

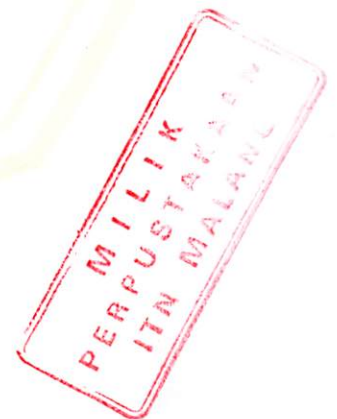
**INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA**



**PERENCANAAN DAN PEMBUATAN SISTEM PENGAMAN MOBIL
DENGAN SISTEM PASSWORD MENGGUNAKAN CARD READER
BERBASIS MIKROKONTROLLER AT89S51**

SKRIPSI

**Disusun Oleh :
R. ERWIEN INDIARTO
0117163**



MARET 2006

UNCLASSIFIED CONFIDENTIAL INFORMATION
EXCLUDED FROM AUTOMATIC DOWNGRADING
AND DECLASSIFICATION SCHEDULES
EXCEPT WHERE SHOWN OTHERWISE

UNCLASSIFIED INFORMATION EXCLUDED FROM AUTOMATIC
DOWNGRADING AND DECLASSIFICATION SCHEDULES
EXCEPT WHERE SHOWN OTHERWISE

CONFIDENTIAL

EXCLUDED FROM AUTOMATIC
DOWNGRADING AND DECLASSIFICATION
SCHEDULES

CONFIDENTIAL

LEMBAR PERSETUJUAN



**PERENCANAAN DAN PEMBUATAN SISTEM PENGAMAN MOBIL
DENGAN SISTEM PASSWORD MENGGUNAKAN CARD READER
BERBASIS MIKROKONTROLLER AT89S51**

SKRIPSI

Disusun dan Diajukan untuk Melengkapi Tugas Akhir dan Sebagai Syarat untuk
Memperoleh Gelar Sarjana Teknik

Disusun Oleh :

NAMA : R. ERWIEN INDIARTO

NIM : 0117163



**Mengetahui
Ketua Jurusan Elektro S-1**

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

**Diperiksa dan Disetujui
Dosen Pembimbing**

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

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



Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Elektro S-1
Konsentrasi Elektronika

BERITA ACARA UJIAN SKRIPSI

Nama : R Erwien Indiarto
NIM : 0117163
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perencanaan dan Pembuatan Sistem Pengaman Mobil dengan Sistem Password Menggunakan Card Reader Berbasis Mikrokontroler AT89S51.

Dipertahankan dihadapan majelis penguji skripsi jenjang Strata Satu (S1) Pada:

Hari : Rabu
Tanggal : 22 Maret 2006
Dengan Nilai : 85,5 (A) *hm*

Ketua Majelis Penguji



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

Sekretaris Majelis Penguji



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

Penguji I

(Ir. Usman Djuanda, MM)
NIP. Y.1018700143

Penguji II

(Ir. Mimin Mustikawati)
NIP.Y.1030000352



BERITA ACARA UJIAN SKRIPSI

Nama : R. Rizka Fadiono
NIM : 0117163
Jurusan : Teknik Elektro 2-1
Konvensi : Teknik Elektronika
Judul Skripsi : Perencanaan dan Pembuatan Sistem Pengaman Mobil dengan Sistem Password Menggunakan Card Reader Berbasis


Mikrokontroler AT89C51


Dipersembahkan dihadapan Majelis Pengaji Skripsi Jurusan (S1) Robotika

Hari : Rabu
Tanggal : 22 Maret 2000
Dengan Nilai : 85,3 (A)

Sekretaris Majelis Pengaji

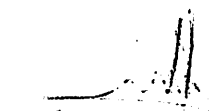
Ketua Majelis Pengaji



(Dr. H. Yudi Haryanto, ST)
NIP. 7.103200374


(Dr. Mochlis Azoni, MSiE)
NIP. 7.101810030

Pengaji II

Pengaji I


(Dr. Minia M. Alimawati)
NIP. 7.103000383


(Dr. Isman Djuanda, MSiE)
NIP. 7.101870013

Lembar Persembahan

Puja dan puji syukur selalu saya panjatkan kehadirat Allah swt. beserta junjungannya Nabi Muhammad SAW Saya sebagai penulis dalam upaya mencapai jenjang saat ini tidak terlepas dari dukungan dan bantuan dari berbagai pihak, karenanya melalui halaman ini saya ingin mengungkapkan rasa terimakasih sebagai rasa hormat saya.

- ◆ Kepada Ibu dan Bapak yang selalu memberikan bimbingan dan dukungan sejak saat saya terlahir di dunia ini dan tidak akan pernah bisa terbalaskan. Wenny, Erien terimakasih untuk semua dukungan dan perhatian, kamu semua adalah semangat saya untuk terus maju dan berkembang.
- ◆ Keluarga Besar R. Soeparno. Om Ndut-Tante Ning dan keluarga, Om Djoko- Tante Yayuk dan keluarga, Om Eddy-Tante Wiwik dan keluarga, Om Yon-tante Ludya dan Keluarga, Om Budi-Tante Tia, terimakasih untuk semua bimbingan, perhatian, dukungan, doa dalam berbagai hal yang luar biasa berarti bagi saya dan tidak akan pernah terganti selamanya.
- ◆ Ibu dan Bapak Trowulan, terimakasih untuk sambutan, dukungannya selama ini yang sangat berarti sebagai saya, dan memperkenankan putrinya untuk mendampingi saya.
- ◆ Afrita Endryanti terimakasih atas semua hal yang diberikan kepadaku, (marahmu, manjamu, tawamu, dan semua darimu) adalah penyemangatku untuk terus maju, aku selalu berusaha memenuhi keinginanmu, selalulah berusaha menjadi yang terbaik untuk keluarga dan hidupku.
- ◆ Farid & Icha terimakasih untuk semuanya, tetep awet sampai tua.
- ◆ Teman-teman Elka 4 yang selalu banyak bantu saya semoga yang belum lulus cepat lulus yang sudah cepat cari kerja., terimakasih banyak
- ◆ Si Kuda yang selalu setia menemani saya dalam suka dan duka tanpa mengeluh. I LUV my YAMAHA.
- ◆ Pak Buncis trims untuk semua bantuan,merawat si kuda sehingga tetap bertenaga, Enis, Andy, Aan, Pipin, trims saya senang sekali bisa berteman. Semua sahabat di Sumber sari yang pernah saya kenal.
- ◆ Semua sahabat dan berbagai pihak yang banyak membantu yang tidak bisa saya sebutkan satu persatu.

ABSTRAKSI

Perencanaan dan Pembuatan Sistem Pengaman Mobil dengan Sistem Password Menggunakan Card Reader Berbasis Mikrokontroler AT89S51.

(Erwien Indiarso, 0117163, Teknik Elektro S1/ Konsentrasi Teknik Elektronika)

(Pembimbing : Ir. Eko Nurcahyo)

Kata Kunci : AT89S51, Password, Card reader

Skripsi ini bertujuan merancang dan membuat sistem pengaman mobil dengan sistem password menggunakan card reader yang pengaturannya dilakukan oleh mikrokontroler AT89S51. Cara kerja alat ini adalah memberikan proteksi pada starter mobil dengan cara memutus hubungan arus pada ignition coil dan motor starter mobil yang interkoneksinya dengan *power supply* dikendalikan oleh pemrosesan *password* dan menggunakan *card reader* yang dilengkapi *card* dengan menggunakan IC EEPROM 26C64 yang berfungsi menyimpan *password* dengan format angka 4 digit yang dapat diubah sesuai keinginan operator, dan dilengkapi alarm pengaman apabila terjadi kesalahan sebanyak 2 kali dalam memasukkan password. Pengamanan yang dimaksud terletak pada kerahasiaan password, dan alarm yang berbunyi apabila terjadi kesalahan dalam menginputkan password, artinya apabila password salah maka langkah starter tidak dapat dilakukan, apabila hal ini diulangi maka alarm akan berbunyi. Sistem ini hanya dapat diakses apabila *card* terkoneksi pada *card reader*.

KATA PENGANTAR

Segala puja dan puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah yang telah dikaruniakan kepada penulis sehingga dapat menyelesaikan Skripsi dengan judul **Perencanaan dan Pembuatan Sistem Pengaman Mobil dengan Sistem Password Menggunakan Card Reader Berbasis Mikrokontroler AT89S51.**

Adapun maksud dan tujuan Penulisan Skripsi ini adalah untuk menyelesaikan studi di Jurusan Elektro S1 konsentrasi Elektronika , Fakultas Teknologi Industri ITN Malang.

Penulis merasa perlu mengucapkan terima kasih kepada Pihak-pihak yang telah membantu proses penyelesaian skripsi ini :

1. Bapak Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Yudi Limpraptono, MT, selaku Ketua Jurusan Elektro ITN Malang.
3. Bapak Ir. Eko Nurcahyo, Selaku Dosen Pembimbing.
4. Berbagai pihak yang berperan dalam proses penyelesaian skripsi ini yang tida mungkin penulis sebut satu-persatu

Semoga skripsi ini bermanfaat bagi semua pihak. Kritik dan saran konstruktif selalu penulis perhatikan sebagai upaya perbaikan kualitas tulisan-tuisan berikutnya.

Malang, 14 Maret 2006

Penulis

DAFTAR ISI

JUDUL.....	i
LEMBAR PERSETUJUAN	ii
BERITA ACARA UJIAN SKRIPSI.....	iii
LEMBAR PERSEMBAHAN	iv
ABSTRAKSI.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	3
1.5 Metodologi	4
1.6 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Mikrokontroler 89S51	6
2.1.2. Organisasi Memori.....	9
2.1.2.1 Memori Program.....	10
2.1.2.2 Memori Data.....	11
2.1.3 Pewaktuan CPU	14

2.1.4 Interupsi	15
2.1.4.1 Interrupt Enable	16
2.1.4.2 Prioritas Interupsi	17
2.1.5 <i>Special Funcion Register</i> (SFR)	19
2.1.6 Reset.....	21
2.1.7 <i>Timer/ Counter</i>	22
2.1.7.1 Menyetel <i>Timer</i>	27
2.2 <i>Liquid Crystal Display</i> (LCD).....	29
2.3 Papan Tombol	30
2.4 Relay.....	31
2.5 EEPROM 26C64.....	33
2.6 Optocoupler.....	34
2.6.1 Laser Dioda.....	34
2.6.2 Photo Dioda	34
BAB III PERENCANAAN DAN PEMBUATAN ALAT.....	36
3.1 Sistem Mikrokontroller	38
3.1.1 Mikrokontroller 89S51.....	38
3.1.1.1 Clock	39
3.1.1.2 Reset.....	40
3.1.1.3 Pembagian Port	41
3.1.1.4 Pengendali (Driver Motor).....	41
3.1.1.5 Liquid Crystal Display	43
3.1.1.6 Perancangan Unit Card Reader	44
3.1.1.7 Papan Tombol	48

3.1.1.8 Optocoupler	49
3.1.1.8.1 LD (Laser Diode)	50
3.1.1.8.2 <i>Photo</i> - Diode	50
3.1.1.9 Perangkat Lunak	51
BAB IV PENGUJIAN DAN ANALISIS	55
4.1 Umum.....	55
4.2 Pengujian Sistem Setiap Blok	55
4.2.1 Pengujian Sistem Mikrokontroller	55
4.2.2 Pengujian Rangkaian Tampilan	58
4.2.3 Pengujian Papan Tombol	59
4.2.4 Pengujian Optocoupler.....	62
4.3 Pengujian Sistem Keseluruhan.....	64
BAB V PENUTUP	68
5.1 Kesimpulan.....	68
5.2 Saran-saran	68
Daftar Pustaka.....	69
Lampiran.....	70

DAFTAR GAMBAR

Gambar 2.1 Blok Diagram Mikrokontroler 89S51	7
Gambar 2.2 Susunan Pena Mikrokontroler 89S51.....	8
Gambar 2.3 Struktur Memori Mikrokontroler 89S51.....	10
Gambar 2.4 Memori Program	11
Gambar 2.5 Memori Data Internal	12
Gambar 2.6 Bagian Bawah 128 Byte RAM Internal	12
Gambar 2.7 Bagian Atas 128 Bytes RAM Internal.....	13
Gambar 2.8 Ruang Special Function Register	14
Gambar 2.9 Penggunaan Osilator Internal	14
Gambar 2.10 Sistem Komtrol Interupsi	18
Gambar 2.11 Rangkaian Power-On Reset.....	21
Gambar 2.12 Timer/ Counter 1 Mode 0.....	25
Gambar 2.13 Timer/ Counter 1 dalam Mode 2	26
Gambar 2.14 Timer/ Counter 0 dalam Mode 3	27
Gambar 2.15 Koneksi Header LCD 16x2 dengan MCU	30
Gambar 2.16 Koneksi Keypad dengan MCU.....	31
Gambar 2.17 Simbol Relay	32
Gambar 2.18 EEPROM 26C64	33
Gambar 2.20 Simbol Photodiode	35
Gambar 3.1 Diagram Blok Sistem	36
Gambar 3.2 Susunan Pena Mikrokontroler.....	39
Gambar 3.3 Rangkaian Pewaktuan	40

