA = 1;
SCL = 1;
_asm
nop
nop
nop
nop
nop
nop
nop
nop
nop
nop
_endasm;
while(SDA);
SCL = 0;
_asm
nop
nop
nop
nop
nop
nop
nop
nop
_endasm;
```

```
signed char i2c_read(unsigned char last_addr)
```

```
unsigned char i,data_output;

SDA = 1;

for(i=8; i!=0; i--){
    data_output = data_output << 1;
    _asm
    nop
    nop
    nop
    _endasm;
    SCL = 1;
    _asm
    nop
    nop
    _endasm;
    data_output |= SDA;
    SCL = 0;
}
//data_output=data_output+0x80;
if (!last_addr) SDA = 1;
else
SDA = 0;
SCL = 1;
```

ADDA.C

```

_asm
nop
nop
nop
nop
_endasm;
SCL = 0;
_asm
nop
nop
nop
nop
nop
nop
nop
nop
nop
_endasm;
return data_output;

```

```

void delay_ms(unsigned int nilai)

```

```

{
    unsigned char angka, angka1;
    do{
        angka1 = 2;
        do{
            angka = 0;
            do{
                angka--;
            }while(angka != 0);
            angka1--;
        }while(angka1 != 0);
        nilai--;
    }while(nilai !=0);
}

```

```

unsigned char read_adc(unsigned char channel)

```

```

{
    unsigned char datax;
    i2c_start();
    i2c_write(ADDA_ADDR);
    //i2c_write((channel+0x10));
    i2c_write((channel+0x0));
    i2c_stop();
    i2c_start();
    i2c_write(ADDA_ADDR | 1);
    datax = i2c_read(0);
    i2c_stop();
    return datax;
}

```

```

void write_dac(unsigned char tmp1, unsigned char tmp2)

```

```

{
    i2c_start();
    i2c_write(ADDA_ADDR);
    i2c_write(tmp1+0x10);
    i2c_write(tmp2);
    i2c_stop();
}

```

```

unsigned char bcd2dec(unsigned char input)

```

```

{
    unsigned char tmp_data, tmp1;

    tmp_data = input;
    tmp1 = tmp_data % 16;
    if (tmp_data > 15) tmp_data = tmp_data / 16;
    else tmp_data = 0;
}

```

ADDA.C

```
tmp_data = (tmp_data * 10)+tmp1;  
return tmp_data;
```

signed char dec2bcd(unsigned char input)

```
unsigned char tmp_data;  
if (input > 9) tmp_data = ((input / 10)*16) + (input % 10);  
else  
    tmp_data = input;  
return tmp_data;
```

signed char read\_rtc(unsigned char alamat)

```
unsigned char tmp_data;  
i2c_start();  
i2c_write(RTC_ADDR);  
i2c_write(alamat);  
i2c_stop();  
i2c_start();  
i2c_write(RTC_ADDR | 1);  
tmp_data = i2c_read(0);  
i2c_stop();  
return bcd2dec(tmp_data);
```

void write\_rtc(unsigned char alamat, unsigned char datax)

```
    i2c_start();  
    i2c_write(RTC_ADDR);  
    i2c_write(alamat);  
    if (alamat < 7)  
        i2c_write(dec2bcd(datax));  
    else  
        i2c_write(datax);  
    i2c_stop();
```

signed char baca(unsigned char alamat)

```
unsigned char datax;  
i2c_start();  
i2c_write(EEPROM_ADDR);  
i2c_write(alamat);  
i2c_stop();  
i2c_start();  
i2c_write(EEPROM_ADDR | 1);  
datax = i2c_read(0);  
i2c_stop();  
return datax;
```

void tulis(unsigned char alamat, unsigned char nilai)

```
i2c_start();  
i2c_write(EEPROM_ADDR);  
i2c_write(alamat);  
i2c_write(nilai);  
i2c_stop();  
delay_ms(10);
```

signed char baca2(unsigned char chip\_addr, unsigned int address)

```
unsigned char datax
```

;

ADDA.C

```
unsigned char addresshi      ;
unsigned char addresslo     ;
unsigned char chip          ;
    addresshi      = address >> 8 ;
    addresslo      = address      ;
    chip           = chip_addr << 1 ;
i2c_start()                ;
i2c_write(EEPROM_ADDR | chip ) ;
i2c_write(addresshi)      ;
i2c_write(addresslo)     ;
    i2c_stop()            ;

i2c_start()                ;
i2c_write(EEPROM_ADDR | (chip | 1 )) ;
datax = i2c_read(0)      ;
i2c_stop()                ;
return datax              ;
```

id tulis2(unsigned char chip\_addr,unsigned int address, unsigned char nilai)

```
unsigned char addresshi      ;
unsigned char addresslo     ;
unsigned char chip          ;
    addresshi      = address >> 8 ;
    addresslo      = address      ;
    chip           = chip_addr << 1 ;
    i2c_start()                ;
i2c_write(EEPROM_ADDR | chip_addr ) ;
i2c_write(addresshi)      ;
i2c_write(addresslo)     ;
i2c_write(nilai)         ;
i2c_stop()                ;
delay_ms(10)              ;
```

ndif

pos1.c

```
#include <at89x51.h>
#include "Pending.c"
#include "myser1.c"
#include "lckdu.c"
#include "adda.c"
```

```
#define tom          P3_5
#define j13          P2_0
#define j12          P2_1
#define j11          P2_2
#define j10          P2_3
```

```
signed char a,id;
signed int i;
t;
time;
```

-----  
Program Utama  
-----

```
main ()
Begin of Main
d=1;
j3=id/8;
j2=id/4%2;
j1=id/2%2;
j0=id%2;
while(1)/* Pengulangan Loop tanpa henti
{/* Begin of While

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

pos 2.c

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

```
#define tom          P3_5
#define j13          P2_0
#define j12          P2_1
#define j11          P2_2
#define j10          P2_3
```

```
signed char a,id;
signed int i;
t;
; time;
```

-----  
Program Utama  
-----

```
int main ()
/* Begin of Main
id=2;
j13=id/8;
j12=id/4%2;
j11=id/2%2;
j10=id%2;
while(1)/* Pengulangan Loop tanpa henti
    {/* Begin of While
        }/* End of While
/* End of Main
```

```

Unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, CPort, ExtCtrls, DBCtrls, Grids, DBGrids, DB, DBTables, Menus,
  StdCtrls;

type
  TForm1 = class(TForm)
    tbid: TTable;
    tbpresensi: TTable;
    DSid: TDataSource;
    DSPresensi: TDataSource;
    ComPort1: TComPort;
    MainMenu1: TMainMenu;
    Pilihan1: TMenuItem;
    Refresh1: TMenuItem;
    Timer1: TTimer;
    tbpresensilkp: TTable;
    AmbilData1: TMenuItem;
    Timer2: TTimer;
    ampilanUtama1: TMenuItem;
    UbahRTC1: TMenuItem;
    Notebook1: TNotebook;
    DBGrid1: TDBGrid;
    Button2: TButton;
    Button3: TButton;
    DBGrid2: TDBGrid;
    Label1: TLabel;
    Label2: TLabel;
    ETGL: TEdit;
    EJAM: TEdit;
    Button4: TButton;
    Memo1: TMemo;
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure Refresh1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure AmbilData1Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure kirimdata(data:string);
    procedure DBGrid1DbClick(Sender: TObject);
    procedure ampilanUtama1Click(Sender: TObject);
    procedure UbahRTC1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  data_lkp:string;

implementation

{$R *.dfm}

procedure TForm1.kirimdata(data:string);
var i:integer;
begin
  ShowMessage(data);
  i:=0;

```

Untitled

```

repeat
inc(i);
ComPort1.WriteString(copy(data,i,1));
Sleep(100);
until length(data)=i;
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data:string;
begin
ComPort1.ReadStr(data,count);
data_lkp:=data_lkp+data;
Memo1.Text:=Memo1.Text+data;
Timer1.Enabled:=false;
Timer1.Enabled:=true;
end;

procedure TForm1.Refresh1Click(Sender: TObject);
begin
data_lkp:='';
Memo1.Clear;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
data_lkp:='';
Memo1.Clear;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
var jam,tgl,id:string;
    no:integer;
begin
Timer1.Enabled:=false;
if pos('a',data_lkp)<>0 then
begin
delete(data_lkp,1,pos('a',data_lkp));
repeat
ShowMessage(copy(data_lkp,1,14));
jam:=copy(data_lkp,1,2)+':' +copy(data_lkp,3,2)+':' +copy(data_lkp,5,2);
tgl:=copy(data_lkp,7,2)+'/' +copy(data_lkp,9,2)+'/20'+copy(data_lkp,11,2);
id:=copy(data_lkp,13,2);
if tbid.Findkey([id])then
begin
tbpresensi.kp.Last;
if tbpresensi.kp.RecordCount<>0 then
no:=tbpresensi.kp.fieldbyname('no').AsInteger+1
else no:=1;
tbpresensi.Append;
tbpresensi.fieldbyname('no').AsInteger:=no;
tbpresensi.fieldbyname('id').AsString:=id;
tbpresensi.fieldbyname('tgl').AsDateTime:=date;
tbpresensi.fieldbyname('jam').AsDateTime:=time;
tbpresensi.Post;
end;
delete(data_lkp,1,14);
until length(data_lkp)<14;
Refresh1.Click;
end;
end;

procedure TForm1.AmbilData1Click(Sender: TObject);
begin
ComPort1.WriteString('a');
end;

procedure TForm1.Button1Click(Sender: TObject);
var jam,tgl:string;

```



## Untitled

```
gin
m:=EJAM.Text;
l:=ETGL.Text;
ComPort1.WriteString('j');
ShowMessage(copy(jam,1,2)+copy(jam,4,2));
ComPort1.WriteString(copy(jam,1,2)+copy(jam,4,2));
ShowMessage(copy(tgl,1,2)+copy(tgl,4,2)+copy(tgl,9,2));
ComPort1.WriteString(copy(tgl,1,2)+copy(jam,4,2)+copy(jam,9,2));
ShowMessage('0'+IntToStr(DayOfWeek(date)));
ComPort1.WriteString('0'+IntToStr(DayOfWeek(date)));

rimdata('j'+copy(jam,1,2)+copy(jam,4,2)+copy(tgl,1,2)+copy(tgl,4,2)+copy(tgl,9
)
0'+IntToStr(DayOfWeek(date)));
d;

procedure TForm1.Timer2Timer(Sender: TObject);
gin
GL.Text:=DateToStr(date);
AM.Text:=timeToStr(time);
d;

procedure TForm1.DBGrid1DbClick(Sender: TObject);
gin
tebook1.ActivePage:='UJI';
d;

procedure TForm1.ampilanUtama1Click(Sender: TObject);
gin
tebook1.ActivePage:='UTAMA';
d;

procedure TForm1.UbahRTC1Click(Sender: TObject);
gin
tebook1.ActivePage:='RTC';
d;

procedure TForm1.Button2Click(Sender: TObject);
gin
id.Edit;
id.Post;
d;

procedure TForm1.Button3Click(Sender: TObject);
gin
[LE tbpresensi.RecordCount<>0 DO tbpresensi.Delete;
tbid.RecordCount<>0 THEN tbid.Delete;
d;

i.
```

Untitled

```
ect Form1: TForm1
Left = 191
Top = 113
Width = 288
Height = 640
Caption = 'ITN MALANG'
Color = clBtnFace
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -11
Font.Name = 'MS Sans Serif'
Font.Style = []
MainMenu = MainMenu1
IdCreateOrder = False
OnCreate = FormCreate
PixelsPerInch = 96
ExtentHeight = 13
object Notebook1: TNotebook
  Left = 8
  Top = 8
  Width = 249
  Height = 561
  TabOrder = 0
  object TPage
    Left = 0
    Top = 0
    Caption = 'UTAMA'
  object DBGrid1: TDBGrid
    Left = 32
    Top = 16
    Width = 185
    Height = 120
    DataSource = DSid
    TabOrder = 0
    TitleFont.Charset = DEFAULT_CHARSET
    TitleFont.Color = clWindowText
    TitleFont.Height = -11
    TitleFont.Name = 'MS Sans Serif'
    TitleFont.Style = []
    OnDblClick = DBGrid1DblClick
    Columns = <
      item
        Expanded = False
        FieldName = 'ID'
        Visible = True
      end
      item
        Expanded = False
        FieldName = 'NAMA'
        Title.Caption = 'NAMA POS'
        Visible = True
      end>
  end
end
object Button2: TButton
  Left = 32
  Top = 152
  Width = 75
  Height = 25
  Caption = 'SIMPAN'
  TabOrder = 1
 OnClick = Button2Click
end
object Button3: TButton
  Left = 136
  Top = 152
  Width = 75
  Height = 25
  Caption = 'HAPUS'
```

Untitled

```
TabOrder = 2
OnClick = Button3Click
end
object DBGrid2: TDBGrid
Left = 32
Top = 200
Width = 169
Height = 353
DataSource = DSpresensi
TabOrder = 3
TitleFont.Charset = DEFAULT_CHARSET
TitleFont.Color = clWindowText
TitleFont.Height = -11
TitleFont.Name = 'MS Sans Serif'
TitleFont.Style = []
Columns = <
  item
    Expanded = False
    FieldName = 'TGL'
    Visible = True
  end
  item
    Expanded = False
    FieldName = 'JAM'
    Visible = True
  end>
end
end
object TPage
Left = 0
Top = 0
Caption = 'RTC'
object Label1: TLabel
Left = 30
Top = 56
Width = 113
Height = 29
Caption = 'TANGGAL'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -24
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
object Label2: TLabel
Left = 28
Top = 96
Width = 47
Height = 29
Caption = 'JAM'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -24
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
object ETGL: TEdit
Left = 176
Top = 48
Width = 121
Height = 37
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -24
Font.Name = 'MS Sans Serif'
Font.Style = []
```

Untitled

```
ParentFont = False
ReadOnly = True
TabOrder = 0
end
object EJAM: TEdit
Left = 176
Top = 88
Width = 121
Height = 37
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -24
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
ReadOnly = True
TabOrder = 1
end
object Button4: TButton
Left = 94
Top = 192
Width = 115
Height = 81
Caption = 'UBAH'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -24
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 2
OnClick = Button1Click
end
end
object TPage
Left = 0
Top = 0
Caption = 'UJI'
object Memol: TMemo
Left = 64
Top = 40
Width = 185
Height = 89
Lines.Strings = (
'Memol')
TabOrder = 0
end
end
end
object tbid: TTable
Active = True
TableName = 'tbid.db'
Left = 152
Top = 24
end
object tbpresensi: TTable
Active = True
IndexName = 'ID_1'
MasterFields = 'ID'
MasterSource = DSid
TableName = 'tbpresensi.db'
Left = 192
Top = 24
end
object DSid: TDataSource
DataSet = tbid
Left = 152
Top = 64
```

Untitled

```
nd
object DSpresensi: TDataSource
  DataSet = tbpresensi
  Left = 192
  Top = 64
nd
object ComPort1: TComPort
  Connected = True
  BaudRate = br1200
  Port = 'COM1'
  Parity.Bits = prNone
  StopBits = sbOneStopBit
  DataBits = dbEight
  Events = [evRxChar, evTxEmpty, evRxFlag, evRing, evBreak, evCTS, evDSR,
  rror, evRLSD, evRx80Full]
  FlowControl.OutCTSFlow = False
  FlowControl.OutDSRFlow = False
  FlowControl.ControlDTR = dtrDisable
  FlowControl.ControlRTS = rtsDisable
  FlowControl.XonXoffOut = False
  FlowControl.XonXoffIn = False
  OnRxChar = ComPort1RxChar
  Left = 248
  Top = 40
nd
object MainMenu1: TMainMenu
  Left = 224
  Top = 96
  object Pilihan1: TMenuItem
    Caption = 'Pilihan'
    object Refresh1: TMenuItem
      Caption = 'Refresh'
      OnClick = Refresh1Click
    end
    object AmbilData1: TMenuItem
      Caption = 'Ambil Data'
      OnClick = AmbilData1Click
    end
    object ampilanUtama1: TMenuItem
      Caption = 'Tampilan Utama'
      OnClick = ampilanUtama1Click
    end
    object UbahRTC1: TMenuItem
      Caption = 'Ubah RTC'
      OnClick = UbahRTC1Click
    end
  end
nd
object Timer1: TTimer
  Enabled = False
  Interval = 2000
  OnTimer = Timer1Timer
  Left = 224
  Top = 160
nd
object tbpresensilkp: TTable
  Active = True
  TableName = 'tbpresensi.db'
  Left = 232
  Top = 24
nd
object Timer2: TTimer
  OnTimer = Timer2Timer
  Left = 224
  Top = 128
nd
```

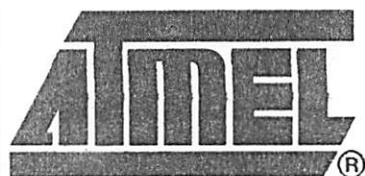
## Features

- Compatible with MCS<sup>®</sup>-51 Products
- 4K Bytes of In-System Programmable (ISP) Flash Memory
- 100,000 Write/Erase Cycles
- 0V to 5.5V Operating Range
- 0 Hz to 33 MHz Supply Static Operation
- Two-level Program Memory Lock
- 3 x 8-bit Internal RAM
- 3 Programmable I/O Lines
- Two 16-bit Timer/Counters
- 5 Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Two Data Pointers
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

## Description

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 cost-effective and flexible solution to many embedded control applications.

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



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

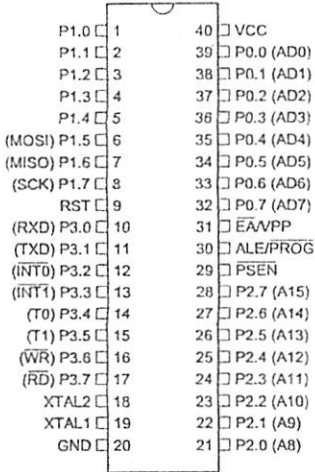
### AT89S51



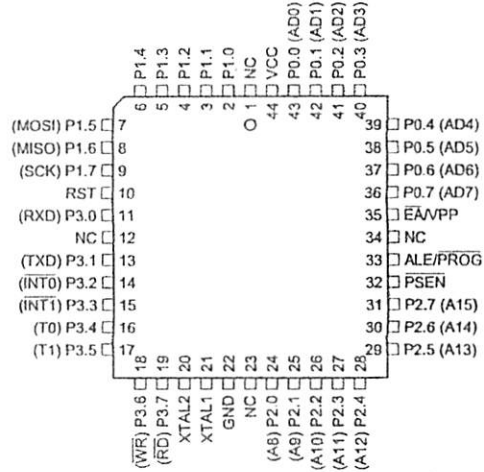


# Configurations

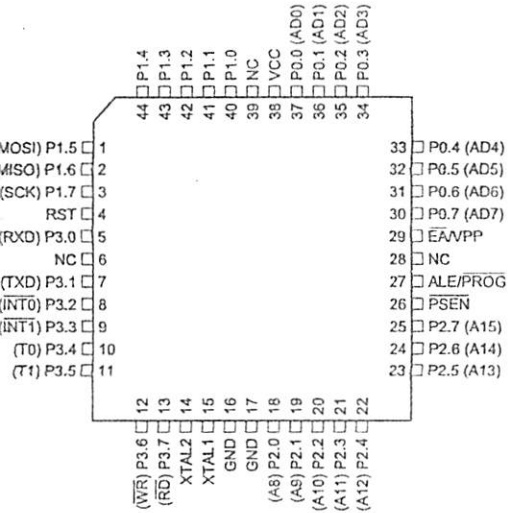
**PDIP**



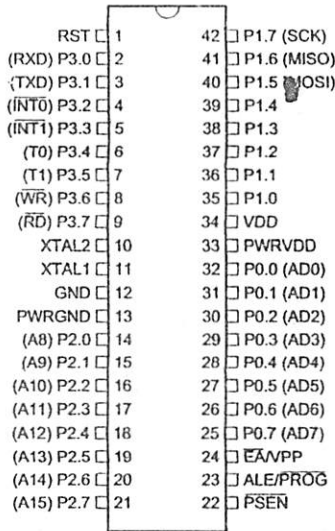
**PLCC**



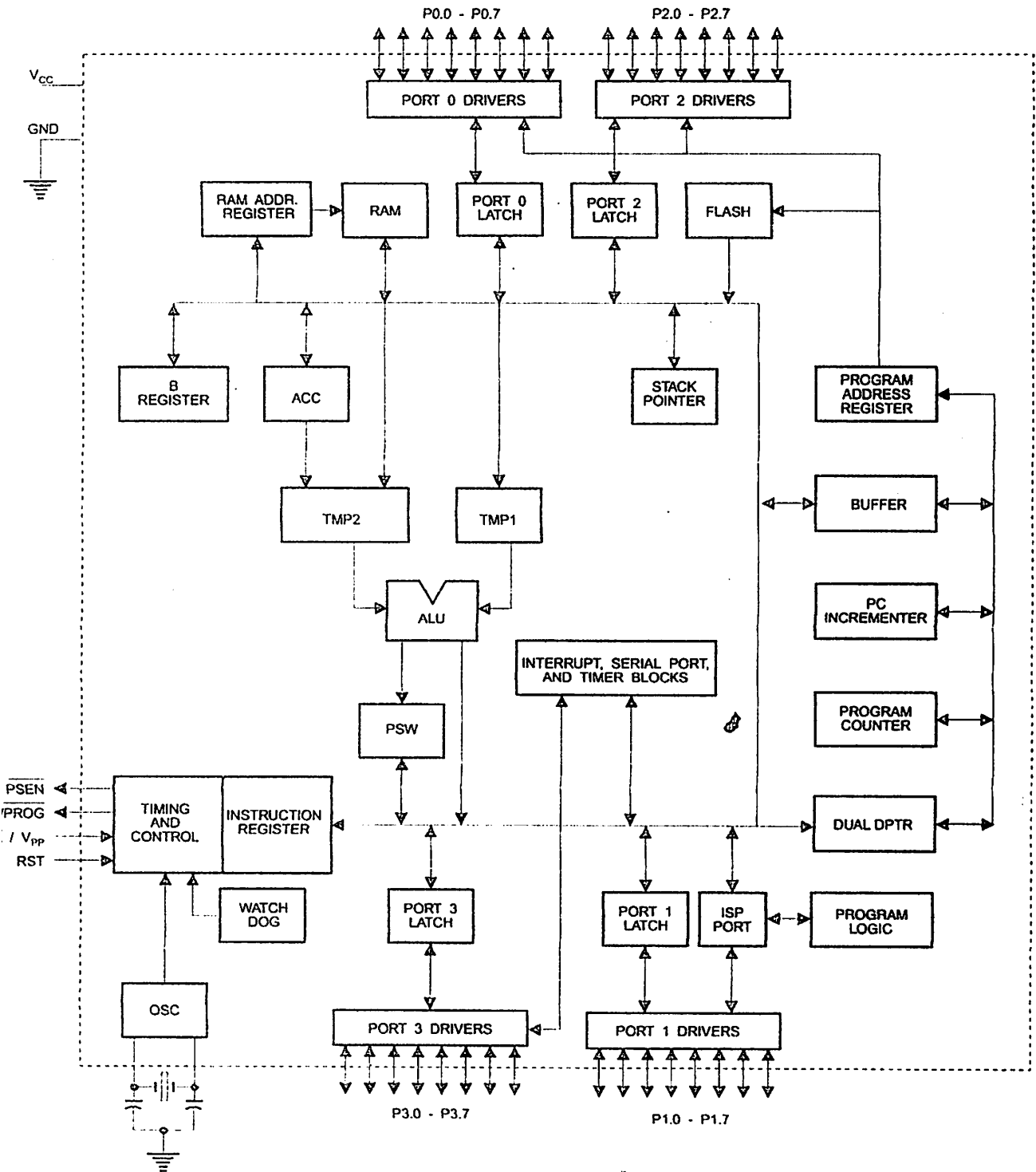
**TQFP**



**PDIP**



Block Diagram







## Description

C Supply voltage (all packages except 42-PDIP).

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

F8H								0FFH
F0H	B 00000000							0F7H
E8H								0EFH
E0H	ACC 00000000							0E7H
D8H								0DFH
D0H	PSW 00000000							0D7H
C8H								0CFH
C0H								0C7H
B8H	IP XX000000							0BFH
B0H	P3 11111111							0B7H
A8H	IE 0X000000							0AFH
A0H	P2 11111111		AUXR1 XXXXXXXX0				WDTRST XXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XX0000	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 = XXXXXXX0B
Not Bit Addressable								
	-	-	-	-	-	-	-	DPS
Bit	7	6	5	4	3	2	1	0
-	Reserved for future expansion							
DPS	Data Pointer Register Select							
	DPS							
	0	Selects DPTR Registers DP0L, DP0H						
	1	Selects DPTR Registers DP1L, DP1H						

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.

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.

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.

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.

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.

Table 4. Interrupt Enable (IE) Register

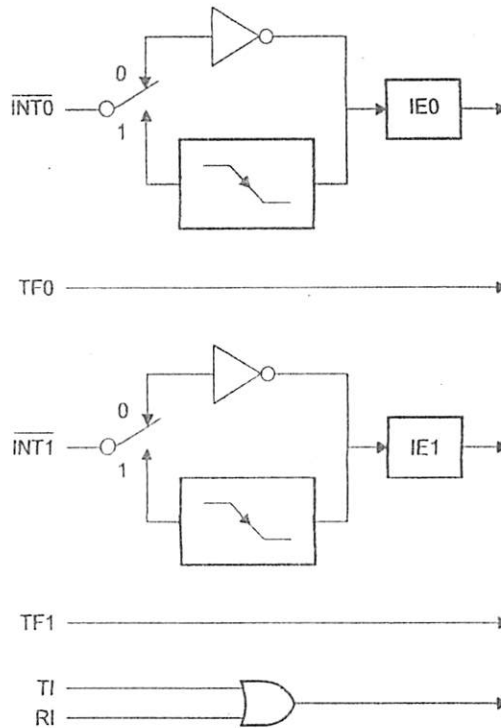
(MSB)				(LSB)			
EA	-	-	ES	ET1	EX1	ET0	EX0

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.

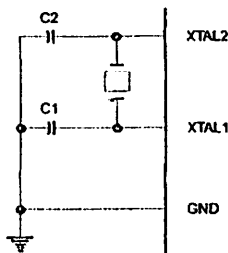
Figure 1. Interrupt Sources



Oscillator Characteristics

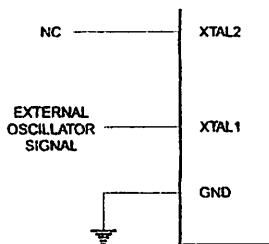
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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 ( $\overline{INT0}$  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

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

Table 6. Lock Bit Protection Modes

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

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

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

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

**Programming Algorithm:** Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash Programming Modes table (Table 7) and Figures 4 and 5. 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  $\overline{\text{RDY/BSY}}$  output signal. P3.0 is pulled low after ALE goes high during programming to indicate  $\overline{\text{BUSY}}$ . P3.0 is pulled high again when programming is done to indicate **READY**.

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

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

- (000H) = 1EH indicates manufactured by Atmel
- (100H) = 51H indicates AT89S51
- (200H) = 06H

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

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

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

## Programming Flash – Serial Mode

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

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

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

## Serial Programming Algorithm

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

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



Power-off sequence (if needed):

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

Set RST to "L".

Turn  $V_{CC}$  power off.

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

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

Serial Programming Instruction Set

Programming Interface – Parallel Mode

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

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

Table 7. Flash Programming Modes

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

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

Figure 4. Programming the Flash Memory (Parallel Mode)

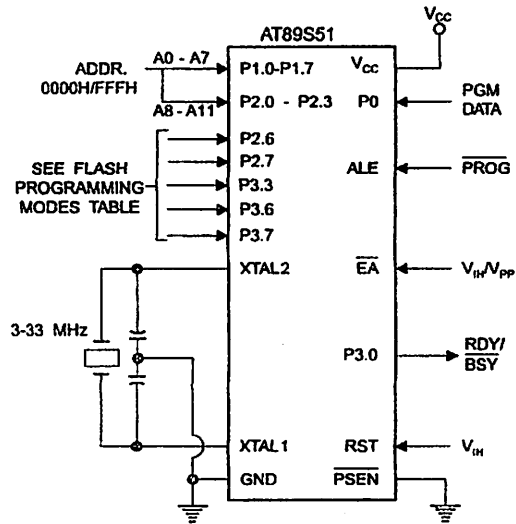
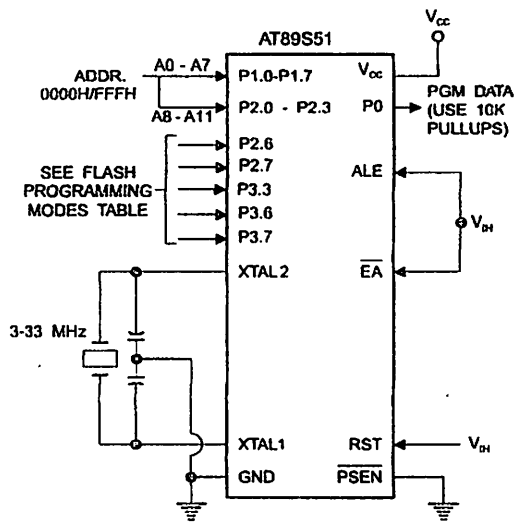


Figure 5. Verifying the Flash Memory (Parallel Mode)





## Flash Programming and Verification Characteristics (Parallel Mode)

$T = 20^{\circ}\text{C}$  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
$f_{CLCL}$	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		$\mu\text{s}$
$t_{HSL}$	$V_{PP}$ 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

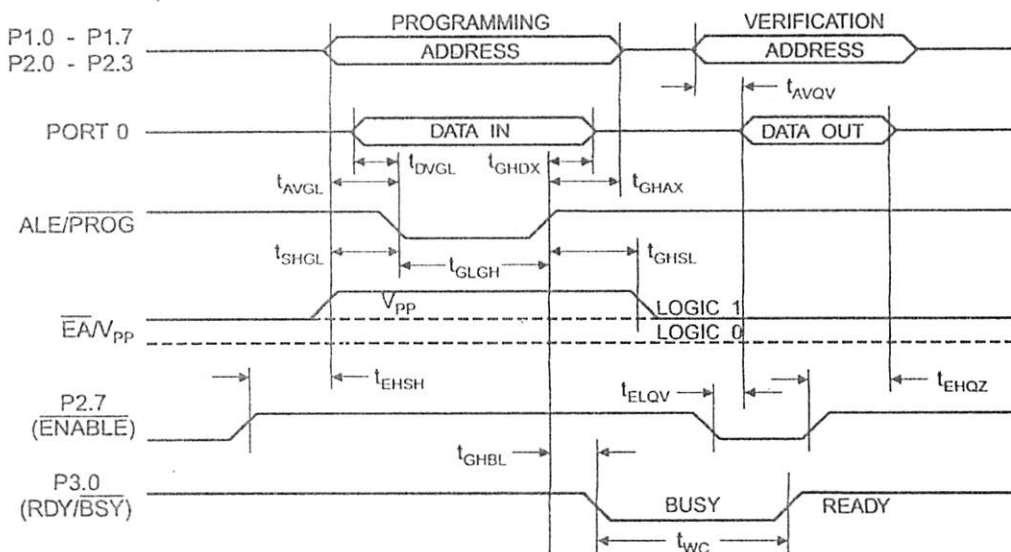
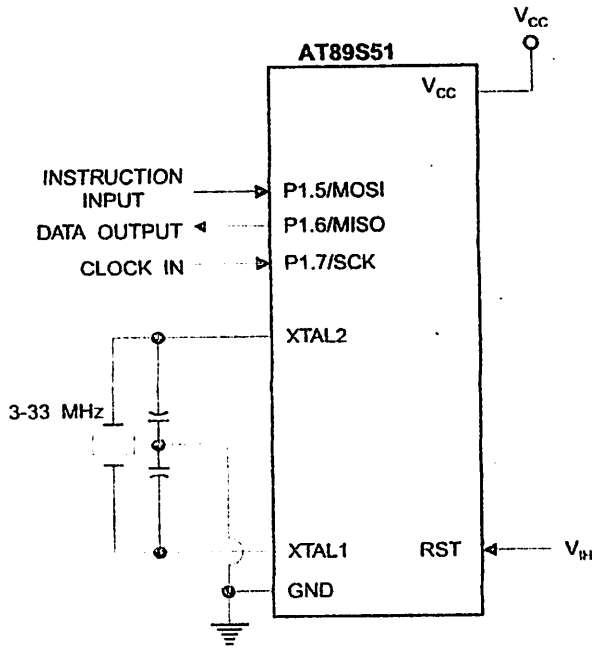


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms

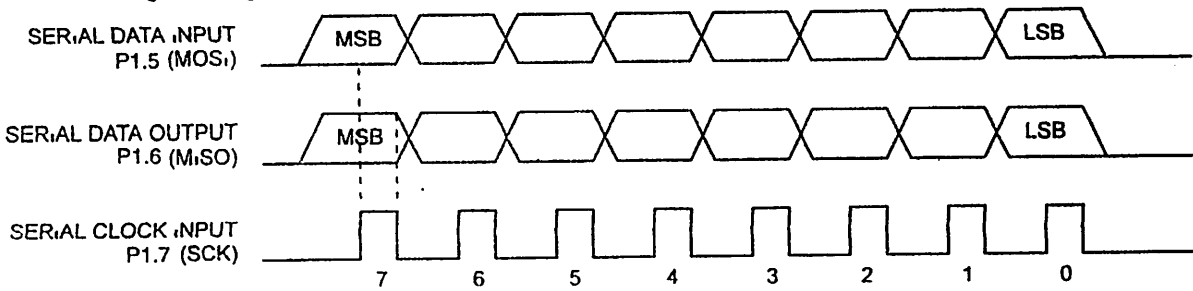




Table 8. Serial Programming Instruction Set

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 on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	7 6 5 4 D D D D 3 2 1 0	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 6 5 4 D D D D 3 2 1 0	Write data to Program memory in the byte mode
Write Lock Bits <sup>(1)</sup>	1010 1100	1110 00 LBL	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxx LBL LBL xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxxx A11 A10 A9 A8	A7 xxx xxx0	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

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

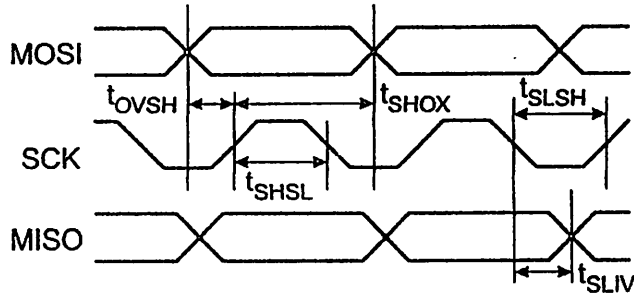
Each of the lock bit modes need to be activated sequentially before Mode 4 can be executed.