Gambar 3.4 Rangkaian Power-On Reset.....	41
Gambar 3.5 Rangkaian Penggerak Relay.....	42
Gambar 3.6 Koneksi LCD 16x2 dengan MCU.....	44
Gambar 3.7 Antarmuka Serail EEPROM dengan MCU.....	44
Gambar 3.8 Diagram Waktu untuk Kondisi Start dan Stop pada Memori ..	45
Gambar 3.9 Diagram waktu yang menunjukkan kapan perubahan data sah	46
Gambar 3.10 Diagram waktu yang menunjukkan <i>zero acknowledge</i>	47
Gambar 3.11 Koneksi Keypad dengan MCU.....	49
Gambar 3.12 Rangkaian Sensor Putaran.....	51
Gambar 4.1 Diagram Blok Pengujian Mikroontroller	57
Gambar 4.2 LED Display Pengujian Mikrokontroller	57
Gambar 4.3 Diagram Blok Pengujian Rangkaian Tampilan.....	59
Gambar 4.4 Pengujian Rangkaian Tampilan.....	59
Gambar 4.5 Pengujian Rangkaian Papan Tombol	60
Gambar 4.6 Rangkaian Pengujian Optocoupler	62
Gambar 4.7 Pengujian Optocoupler saat tidak terhalang	63
Gambar 4.8 Pengujian Optocoupler saat terhalang.....	63
Gambar 4.9 Alat Nampak Luar	67
Gambar 4.10 Rangkaian Elektronik Sistem	67
Gambar 4.11 Rangkaian Pemroses Elektronik.....	68

DAFTAR TABEL

Tabel 2.1 Alamat Layanan Rutin Interupsi	16
Tabel 2.2 Definisi Bit-bit Interupt Enable.....	16
Tabel 2.3 Keterangan Interupt Enable.....	17
Tabel 2.4 Definisi Bit-bit Interupt Priority.....	18
Tabel 2.5 Keterangan Interupt Priority	18
Tabel 2.6 Special Funcion Register.....	20
Tabel 2.7 Definisi Bit-bit pada Timer Control.....	23
Tabel 2.8 Keterangan Bit-bit Timer Control.....	23
Tabel 2.9 Definisi Bit-bit pada Timer Code.....	24
Tabel 2.10 Kombinasi M0 dan M1	24
Tabel 2.11 Fungsi Timer 0	28
Tabel 2.12 Fungsi Counter 0	28
Tabel 2.13 Fungsi Timer 1	28
Tabel 2.14 Fungsi Counter 1	29
Tabel 4.1 Hasil Pengujian Sistem Mikrokontroller.....	57
Tabel 4.2 Hasil Pengujian Papan Tombol.....	61
Tabel 4.3 Pengujian Keypad	61
Tabel 4.4 Hasil Pengujian Optocoupler	63
Tabel 4.5 Hasil Pengujian tegangan Input	65

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dewasa ini sangat pesat. Hal ini dibuktikan dengan banyaknya pemakaian dan pemanfaatan teknologi modern yang canggih diberbagai bidang. Pemanfaatan teknologi tersebut dimaksudkan untuk mencapai hasil yang memiliki efektifitas dan efisiensi yang semakin baik. Bidang elektronika adalah salah satu bidang teknologi yang mendominasi berbagai kemajuan yang ada dan memberikan andil yang besar dalam memenuhi kebutuhan hidup manusia yang selalu mengupayakan ketepatan, kecepatan dan kemudahan dalam menyelesaikan berbagai masalah. Alasan tersebut membuat teknologi elektronika saat ini banyak dipergunakan di berbagai bidang kehidupan manusia, meliputi bidang perkantoran, perindustrian, kedokteran, transportasi, hingga lingkup rumah tangga. Salah satu implementasi dalam mencapai hal tersebut adalah diciptakannya mikrokontroller yang saat ini banyak digunakan.

Berdasarkan alasan tersebut diatas, maka penulis sebagai mahasiswa jurusan Elektronika S1 mengimplementasikan ilmu pengetahuan yang telah dipelajari dengan merencanakan dan membuat instrumen elektronika yang aplikatif untuk keperluan menjalani skripsi. Instrumen yang penulis maksud berfungsi sebagai pengaman mobil dengan memberikan proteksi pada starter mobil yang dikontrol mikrokontroller AT89S51 sebagai pengidentitas *password* yang dianggap valid untuk menjalankan fungsi starter mobil dengan menggunakan *card reader* sebagai prosedur pengamanan, sehingga mobil tersebut hanya dapat dioperasikan oleh

pemilikinya saja dengan menggunakan kartu khusus dengan menggunakan IC 24C64 yang dimilikinya, selain kunci kontak yang dalam hal ini masih dipergunakan.

1.2 Rumusan Masalah

Sistem pengaman mobil yang ada pada saat ini pada umumnya hanya mengacu pada pengamanan disekitar pintu dan kemudi mobil saja tanpa adanya proteksi yang khusus pada salah satu sistem yang paling vital dari kendaraan tersebut. Salah satu sistem yang penulis maksud adalah starter. Karena tanpa adanya starter, tentunya mesin mobil tidak dapat dihidupkan. Atas dasar inilah penulis menyusun sebuah skripsi dengan judul **PERENCANAAN DAN PEMBUATAN SISTEM PENGAMAN MOBIL DENGAN SISTEM PASSWORD MENGGUNAKAN CARD READER BERBASIS MIKROKONTROLLER AT89S51.**

1.3 Tujuan

Tujuan dari perencanaan dan pembuatan alat ini adalah menambah salah satu alternatif lain dari sistem-sistem pengaman pada mobil yang sudah ada sebelumnya, sehingga dapat menciptakan suatu pengaman pada mobil agar jauh lebih aman dari tindak pencurian.

Pengamanan ini penulis terapkan dengan cara memutus hubungan arus pada *ignition coil* dan motor starter mobil yang interkoneksinya dengan *power supply* dikendalikan oleh pemrosesan *password* dan menggunakan *cardreader* yang pengaturannya dilakukan oleh mikrokontroller AT89S51.

1.4 Batasan Masalah

Pembahasan materi yang penulis kemukakan perlu dibatasi sehingga tidak melebar ke pokok bahasan lain. Adapun batasan masalah tersebut adalah :

- Sistem terbatas hanya pada jenis mobil berbahan bakar bensin.
- Sistem pengamanan diterapkan pada starter mobil dengan cara membuka dan menutup jalur pada ignition coil dan motor starter.
- Cara kerja Mikrokontroller AT89S51 dan rangkaian pendukungnya.
 - Rangkaian driver sebagai penggerak motor DC sebagai simulasi mesin hanya dibahas secara garis besar.
 - Rangkaian display dan keyboard.
 - Buzzer digunakan sebagai pengganti alarm
 - Motor DC sebagai simulasi mesin tidak dibahas
- *Password* hanya akan dibaca oleh mikrokontroller dalam format angka 4 digit.
- Menggunakan rangkaian *card reader* dengan menggunakan IC EEPROM 24C64 sebagai *card*.
- Kontak manual tetap digunakan.
- Tidak membahas power supply. Pada kondisi sebenarnya alat ini menggunakan power supply yang terpisah dari power supply utama dalam hal ini *accu* sehingga sistem dapat bekerja dalam kondisi tegangan yang stabil.

- Tidak membahas hal-hal yang mendukung pengamanan, seperti pengabelan.

1.5 Metodologi

Penulisan tugas akhir ini menggunakan beberapa metode :

- Studi lapangan** : melihat dan mempelajari cara kerja sistem starter pada mobil berbahan bakar bensin.
- Studi literatur** : mencari bahan literatur, analisa dan pengumpulan data mengenai mikrokontroller AT89S51 dan IC-IC pendukungnya serta komponen yang menunjang.
- Perancangan** : melakukan perencanaan dan pembuatan perangkat keras dan perangkat lunak.
- Eksperimen** : melakukan serangkaian percobaan untuk perangkat keras dan perangkat lunak yang berhubungan dengan peralatan yang dibuat.
- Penyusunan** : menulis laporan skripsi berdasarkan analisa dan hasil eksperimen pembuatan alat yang telah diuji kelayakannya sesuai dengan tujuan yang diinginkan.

1.6 Sistematika Penulisan

Sistematika penulisan tugas ini dibagi dalam beberapa bab yaitu :

BAB I : PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, sistematika penulisan.

BAB II : LANDASAN TEORI

Bab ini membahas tentang arsitektur mikrokontroler AT89S51 serta komponen-komponen pendukung seperti EEPROM AT24C64, LCD, relay, keypad, dll.

BAB III : PERENCANAAN DAN PEMBUATAN ALAT

Bab ini meliputi pembahasan setiap bagian dari sistem, meliputi sistem mikrokontroler yang terdiri komponen-komponen pendukung sehingga sistem dapat bekerja dengan maksimal, secara perangkat keras dan perangkat lunak.

BAB IV : PENGUJIAN DAN ANALISIS

Membahas tentang pengujian dan analisis alat yang telah dibuat. Untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan.

BAB V : PENUTUP

Bab ini berisi tentang kesimpulan dan saran-saran yang diperlukan untuk penyempurnaan alat yang telah dibuat.

BAB II

LANDASAN TEORI

2.1 Mikrokontroler 89S51

Mikrokontroler 89S51 merupakan salah satu anggota keluarga MCS-51, yaitu suatu komponen produksi Intel yang berorientasi pada kontrol (*microcontroller*) dan oleh Intel diklasifikasikan dalam kelompok *embedded microcontroller*, yang berarti mikrokontroler yang dapat diprogram ulang (*reprogrammable*).

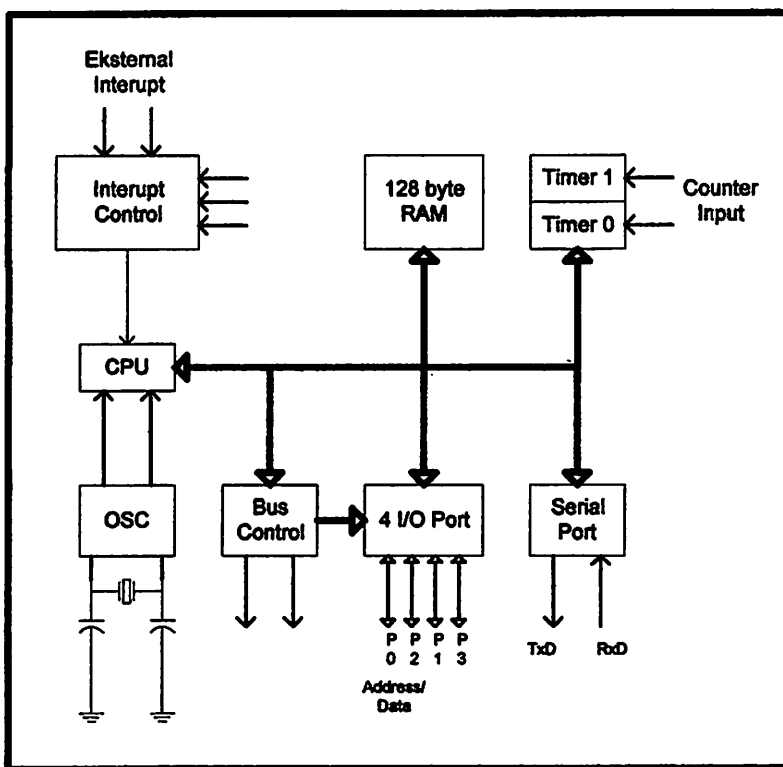
Mikrokontroler 89S51 memiliki *Electrical Programmable Read Only Memory* (EPROM) dengan kapasitas 4 Kbyte. tetapi instruksi-instruksi yang digunakan oleh mikrokontroler 89S51 juga bisa disimpan di memori luar, misalnya dengan menggunakan EPROM eksternal bila memori internal tidak memadai. Spesifikasi umum yang dimiliki oleh mikrokontroler 89S51 antara lain:

- ↳ Sebuah *Central Processing Unit* (CPU) dengan lebar data 8-bit.
- ↳ Osilator internal dan rangkaian pewaktu.
- ↳ 4 Kbyte Flash Memori.
- ↳ RAM internal 128 bytes (*on chip*).
- ↳ Empat buah *programmable port I/O*, masing-masing terdiri atas 8 buah jalur I/O.
- ↳ Dua buah pencacah (*counter*) dan pewaktu (*timer*) 16-bit.
- ↳ Lima buah jalur interupsi (2 buah interupsi eksternal dan 3 buah interupsi internal).
- ↳ Sebuah *port* serial dengan kontrol serial *full duplex Universal Asynchronous Receiver Transmitter* (UART).

↳ Kemampuan melaksanakan operasi penjumlahan, pengurangan, perkalian, pembagian, dan operasi logika (bit).

↳ Kecepatan pelaksanaan instruksi per siklus 1 mikrodetik pada frekuensi *clock* 12 MHz.

Dengan spesifikasi umum tersebut maka mikrokontroler 89S51 ini dapat digunakan untuk merealisasikan peralatan yang akan direncanakan. Blok diagram mikrokontroler 89S51 diperlihatkan dalam Gambar 2.1.



Gambar 2.1 Blok Diagram Mikrokontroler 89S51

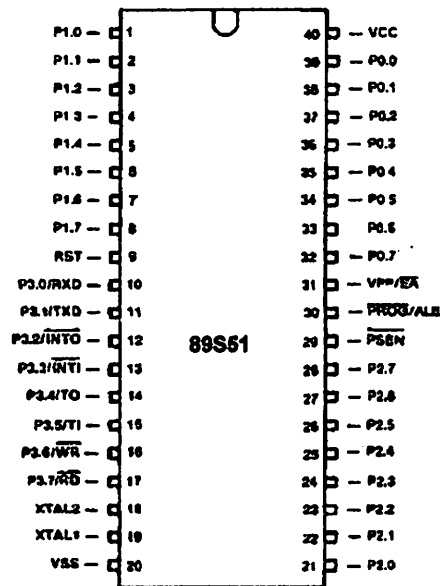
Sumber: Malik & Anistardi, 1997: 7

2.1.1 Susunan Pena-pena Mikrokontroler 89S51

Susunan pena-pena mikrokontroler 89S51 diperlihatkan dalam Gambar 2.2.

Penjelasan dari masing-masing pena adalah sebagai berikut:

- Pena 1 sampai 8 (*Port 1*) merupakan *port* paralel 8-bit dua arah (*bidirectional*).
- Pena 9 (*Reset*) adalah masukan *reset* (aktif tinggi).
- Pena 10 sampai 17 (*Port 3*) adalah *port* paralel 8-bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TxD (*Transmit Data*), RxD (*Receive Data*), $\overline{\text{INT}}_0$ (*Interrupt 0*), $\overline{\text{INT}}_1$ (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), $\overline{\text{WR}}$ (*Write*), dan $\overline{\text{RD}}$ (*Read*).



Gambar 2.2 Susunan Pena Mikrokontroler 89S51

Sumber: Intel, Agustus 1998

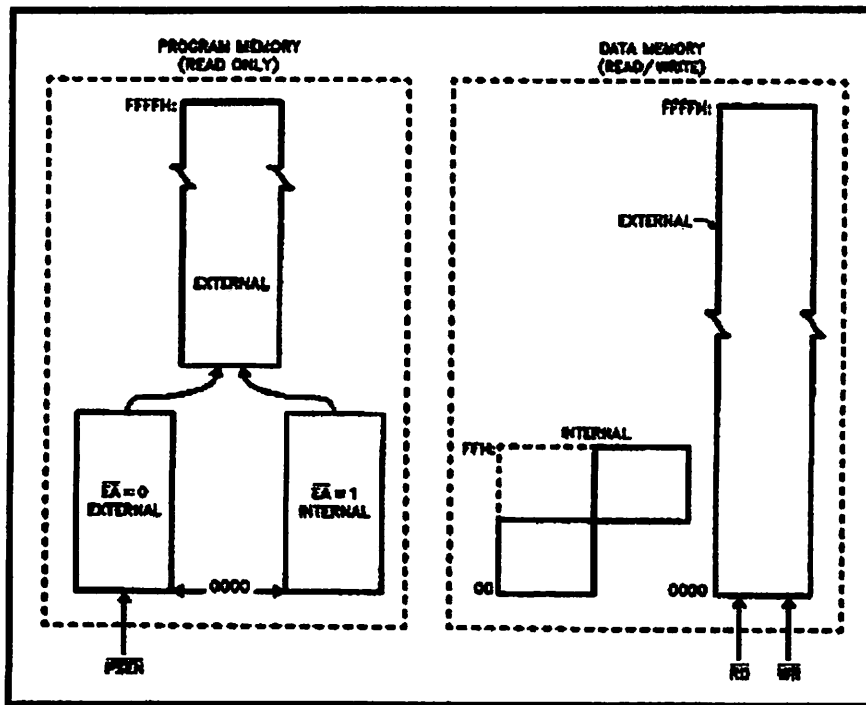
- Pena 18 (XTAL 1) adalah pena masukan ke rangkaian osilator internal.
- Pena 19 (XTAL 2) adalah pena keluaran ke rangkaian osilator internal.
- Pena 20 (*ground*) dihubungkan ke *ground*.

- Pena 21 sampai 28 (*Port 2*) adalah *port* paralel 2 (P2) dengan lebar 8-bit dua arah (*bidirectional*). *Port 2* ini mengirimkan *byte* alamat bila dilakukan pengaksesan memori eksternal.
- Pena 29 adalah pena $\overline{\text{PSEN}}$ (*program store enable*) yang berfungsi untuk mengambil instruksi atau data yang berada di jangkauan alamat memori ROM.
- Pena 30 adalah pena ALE (*Address Latch Enable*) yang digunakan untuk memisahkan data dan alamat yang termultipleks di *port 0* serta menahan alamat A0-A7.
- Pena 31 adalah pena $\overline{\text{EA}}$ (*External Access Enable*). Bila pena ini diberi logika tinggi (H), mikrokontroller akan melaksanakan instruksi dari ROM/EPROM internal. Bila diberi logika rendah (L), mikrokontroler akan melaksanakan seluruh instruksi dari memori program luar.
- Pena 32 sampai 39 (*Port 0*) merupakan *port* paralel 8-bit dua arah. Bila digunakan untuk mengakses memori luar, *port* ini akan memultipleks alamat memori dengan data.
- Pena 40 (Vcc) dihubungkan ke Vcc (+5 volt).

2.1.2 Organisasi Memori

Semua mikrokontroller dalam keluarga MCS-51 memiliki pembagian ruang alamat (*address space*) untuk program dan data seperti yang ditunjukkan dalam Gambar 2.3. Pemisahan memori program dan memori data membolehkan memori data untuk diakses oleh alamat 8-bit. Sekalipun demikian, alamat data memori 16 bit dapat dihasilkan melalui register DPTR (*data pointer register*).

Data pada memori program hanya dapat dibaca. Memori program sebesar 64 *kbytes* dapat dimasukkan ke dalam jangkauan alamat ROM eksternal.



Gambar 2.3 Struktur Memori Mikrokontroler 89S51

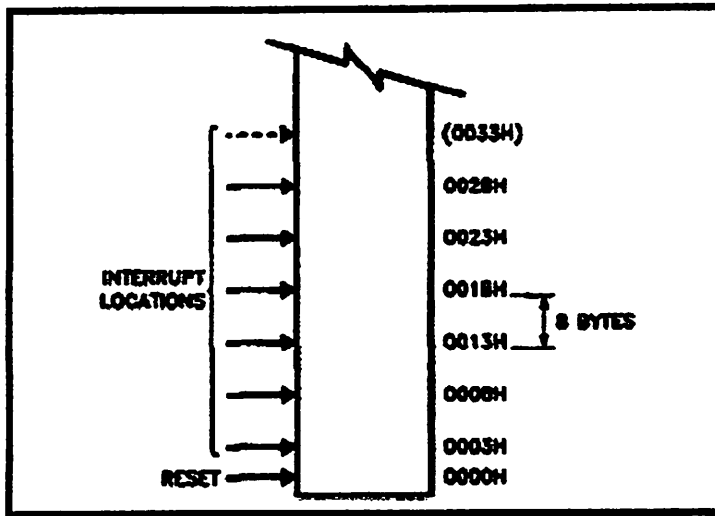
Sumber: Intel, Agustus 1998

Pena $\overline{\text{PSEN}}$ (*program store enable*) mengijinkan pembacaan memori program eksternal. Memori data terletak pada ruang alamat terpisah dari memori program. RAM (*random access memory*) dapat diletakkan dalam jangkauan alamat memori data eksternal. CPU mengaktifkan sinyal $\overline{\text{WR}}$ (*write*) atau $\overline{\text{RD}}$ (*read*) selama menghubungi memori data eksternal.

2.1.2.1 Memori Program

Gambar 2.4 memperlihatkan bagian bawah dari memori program. Setelah *reset*, CPU mengambil data yang dianggap sebagai instruksi pada lokasi 0000H. Setiap interupsi mempunyai alamat tetap dalam memori program. Interupsi menyebabkan PC

(*program counter*) menuju ke alamat tersebut yang berisikan subrutin penanganan interupsi.

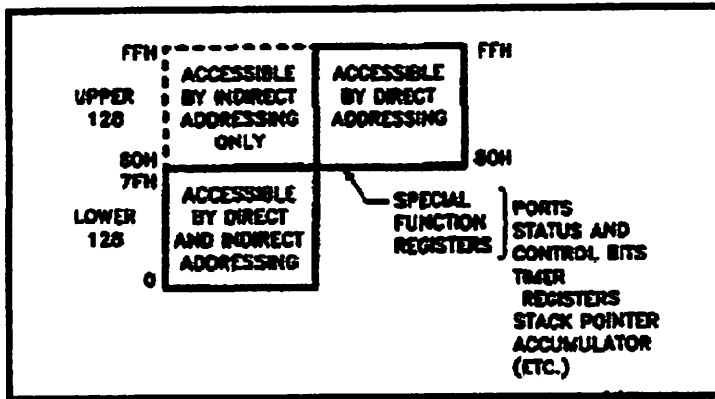


Gambar 2.4 Memori Program

Sumber: Intel, Agustus 1998

2.1.2.2 Memori Data

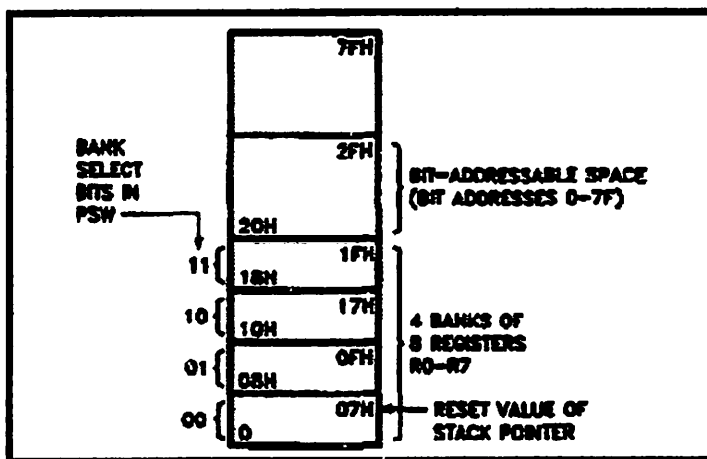
Memori data internal dipetakan seperti dalam Gambar 2.5. Ruang memorinya dibagi menjadi dua blok, yaitu sebagai *lower* sebanyak 128 bytes yang dapat digunakan sebagai memori baca tulis dan *upper* sebanyak 128 bytes yang berfungsi sebagai SFR (*special function register*).



Gambar 2.5 Memori Data Internal

Sumber: Intel, Agustus 1998

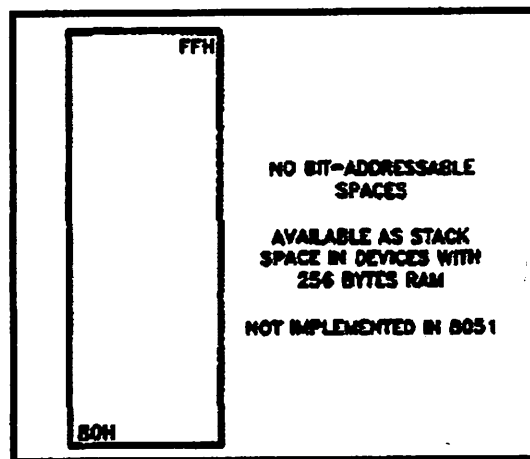
Bagian bawah dari 128 bytes RAM dipetakan seperti terlihat dalam Gambar 2.6. Tiga puluh dua byte paling bawah dikelompokkan dalam 4 bank (masing-masing terdiri atas 8 register, yaitu R0 sampai R7). Dua bit dalam PSW (*program status word*) digunakan untuk memilih register bank, yaitu RS0 dan RS1. Bank 0 dipilih secara *default* pada saat reset.



Gambar 2.6 Bagian Bawah 128 Byte RAM Internal

Sumber: Intel, Agustus 1998

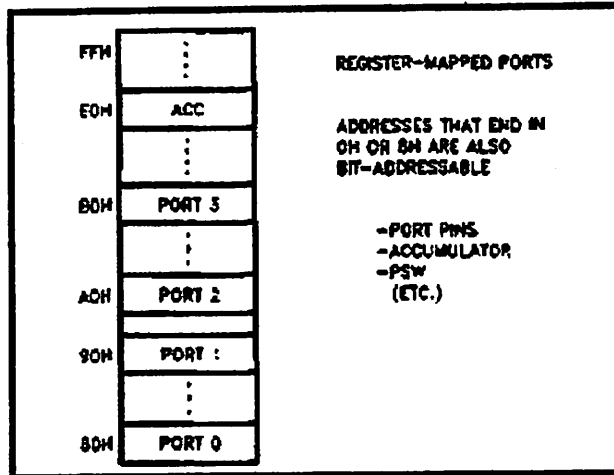
Enam belas *bytes* di atas bank register (alamat 20h sampai 2Fh) membentuk blok ruang memori yang dapat dialamat perbit sebanyak maksimal 128 bit. Semua *byte* pada bagian bawah 128 *byte* RAM internal dapat diakses oleh pengalamatan langsung dan tak langsung. Sedangkan bagian atas 128 *byte* RAM internal (diperlihatkan dalam Gambar 2.7) hanya dapat diakses dengan pengalamatan tak langsung.