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

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

## Serial Programming Characteristics

### 9. Serial Programming Timing



### 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)

Parameter	Min	Typ	Max	Units
Oscillator Frequency	3		33	MHz
Oscillator Period	30			ns
SCK Pulse Width High	$8 t_{CLCL}$			ns
SCK Pulse Width Low	$8 t_{CLCL}$			ns
MOSI Setup to SCK High	$t_{CLCL}$			ns
MOSI Hold after SCK High	$2 t_{CLCL}$			ns
SCK Low to MISO Valid	10	16	32	ns
Chip Erase Instruction Cycle Time			500	ms
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 Respect to Ground.....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
Maximum Output Current.....	15.0 mA

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

## Characteristics

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

Symbol	Parameter	Condition	Min	Max	Units
	Input Low Voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V
	Input Low Voltage ( $\overline{EA}$ )		-0.5	$0.2 V_{CC} - 0.3$	V
	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, $\overline{PSEN}$ )	$I_{OL} = 3.2 \text{ mA}$		0.45	V
	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$ )	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	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
	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
	Input Leakage Current (Port 0, $\overline{EA}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode <sup>(2)</sup>	$V_{CC} = 5.5\text{V}$		50	$\mu\text{A}$

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



Characteristics

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

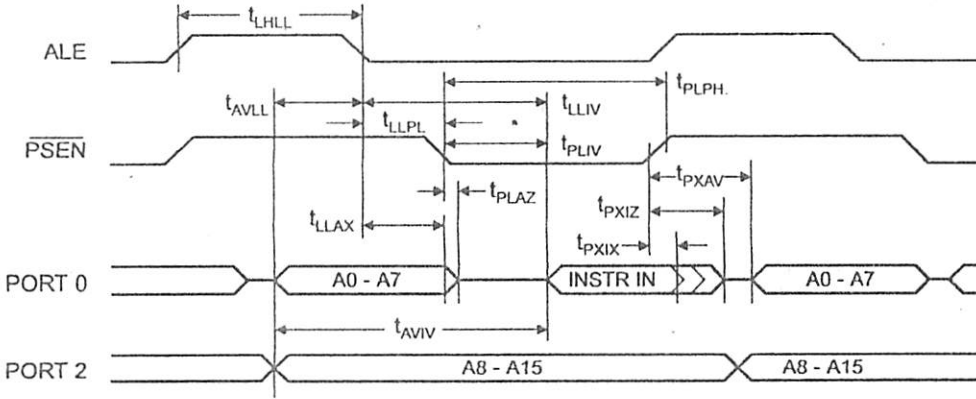
Internal Program and Data Memory Characteristics

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

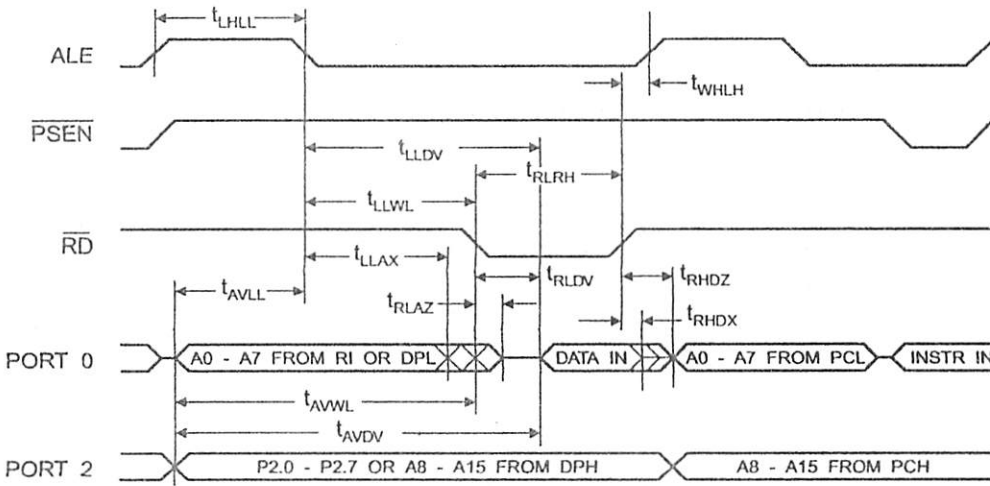




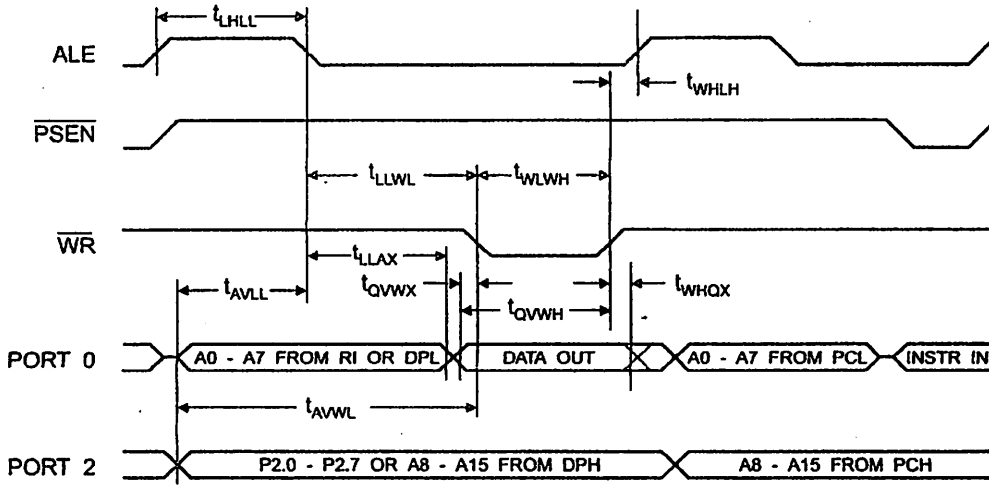
## Internal Program Memory Read Cycle



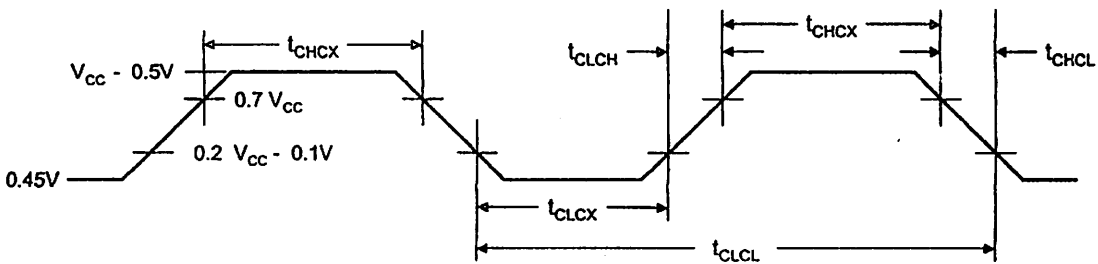
## Internal Data Memory Read Cycle



Internal Data Memory Write Cycle



Internal Clock Drive Waveforms



Internal Clock Drive

Symbol	Parameter	Min	Max	Units
f <sub>CL</sub>	Oscillator Frequency	0	33	MHz
T <sub>CL</sub>	Clock Period	30		ns
t <sub>CH</sub>	High Time	12		ns
t <sub>CL</sub>	Low Time	12		ns
t <sub>r</sub>	Rise Time		5	ns
t <sub>f</sub>	Fall Time		5	ns

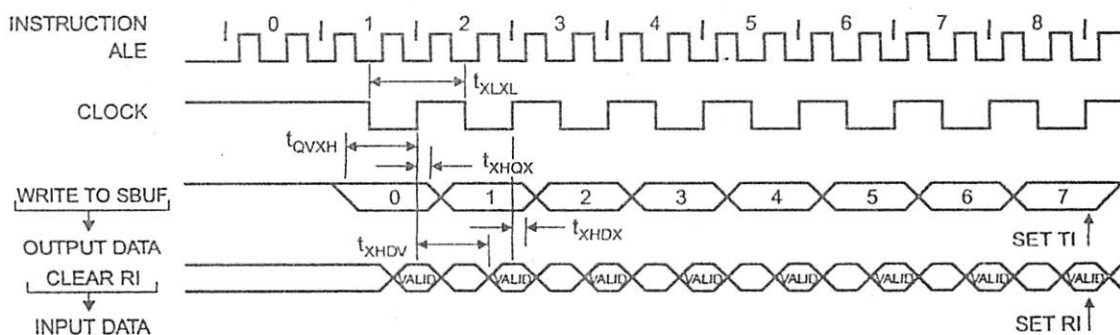


## Serial Port Timing: Shift Register Mode Test Conditions

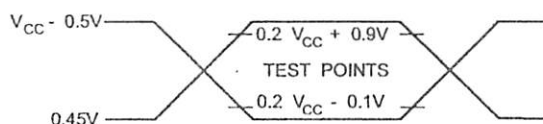
Values in this table are valid for  $V_{CC} = 4.0V$  to  $5.5V$  and Load Capacitance =  $80\text{ pF}$ .

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

## Shift Register Mode Timing Waveforms

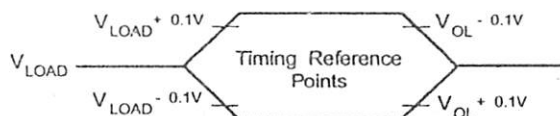


## Testing Input/Output Waveforms<sup>(1)</sup>



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

## Timing Waveforms<sup>(1)</sup>



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

## Ordering Information

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

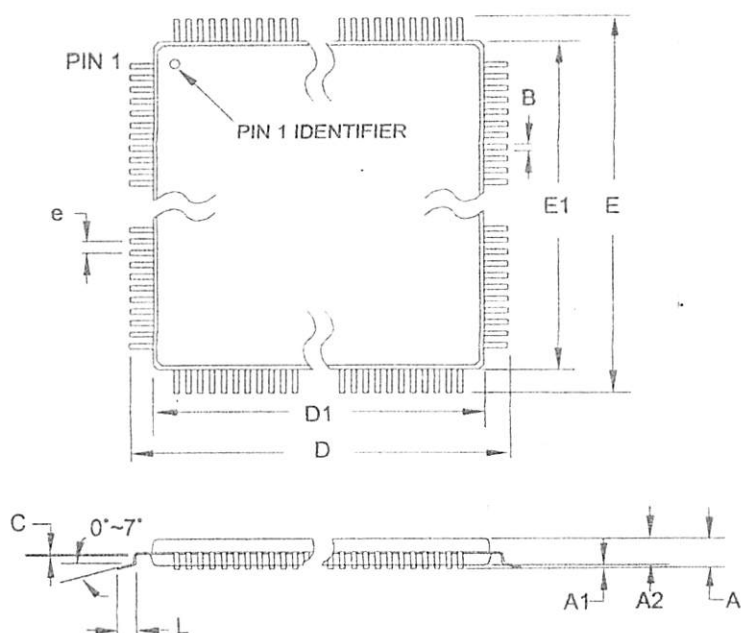
Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
42PS6	42-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)





# Packaging Information

44A - TQFP



**COMMON DIMENSIONS**  
(Unit of Measure = mm)

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

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

10/5/2001

AMEL 2325 Orchard Parkway  
San Jose, CA 95131

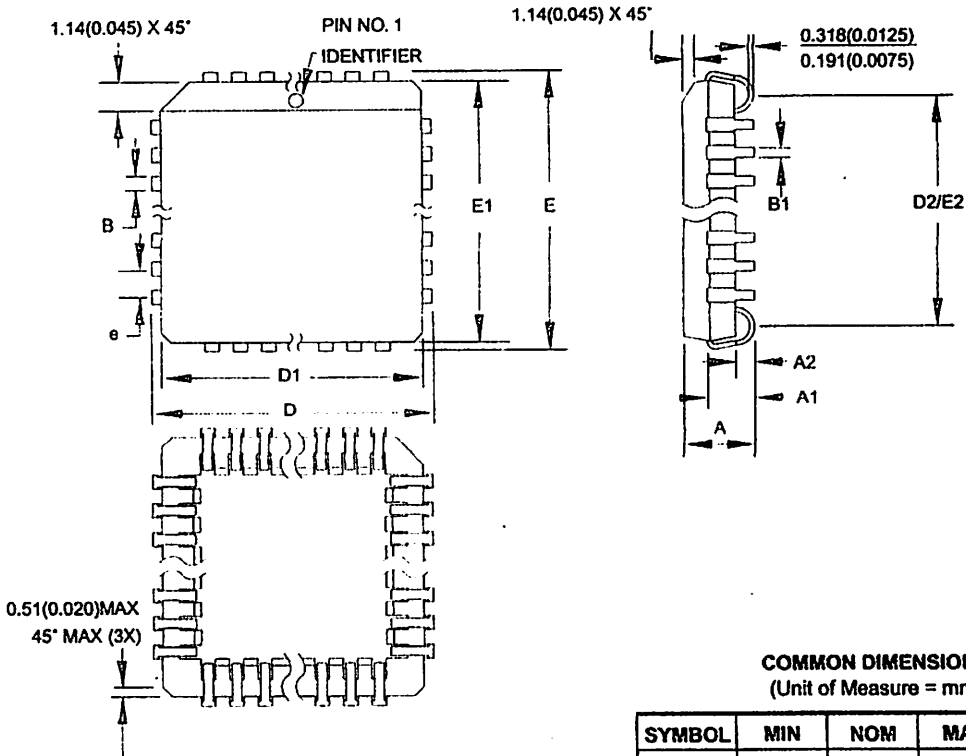
**TITLE**  
44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness,  
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)

**DRAWING NO.** 44A  
**REV.** B

**AT89S51**

2487B-MICRO-12/03

- PLCC



COMMON DIMENSIONS  
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	-	4.572	
A1	2.286	-	3.048	
A2	0.508	-	-	
D	17.399	-	17.653	
D1	16.510	-	16.662	Note 2
E	17.399	-	17.653	
E1	16.510	-	16.662	Note 2
D2/E2	14.986	-	16.002	
B	0.660	-	0.813	
B1	0.330	-	0.533	
e	1.270 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
  2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010" (0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
  3. Lead coplanarity is 0.004" (0.102 mm) maximum.

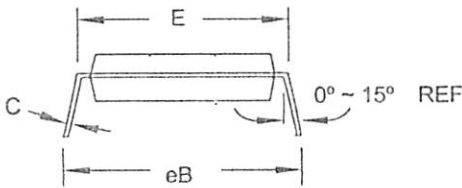
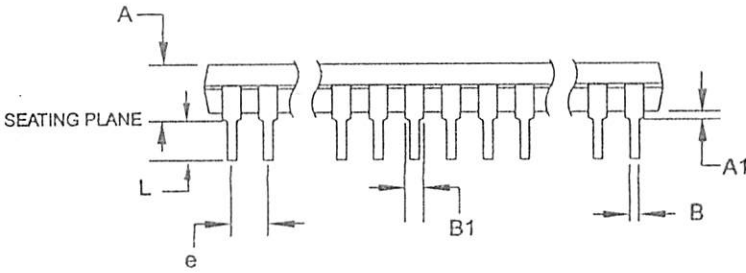
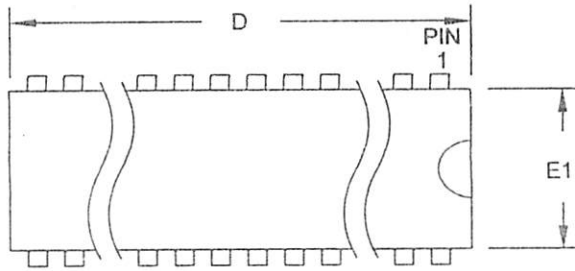
10/04/01

 2325 Orchard Parkway San Jose, CA 95131	<b>TITLE</b> 44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)	<b>DRAWING NO.</b> 44J	<b>REV.</b> B
---	--	---------------------------	------------------





40P6 - PDIP



COMMON DIMENSIONS  
(Unit of Measure = mm)

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

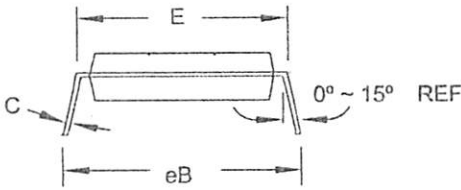
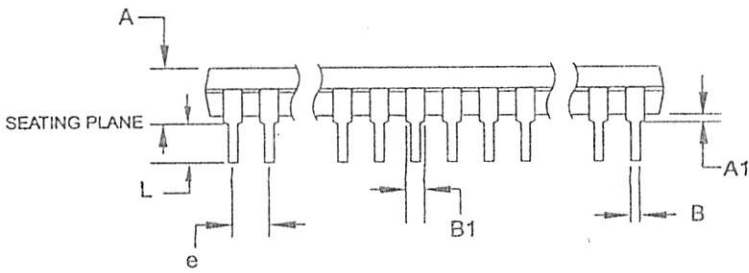
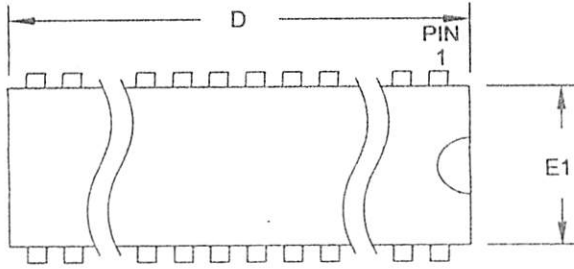
- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
  2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

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



42PS6 - PDIP




COMMON DIMENSIONS  
(Unit of Measure = mm)

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

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

11/6/03

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





## Atmel Corporation

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

## Regional Headquarters

### Europe

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

### Asia

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

### Japan

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

## Atmel Operations

### Memory

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

### Microcontrollers

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

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

### ASIC/ASSP/Smart Cards

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

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

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

### RF/Automotive

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

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

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

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

**Literature Requests**  
[www.atmel.com/literature](http://www.atmel.com/literature)

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

© Atmel Corporation 2003. All rights reserved. Atmel® and combinations thereof are the registered trademarks of Atmel Corporation or its subsidiaries. MCS® is a registered trademark of Intel Corporation. Adobe® and Acrobat® are the registered trademarks of Adobe Systems Inc. Other terms and product names may be the trademarks of others.



Printed on recycled paper.

2487B-MICRO-12/03

## **Komunikasi Antar IC dengan IIC**

### **1. Latar Belakang dan Konsep IIC**

Pada saat ini desain elektronik dituntut untuk semakin ringkas dan fleksibel, dimana ukuran fisik IC semakin diperkecil dan jumlah pin diminimalkan dengan tetap menjaga fleksibilitas dan kompatibilitas IC sehingga mudah untuk digunakan dalam berbagai keperluan desain yang berbeda, oleh karenanya banyak perusahaan semikonduktor yang berusaha mengembangkan cara baru komunikasi antar IC yang lebih akomodatif terhadap tuntutan diatas sebagai alternatif dari hubungan antar IC secara paralel (parallel bus) yang sudah kita kenal luas. Salah satu metode yang telah matang dan dipakai secara luas adalah IIC (sering ditulis juga I2C) singkatan dari Inter Integrated Circuit bus yang dikembangkan oleh Philips Semiconductor sejak tahun 1992, dengan konsep dasar komunikasi 2 arah antar IC dan/atau antar sistem secara serial menggunakan 2 kabel.

### **2. Fitur Utama IIC**

Fitur utama I2C bus adalah sebagai berikut :

- Hanya melibatkan dua kabel yaitu serial data line (selanjutnya disebut SDA) dan serial clock line (selanjutnya disebut SCL).
- Setiap IC yang terhubung dalam I2C memiliki alamat yang unik yang dapat diakses secara software dengan master/slave protocol yang sederhana, dan mampu mengakomodasikan multi master (akan dijelaskan lebih detil pada bagian lain).
- I2C merupakan serial bus dengan orientasi data 8 bit (byte), komunikasi 2 arah, dengan kecepatan transfer data sampai 100Kbit/s pada mode standart dan 3,4 Mbit/s pada mode kecepatan tinggi.
- Jumlah IC yang dapat dihubungkan pada I2C bus hanya dibatasi oleh beban kapasitansi pada bus yaitu maksimum 400pF.

### **3. Keuntungan I2C**

Keuntungan yang didapat dengan menggunakan I2C antara lain :

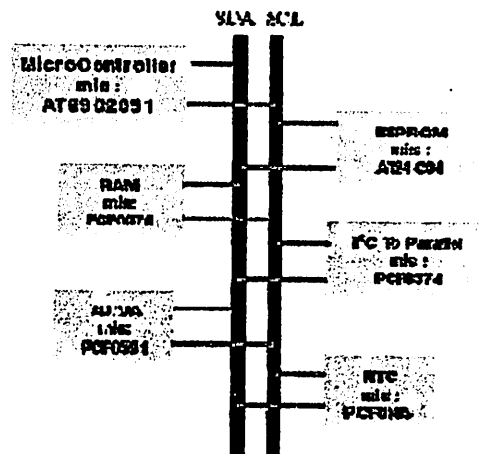
- Meminimalkan jalur hubungan antar IC (bandingkan dengan parallel bus !).
- Menghemat luasan PCB yang dibutuhkan.
- Membuat sistem yang didesain berorientasi software (mudah diekspan dan diupgrade).
- Membuat sistem yang didesain menjadi standart, sehingga dapat dihubungkan dengan sistem lain yang juga menggunakan I2C bus.

### **4. Cara kerja I2C Bus**

Sebelum memahami cara kerjanya, ada beberapa terminologi, karakter dan kondisi penting dalam I2C yang harus dipahami terlebih dahulu, yaitu :

## 4.1 Terminologi

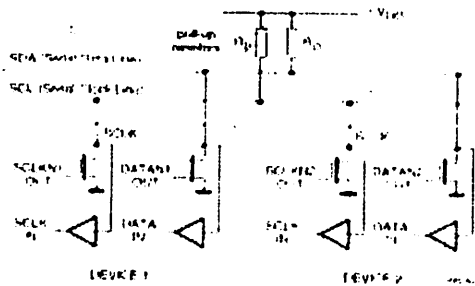
- **Transmitter** yaitu device yang mengirim data ke bus.
- **Receiver** yaitu device yang menerima data dari bus.
- **Master** yaitu device yang memiliki inisiatif (memulai dan mengakhiri) transfer data dan yang membangkitkan sinyal clock.
- **Slave** yaitu device yang dialamati (diakses berdasarkan alamatnya) oleh Master.
- **Multi-master** yaitu sistem yang memungkinkan lebih dari satu Master melakukan inisiatif transfer data dalam waktu yang bersamaan tanpa terjadi korupsi data.
- **Arbitration** yaitu prosedur yang memastikan bahwa jika ada lebih dari satu Master melakukan inisiatif transfer data secara bersamaan, hanya akan ada satu Master yang diperbolehkan dengan tanpa merusak data yang sedang ditransfer.
- **Synchronization** yaitu prosedur untuk menyelaraskan sinyal clock dari dua atau lebih device.



**Gambar 1**  
**Contoh Sistem dengan IIC Bus**

## 4.2 Karakter perangkat keras

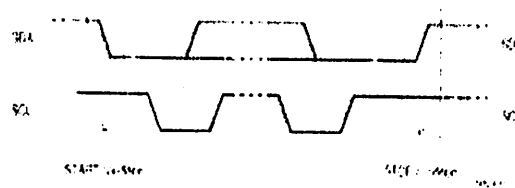
- Kedua pin pada I2C yaitu SDA dan SCL harus memiliki kemampuan input dan output serta bersifat open drain atau open collector.
- Kedua pin tersebut terhubung pada I2C bus yang telah di pull-up dengan resistor ke suplai positif dari sistem
- Semua device yang terhubung pada bus harus terhubung pada ground yang sama sebagai referensi.



**GAMBAR 2**  
**KONEKSI IIC BUS**  
**Komunikasi Antar IC dengan IIC - Halaman 1**

### 4.3 Karakter Transfer Data Bit

Data bit dikirim/diterima melalui SDA, sedangkan sinyal clock dikirim/diterima melalui SCL, dimana dalam setiap transfer data bit satu sinyal clock dihasilkan, transfer data bit dianggap valid jika data bit pada SDA tetap stabil selama sinyal clock high, data bit hanya boleh berubah jika sinyal clock dalam kondisi low, lihat gambar 3.



**GAMBAR 3**  
**TRANSFER DATA BIT PADA I2C BUS**

### 4.4 Karakter Transfer Data Byte

I2C Bus berorientasi pada 8 bit data (byte), dengan most significant bit / MSB ditransfer terlebih dulu, serta 2 macam data byte yaitu Address Byte dan Data Byte.

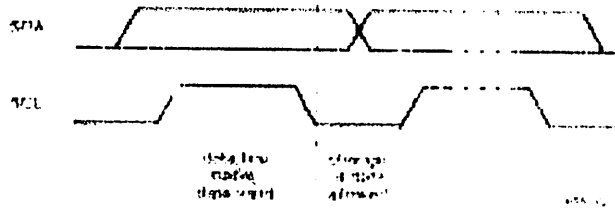
### 4.5 Kondisi

#### 4.5.1 START dan STOP

Apabila pada SDA terjadi transisi dari kondisi high ke kondisi low pada saat SCL berkondisi high, maka terjadilah kondisi START.

Apabila pada SDA terjadi transisi dari kondisi low ke kondisi high pada saat SCL berkondisi high, maka terjadilah kondisi STOP.

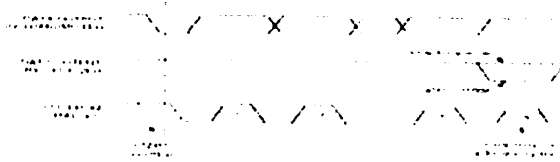
Kondisi START dan STOP selalu dibangkitkan oleh Master, dan bus dikatakan sibuk setelah START dan dikatakan bebas setelah STOP.



**GAMBAR 4**  
**KONDISI START DAN STOP**

#### 4.5.2 ACK dan NACK

Kondisi ACK terjadi apabila receiver "menarik" SDA pada kondisi low selama 1 sinyal clock. Kondisi NACK terjadi apabila receiver "membebaskan" SDA pada kondisi high selama 1 sinyal clock.



**GAMBAR 5**  
**KONDISI ACK DAN NACK**

#### 4.5.3 Cara kerja I2C Bus (Format 7 bit address)

Cara kerja I2C bus dapat dibedakan menjadi format 7 bit addressing dan format 10 bit addressing, dalam artikel ini hanya akan dibahas format 7 bit addressing, untuk format 10 bit addressing dapat anda baca keterangan lengkapnya dalam "The I2C-BUS Specification Version 2.1 January 2000" yang ditulis oleh Philips Semiconductor. Inisiatif komunikasi/transfer data selalu oleh Master dengan mengirimkan kondisi START diikuti dengan address byte (7 bit address + 1 bit pengarah/data direction bit) seperti pada ilustrasi dibawah ini



**GAMBAR 6**  
**FORMAT ADDRESS BYTE**



Sinyal Acknowledge (ACK) terjadi :

Dari Slave ke Master Transmitter :

- Sesudah address byte diterima dengan baik oleh slave
- Setiap kali slave selesai menerima data byte dengan baik

Dari Master Receiver ke Slave :

- Setiap kali Master selesai menerima data byte dengan baik

Sinyal Negative Acknowledge (NACK) terjadi :

Dari Slave ke Master Transmitter :

- Setelah slave gagal menerima address byte dengan baik
- Setiap kali slave gagal menerima data byte dengan baik
- Slave tidak terhubung pada bus

Dari Master Receiver ke slave :

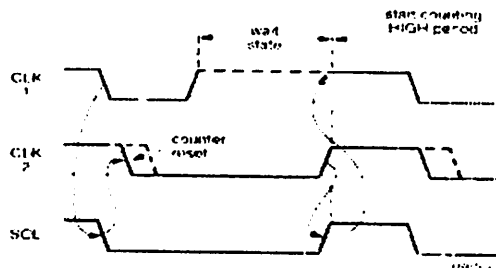
- Setelah Master menerima data byte yang terakhir dari slave

## 5. Multi Master

Pada I2C bus bisa terjadi situasi dimana lebih dari 1 device mengambil inisiatif transfer data sebagai Master, dengan protocol Master/slave dan karakter hardware open drain/open collector yang bersifat wired AND, hal ini tidak menyebabkan terjadinya korupsi data, inilah yang disebut dengan Multi Master.

Untuk dapat melakukan Multi Master ada 2 hal yang penting yaitu Clock Synchronization dan Arbitration.

### 5.1 Clock Synchronization



**GAMBAR 11**  
**CLOCK SYNCHRONIZATION**

Karena sifat wired AND dari I2C bus, dimana jika salah satu device menarik bus dalam kondisi low maka device lain tidak dapat membuat bus tersebut menjadi high (sifat dari logika AND), sehingga jika ada lebih dari satu device yang melakukan inisiatif transfer data sebagai Master dengan membangkitkan sinyal clock pada SCL pada saat yang bersamaan harus ada sinkronisasi clock yang dapat dijelaskan (seperti terlihat pada gambar 10 diatas) sebagai berikut :

- Jika Master1 (Clock 1) memulai periode low sinyal clock-nya, maka SCL menjadi low,



Master2 mendeteksi kondisi tersebut dan harus juga memulai menghitung periode low sinyal clock-nya.

- Saat Master1 (Clock 1) akan memulai periode high sinyal clock-nya dan mendeteksi bahwa SCL masih dalam kondisi low (disebabkan periode low sinyal clock dari Master 2 (Clock 2) masih belum selesai) maka dia harus menunggu dan tidak menghitung periode high sinyal clock-nya terlebih dahulu.
- Saat Master2 (Clock 2) memulai periode high sinyal clock-nya, maka kondisi SCL menjadi high, Master1 (Clock1) yang mendeteksi kondisi tersebut juga harus memulai menghitung periode high sinyal clock-nya.
- Karena Master 1 (Clock1) terlebih dahulu menyelesaikan periode high sinyal clock-nya dan memulai periode low maka kondisi SCL menjadi low, maka Master 2 (Clock 2) yang mendeteksi kondisi tersebut juga harus memulai menghitung periode low sinyal clock-nya, demikian seterusnya sehingga terjadilah sinkronisasi sinyal clock antara Master1 dan Master2.

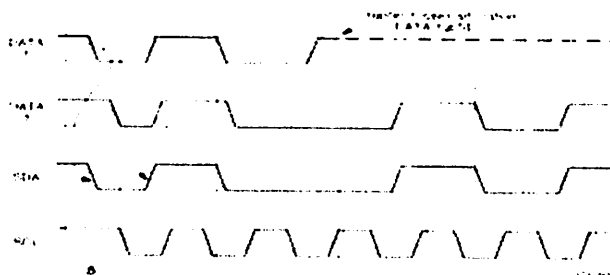
Secara singkat sinkronisasi clock dapat disimpulkan sbb:

1. Jika kondisi SCL tetap low pada saat Master mencoba membuatnya high, maka Master tersebut harus memulai menghitung periode low sinyal clock-nya
2. Jika Master akan memulai periode high sinyal clock-nya, maka Master tersebut harus menunggu kondisi SCL menjadi high sebelum memulai menghitung periode high sinyal clock-nya.

Sehingga sinkronisasi clock yang terbentuk sbb:

- Periode low akan mengikuti periode low dari device yang membangkitkan sinyal clock dengan periode low yang terpanjang.
- Periode high akan mengikuti periode high dari device yang membangkitkan sinyal clock dengan periode high yang terpendek.

## 5.2 Arbitration



**GAMBAR 12**  
**ARBITRATION**

Dalam Multi Master, bisa terjadi kemungkinan lebih dari satu device melakukan inisiatif transfer data menjadi Master, walaupun transfer data hanya bisa dilakukan jika kondisi bus bebas, tetapi sangat memungkinkan lebih dari satu device mendeteksi kondisi bus sebagai bebas dan membangkitkan kondisi START sedikit berselisih waktu tetapi masih dalam batas-

batas kondisi START yang valid. Untuk kondisi seperti dijelaskan diatas, maka arbitration diberlakukan bit demi bit hingga selesai, dimana sekali lagi sifat/karakter bus yang wired AND memungkinkan hal tersebut terjadi.

Untuk jelasnya perhatikan gambar 11 diatas yang dapat dijelaskan sebagai berikut :

- Ambil contoh Master 1 (Data 1) akan mentransfer data 101xxxxB sedangkan Master 2 (Data 2) akan mentransfer data 100101xxB.
- Kedua Master mendeteksi bus dalam keadaan bebas, dan membangkitkan sinyal START yang hampir bersamaan, Master 1 lebih dahulu membangkitkan START sehingga kondisi SDA mengikuti Master 1, baru kemudian Master 2 membangkitkan START, tetapi kondisi START pada SDA masih valid untuk Master2.
- Kedua Master mentransfer 1MSbit (sama-sama "1"), kemudian data bit berikutnya (sama-sama "0"), pada bit yang berikutnya Master1 berusaha untuk membuat SDA high sesuai dengan data bit-nya, sedangkan Master 2 berusaha untuk membuat SDA low (sesuai dengan data bit-nya), karena sifat wired AND dari SDA, maka kondisi SDA menjadi low, karena itu Master 1 dikatakan kehilangan arbitras (dengan kata lain bisa disebut sebagai kehilangan kontrol) atas SDA.
- Bagi Master 1 yang kehilangan arbitras bisa terus menghancurkan sinyal clock sampai transfer data selesai dan bus dalam kondisi bebas lagi, bagi Master 2 yang memenangkan arbitras (mendapat kontrol) atas SDA dapat menyelesaikan transfer data-nya tanpa ada data yang terkorupsi sama sekali.

## 6. Kesimpulan

Komunikasi I2C dapat memenuhi target desain elektronika saat ini, dengan karakter hardware dan master/slave protocol yang sederhana tetapi tangguh.

Jika digunakan device yang telah dilengkapi dengan kemampuan I2C secara built-in, akan sangat mudah untuk melaksanakan komunikasi baik single Master maupun multi Master, tetapi bagi device yang tidak, masih dapat mengemulasikan I2C protocol pada pin-pin I/O-nya asalkan memenuhi syarat karakter hardware dengan cukup mudah untuk single Master tetapi sedikit rumit untuk multi Master.

Artikel ini tidak membahas secara lengkap seluruh kemampuan I2C seperti misalnya I2C mode kecepatan tinggi, tetapi cukup detil memberikan gambaran tentang I2C, untuk mengetahui secara lengkap bisa didapatkan dari referensi untuk artikel ini yang dicantumkan dibawah.

## 7. Referensi

1. THE I2C-BUS SPECIFICATION VERSION 2.1 JANUARY 2000, PHILIPS SEMICONDUCTOR
2. APPLICATION NOTE AN435A, PHILIPS SEMICONDUCTOR.
3. APPLICATION NOTES I2C SPECIFIC INFORMATION, PHILIPS SEMICONDUCTOR
4. DATA SHEET PCF3591 8-BIT A/D AND D/A CONVERTER, PHILIPS SEMICONDUCTOR

## Features

### Wide-Voltage and Standard-Voltage Operation

5.0 ( $V_{CC} = 4.5V$  to  $5.5V$ )

2.7 ( $V_{CC} = 2.7V$  to  $5.5V$ )

2.5 ( $V_{CC} = 2.5V$  to  $5.5V$ )

1.8 ( $V_{CC} = 1.8V$  to  $5.5V$ )

Low-Power Devices ( $I_{SD} = 2 \mu A$  @  $5.5V$ ) Available

Internally Organized 4096 x 8, 8192 x 8

2-Wire Serial Interface

Interrupt Trigger, Filtered Inputs for Noise Suppression

Bi-Directional Data Transfer Protocol

100 kHz (1.8V, 2.5V, 2.7V) and 400 kHz (5V) Compatibility

Tri-State Protect Pin for Hardware Data Protection

Byte Page Write Mode (Partial Page Writes Allowed)

Programmable Write Cycle (10 ms max)

High Reliability

Endurance: 1 Million Write Cycles

Data Retention: 100 Years

ESD Protection: >3,000V

Automotive Grade and Extended Temperature Devices Available

Available in JEDEC PDIP, 8-Pin and 14-Pin JEDEC SOIC, 8-Pin EIAJ SOIC,

and 8-pin TSSOP Packages

## Description

AT24C32/64 provides 32,768/65,536 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 4096/8192 words of 8 bits

The device's cascadable feature allows up to 8 devices to share a common 2-wire bus.

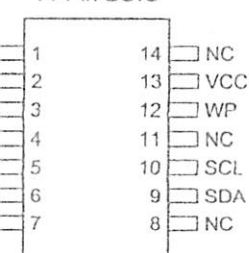
The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The AT24C32/64 is available in space saving 8-pin JEDEC PDIP, 8-pin and 14-pin JEDEC SOIC, 8-pin TSSOP, and 8-pin TSSOP packages and is accessed via a 2-wire serial interface.

In addition, the entire family is available in 5.0V (4.5V to 5.5V), 2.7V (2.7V to 5.5V), 2.5V (2.5V to 5.5V) and 1.8V (1.8V to 5.5V) versions.

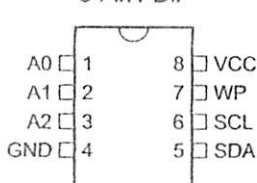
## Configurations

Name	Function
A0-A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect

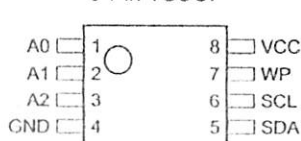
14-Pin SOIC



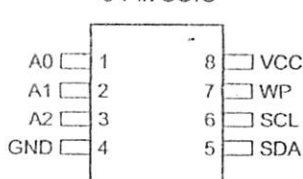
8-Pin PDIP



8-Pin TSSOP



8-Pin SOIC



## 2-Wire Serial EEPROM

32K (4096 x 8)

64K (8192 x 8)

AT24C32

AT24C64



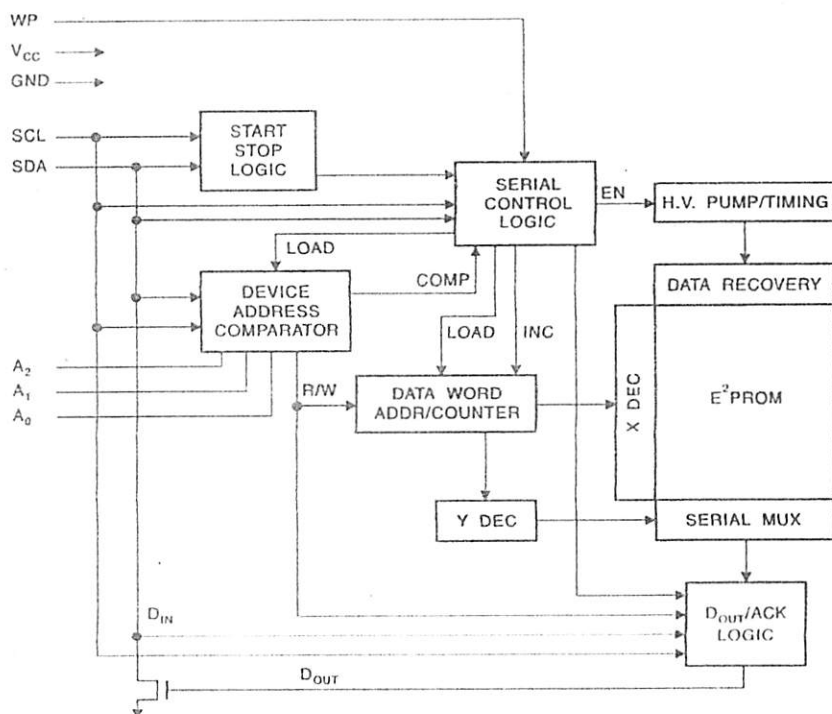


## Absolute Maximum Ratings\*

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.....	6.25V
IO Pin Output Current.....	5.0 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.