Gambar 2.7 Bagian Atas 128 Bytes RAM Internal

Sumber: Intel, Agustus 1998

Gambar 2.8 menunjukkan ruang SFR (*SFR space*). SFR berisi penahan *port* (*port latch*), pewaktu (*timer*), pengontrol perifer, dan lain-lain. Register ini hanya dapat diakses oleh pengalamatan langsung.

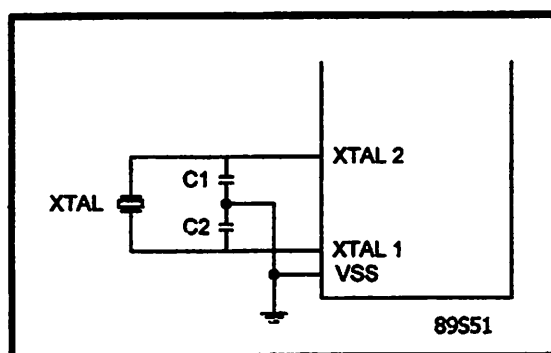


Gambar 2.8 Ruang *Special Function Register*

Sumber: Intel, Agustus 1998

2.1.3 Pewaktuan CPU

Mikrokontroler 89S51 memiliki osilator internal (*on chip oscillator*) yang dapat digunakan sebagai sumber clock bagi CPU. Untuk menggunakan osilator internal diperlukan sebuah kristal antara pena XTAL 1 dan pena XTAL 2 dan dua buah kapasitor ke ground seperti terlihat dalam Gambar 2.9. Untuk kristal dapat digunakan frekuensi dari 6 sampai 12 MHz, sedangkan untuk kapasitor dapat bernilai antara 27 pF sampai 33 pF [Malik & Anistardi, 1997:14].



Gambar 2.9 Menggunakan Osilator Internal

Sumber: Malik & Anistardi, 1997: 14

2.1.4 Interupsi

Apabila CPU dalam mikrokontroler 89S51 sedang melaksanakan suatu program, pelaksanaan program tersebut dapat disela sementara dengan meminta interupsi. Apabila CPU mendapat permintaan interupsi, alamat berikutnya pada program yang disela akan disimpan di *stack* pada RAM internal dan selanjutnya *program counter* (PC) akan diisi oleh alamat interupsi yang telah ditentukan secara internal. CPU kemudian melaksanakan rutin pelayanan interupsi mulai dari alamat awal interupsi. PC akan diisi oleh alamat yang diambilkan dari *stack* jika instruksi RETI yang berada di akhir subrutin penanganan interupsi telah dieksekusi.

Pada mikrokontroler 89S51 terdapat beberapa saluran interupsi. Interupsi pada 89S51 dibedakan dalam 2 jenis, yaitu:

1. Interupsi yang tidak dapat dihalangi oleh perangkat lunak (*non maskable interrupt*), misalnya *reset*.
2. Interupsi yang dapat dihalangi oleh perangkat lunak (*maskable interrupt*). Contoh interupsi jenis ini adalah $\overline{INT_0}$ dan $\overline{INT_1}$ (eksternal) serta *Timer/counter* 0, *Timer/counter* 1, dan interupsi dari *port* serial (internal).

Instruksi RETI (*return from interrupt routine*) harus digunakan untuk kembali dari subrutin layanan interupsi. Instruksi RETI dipakai agar saluran interupsi kembali dapat digunakan kembali. Alamat awal layanan rutin interupsi dari setiap sumber interupsi diperlihatkan dalam Tabel 2.1.

Tabel 2.1 Alamat layanan rutin interupsi

Nama	Lokasi
Reset	00H
\overline{INT}_0	03H
Timer 0	0BH
\overline{INT}_1	13H
Timer 1	1BH
Sint	23H

Sumber: Malik & Anistardi, 1997: 16

Mikrokontroler 89S51 menyediakan lima sumber interupsi, yaitu: dua interupsi eksternal, dua interupsi *timer*, dan satu interupsi *port* serial. Interupsi eksternal \overline{INT}_0 dan \overline{INT}_1 , masing-masing dapat diaktifkan berdasarkan level atau transisi, tergantung pada bit IT0 dan IT1 dalam TCON. Apabila ITX (mengacu pada IT0 atau IT1) diset logika 1 maka sinyal transisi dari keadaan tinggi ke rendah akan mengaktifkan interupsi eksternal, sebaliknya jika diset logika 0, maka sinyal dengan logika rendah akan mengaktifkan interupsi. Interupsi *timer* 0 dan *timer* 1 dihasilkan oleh TF0 dan TF1. Interupsi *port* serial dihasilkan oleh logika OR dari RI dan TI. Ada dua buah register yang mengontrol interupsi, yaitu IE (*interrupt enable*) dan IP (*interrupt priority*).

2.1.4.1 Interrupt Enable

Setiap sumber interupsi dapat diaktifkan atau dilumpuhkan secara individual dengan mengatur satu bit di SFR yang bernama IE (*interrupt enable*). Bit-bit IE didefinisikan sebagai berikut:

Tabel 2.2 Definisi Bit-bit IE (*interrupt enable*)

MSB							LSB
EA	—	—	ES	ET1	EX1	ET0	EX0

Tabel 2.3 Keterangan IE (*interrupt enable*)

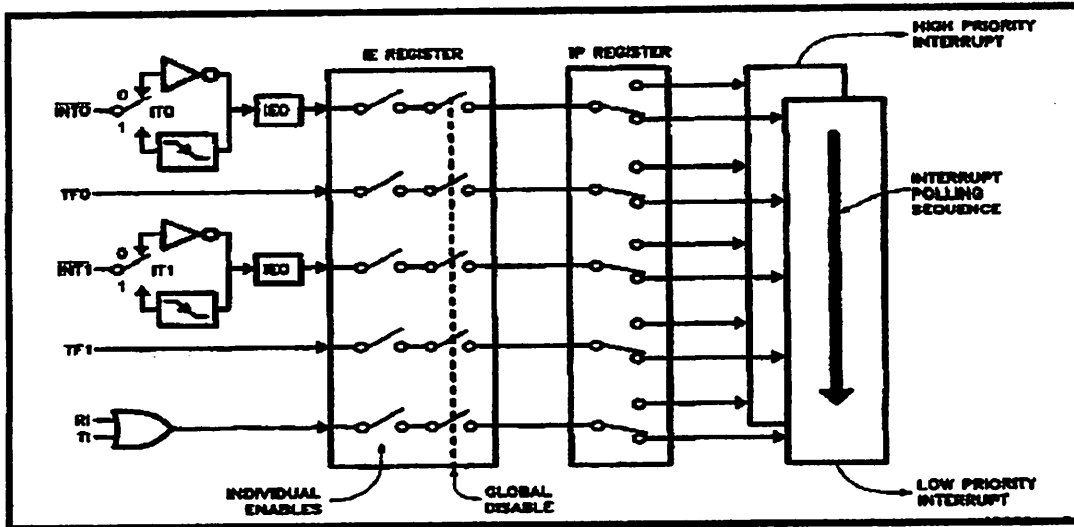
Simbol	Posisi	Fungsi
EA	IE.7	Melumpuhkan semua interupsi. Jika EA=0 tidak ada interupsi yang akan dilayani. Jika EA=1 setiap sumber interupsi dapat dijalankan atau dilumpuhkan secara individual.
—	IE.6	Kosong
—	IE.5	Kosong
ES	IE.4	Bit pembuat enable <i>port</i> serial
ET1	IE.3	Bit pembuat enable <i>timer</i> 1
EX1	IE.2	Bit pembuat enable $\overline{\text{INT}}1$
ET0	IE.1	Bit pembuat enable <i>timer</i> 0
EX0	IE.0	Bit pembuat enable $\overline{\text{INT}}0$

Jika akan mengaktifkan interupsi 0 ($\overline{\text{INT}}_0$), misalnya, nilai yang harus diberikan ke IE adalah 81H (yaitu memberikan logika 1 ke EA dan EX0).

2.1.4.2 Prioritas Interupsi

Setiap sumber interupsi dapat diprogram secara individual (sendiri-sendiri) menjadi satu atau dua tingkat prioritas dengan mengatur bit pada SFR yang bernama IP (*interrupt priority*). Interupsi dengan prioritas rendah (*low priority*) dapat diinterupsi oleh interupsi yang memiliki prioritas lebih tinggi (*high priority*), tetapi tidak dapat diinterupsi oleh interupsi yang memiliki prioritas lebih rendah. Interupsi yang memiliki prioritas tertinggi tidak dapat diinterupsi oleh sumber interupsi lainnya.

Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas tertinggi yang akan dilayani. Jika permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan *polling* untuk menentukan mana yang akan dilayani.



Gambar 2.10 Sistem Kontrol Interupsi

Bit-bit pada IP adalah sebagai berikut:

Tabel 2.4 Definisi Bit-bit IP (*interrupt priority*)

MSB							LSB
—	—	—	PS	PT1	PX1	PT0	PX0

Priority bit = 1 menandakan prioritas tinggi

Priority bit = 0 menandakan prioritas rendah

Tabel 2.5 Keterangan IP (*interrupt priority*)

Simbol	Posisi	Fungsi
—	IP.7	Kosong
—	IP.6	Kosong
—	IP.5	Kosong
PS	IP.4	Bit prioritas interupsi <i>port serial</i>
PT1	IP.3	Bit prioritas interupsi <i>timer 1</i>
PX1	IP.2	Bit prioritas interupsi <i>/INT.1</i>
PT0	IP.1	Bit prioritas interupsi <i>timer 0</i>
PX0	IP.0	Bit prioritas interupsi <i>/INT.0</i>

2.1.5 *Special Function Register (SFR)*

SFR berisi register-register dengan fungsi tertentu. Masing-masing register SFR ditunjukkan dalam Tabel 2.6 yang meliputi simbol, nama, dan alamatnya serta keadaannya dalam nilai biner pada saat terjadi *reset*.

Beberapa fungsi SFR yang penting, yaitu:

- *Accumulator (ACC)*: merupakan register penting dalam operasi penambahan dan pengurangan.
- *Register B*: merupakan register khusus yang berfungsi melayani operasi perkalian dan operasi pembagian.
- *Program Status Word (PSW)*: berisi beberapa bit status yang menggambarkan keadaan register akumulator pada saat itu. PSW terdiri atas bit *carry*, *auxiliary carry*, dua bit pemilih *bank*, bendera *overflow*, bit paritas, dan dua bendera yang dapat didefinisikan sendiri oleh pemakai.
- *Stack Pointer (SP)*: merupakan register 8-bit. Register SP dapat diletakkan pada alamat manapun pada RAM internal. Isi register ini ditambah sebelum data disimpan, selama instruksi *PUSH* dan *CALL*. Pada saat *reset*, register SP diinisialisasi pada alamat 07h, sehingga *stack* akan dimulai pada lokasi 08h.

Tabel 2.6 *Special Function Register*

Simbol	Nama	Alamat	Reset
ACC	Akumulator	E0h	0000 0000
B	Register B	F0h	0000 0000
PSW	Program Status Word	D0h	0000 0000
SP	Stack Pointer	81h	0000 0111
DPTR	Data Pointer 16 bit		
DPL	Data Pointer byte rendah	82h	0000 0000
DPH	Data Pointer byte tinggi	83h	0000 0000
P0	Port 0	80h	1111 1111
P1	Port 1	90h	1111 1111
P2	Port 2	A0h	1111 1111
P3	Port 3	B0h	1111 1111
IP	Interrupt Priority Control	B8h	xxx0 0000
IE	Interrupt Enable Control	A8h	0xx0 0000
TMOD	Timer/counter Mode Control	89h	0000 0000
TCON	Timer/counter Control	88h	0000 0000
TH0	Timer/counter 0 high byte	8Ch	0000 0000
TL0	Timer/counter 0 low byte	8Ah	0000 0000
TH1	Timer/counter 1 high byte	8Dh	0000 0000
TL1	Timer/counter 0 low byte	8Bh	0000 0000
SCON	Serial Control	98h	0000 0000
SBUF	Serial Data Buffer	99h	Xxxx xxxx
PCON	Power Control	87h	0xxx xxxx

Sumber: Malik & Anistardi, 1997: 19-20

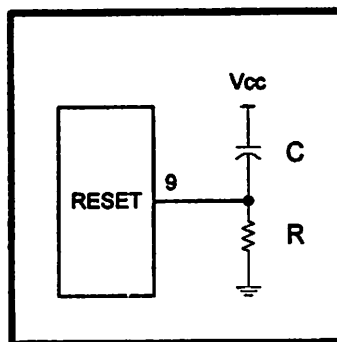
- *Data Pointer (DPTR)*: terdiri dari dua register, yaitu untuk *byte* tinggi (*Data Pointer High*, DPH) dan *byte* rendah (*Data Pointer Low*, DPL). Fungsi DPTR adalah untuk

menahan alamat memori eksternal 16-bit. DPTR dapat dimanipulasi sebagai register 16-bit atau sebagai dua buah register 8-bit.