## Block Diagram



## Pin Description

**SERIAL CLOCK (SCL):** The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

**SERIAL DATA (SDA):** The SDA pin is bidirectional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open collector devices.

**DEVICE/PAGE ADDRESSES (A<sub>2</sub>, A<sub>1</sub>, A<sub>0</sub>):** The A<sub>2</sub>, A<sub>1</sub> and A<sub>0</sub> pins are device address inputs that are hard wired to GND or V<sub>CC</sub>. If left not connected for hardware compatibility with AT24C16. When the pins are hardwired, as many as eight 32K/64K devices may be addressed on a single bus system (device addressing is discussed in detail under the

Device Addressing section). When the pins are not hardwired, the default A<sub>2</sub>, A<sub>1</sub>, and A<sub>0</sub> are zero.

**WRITE PROTECT (WP):** The write protect input, when tied to GND, allows normal write operations. When WP is tied high to V<sub>CC</sub>, all write operations to the upper quadrant (8/16K bits) of memory are inhibited. If left unconnected, WP is internally pulled down to GND.

## Memory Organization

**AT24C32/64, 32K/64K SERIAL EEPROM:** The 32K/64K is internally organized as 256 pages of 32 bytes each. Random word addressing requires a 12/13 bit data word address.

# AT24C32/64

## Capacitance<sup>(1)</sup>

measured over recommended operating range from  $T_A = 25^\circ\text{C}$ ,  $f = 1.0\text{ MHz}$ ,  $V_{CC} = +1.8\text{V}$ .

Symbol	Test Condition	Max	Units	Conditions
	Input/Output Capacitance (SDA)	8	pF	$V_{IO} = 0\text{V}$
	Input Capacitance ( $A_0, A_1, A_2, \text{SCL}$ )	6	pF	$V_{IN} = 0\text{V}$

1. This parameter is characterized and is not 100% tested.

## Characteristics

measured over recommended operating range from:  $T_{AI} = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{CC} = +1.8\text{V}$  to  $+5.5\text{V}$ ,  $T_{AC} = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $V_{IO} = +1.8\text{V}$  to  $+5.5\text{V}$  (unless otherwise noted).

Symbol	Parameter	Test Condition	Min	Typ	Max	Units
	Supply Voltage		1.8		5.5	V
	Supply Voltage		2.5		5.5	V
	Supply Voltage		2.7		5.5	V
	Supply Voltage		4.5		5.5	V
	Supply Current $V_{CC} = 5.0\text{V}$	READ at 100 kHz		0.4	1.0	mA
	Supply Current $V_{CC} = 5.0\text{V}$	WRITE at 100 kHz		2.0	3.0	mA
	Standby Current (1.8V option)	$V_{CC} = 1.8\text{V}$	$V_{IN} = V_{CC}$ or $V_{SS}$		0.1	$\mu\text{A}$
		$V_{CC} = 5.5\text{V}$			2.0	
	Standby Current (2.5V option)	$V_{CC} = 2.5\text{V}$	$V_{IN} = V_{CC}$ or $V_{SS}$		0.5	$\mu\text{A}$
		$V_{CC} = 5.5\text{V}$			2.0	
	Standby Current (2.7V option)	$V_{CC} = 2.7\text{V}$	$V_{IN} = V_{CC}$ or $V_{SS}$		0.5	$\mu\text{A}$
		$V_{CC} = 5.5\text{V}$			2.0	
	Standby Current (5V option)	$V_{CC} = 4.5 - 5.5\text{V}$	$V_{IN} = V_{CC}$ or $V_{SS}$	20	35	$\mu\text{A}$
	Input Leakage Current	$V_{IN} = V_{CC}$ or $V_{SS}$		0.10	3.0	$\mu\text{A}$
	Output Leakage Current	$V_{OUT} = V_{CC}$ or $V_{SS}$		0.05	3.0	$\mu\text{A}$
	Input Low Level <sup>(1)</sup>		-0.6		$V_{CC} \times 0.3$	V
	Input High Level <sup>(1)</sup>		$V_{CC} \times 0.7$		$V_{CC} + 0.5$	V
	Output Low Level $V_{CC} = 3.0\text{V}$	$I_{OL} = 2.1\text{ mA}$			0.4	V
	Output Low Level $V_{CC} = 1.8\text{V}$	$I_{OL} = 0.15\text{ mA}$			0.2	V

1.  $V_{IL}$  min and  $V_{IH}$  max are reference only and are not tested.





## Characteristics

able over recommended operating range from  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ ,  $V_{CC} = +1.8\text{V}$  to  $+5.5\text{V}$ ,  $CL = 1$  TTL Gate and 100 pF (unless otherwise noted).

Parameter	1.8-volt		2.7-, 2.5-volt		5.0-volt		Units
	Min	Max	Min	Max	Min	Max	
Clock Frequency, SCL		100		100		400	kHz
Clock Pulse Width Low	4.7		4.7		1.2		$\mu\text{s}$
Clock Pulse Width High	4.0		4.0		0.6		$\mu\text{s}$
Noise Suppression Time <sup>(1)</sup>		100		100		50	ns
Clock Low to Data Out Valid	0.1	4.5	0.1	4.5	0.1	0.9	$\mu\text{s}$
Time the bus must be free before a new transmission can start <sup>(1)</sup>	4.7		4.7		1.2		$\mu\text{s}$
Start Hold Time	4.0		4.0		0.6		$\mu\text{s}$
Start Set-up Time	4.7		4.7		0.6		$\mu\text{s}$
Data In Hold Time	0		0		0		$\mu\text{s}$
Data In Set-up Time	200		200		100		ns
Inputs Rise Time <sup>(1)</sup>		1.0		1.0		0.3	$\mu\text{s}$
Inputs Fall Time <sup>(1)</sup>		300		300		300	ns
Stop Set-up Time	4.7		4.7		0.6		$\mu\text{s}$
Data Out Hold Time	100		100		50		ns
Write Cycle Time		20		10		10	ms
Endurance <sup>(1)</sup> 5.0V, 25°C, Page Mode	1M		1M		1M		Write Cycles

1. This parameter is characterized and is not 100% tested.

## Sequence Operation

**START and DATA TRANSITIONS:** The SDA pin is normally pulled high with an external device. Data on the SDA pin changes only during SCL low time periods (refer to Validity timing diagram). Data changes during SCL low periods will indicate a start or stop condition as shown below.

**START CONDITION:** A high-to-low transition of SDA with SCL high is a start condition which must precede any other condition (refer to Start and Stop Definition timing diagram).

**STOP CONDITION:** A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the master will place the EEPROM in a standby power mode (refer to Start and Stop Definition timing diagram).

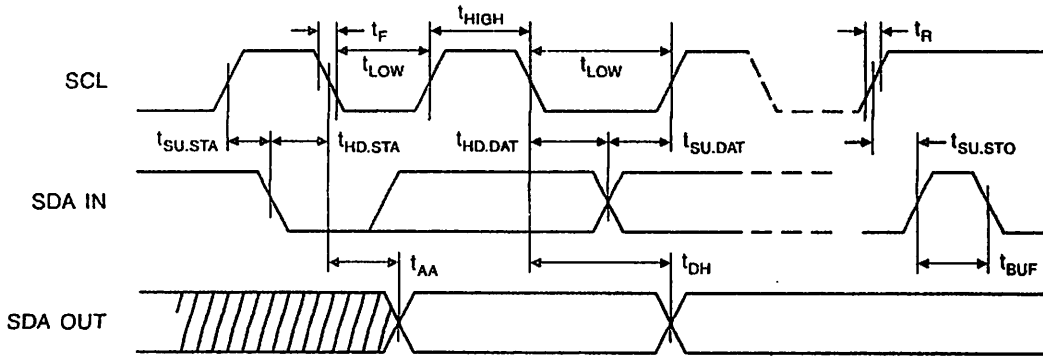
**ACKNOWLEDGE:** All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero during the ninth clock cycle to acknowledge that it has received each word.

**STANDBY MODE:** The AT24C32/64 features a low power standby mode which is enabled: a) upon power-up and b) after the receipt of the STOP bit and the completion of any internal operations.

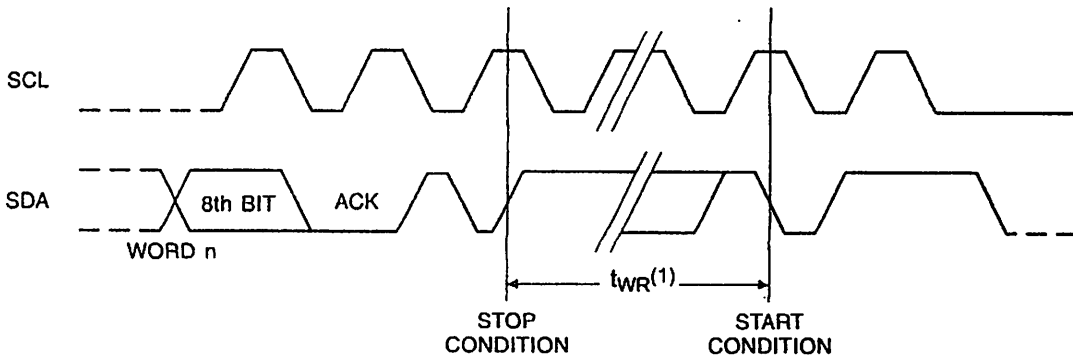
**MEMORY RESET:** After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

(a) Clock up to 9 cycles, (b) look for SDA high in each cycle while SCL is high and then (c) create a start condition as SDA is high.

Timing  
SCL: Serial Clock, SDA: Serial Data I/O



Write Cycle Timing  
SCL: Serial Clock, SDA: Serial Data I/O

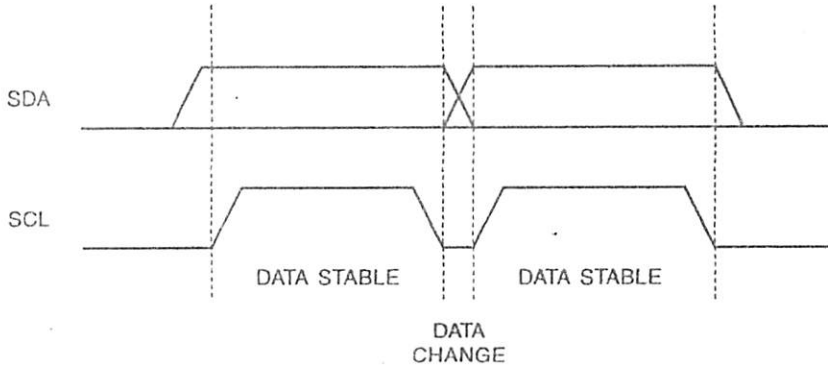


1. The write cycle time  $t_{WR}$  is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

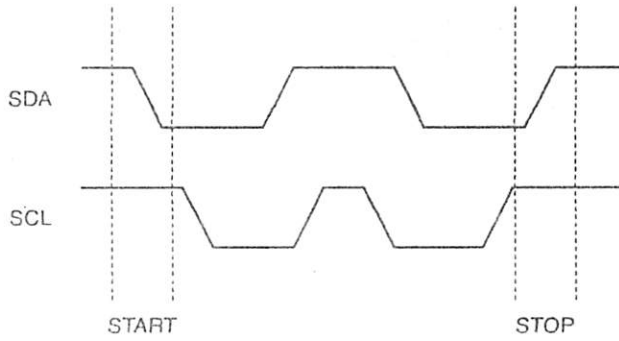




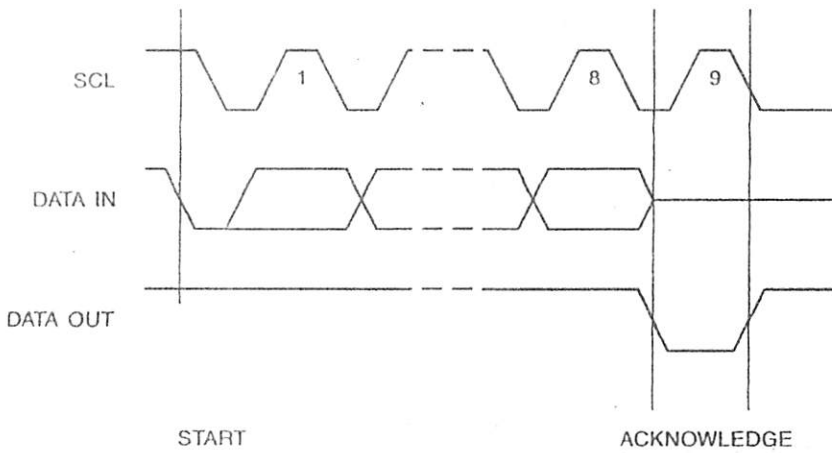
### Data Validity



### Start and Stop Definition



### Output Acknowledge





## Device Addressing

The AT24C32/64 EEPROM requires an 8-bit device address following a start condition to enable the chip for a read or write operation (refer to Figure 1). The device address consists of a mandatory one, zero sequence for the four most significant bits as shown. This is common to all wire EEPROM devices.

The AT24C32/64 uses the three device address bits A2, A1, A0 to allow as many as eight devices on the same bus. These must compare to their corresponding hardwired input pins. The A2, A1, and A0 pins use an internal proprietary circuit that biases them to a logic low condition if the pins are allowed to float.

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

On a compare of the device address, the EEPROM will output a zero. If a compare is not made, the device will return to standby state.

**NOISE PROTECTION:** Special internal circuitry placed on the I<sup>2</sup>C (SDA and SCL) pins prevent small noise spikes from disabling the device. A low-V<sub>CC</sub> detector (5-volt option) allows the device to prevent data corruption in a noisy environment.

**DATA SECURITY:** The AT24C32/64 has a hardware data protection scheme that allows the user to write protect the first quadrant (8/16K bits) of memory when the WP pin is driven low.

## Write Operations

**BYTE WRITE:** A write operation requires two 8-bit data addresses following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero and the addressing sequence, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM completes an internally-timed write cycle, t<sub>WR</sub>, to the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (refer to Figure 2).

**PAGE WRITE:** The 32K/64K EEPROM is capable of 32-page writes.

A page write is initiated the same way as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to 31 more data words. The EEPROM will respond with a zero after each data word is received. The microcontroller must terminate the page write sequence with a stop condition (refer to Figure 3).

The data word address lower 5 bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than 32 data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten.

**ACKNOWLEDGE POLLING:** Once the internally-timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero, allowing the read or write sequence to continue.

## Read Operations

Read operations are initiated the same way as write operations with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

**CURRENT ADDRESS READ:** The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "roll over" during a read is from the last byte of the last memory page, to the first byte of the first page. The address "roll over" during a write is from the last byte of the current page to the first byte of the same page.

Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero but does generate a following stop condition (refer to Figure 4).

**RANDOM READ:** A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 5).

**SEQUENTIAL READ:** Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledge. As long as the EEPROM receives an





knowledge, it will continue to increment the data word address and serially clock out sequential data words. When memory address limit is reached, the data word address will "roll over" and the sequential read will con-

tinue. The sequential read operation is terminated when the microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 6).