- *Port 0* sampai *Port 3*: merupakan register yang berfungsi untuk membaca dan mengeluarkan data pada *port 0*, *1*, *2*, dan *3*. Masing-masing register ini dapat dialamati secara per-bit maupun per-byte.
- *Control Register*: terdiri atas register yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*Interrupt Priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat register khusus, yaitu register TMOD (*Timer/Counter Mode Control*) dan register TCON (*Timer/Counter Control*), serta untuk pelayanan *port* serial menggunakan register SCON (*Serial Port Control*).

2.1.6 Reset

Untuk me-*reset* rangkaian mikrokontroler, digunakan rangkaian *power-on reset*. Rangkaian ini akan me-*reset* mikrokontroler secara otomatis setiap kali catu daya dihidupkan. Gambar 2.11 menunjukkan rangkaian *power-on reset*.



Gambar 2.11 Rangkaian *Power-on Reset*

Sumber: Malik & Anistardi, 1997: 8

Prinsip kerja rangkaian *reset* adalah: saat catu daya diaktifkan, rangkaian *reset* akan menahan logika tinggi pada penyemat RST untuk jangka waktu tertentu. Jangka waktu tersebut ditentukan oleh pengosongan muatan pada kondensator. Untuk memastikan keabsahan *reset*, logika tinggi tersebut harus ditahan untuk waktu yang lebih lama dari dua siklus mesin ditambah waktu mulai hidup (*start-on*) osilator.

2.1.7 Timer/counter

Pada mikrokontroler 89S51 terdapat dua buah *timer/counter* 16 bit yang dapat diatur melalui perangkat lunak, yaitu *timer/counter* 0 dan *timer/counter* 1. Apabila *timer/counter* diaktifkan pada frekuensi kerja mikrokontroler 12 MHz, *timer/counter* akan melakukan perhitungan waktu sekali setiap 1 mikrodetik secara independen, tidak tergantung pada pelaksanaan suatu instruksi. Satu siklus pencacahan waktu berpadanan dengan satu siklus pelaksanaan instruksi, sedangkan satu siklus diselenggarakan dalam waktu 1 mikrodetik. Bila dimisalkan suatu urutan instruksi telah selesai dilaksanakan dalam waktu 5 mikrodetik, pada saat itu pula *timer/counter* telah menunjukkan periode waktu 5 mikrodetik.

Apabila periode waktu tertentu telah dilampaui, *timer/counter* segera menginterupsi mikrokontroler untuk memberitahukan bahwa perhitungan periode waktu telah selesai dilaksanakan. Periode waktu *timer/counter* secara umum ditentukan oleh persamaan berikut:

- Sebagai *timer/counter* 8 bit:

$$T = (255 - TLx) \times 1 \mu s \dots\dots\dots (2-1)$$

dengan TLx adalah isi register TL0 atau TL1.

- Sebagai *timer/counter* 16 bit:

$$T = (65535 - THx \times TLx) \times 1 \mu s \dots\dots\dots (2-2)$$

THx = isi register TH0 atau TH1

TLx = isi register TL0 atau TL1

Pengontrol kerja *timer/counter* adalah register *timer control* (TCON). Definisi dari bit-bit pada *timer control* adalah sebagai berikut:

Tabel 2.7 Definisi Bit-bit pada Timer Control

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Tabel 2.8 Keterangan Bit-bit pada Timer Control

Simbol	Posisi	Fungsi
TF1	TCON.7	<i>Timer 1 overflow flag. Di-set oleh perangkat keras saat timer/counter menghasilkan limpahan (overflow).</i>
TR1	TCON.6	Bit untuk menjalankan <i>Timer 1</i> . Di-set/clear oleh <i>software</i> untuk membuat <i>timer on</i> atau <i>off</i> .
TF0	TCON.5	<i>Timer 0 overflow flag. Di-set oleh perangkat keras.</i>
TR0	TCON.4	Bit untuk menjalankan <i>timer 0</i> . Di-set/clear oleh <i>software</i> untuk membuat <i>timer on</i> atau <i>off</i> .
IE1	TCON.3	<i>External interrupt 1 edge flag.</i>
IT1	TCON.2	<i>Interrupt 1 type control bit. Set/clear oleh software untuk menspesifikasikan sisi turun/level rendah trigger dari interupsi eksternal.</i>
IE0	TCON.1	<i>External interrupt 0 edge flag.</i>
IT0	TCON.0	<i>Interrupt 0 type control bit.</i>

Pengontrol pemilihan mode operasi *timer/counter* adalah register *timer mode* (TMOD) yang bit-bitnya didefinisikan sebagai berikut:

Tabel 2.9 Definisi Bit-bit pada Timer Mode

MSB							LSB
GATE	C/T	M1	M0	GATE	C/T	M1	M0

Keterangan:

GATE : Saat GATE = 1, *Timer/counter* x akan berjalan ketika TRx dalam TCON di-set 1 (*timer* dikontrol *software*).

C/T : Pemilih fungsi *timer* atau *counter*. *Clear* (0) untuk operasi *timer* dengan masukan dari sistem clock internal. *Set* (1) untuk operasi *counter* dengan masukan dari pena T0 atau T1.

M1 : Bit pemilih mode.

M0 : Bit pemilih mode.

Tabel 2.10 Kombinasi M0 dan M1 adalah sebagai berikut:

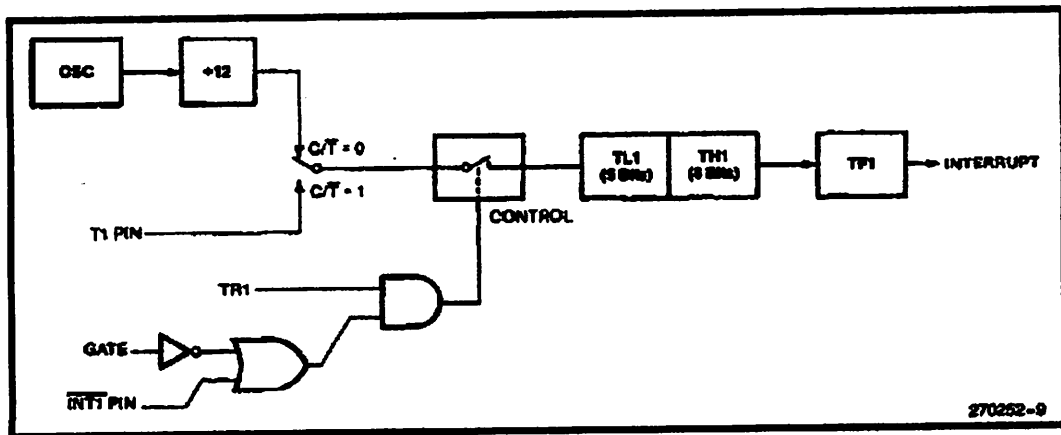
M1	M0	Mode	Operasi
0	0	0	<i>Timer</i> 13 bit
0	1	1	<i>Timer/counter</i> 16 bit
1	0	2	<i>Timer auto reload</i> 8 bit (pengisian otomatis)
1	1	3	TL0 adalah <i>timer/counter</i> 8 bit yang dikontrol oleh kontrol bit standar <i>Timer</i> 0 TH0 adalah <i>timer</i> 8 bit dan dikontrol oleh kontrol bit <i>Timer</i> 1

Keterangan:

Mode 0:

Dalam mode ini register *timer* disusun sebagai register 13 bit. Setelah semua perhitungan selesai, mikrokontroler akan men-set *Timer Interrupt Flag* (TF1). Dengan

membuat $GATE=1$, *timer* dapat dikontrol oleh masukan luar \overline{INT}_1 , untuk fasilitas pengukuran lebar pulsa. Gambar 2.12 menunjukkan *Timer/counter* 1 dalam mode 0.



Gambar 2.12 *Timer/counter* 1 mode 0

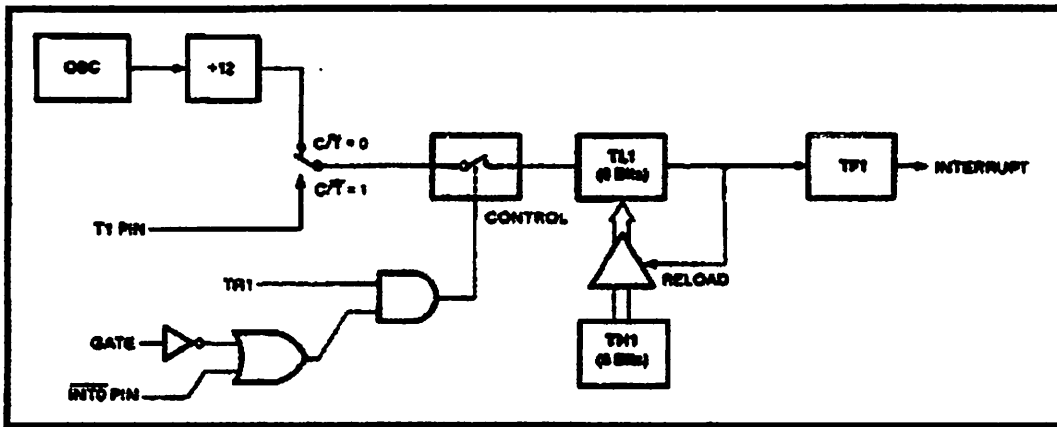
Sumber: Intel, Agustus 1998

Mode 1:

Mode 1 sama dengan mode 0 kecuali register *timer* akan bekerja dalam 16 bit.

Mode 2:

Mode 2 menyusun register *timer* sebagai 8 bit *counter*. Limpahan (*overflow*) dari TL1 tidak hanya men-*set* TF1 tetapi juga mengisi TL1 dengan isi TH1 yang diatur secara *software*. Pengisian ini tidak mengubah TH1. *Timer/counter* 1 dalam mode 2 dapat dilihat dalam Gambar 2.13.



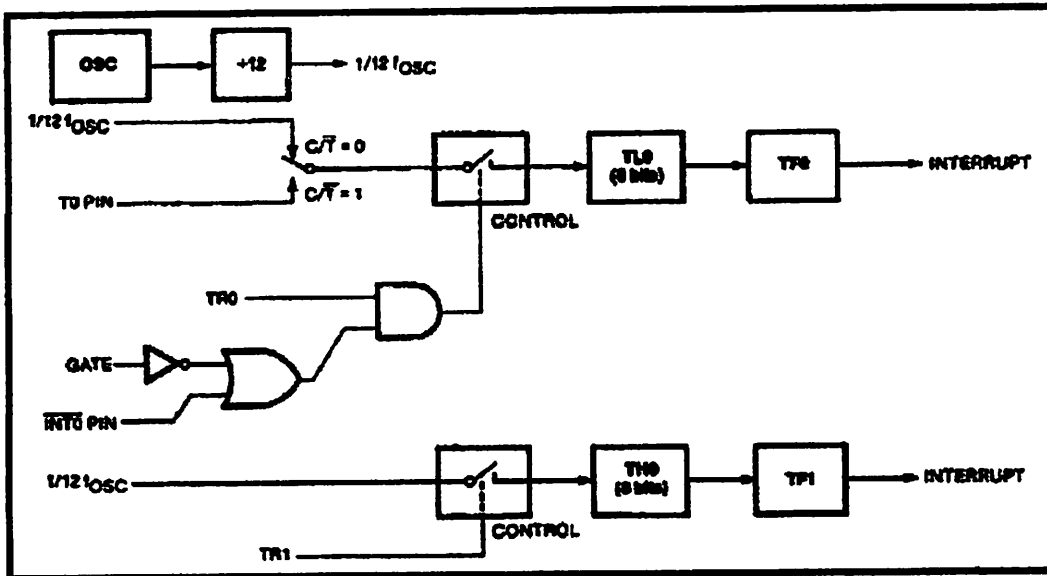
Gambar 2.13 *Timer/counter 1 dalam mode 2*

Sumber: Intel, Agustus 1998

Mode 3:

Timer 1 dalam mode 3 semata-mata memegang hitungan. Efeknya sama seperti men-set $TR1=0$. *Timer 0* dalam mode 3 menetapkan $TL0$ dan $TH0$ sebagai dua *counter* terpisah. $TL0$ menggunakan kontrol bit *timer 0* yaitu C/T , $GATE$, $TR0$, $\overline{INT_0}$, dan $TF0$. $TH0$ ditetapkan sebagai fungsi *timer*.

Mode 3 diperlukan untuk aplikasi yang membutuhkan *timer/counter* 8 bit. Dengan *timer 0* dalam mode 3, mikrokontroler 89S51 memiliki 3 *timer/counter*. Saat *timer 0* dalam mode 3, *timer 1* dapat dihidupkan atau dimatikan, atau dapat digunakan oleh *port serial* sebagai pembangkit baud rate. *Timer/counter 0* dalam mode 3 dapat dilihat dalam Gambar 2.14.



Gambar 2.14 *Timer/counter 0 dalam mode 3*

Sumber: Intel, Agustus 1998

2.1.7.1 Menyetel *Timer*

Tabel 2.11 sampai 2.14 memberikan beberapa nilai bagi TMOD yang dapat digunakan untuk menyetel *timer* dalam mode yang berbeda.

Diasumsikan hanya satu *timer* yang digunakan. Jika diinginkan untuk menjalankan *timer 0* dan *timer 1* secara bersamaan dalam beberapa mode, maka nilai TMOD dalam *timer 0* (Tabel 2.11) harus di-OR-kan dengan nilai TMOD dalam *timer 1* (Tabel 2.13). Sebagai contoh, jika diinginkan untuk menjalankan *timer 0* dalam mode 1 (kontrol eksternal) dan *timer 1* dalam mode 2 *counter* (kontrol internal), maka nilai yang harus diisikan pada TMOD adalah 69H (nilai 09H dari Tabel 2.11 di-OR-kan dengan 60H dari Tabel 2.14).

Pada kontrol internal, *timer* dihidup-matikan dengan men-*set* atau men-*clear* bit TR0 (untuk *timer 0*) dan TR1 (untuk *timer 1*). Hal ini dilakukan lewat kontrol secara *software*. Sedangkan, pada kontrol eksternal, *timer* dihidup-matikan dengan

memberikan logika 0 pada pena $\overline{INT_0}$ (untuk *timer* 0) dan $\overline{INT_1}$ (untuk *timer* 1). Hal ini dilakukan lewat kontrol secara *hardware*.

Tabel 2.11 Fungsi *Timer* 0

Mode	Fungsi <i>Timer</i> 0	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	00H	08H
1	16 bit <i>timer</i>	01H	09H
2	8 bit <i>auto reload</i>	02H	0AH
3	Dua 8 bit <i>timer</i>	03H	0BH

Sumber: Malik & Anistardi, 1997: 24

Tabel 2.12 Fungsi *Counter* 0

Mode	Fungsi <i>Counter</i> 0	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	04H	0CH
1	16 bit <i>timer</i>	05H	0DH
2	8 bit <i>auto reload</i>	06H	0EH
3	Satu 8 bit <i>counter</i>	07H	0FH

Sumber: Malik & Anistardi, 1997: 24

Tabel 2.13 Fungsi *Timer* 1

Mode	Fungsi <i>Timer</i> 1	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit <i>timer</i>	00H	80H
1	16 bit <i>timer</i>	10H	90H
2	8 bit <i>auto reload</i>	20H	A0H

Sumber: Malik & Anistardi, 1997: 24

Tabel 2.14 Fungsi Counter 1

Mode	Fungsi Timer 0	TMOD	
		Kontrol Internal	Kontrol Eksternal
0	13 bit timer	40H	C0H
1	16 bit timer	50H	D0H
2	8 bit auto reload	60H	E0H

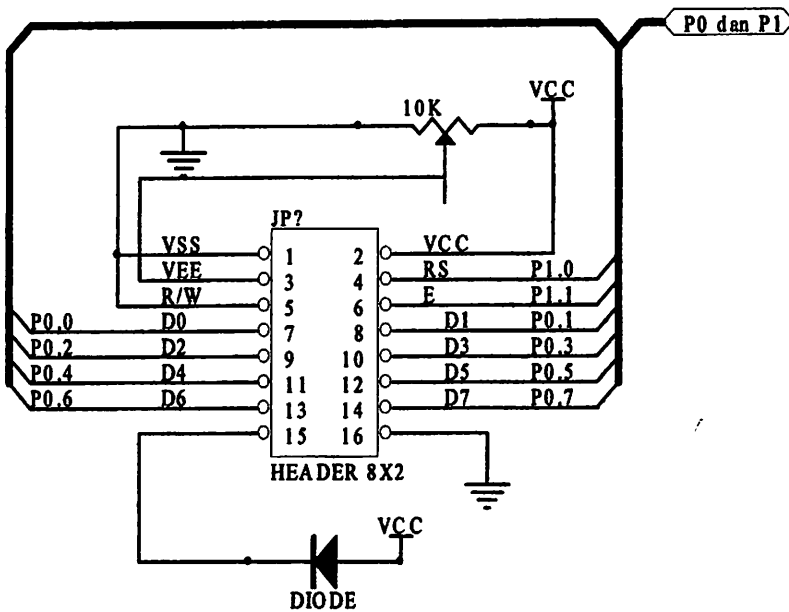
Sumber: Malik & Anistardi, 1997: 24

2.2 Liquid Crystal Display (LCD)

Pada sistem yang direncanakan akan digunakan LCD (Liquid Crystal Display) sebagai tampilan. LCD yang digunakan adalah jenis TM162ABC yang merupakan LCD dua baris dengan tiap barisnya terdiri dari 16 karakter.

LCD ini membutuhkan 3 sinyal kontrol, R/W (*read/write*) untuk menentukan apakah data akan dibaca atau ditulis, E (*Enable*) yang merupakan sinyal untuk meng-enable-kan dan RS (*Register Select*) untuk memilih register yang diakses. LCD TM162ABC memiliki 2 register yaitu register data dan register instruksi.

Pin R/W dihubungkan ke *ground* atau selalu berlogika 0 karena dalam perancangan. LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin P1.0 sistem mikrokontroler. Pengaktifan LCD ini selanjutnya tergantung pada pin E. Dimana pin E ini tergantung dari P1.1 dari mikrokontroler. Bus data LCD dihubungkan dengan port 0. Rangkaian LCD seperti terlihat dalam Gambar 2.15.



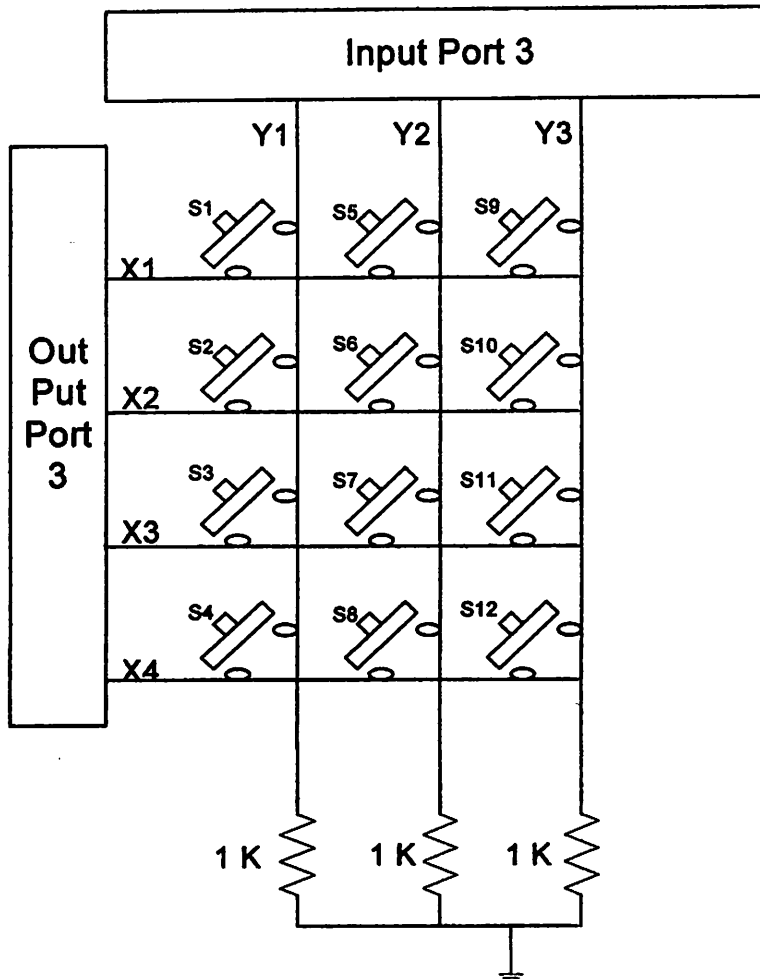
Gambar 2.15 Koneksi Header LCD 16x2 dengan MCU

2.3 Papan Tombol (*Keypad*)

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port 3*.

Papan tombol tersebut mempunyai matrik 4 baris dan 4 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port 3* yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan *ground* (berlogika 0) dan *port 3* (P34-P37) yang difungsikan sebagai *input* mikrokontroler. Sedangkan deretan baris dihubungkan ke *port 3* (P30-P33) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri. Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. *Port* ini

difungsikan sebagai *output* dari mikrokontroler. Rangkaian papan tombol tersebut dapat dilihat dalam Gambar 3.3.

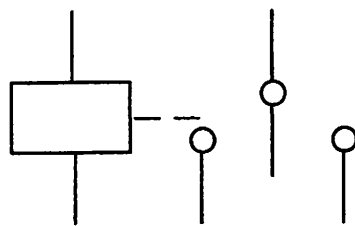


Gambar 2.16 Koneksi *Keypad* dengan MCU

2.4 Relay

Relay adalah komponen elektronika yang umumnya digunakan untuk menghidupkan rangkaian kontrol dan peralatan listrik lainnya yang menarik arus relatif kecil. Namun demikian relay dapat mengontrol tegangan dan arus yang lebih besar

dengan menggunakan efek penguatan (*Amplifying effect*). *Amplifying effect* didapat dengan memanfaatkan tegangan kecil (5-24 Volt) untuk mengoperasikan koil dan relay. Kemudian koil tersebut digunakan untuk mengubah-ubah posisi kontak. Kontak pada relay dapat digunakan untuk mensaklar (*switching*) tegangan yang lebih besar sampai 460 Watt. Aliran arus yang digunakan untuk mengatur koil. Relay terpisah dari arus listrik yang dikontrol oleh kontak-kontak pada relay tersebut.



Gambar 2.17 Simbol Relay

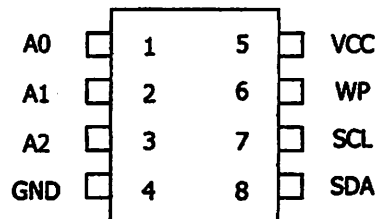
Pada dasarnya relay dapat dikatakan sebagai kuda beban elektrik. (*Electrical work horses*) yang mengontrol suatu rangkaian elektrik dengan cara membuka dan menutup kontak pada rangkaian lain (rangkaiannya Internal). Apabila kontak relay adalah Normally Open (NO), maka akan terbuka bila relay tidak dialiri energi listrik. Sebaliknya pada titik kontak yang tergolong Normally Closed (NC) akan tertutup apabila relay tidak dialiri energi listrik, pada kedua kondisi tersebut kontak-kontak dari relay akan berubah kedaannya apabila relay dialiri energi listrik.

Berdasarkan bedanya relay dapat dibagi menjadi dua golongan besar, yaitu *elektro mechanical relay* dan *solid state relay*, pada *elektro mechanical relay* kontak-kontak dibuka dan ditutup dengan kekuatan daya magnet (*magnetic forces*), sedangkan pada *solid state relay* tidak terdapat kontaktor, semua pensaklaran (*switching*) dilakukan secara elektronik (memanfaatkan sifat listrik bahan semikonduktor).

Terdapat berbagai jenis konfigurasi yang berbeda dari relay bergantung pada jumlah dari *break pole* dan *throw*, sebagai contoh sebuah relay dapat digambarkan sebagai *Single Pole Double Throw (SPDT)* atau *Double Pole Double Throw (DPDT)*.

2.5 EEPROM 24C64

EEPROM (Electrically Erasable and Programmable Read Only Memory) adalah memory yang dapat dihapus dan diprogram ulang. Berbeda dengan EEPROM biasa, 24C64 dapat diperlakukan sebagai RAM pada umumnya, namun pada EEPROM jenis ini program yang telah ditulis didalamnya dapat bertahan walaupun catu daya telah diputus (dimatikan).



Gambar 2.18 EEPROM 24C64

Fungsi-fungsi dari pin EEPROM 24C64 ini antara lain :

SCL (SerialClock), SCL berfungsi sebagai input apabila ada clock tepi naik pada EEPROM dan clock tepi turun untuk data out.

SDA (Seral Data), Pin SDA ini merupakan jalur bidirectional untuk transfer data secara serial.

A0, A1, A2, merupakan pin-pin alamat input pada EEPROM 24C64 tidak menggunakan pin alamat, jadi pin A0, A1, A2 tidak dihubungkan.

WP (Write Protec) untuk memberikan proteksi pada data pada saat dihubungkan ke ground. Operasi baca tulis bisa dilaksanakan ketika pin

WP dihubungkan ke Vcc, jika dihubungkan ke ground maka operasi baca tulis tidak dapat dilaksanakan.

2.6 Optocoupler

2.6.1 Laser Diode

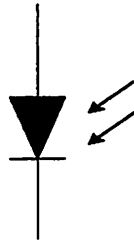
Sinar laser (*Light Amplification By Stimulated Emission Of Radiation*) merupakan cahaya yang intensitasnya digandakan dan difokuskan pada arah tertentu. Sinar laser bersifat koheren dan mempunyai intensitas yang sangat tinggi. Tahun 1960 untuk pertama kalinya sinar laser He-Ne di demontrasikan oleh Javan, Bennet dan Heriot. Setelah itu berkembang sinar laser jenis gas seperti kripton dan sinar laser jenis zat cair seperti laser dyne. Salah satu dari jenis laser adalah laser diode atau disebut juga laser injeksi, karena pemompaannya dilakukan dengan injeksi arus listrik lewat sambungan PN semikonduktornya. Jadi laser ini tidak lain adalah sebuah diode dengan bias maju biasa.