Figure 1. Device Address

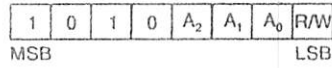


Figure 2. Byte Write

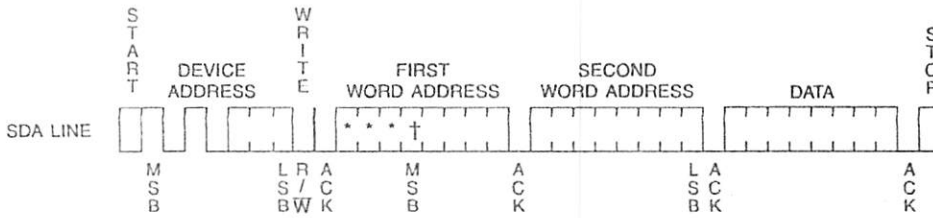
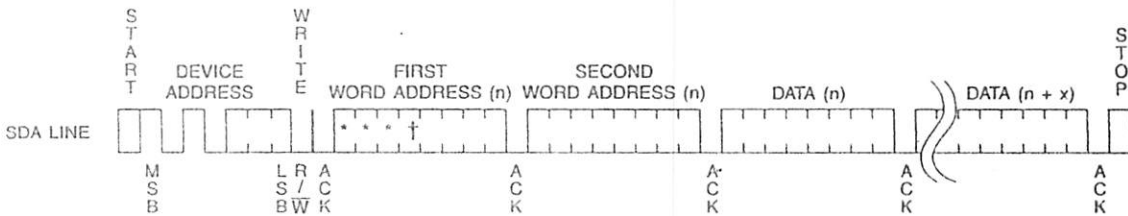


Figure 3. Page Write



Notes: 1. \* = DON'T CARE bits

2. † = DON'T CARE bits for the 32K

Figure 4. Current Address Read

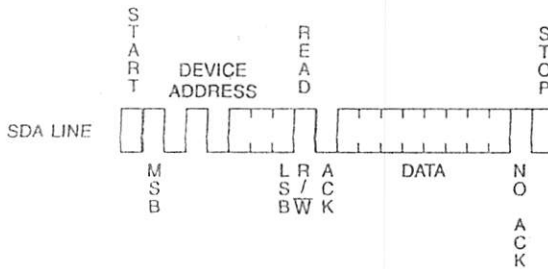
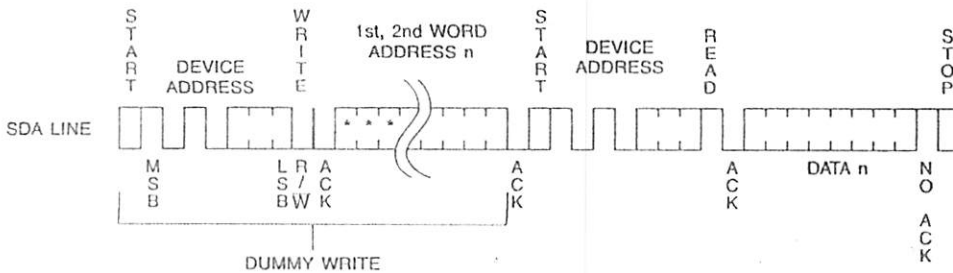
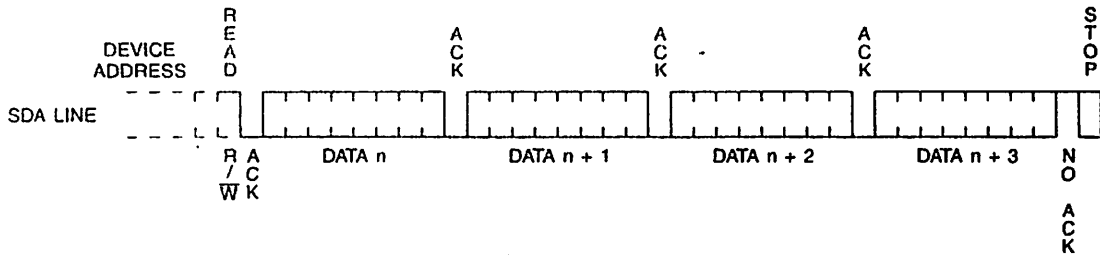


Figure 5. Random Read



Notes: 1. \* = DON'T CARE bits

Figure 6. Sequential Read





## 24C32 Ordering Information

$t_{WR}$ (max) (ms)	$I_{CC}$ (max) ( $\mu$ A)	$I_{SB}$ (max) ( $\mu$ A)	$f_{MAX}$ (kHz)	Ordering Code	Package	Operation Range
10	3000	35	400	AT24C32-10PC	8P3	Commercial (0°C to 70°C)
				AT24C32N-10SC	8S1	
				AT24C32W-10SC	8S2	
				AT24C32-10TC	8T	
				AT24C32-10SC	14S	
	3000	35	400	AT24C32-10PI	8P3	Industrial (-40°C to 85°C)
				AT24C32N-10SI	8S1	
				AT24C32W-10SI	8S2	
				AT24C32-10TI	8T	
				AT24C32-10SI	14S	
10	1500	0.5	100	AT24C32-10PC-2.7	8P3	Commercial (0°C to 70°C)
				AT24C32N-10SC-2.7	8S1	
				AT24C32W-10SC-2.7	8S2	
				AT24C32-10TC-2.7	8T	
				AT24C32-10SC-2.7	14S	
	1500	0.5	100	AT24C32-10PI-2.7	8P3	Industrial (-40°C to 85°C)
				AT24C32N-10SI-2.7	8S1	
				AT24C32W-10SI-2.7	8S2	
				AT24C32-10TI-2.7	8T	
				AT24C32-10SI-2.7	14S	

Package Type	
3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
2	8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline (EIAJ SOIC)
	8-Lead, 0.170" Wide, Plastic Gull Wing Small Outline (TSSOP)
5	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (SOIC)
Options	
ank	Standard Operation (4.5V to 5.5V)
7	Low Voltage (2.7V to 5.5V)
5	Low Voltage (2.5V to 5.5V)
3	Low Voltage (1.8V to 5.5V)

# AT24C32/64

# AT24C32/64

## 4C32 Ordering Information (Continued)

(max) ns	I <sub>CC</sub> (max) ( $\mu$ A)	I <sub>SB</sub> (max) ( $\mu$ A)	f <sub>MAX</sub> (kHz)	Ordering Code	Package	Operation Range
10	1000	0.5	100	AT24C32-10PC-2.5	8P3	Commercial (0°C to 70°C)
				AT24C32N-10SC-2.5	8S1	
				AT24C32W-10SC-2.5	8S2	
				AT24C32-10TC-2.5	8T	
				AT24C32-10SC-2.5	14S	
	1000	0.5	100	AT24C32-10PI-2.5	8P3	Industrial (-40°C to 85°C)
				AT24C32N-10SI-2.5	8S1	
				AT24C32W-10SI-2.5	8S2	
				AT24C32-10TI-2.5	8T	
				AT24C32-10SI-2.5	14S	
10	800	0.1	100	AT24C32-10PC-1.8	8P3	Commercial (0°C to 70°C)
				AT24C32N-10SC-1.8	8S1	
				AT24C32W-10SC-1.8	8S2	
				AT24C32-10TC-1.8	8T	
				AT24C32-10SC-1.8	14S	
	800	0.1	100	AT24C32-10PI-1.8	8P3	Industrial (-40°C to 85°C)
				AT24C32N-10SI-1.8	8S1	
				AT24C32W-10SI-1.8	8S2	
				AT24C32-10TI-1.8	8T	
				AT24C32-10SI-1.8	14S	

### Package Type

8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline (EIAJ SOIC)
8-Lead, 0.170" Wide, Plastic Gull Wing Small Outline (TSSOP)
14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (SOIC)

### Options

Standard Operation (4.5V to 5.5V)
Low Voltage (2.7V to 5.5V)
Low Voltage (2.5V to 5.5V)
Low Voltage (1.8V to 5.5V)





## 24C64 Ordering Information

$t_{VR}$ (max) (ms)	$I_{CC}$ (max) ( $\mu$ A)	$I_{SB}$ (max) ( $\mu$ A)	$f_{MAX}$ (kHz)	Ordering Code	Package	Operation Range
10	3000	35	400	AT24C64-10PC AT24C64N-10SC AT24C64W-10SC AT24C64-10TC AT24C64-10SC	8P3 8S1 8S2 8T 14S	Commercial (0°C to 70°C)
	3000	35	400	AT24C64-10PI AT24C64N-10SI AT24C64W-10SI AT24C64-10TI AT24C64-10SI	8P3 8S1 8S2 8T 14S	Industrial (-40°C to 85°C)
10	1500	0.5	100	AT24C64-10PC-2.7 AT24C64N-10SC-2.7 AT24C64W-10SC-2.7 AT24C64-10TC-2.7 AT24C64-10SC-2.7	8P3 8S1 8S2 8T 14S	Commercial (0°C to 70°C)
	1500	0.5	100	AT24C64-10PI-2.7 AT24C64N-10SI-2.7 AT24C64W-10SI-2.7 AT24C64-10TI-2.7 AT24C64-10SI-2.7	8P3 8S1 8S2 8T 14S	Industrial (-40°C to 85°C)

### Package Type

3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
2	8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline (EIAJ SOIC)
	8-Lead, 0.170" Wide, Plastic Gull Wing Small Outline (TSSOP)
S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (SOIC)

### Options

ank	Standard Operation (4.5V to 5.5V)
7	Low Voltage (2.7V to 5.5V)
5	Low Voltage (2.5V to 5.5V)
8	Low Voltage (1.8V to 5.5V)

# AT24C32/64

## AT24C64 Ordering Information (Continued)

$t_{WR}$ (max) (ms)	$I_{CC}$ (max) ( $\mu$ A)	$I_{SB}$ (max) ( $\mu$ A)	$f_{MAX}$ (kHz)	Ordering Code	Package	Operation Range
10	1000	0.5	100	AT24C64-10PC-2.5 AT24C64N-10SC-2.5 AT24C64W-10SC-2.5 AT24C64-10TC-2.5 AT24C64-10SC-2.5	8P3 8S1 8S2 8T 14S	Commercial (0°C to 70°C)
	1000	0.5	100	AT24C64-10PI-2.5 AT24C64N-10SI-2.5 AT24C64W-10SI-2.5 AT24C64-10TI-2.5 AT24C64-10SI-2.5	8P3 8S1 8S2 8T 14S	Industrial (-40°C to 85°C)
10	800	0.1	100	AT24C64-10PC-1.8 AT24C64N-10SC-1.8 AT24C64W-10SC-1.8 AT24C64-10TC-1.8 AT24C64-10SC-1.8	8P3 8S1 8S2 8T 14S	Commercial (0°C to 70°C)
	800	0.1	100	AT24C64-10PI-1.8 AT24C64N-10SI-1.8 AT24C64W-10SI-1.8 AT24C64-10TI-1.8 AT24C64-10SI-1.8	8P3 8S1 8S2 8T 14S	Industrial (-40°C to 85°C)

### Package Type

P3	8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
S1	8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)
S2	8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline (EIAJ SOIC)
T	8-Lead, 0.170" Wide, Plastic Gull Wing Small Outline (TSSOP)
4S	14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (SOIC)

### Options

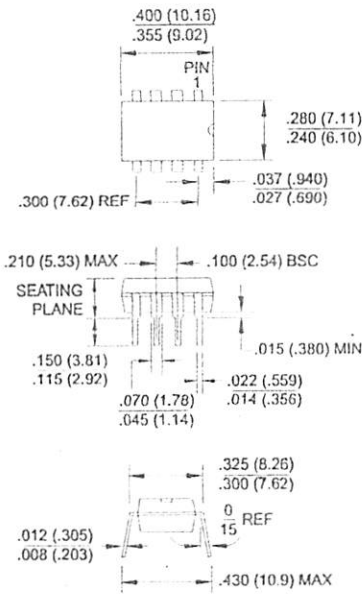
Bank	Standard Operation (4.5V to 5.5V)
2.7	Low Voltage (2.7V to 5.5V)
2.5	Low Voltage (2.5V to 5.5V)
1.8	Low Voltage (1.8V to 5.5V)



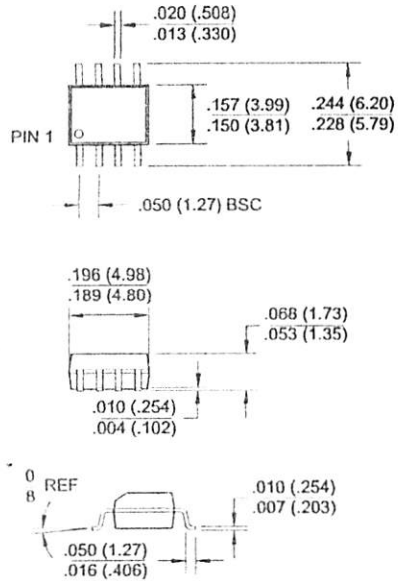


## Packaging Information

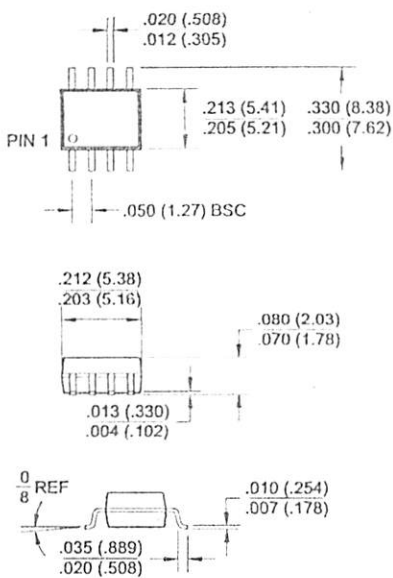
**8P3, 8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)**  
 Dimensions in Inches and (Millimeters)  
 JEDEC STANDARD MS-001 BA



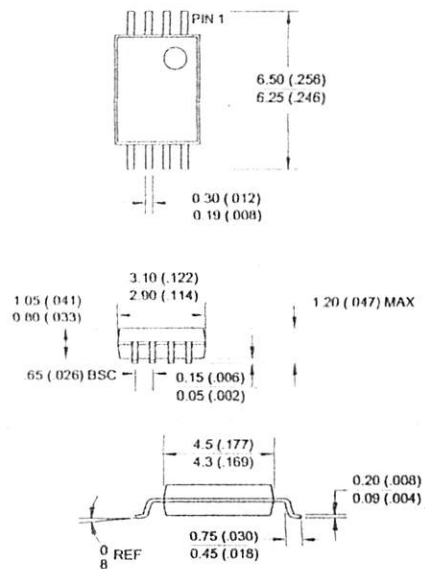
**8S1, 8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (JEDEC SOIC)**  
 Dimensions in Inches and (Millimeters)



**8S2, 8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline (EIAJ SOIC)**  
 Dimensions in Inches and (Millimeters)



**8T, 8-Lead, Plastic Thin Small Outline Package (TSSOP)**  
 Dimensions in Millimeters and (Inches)\*

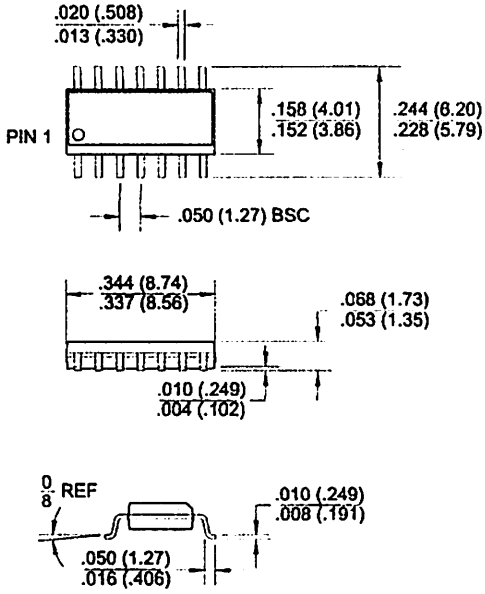


\*Controlling dimension: millimeters



**Packaging Information**

**AT24C32/64, 14-Lead, 0.150" Wide, Plastic Gull Wing Small Outline (SOIC)**  
 Dimensions in Inches and (Millimeters)



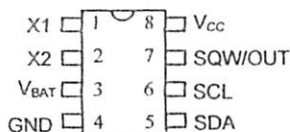
**FEATURES**

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode with oscillator running
- Optional industrial temperature range -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Recognized by Underwriters Laboratory

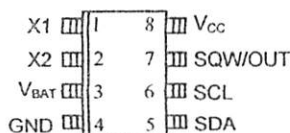
**ORDERING INFORMATION**

DS1307	8-Pin DIP
DS1307Z	8-Pin SOIC (150 mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

**PIN ASSIGNMENT**



DS1307 8-Pin DIP (300 mil)



DS1307Z 8-Pin SOIC (150 mil)

**PIN DESCRIPTION**

- V<sub>CC</sub> - Primary Power Supply
- X1, X2 - 32.768 kHz Crystal Connection
- V<sub>BAT</sub> - +3V Battery Input
- GND - Ground
- SDA - Serial Data
- SCL - Serial Clock
- SQW/OUT - Square wave/Output Driver

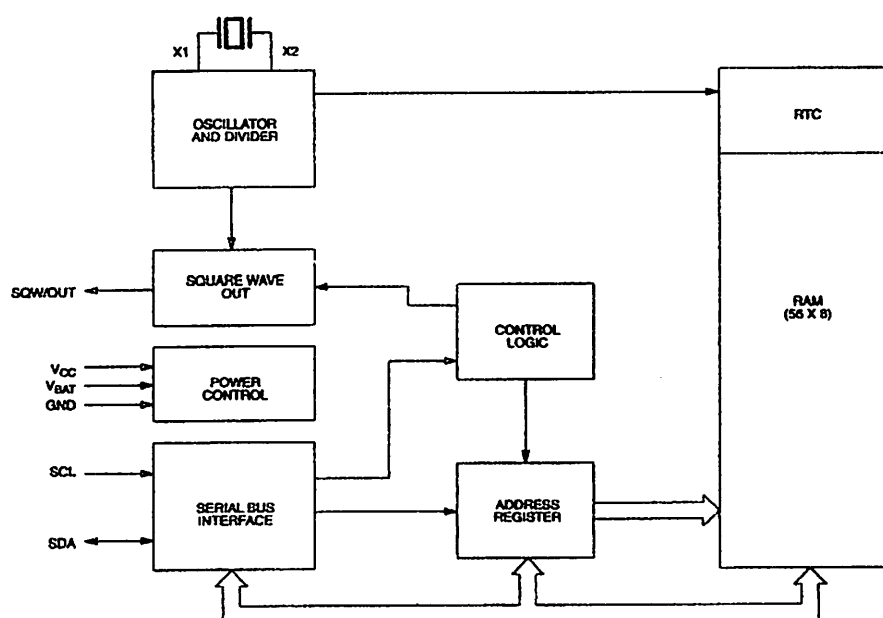
**DESCRIPTION**

The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit which detects power failures and automatically switches to the battery supply.

## OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When  $V_{CC}$  falls below  $1.25 \times V_{BAT}$  the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When  $V_{CC}$  falls below  $V_{BAT}$  the device switches into a low current battery backup mode. Upon power up, the device switches from battery to  $V_{CC}$  when  $V_{CC}$  is greater than  $V_{BAT} + 0.2V$  and recognizes inputs when  $V_{CC}$  is greater than  $1.25 \times V_{BAT}$ . The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock.

## DS1307 BLOCK DIAGRAM Figure 1



## SIGNAL DESCRIPTIONS

$V_{CC}$ , **GND** - DC power is provided to the device on these pins.  $V_{CC}$  is the +5 volt input. When 5 volts is applied within normal limits, the device is fully accessible and data can be written and read. When a 3-volt battery is connected to the device and  $V_{CC}$  is below  $1.25 \times V_{BAT}$ , reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As  $V_{CC}$  falls below  $V_{BAT}$  the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at  $V_{BAT}$ .

$V_{BAT}$  - Battery input for any standard 3-volt lithium cell or other energy source. Battery voltage must be held between 2.0 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as  $1.25 \times V_{BAT}$  nominal. A lithium battery with 48 mAhr or greater will back up the DS1307 for more than 10 years in the absence of power at 25 degrees C.

**SCL (Serial Clock Input)** - SCL is used to synchronize data movement on the serial interface.

**SDA (Serial Data Input/Output)** - SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

**SQW/OUT (Square Wave/ Output Driver)** - When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 kHz, 8 kHz, 32 kHz). The SQW/OUT pin is open drain which requires an external pullup resistor. SQW/OUT will operate with either Vcc or Vbat applied.

**X1, X2** - Connections for a standard 32.768 kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks." The DS1307 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

## RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

### DS1307 ADDRESS MAP Figure 2

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

## LOCK AND CALENDAR

Real time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary-Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

Please note that the initial power on state of all registers is not defined. Therefore it is important to enable the oscillator (CH bit=0) during initial configuration.

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

### DS1307 TIMEKEEPER REGISTERS Figure 3

		BIT 7							BIT 0		
00H	CH	10 SECONDS			SECONDS					00-59	
	X	10 MINUTES			MINUTES					00-59	
	X	12/24	10 HR A/P	10 HR	HOURS					01-12 00-23	
	X	X	X	X	X	DAY					1-7
	X	X	10 DATE		DATE					01-28/29 01-30 01-31	
	X	X	X	10 MONTH	MONTH					01-12	
	10 YEAR			YEAR					00-99		
07H	OUT	X	X	SQWE	X	X	RS1	RS0			

### CONTROL REGISTER

The DS1307 Control Register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	X	X	SQWE	X	X	RS1	RS0

**OUT (Output control):** This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE=0, the logic level on the SQW/OUT pin is 1 if OUT=1 and is 0 if OUT=0.

**SQWE (Square Wave Enable):** This bit, when set to a logic 1, will enable the oscillator output. The frequency of the square wave output depends upon the value of the RS0 and RS1 bits.

**RS (Rate Select):** These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

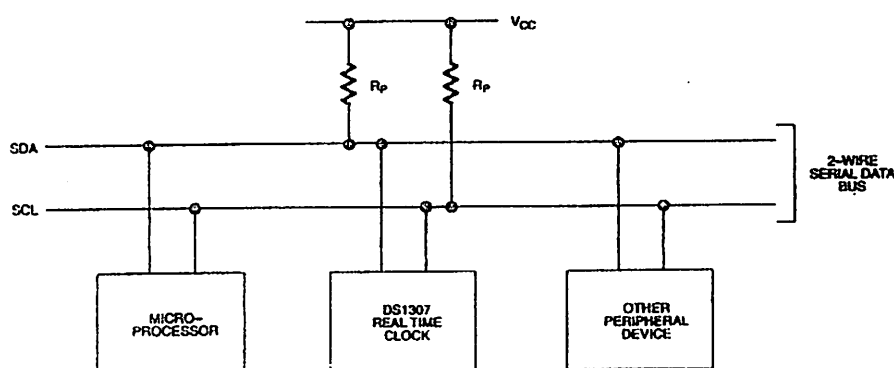
### SQUAREWAVE OUTPUT FREQUENCY Table 1

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

## 2-WIRE SERIAL DATA BUS

The DS1307 supports a bi-directional 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2-wire bus. A typical bus configuration using this 2-wire protocol is shown in Figure 4.

### TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



Figures 5, 6, and 7 detail how data is transferred on the 2-wire bus.

Data transfer may be initiated only when the bus is not busy.

During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

**Bus not busy:** Both data and clock lines remain HIGH.

**Start data transfer:** A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

**Stop data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

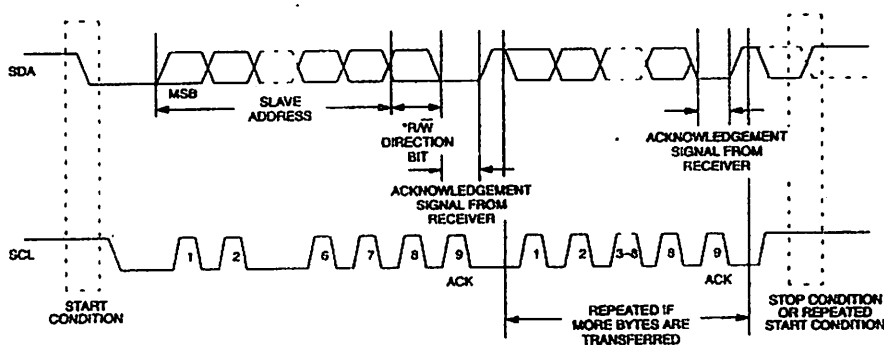
**Data valid:** The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the 2-wire bus specifications a regular mode (100 kHz clock rate) and a fast mode (400 kHz clock rate) are defined. The DS1307 operates in the regular mode (100 kHz) only.

**Acknowledge:** Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

## DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



Depending upon the state of the  $\overline{R/\overline{W}}$  bit, two types of data transfer are possible:

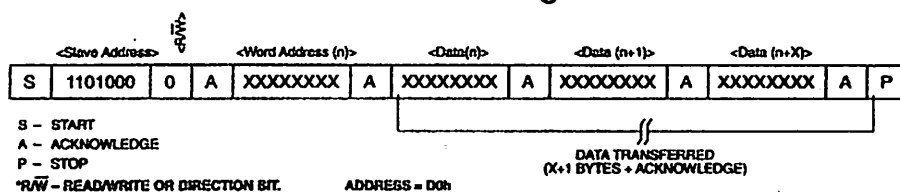
1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

DS1307 may operate in the following two modes:

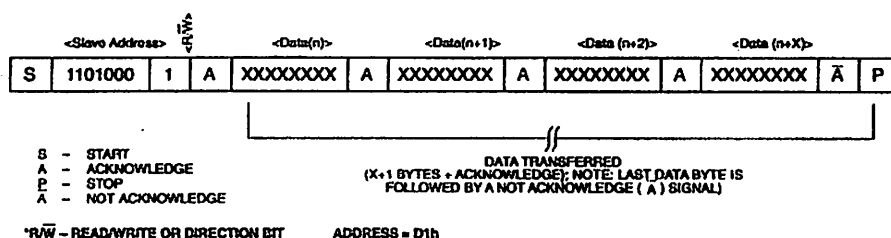
**Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and \*direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the \*direction bit ( $\overline{R/W}$ ) which, for a write, is a 0. After receiving and decoding the address byte the device outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307. This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

### DATA WRITE - SLAVE RECEIVER MODE Figure 6



**Slave transmitter mode (DS1307 read mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the \*direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7-bit DS1307 address, which is 1101000, followed by the \*direction bit ( $\overline{R/W}$ ) which, for a read, is a 1. After receiving and decoding the address byte the device inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a Not Acknowledge to end a read.

### DATA READ - SLAVE TRANSMITTER MODE Figure 7





**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature	0°C to 70°C (-40°C to 85°C for industrial)
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds DIP
	See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices

This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

**RECOMMENDED DC OPERATING CONDITIONS**

(0°C to 70°C or -40°C to +85°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V <sub>CC</sub>	4.5	5.0	5.5	V	1
Logic 1	V <sub>IH</sub>	2.2		V <sub>CC</sub> +0.3	V	1
Logic 0	V <sub>IL</sub>	-0.3		+0.8	V	1
BAT Battery Voltage	V <sub>BAT</sub>	2.0		3.5	V	1

**DC ELECTRICAL CHARACTERISTICS**(0°C to 70°C or -40°C to +85°C; V<sub>CC</sub>=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I <sub>LI</sub>			1	μA	10
IO Leakage	I <sub>LO</sub>			1	μA	11
Logic 0 Output	V <sub>OL</sub>			0.4	V	2
Active Supply Current	I <sub>CCA</sub>			1.5	mA	9
Standby Current	I <sub>CCS</sub>			200	μA	3
Battery Current (OSC ON); QW/OUT OFF	I <sub>BAT1</sub>		300	500	nA	4
Battery Current (OSC ON); QW/OUT ON (32 kHz)	I <sub>BAT2</sub>		480	800	nA	4

**ELECTRICAL CHARACTERISTICS**(0°C to 70°C or -40°C to +85°C;  $V_{CC}=4.5V$  to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Clock Frequency	$f_{SCL}$	0		100	kHz	
Free Time Between a STOP and START Condition	$t_{BUF}$	4.7			$\mu s$	
Time (Repeated) START Condition	$t_{HD:STA}$	4.0			$\mu s$	5
Period of SCL Clock	$t_{LOW}$	4.7			$\mu s$	
High Period of SCL Clock	$t_{HIGH}$	4.0			$\mu s$	
Setup Time for a Repeated START Condition	$t_{SU:STA}$	4.7			$\mu s$	
Hold Time	$t_{HD:DAT}$	0			$\mu s$	6, 7
Set-up Time	$t_{SU:DAT}$	250			ns	
Time of Both SDA and SCL Signals	$t_R$			1000	ns	
Fall Time of Both SDA and SCL Signals	$t_F$			300	ns	
Setup Time for STOP Condition	$t_{SU:STO}$	4.7			$\mu s$	
Capacitive Load for each Bus Line	$C_B$			400	pF	8
Capacitance	$C_{I/O}$		10		pF	
Maximum Specified Load Capacitance			12.5		pF	

**NOTES:**

All voltages are referenced to ground.

Logic zero voltages are specified at a sink current of 5 mA at  $V_{CC}=4.5V$ ,  $V_{OL}=GND$  for capacitive loads.

Tests specified with  $V_{CC}=5.0V$  and SDA, SCL=5.0V.

$V_{CC}=0V$ ,  $V_{BAT}=3V$ .

After this period, the first clock pulse is generated.

A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the  $t_{HMIN}$  of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.

The maximum  $t_{HD:DAT}$  has only to be met if the device does not stretch the LOW period ( $t_{LOW}$ ) of the SCL signal.

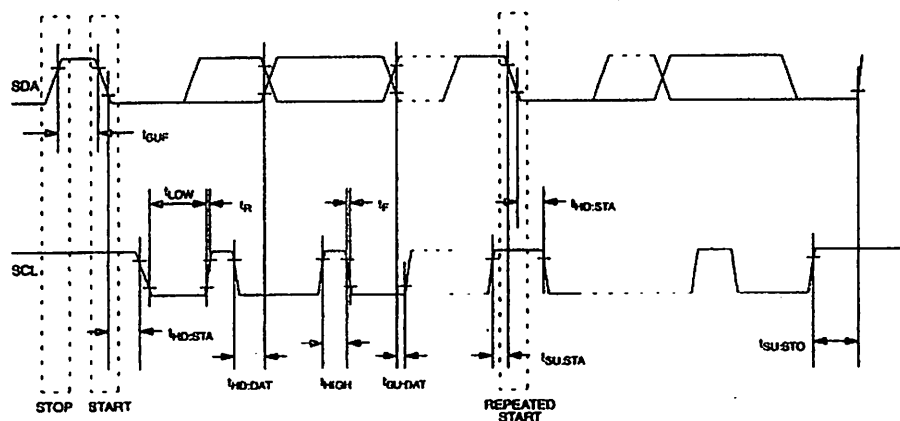
$C_B$  - total capacitance of one bus line in pF.

$f_{CCA}$  - SCL clocking at max frequency = 100 kHz.

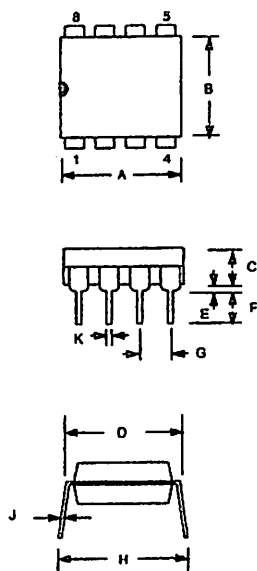
CI, only.

SDA and SQW/OUT

**TIMING DIAGRAM Figure 8**



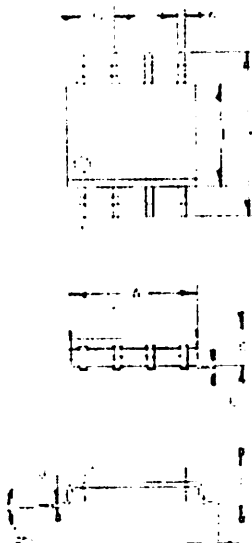
**DS1307 64 X 8 SERIAL REAL TIME CLOCK  
8-PIN DIP MECHANICAL DIMENSIONS**



PKG DIM	8-PIN	
	MIN	MAX
A IN.	0.360	0.400
MM	9.14	10.16
B IN.	0.240	0.260
MM	6.10	6.60
C IN.	0.120	0.140
MM	3.05	3.56
D IN.	0.300	0.325
MM	7.62	8.26
E IN.	0.015	0.040
MM	0.38	1.02
F IN.	0.120	0.140
MM	3.04	3.56
G IN.	0.090	0.110
MM	2.29	2.79
H IN.	0.320	0.370
MM	8.13	9.40
J IN.	0.008	0.012
MM	0.20	0.30
K IN.	0.015	0.021
MM	0.38	0.53

# DS1307Z 64 X 8 SERIAL REAL TIME CLOCK

## 8-PIN SOIC (150-MIL) MECHANICAL DIMENSIONS



PKG	8-PIN (150 MIL)	
	MIN	MAX
A IN.	0.188	0.196
MM	4.78	4.98
B IN.	0.150	0.158
MM	3.81	4.01
C IN.	0.048	0.062
MM	1.22	1.57
E IN.	0.004	0.010
MM	0.10	0.25
F IN.	0.053	0.069
MM	1.35	1.75
G IN.	0.050 BSC	
MM	1.27 BSC	
H IN.	0.230	0.244
MM	5.84	6.20
J IN.	0.007	0.011
MM	0.18	0.28
K IN.	0.012	0.020
MM	0.30	0.51
L IN.	0.016	0.050
MM	0.41	1.27
phi	0°	8°

56-G2008-001

# MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

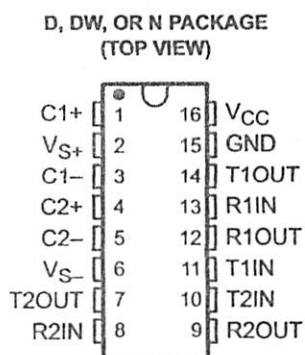
SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

- Operates With Single 5-V Power Supply
- LinBiCMOS™ Process Technology
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Designed to be Interchangeable With Maxim MAX232
- Applications
  - TIA/EIA-232-F
  - Battery-Powered Systems
  - Terminals
  - Modems
  - Computers
- ESD Protection Exceeds 2000 V Per MIL-STD-883, Method 3015
- Package Options Include Plastic Small-Outline (D, DW) Packages and Standard Plastic (N) DIPs

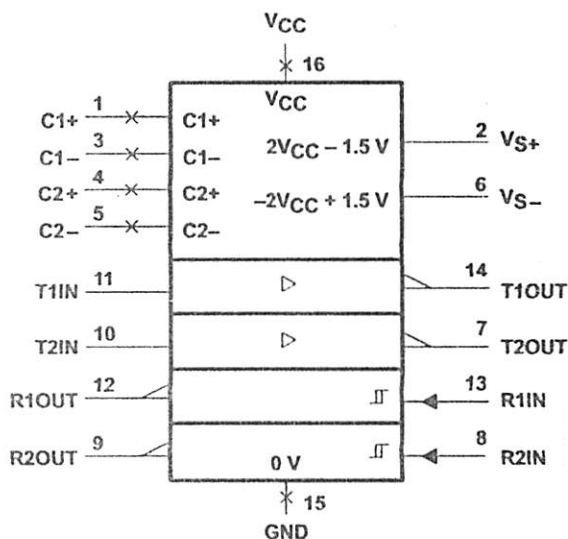
## description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

The MAX232 is characterized for operation from 0°C to 70°C. The MAX232I is characterized for operation from -40°C to 85°C.



## logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

## AVAILABLE OPTIONS

T <sub>A</sub>	PACKAGED DEVICES		
	SMALL OUTLINE (D)	SMALL OUTLINE (DW)	PLASTIC DIP (N)
0°C to 70°C	MAX232D†	MAX232DW†	MAX232N
-40°C to 85°C	MAX232ID†	MAX232IDW†	MAX232IN

† This device is available taped and reeled by adding an R to the part number (i.e., MAX232DR).



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

LinASIC and LinBiCMOS are trademarks of Texas Instruments Incorporated.

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

**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 1998, Texas Instruments Incorporated

# MAX232, MAX232I

## DUAL EIA-232 DRIVER/RECEIVER

SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, $V_{CC}$ (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	-0.3 V to -15 V
Input voltage range, $V_I$ : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	$\pm 30$ V
Output voltage range, $V_O$ : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, $\theta_{JA}$ (see Note 2): D package	113°C/W
DW package	105°C/W
N package	78°C/W
Storage temperature range, $T_{stg}$	-65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

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

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JESD 51, except for through-hole packages, which use a trace length of zero.

### recommended operating conditions

	MIN	NOM	MAX	UNIT
Supply voltage, $V_{CC}$	4.5	5	5.5	V
High-level input voltage, $V_{IH}$ (T1IN, T2IN)	2			V
Low-level input voltage, $V_{IL}$ (T1IN, T2IN)			0.8	V
Receiver input voltage, R1IN, R2IN			$\pm 30$	V
Operating free-air temperature, $T_A$	MAX232	0	70	°C
	MAX232I	-40	85	



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

# MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

**electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (unless otherwise noted)**

PARAMETER		TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND	5	7		V
		R1OUT, R2OUT	I <sub>OH</sub> = -1 mA	3.5			
V <sub>OL</sub>	Low-level output voltage‡	T1OUT, T2OUT	R <sub>L</sub> = 3 kΩ to GND		-7	-5	V
		R1OUT, R2OUT	I <sub>OL</sub> = 3.2 mA			0.4	
V <sub>IT+</sub>	Receiver positive-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C		1.7	2.4	V
V <sub>IT-</sub>	Receiver negative-going input threshold voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V, T <sub>A</sub> = 25°C	0.8	1.2		V
V <sub>hys</sub>	Input hysteresis voltage	R1IN, R2IN	V <sub>CC</sub> = 5 V	0.2	0.5	1	V
r <sub>i</sub>	Receiver input resistance	R1IN, R2IN	V <sub>CC</sub> = 5, T <sub>A</sub> = 25°C	3	5	7	kΩ
r <sub>o</sub>	Output resistance	T1OUT, T2OUT	V <sub>S+</sub> = V <sub>S-</sub> = 0, V <sub>O</sub> = ±2 V	300			Ω
I <sub>OS</sub> §	Short-circuit output current	T1OUT, T2OUT	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 0		±10		mA
I <sub>IS</sub>	Short-circuit input current	T1IN, T2IN	V <sub>I</sub> = 0			200	μA
I <sub>CC</sub>	Supply current		V <sub>CC</sub> = 5.5 V, T <sub>A</sub> = 25°C, All outputs open,		8	10	mA

† All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

### switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C

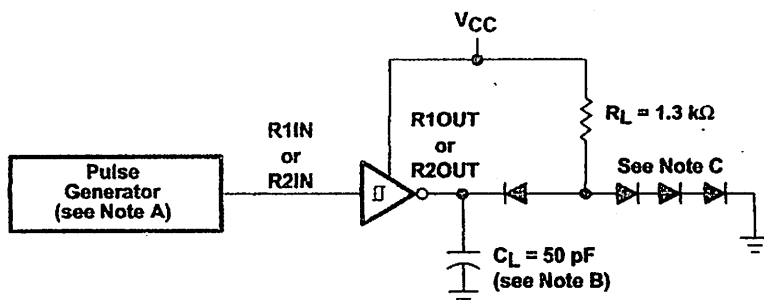
PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
t <sub>PLH(R)</sub>	Receiver propagation delay time, low- to high-level output	See Figure 1		500		ns
t <sub>PHL(R)</sub>	Receiver propagation delay time, high- to low-level output	See Figure 1		500		ns
SR	Driver slew rate	R <sub>L</sub> = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(tr)	Driver transition region slew rate	See Figure 3		3		V/μs



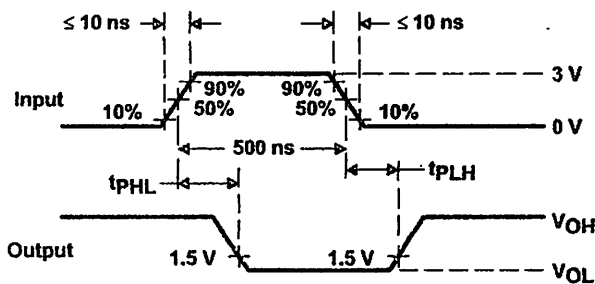
**MAX232, MAX232I**  
**DUAL EIA-232 DRIVER/RECEIVER**

LS047G - FEBRUARY 1989 - REVISED AUGUST 1998

**PARAMETER MEASUREMENT INFORMATION**



**TEST CIRCUIT**



**WAVEFORMS**

- NOTES: A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
 B.  $C_L$  includes probe and jig capacitance.  
 C. All diodes are 1N3064 or equivalent.