Laser semikonduktor yang pertama diciptakan secara bersamaan oleh tiga kelompok pada tahun 1962. Mereka adalah R.H. Rediker dkk. (Lincoln Lab, MIT), M.I. Nathan dkk. (Yorktown Heights, IBM) dan R.N. Hall dkk. (General Electric Research Lab.). Diode-diode yang digunakan adalah galiun arsenida-flofida GaAsP (sinar-tampak merah). Proses laser jenis ini mirip dengan kerja LED biasa. Pancaran fotonnya disebabkan oleh bergabungnya kembali elektron dan lubang (hole) di daerah sambungan PN-nya.

2.6.2 PhotoDiode

Photodiode adalah diode yang dirancang khusus untuk mendeteksi cahaya. Simbol dari photodiode dapat dilihat pada gambar 2.2. Energi cahaya lewat melalui

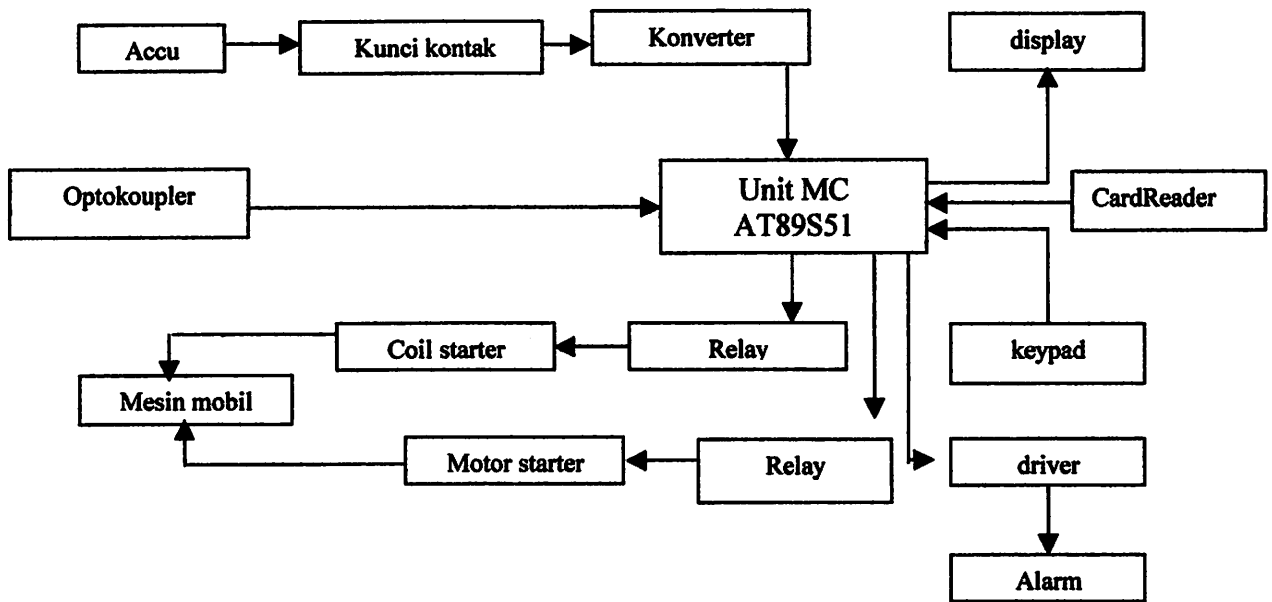
lensa dan dikumpulkan menjadi satu titik sehingga jatuh pada sambungan P dan N. Photodiode dirancang dengan beroperasi pada bias mundur, pada alat ini arus bocor bias mundur meningkat sebanding dengan peningkatan cahaya. Harga arus umumnya adalah dalam rentang mikroampere. Keuntungan dari photo-diode adalah mempunyai respon waktu yang cepat terhadap cahaya.



Gambar 2.20 Simbol Photodiode

BAB III

PERENCANAAN DAN PEMBUATAN ALAT



Gambar 3.1 Diagram blok sistem

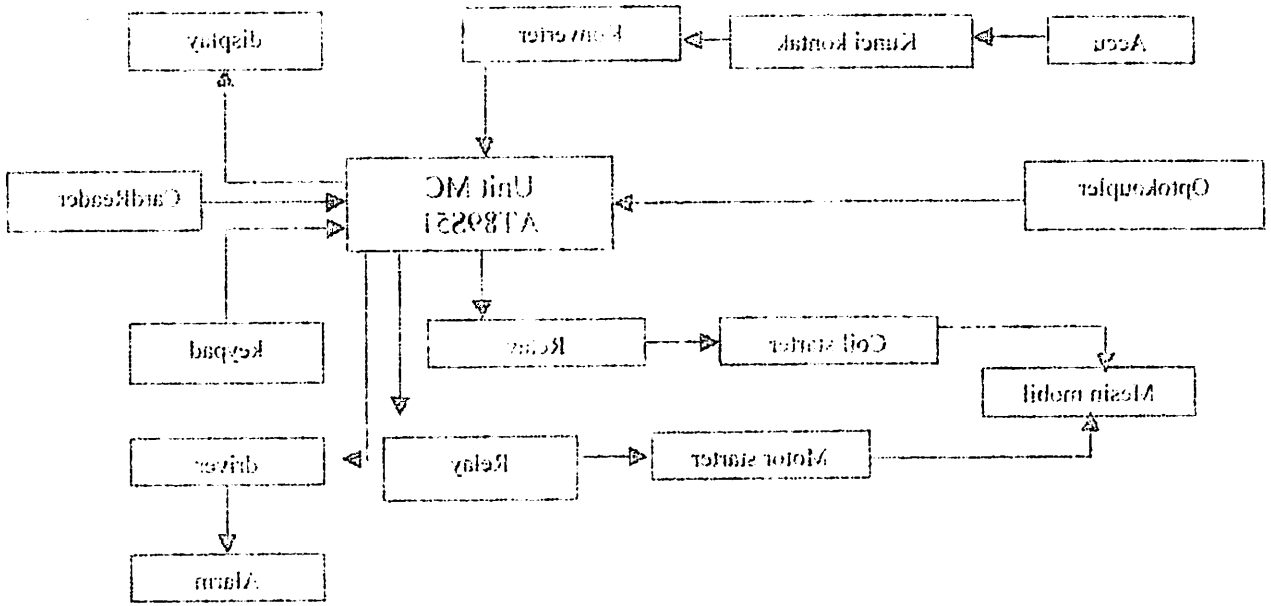
Fungsi setiap blok diagram dijelaskan sebagai berikut :

Accu : sebagai sumber arus DC yang telah tersedia pada mobil.

Kunci kontak : sebagai penghubung kelistrikan untuk melakukan langkah starter secara manual, sebagai antisipasi bila *card* rusak atau nomor PIN secara tidak sengaja dilupakan operator yang pengaktifannya harus melalui prosedur tertentu yang hanya diketahui oleh operator.

BAB III

PERENCANAAN DAN PEMBUATAN ALAT



Gambar 3.1 Diagram blok sistem

Fungsi setiap blok diagram dijelaskan sebagai berikut :

- Acara : sebagai sumber data DC yang telah tersedia pada mobil.
- Kunci kontak : sebagai penghubung kestabilan untuk melakukan langkah starter secara manual, sebagai antisipasi bila rawat masuk akan nomor PIN secara tidak sengaja dilakukan operator yang pengertiannya harus melalui prosedur tertentu yang hanya diketahui oleh operator.

- Konverter** : digunakan untuk menurunkan tegangan Accu yang bertegangan 12 V menjadi 5 V DC yang dibutuhkan mikrokontroller.
- Display** : digunakan untuk menampilkan informasi suatu kondisi seperti masukkan password, aktif atau tidaknya prosedur pengamanan, mesin hidup, mesin mati dan lain-lain.
- Keypad** : digunakan untuk memberikan inputan yang dibutuhkan untuk mengaktifkan sistem yang berupa password dalam format angka dan pengaktifan menu-menu yang lain yang berhubungan dengan prosedur pengamanan
- Card reader** : digunakan sebagai pengganti kunci kontak artinya dengan memasukkan kartu khusus dengan menggunakan IC EEPROM 24C64 maka sistem pengaman bisa diakses dan lebih lanjut pengaktifan starter dapat dilakukan dengan menginputkan password terlebih dahulu, kecuali melalui sakelar rahasia mode pengamanan ini tidak diaktifkan maka operator dapat langsung melakukan langkah starter tanpa menginputkan password. EEPROM 24C64 digunakan sebagai memori eksternal untuk menyimpan program yang berhubungan dengan password.
- Sensor** : digunakan untuk mengetahui status mesin sudah menyala atau belum untuk kemudian mengirim sinyal ke mikrokontroler untuk

mendisplaykan status kondisi, apakah mesin menyala atau belum.

Mikrokontroler : AT 89S51 digunakan sebagai prosesor yang mengatur langkah kerja sistem sesuai prosedur yang telah penulis rancang.

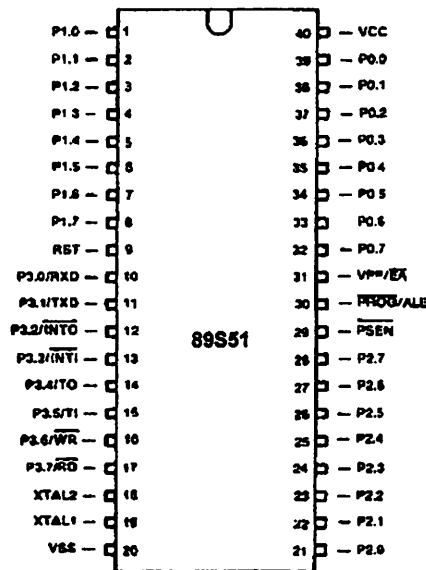
Relay : digunakan untuk menghidupkan rangkaian kontrol dan peralatan listrik lainnya dengan input arus relatif kecil. Namun demikian relay dapat mengontrol tegangan dan arus yang lebih besar dengan menggunakan efek penguatan (Amplifying effect).

3.1 Sistem Mikrokontroller

Sistem mikrokontroller terdiri atas mikrokontroller dan komponen-komponen pendukung agar sistem dapat bekerja dengan optimal..

3.1.1 Mikrokontroller 89S51

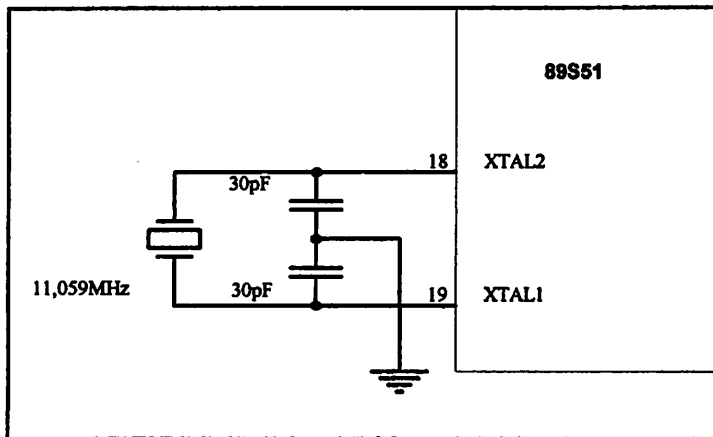
Mikrokontroller 89S51 harus didukung oleh beberapa rangkaian lain agar dapat melakukan prosesnya, yaitu berupa rangkaian clock dan reset. Selain itu juga harus ditentukan penggunaan port-portnya dan sinyal-sinyal yang digunakan untuk mendukung proses yang akan dilakukan. Berikut ini adalah gambar pin-pin dalam Mikrokontoller AT89S51.



Gambar 3.2 Susunan Pena Mikrokontroler 89S51

3.1.1.1 Clock

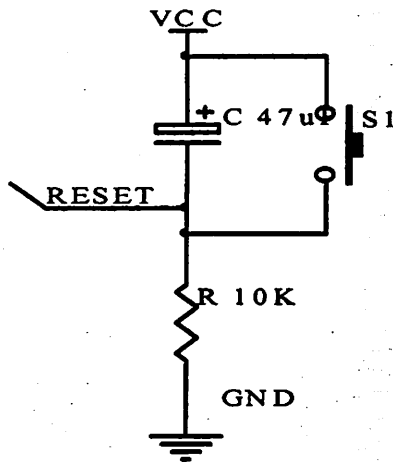
Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* (pewaktuan) yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini akan menggunakan osilator internal yang sudah tersedia di dalam chip 89S51. Untuk menentukan frekuensi osilatornya cukup dengan cara menghubungkan kristal pada pin XTAL1 dan XTAL2 serta dua buah kapasitor ke ground. Besar kapasitansinya disesuaikan dengan spesifikasi pada lembar data 89S51 yaitu 30 pF. Pemilihan besar frekuensi kristal disesuaikan dengan pemilihan kecepatan yang diharapkan untuk transfer data melalui pin *serial interface* 89S51 tersebut. Gambar 3.3 memperlihatkan rangkaian pewaktu yang digunakan.