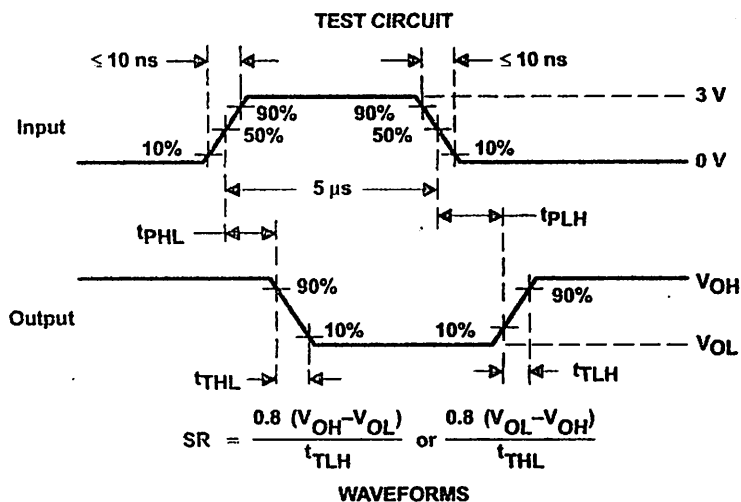
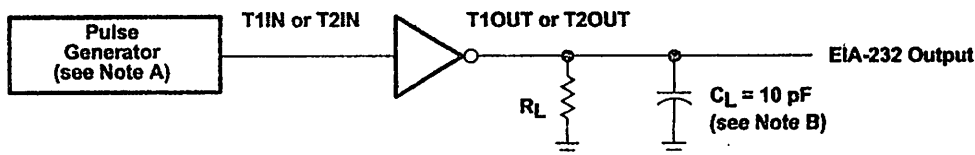
**Figure 1. Receiver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements**



# MAX232, MAX232I DUAL EIA-232 DRIVER/RECEIVER

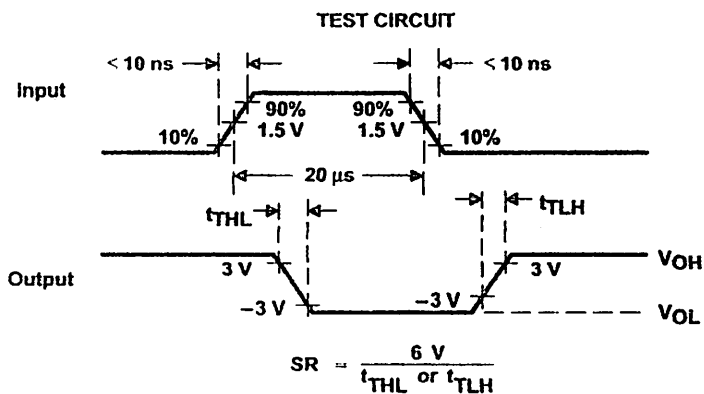
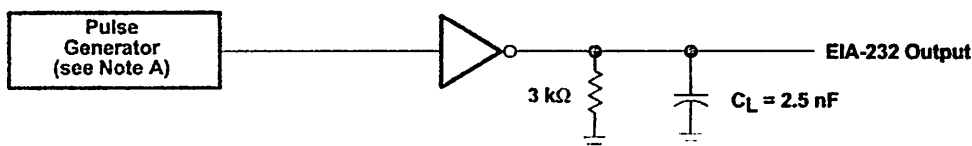
SLLS047G - FEBRUARY 1989 - REVISED AUGUST 1998

## PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics:  $Z_0 = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
B.  $C_L$  includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for  $t_{pHL}$  and  $t_{pLH}$  Measurements (5- $\mu$ s input)



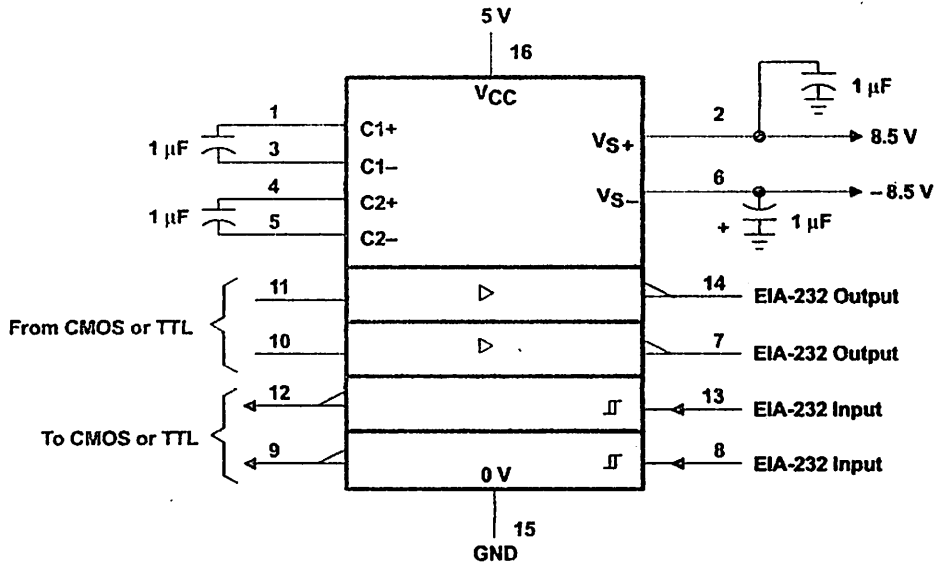
NOTE A: The pulse generator has the following characteristics:  $Z_0 = 50 \Omega$ , duty cycle  $\leq 50\%$ .

Figure 3. Test Circuit and Waveforms for  $t_{THL}$  and  $t_{TLH}$  Measurements (20- $\mu$ s input)

**MAX232, MAX232I**  
**DUAL EIA-232 DRIVER/RECEIVER**

LS047G - FEBRUARY 1989 - REVISED AUGUST 1998

**APPLICATION INFORMATION**



**Figure 4. Typical Operating Circuit**



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

## IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

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

**CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.**

*In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.*

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty or endorsement thereof.

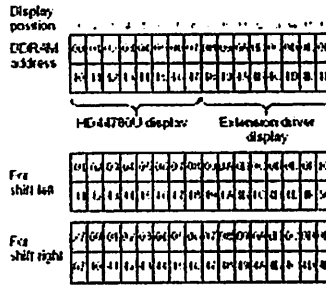
## M1632 MODULE LCD 16 X 2 BARIS (M1632)

### Deskripsi:

M1632 adalah merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGRAM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory) dan DDRAM (Display Data Random Access Memory).

### DDRAM

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Contoh, untuk karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.



Gambar 1  
DDRAM M1632 (diambil dari data sheet HD44780)

### CGRAM

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

### CGROM

CGROM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubahnya. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif.

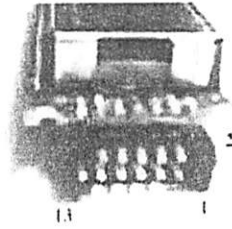
Pada gambar 2, tampak terlihat pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Pada saat HD44780 akan menampilkan data 41H yang tersimpan pada DDRAM, maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter 'A'.

	V000	G001	V010	D011	G100	G101	V110	D111	T000	T001	V110	D111	T100	T101	V110	D111	T110	T111	
V000	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G001	1			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V010	2			2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
D011	3			3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
G100	4			4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
G101	5			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
V110	6			6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
D111	7			7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
T000	8			8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
T001	9			9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
V110	10			10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
D111	11			11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
T100	12			12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
T101	13			13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
V110	14			14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
D111	15			15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
T110	16			16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
T111	17			17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17

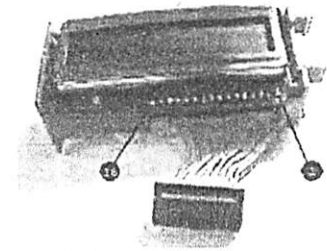
Gambar 2  
Hubungan antara CGROM dan DDRAM (diambil dari data sheet HD44780)

Pin Out

No	Nama Pin	Deskripsi
1	VCC	+5V
2	GND	0V
3	VEE	Tegangan Kontras LCD
4	RS	Register Select, 0 = Register Perintah, 1 = Register I
5	R/W	1 = Read, 0 = Write
6	E	Enable Clock LCD, logika 1 setiap kali pengiriman : pembacaan data
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	Anoda (Kabel coklat untuk LCD Hitachi)	Tegangan positif backlight
16	Katoda (Kabel merah untuk LCD Hitachi)	Tegangan negatif backlight



Gambar 3  
Pin Out M1632 LCD Hitachi



Gambar 4  
Pin Out LCD M1632 Standard

#### Register

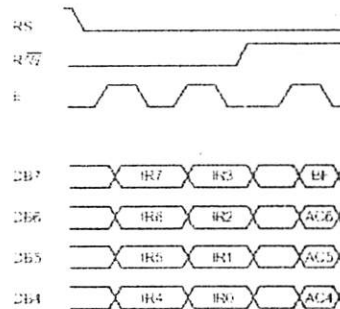
HD44780, mempunyai dua buah Register yang aksesnya diatur dengan menggunakan kaki. Pada saat RS berlogika 0, maka register yang diakses adalah Register Perintah dan pada saat RS berlogika 1, maka register yang diakses adalah Register Data.

#### Register Perintah

Register ini adalah register di mana perintah-perintah dari mikrokontroler ke HD44780 pada proses penulisan data atau tempat status dari HD44780 dapat dibaca pada saat pembacaan data.

#### Penulisan Data ke Register Perintah

Penulisan data ke Register Perintah dilakukan dengan tujuan mengatur tampilan LCD, inisial, dan mengatur Address Counter maupun Address Data. Gambar 5 menunjukkan proses penulisan data register perintah dengan menggunakan mode 4 bit interface. Kondisi RS berlogika 0 menunjukkan akses data ke Register Perintah. RW berlogika 0 yang menunjukkan proses penulisan data akan dilakukan. Nibble tinggi (bit 7 sampai bit 4) terlebih dahulu dikirimkan dengan diawali pulsa logika 1 pada E Clock. Kemudian Nibble rendah (bit 3 sampai bit 0) dikirimkan dengan diawali pulsa logika 1 pada E Clock. Untuk mode 8 bit interface, proses penulisan dapat langsung dilakukan secara 8 bit (bit 7 ... bit 0) diawali sebuah pulsa logika 1 pada E Clock.



Gambar 5  
Timing diagram Penulisan Data ke Register Perintah Mode 4 bit Interface

Tabel 1  
Perintah-perintah M1632

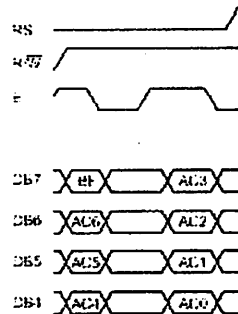
Perintah	D7	D6	D5	D4	D3	D2	D1	D0	Deskripsi
Hapus Display	0	0	0	0	0	0	0	1	Hapus Display dan DDRAM
Posisi Awal	0	0	0	0	0	0	1	X	Set Alamat DDRAM di 0
Set Mode	0	0	0	0	0	1	I/D	S	Atur arah pergeseran cursor dan displ

Display On/OFF	0	0	0	0	1	D	C	B	Atur display (D) On/OFF, cursor ON/OFF, Blinking (B)
Geser Cursor/Display	0	0	0	1	S/C	R/L	X	X	Geser Cursor atau display tanpa men alamat DDRAM
Set Fungsi	0	0	1	DL	N	F	X	X	Atur panjang data, jumlah baris : tampil, dan font karakter
Set CGRAM Alamat	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Data dapat dibaca atau ditulis set alamat diatur
Set DDRAM Alamat	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Data dapat dibaca atau ditulis set alamat diatur

X = diabaikan  
 I/D 1=Increment, 0=Decrement  
 S 0=Display tidak geser  
 S/C 1=Display Shift, 0=Geser Cursor  
 R/L 1=Geser Kiri, 0=Geser Kanan  
 DL 1=8 bit, 0=4bit  
 N 1=2 baris, 0=1 baris  
 F 1=5x10, 0=5x8  
 D 0=Display OFF, 1=Display ON  
 C 0=Cursor OFF, 1=Cursor ON  
 B 0=Blinking OFF, 1=Blinking ON

**Pembacaan Data dari Register Perintah**

Proses pembacaan data pada register perintah biasa digunakan untuk melihat status busy dari L atau membaca Address Counter. RS diatur pada logika 0 untuk akses ke Register Perintah, R/W diatur p logika 1 yang menunjukkan proses pembacaan data. 4 bit nibble tinggi dibaca dengan diawali pulsa log 1 pada E Clock dan kemudian 4 bit nibble rendah dibaca dengan diawali pulsa logika 1 pada E Clc Untuk Mode 8 bit interface, pembacaan 8 bit (nibble tinggi dan rendah) dilakukan sekaligus dengan diaw sebuah pulsa logika 1 pada E Clock.



**Gambar 6**  
**Timing Diagram Pembacaan Register Perintah Mode 4 bit Interface**

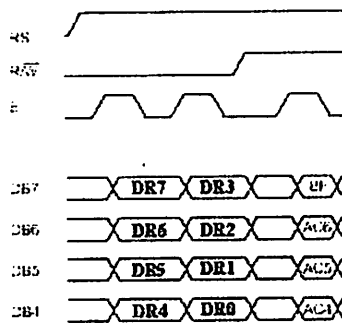
**Register Data**

Register ini adalah register di mana mikrokontroler dapat menuliskan atau membaca data ke : dari DDRAM. Penulisan data pada register ini akan menempatkan data tersebut ke DDRAM sesuai den; alamat yang telah diatur sebelumnya

**Penulisan Data ke Register Data**

Penulisan data pada Register Data dilakukan untuk mengirimkan data yang akan ditampilkan pa LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke Register Data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Data 4 bit nibble tinggi (bit 7 hingga

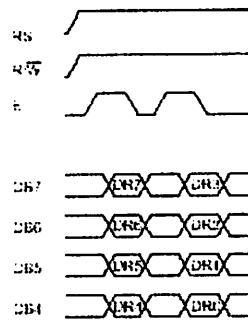
bit 4) dikirim dengan diawali pulsa logika 1 pada sinyal E Clock dan kemudian diikuti 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali pulsa logika 1 pada sinyal E Clock.



**Gambar 7**  
Timing Diagram Penulisan Data ke Register Data Mode 4 bit Interface

**Pembacaan Data dari Register Data**

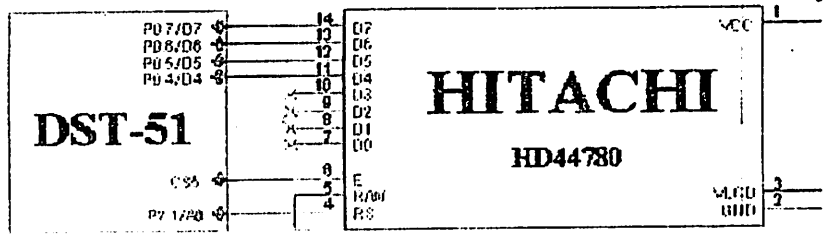
Pembacaan data dari Register Data dilakukan untuk membaca kembali data yang tampil p LCD. Proses dilakukan dengan mengatur RS pada logika 1 yang menunjukkan adanya akses ke Regi Data. Kondisi R/W diatur pada logika tinggi yang menunjukkan adanya proses pembacaan data. Data 4 nibble tinggi (bit 7 hingga bit 4) dibaca dengan diawali adanya pulsa logika 1 pada E Clock dan dilanjutkan dengan data 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali dengan pulsa logika 1 pada E Clock



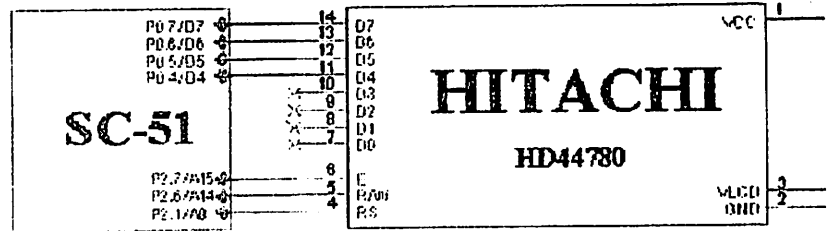
**Gambar 8**  
Timing Diagram Pembacaan Data dari Register Data Mode 4 bit Interface

Antar muka LCD dengan mikrokontroler





**Gambar 9**  
Antar muka dengan Modul DST-51



**Gambar 10**  
Antar Muka dengan Modul SC-51 atau AT8951

**Program**

Rutin-rutin Program untuk DST-51 yang diassembly dengan ALDS atau ASM51

Rutin-rutin Program untuk SC-51/AT8951 yang diassembly dengan ALDS atau ASM51

Rutin delay yang diassembly dengan ALDS atau ASM51

Datasheet HD44780