Gambar 3.3 Rangkaian Pewaktuan

3.1.1.2 Reset

Untuk mereset mikrokontroler, pin RST harus diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset pada saat awal catu daya dihidupkan, suatu reset otomatis dapat dilakukan dengan menghubungkan pin RST ke rangkaian *Power-On Reset*, seperti dalam Gambar 3.4 Ketika catu daya dinyalakan, rangkaian akan menahan pin RST dalam kondisi logika tinggi selama selang beberapa saat tergantung nilai kapasitor dan kecepatan pengisian muatannya. Pemberian catu daya pada mikrokontroler tanpa suatu sinyal reset dapat menyebabkan CPU memulai eksekusi instruksinya dari lokasi yang tak tertentu. Ini disebabkan karena *Program Counter* tidak terinisialisasi.



Gambar 3.4 Rangkaian *Power-On Reset*

3.1.1.3 Pembagian Port

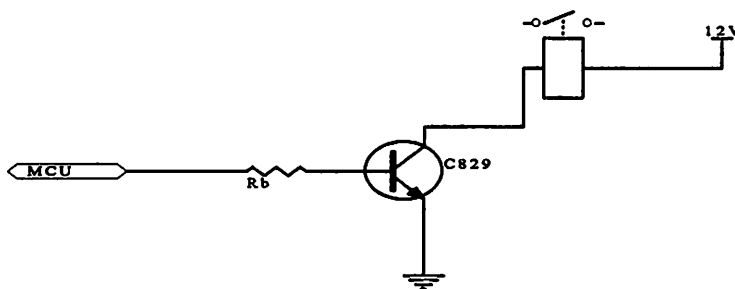
Mikrokontroler 89S51 mempunyai 4 buah port yaitu port 0 (P0), port 1 (P1), port 2 (P2) dan port 3 (P3).

- Port 0 : Bus Data LCD
- Port 1 : Kontrol LCD
- Port 2 : Sensor, card reader, dan driver relay.
- Port 3 : Masukan Keypad

3.1.1.4 Pengendali (Driver Motor).

Bagian ini digunakan untuk mengendalikan relay yang akan memutus dan menghubungkan motor DC dengan catu daya.

Rangkaian penggerak relay ditunjukkan dalam Gambar 3.5 karena arus keluaran mikrokontroler terlalu kecil untuk menggerakkan relay, maka diperlukan pengendali relay. Untuk keperluan ini maka digunakan transistor C829.



Gambar 3.5 Rangkaian Penggerak relay

Apabila V_1 (tegangan masukan transistor) = 0 v, transistor bekerja dalam keadaan terputus. Dalam hal ini tidak ada arus mengalir melalui beban. Tetapi bila V_1 cukup besar, titik operasi beralih dari keadaan putus ke keadaan jenuh. Besar arusnya jika resistansi relay 165Ω (pengukuran) adalah:

$$I_{C(Sat)} = \frac{V_{cc}}{RI}$$

$$= \frac{12}{165} = 72,7 \text{ mA}$$

Dengan $I_{C(Sat)}$ 72,7 mA maka besarnya I_B :

$$I_{B(Sat)} = \frac{I_{C(Sat)}}{\beta_{dc}}$$

Besarnya β transistor C829 didesain pada penguatan 100, sehingga $I_{B(Sat)}$ diperoleh 0,727 mA. Tegangan masukan basis merupakan keluaran mikrokontroler, sehingga

tegangan masukan basis untuk level high minimum sekitar 2,4 volt. Tegangan sambungan BE yaitu 0,7 volt. Maka untuk menghasilkan kejenuhan, besarnya R_B :

$$V_1 = I_{B(\text{Sat})} \cdot R_B + V_{BE}$$

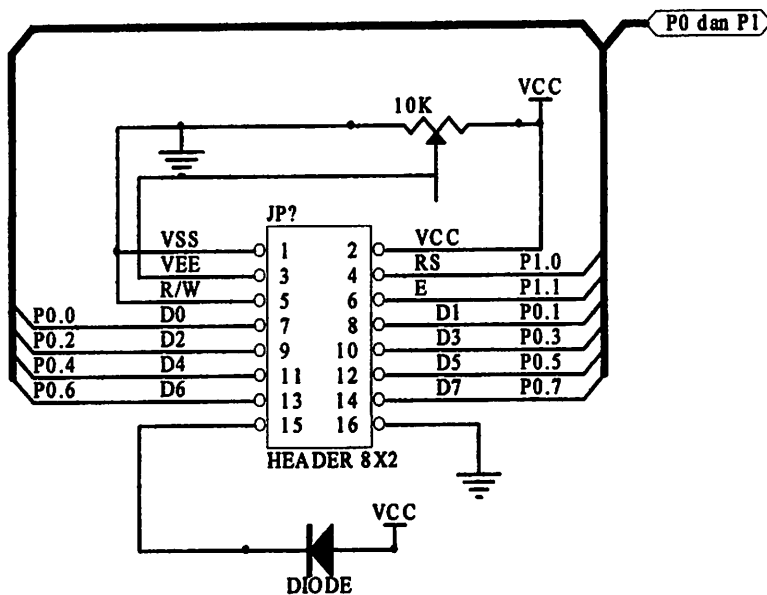
Dengan memasukkan nilai yang telah diperoleh sebelumnya, maka didapatkan R_B sebesar 2,338 k Ω . Dari nilai paling mendekati yang mudah didapat yaitu 2,2 k Ω .

3.1.1.5 Liquid Crystal Display (LCD)

Pada sistem yang direncanakan akan digunakan LCD (Liquid Crystal Display) sebagai tampilan. LCD yang digunakan adalah jenis TM162ABC yang merupakan LCD dua baris dengan tiap barisnya terdiri dari 16 karakter.

LCD ini membutuhkan 3 sinyal kontrol, R/W (*read/write*) untuk menentukan apakah data akan dibaca atau ditulis, E (*Enable*) yang merupakan sinyal untuk meng-enable-kan dan RS (*Register Select*) untuk memilih register yang diakses. LCD TM162ABC memiliki 2 register yaitu register data dan register instruksi.

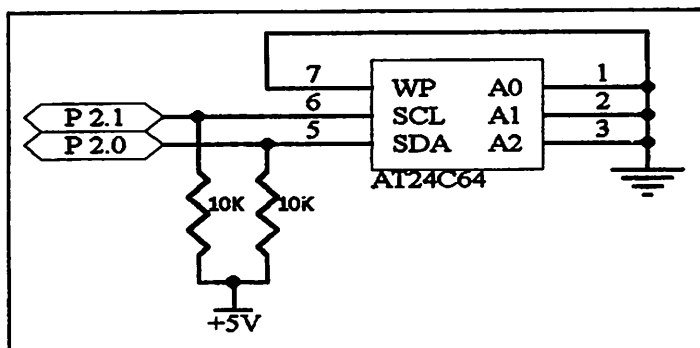
Pin R/W dihubungkan ke *ground* atau selalu berlogika 0 karena dalam perancangan. LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin P1.0 sistem mikrokontroler. Pengaktifan LCD ini selanjutnya tergantung pada pin E. Dimana pin E ini tergantung dari P1.1 dari mikrokontroler. Bus data LCD dihubungkan dengan port 0. Rangkaian LCD seperti terlihat dalam Gambar 3.6



Gambar 3.6 Koneksi LCD 16x2 dengan MCU

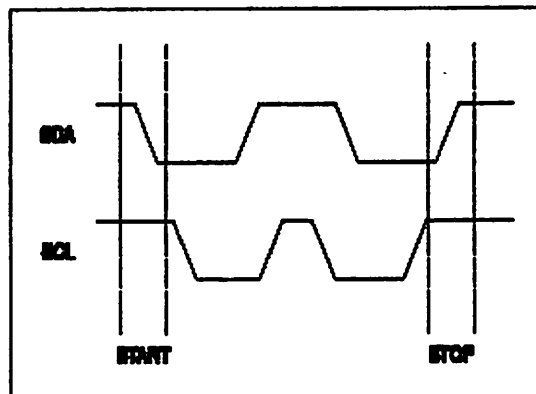
3.1.1.6 Perancangan Unit Card Reader

Gambar antarmuka serial EEPROM AT24C64 dengan mikrokontroler AT89S51 dapat dilihat pada Gambar 3.7



Gambar 3.7 Antarmuka Serial EEPROM dengan
----- Mikrokontroler AT89S51

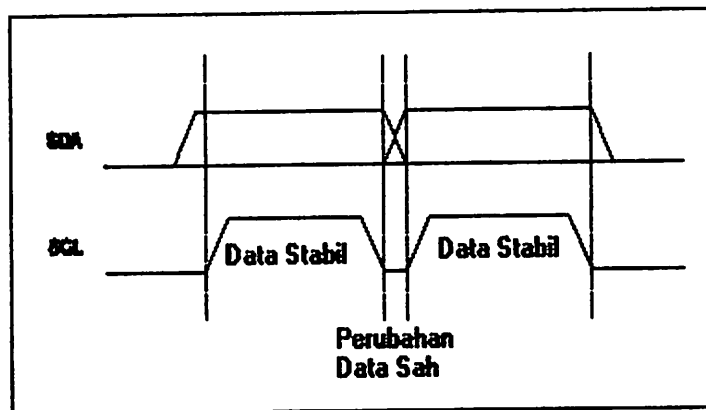
Pin *Serial Data* (SDA) dan pin *Serial Clock* (SCL) memegang peranan di dalam mengendalikan kerja dari memori *serial* AT24C64. Kedua pin tersebut memerlukan tahanan *pull-up* sebesar $10K\Omega$. *Serial data* digunakan untuk mentransfer data dua arah secara serial. *Serial clock* digunakan untuk meng-clock data. Clock sisi positif digunakan untuk memasukkan setiap data ke dalam EEPROM dan clock sisi negatif digunakan untuk mengeluarkan data dari EEPROM. Perubahan data pada SDA selama kondisi pin SCL dalam keadaan *high* menunjukkan adanya kondisi mulai dan kondisi berhenti dari memori. Diagram waktu untuk kondisi mulai kerja dan kondisi berhenti dari memori dapat dilihat pada Gambar 3.8



Gambar 3.8 Diagram waktu untuk kondisi start dan stop pada memori.

Sumber: Atmel Serial Memori Data Sheet: 6

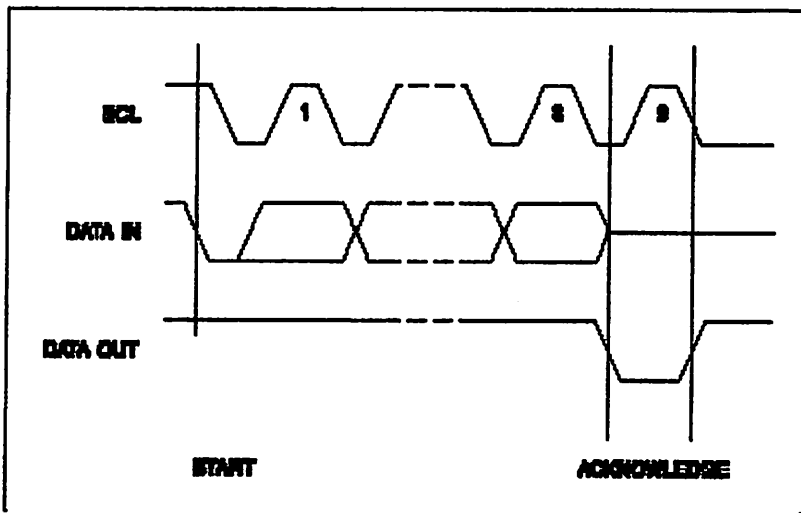
Selain untuk kondisi mulai dari kondisi berhenti, perubahan data pada pin SDA hanya akan sah apabila pin SCL dan keadaan kondisi *low*. Diagram waktu yang menunjukkan kapan perubahan data pada SDA akan sah dapat dilihat pada Gambar 3.9



Gambar 3.9 Diagram waktu yang menunjukkan kapan perubahan data sah.

Sumber: Atmel Serial Memori Data Sheet: 6

Seperti yang telah disebutkan didepan bahwa EEPROM 24C64 memiliki kapasitas sebesar 8192 word dengan masing-masing word sebesar 8 bit. Selain itu EEPROM yang dipakai adalah jenis serial EEPROM. Oleh sebab itu, setiap kali pengiriman data sebesar 8 bit dari mikrokontroller maka data tersebut akan dikirim tiap bit-nya secara serial. Setiap selesai menerima masing-masing word EEPROM secara otomatis akan mengirimkan *zero acknowledge* pada clock ke sembilan. Timing diagram yang menunjukkan *zero acknowledge* dapat dilihat pada Gambar 3.10



Gambar 3.10 Diagram waktu yang menunjukkan *zero acknowledge*

Sumber: Atmel Serial Memori Data Sheet: 6

Setiap mikrokontroller selesai mengirimkan delapan buah data maka secara otomatis EEPROM akan mengirimkan *zero acknowledge*. *Zero acknowledge* akan memiliki kondisi *low* dan setelah diletakkan oleh EEPROM pada SDA akan dipindahkan pada *register C* pada mikrokontroller. Setelah itu diadakan penyeleksian kondisi oleh mikrokontroller. Apabila data pada *register C* mempunyai kondisi *low* maka EEPROM memberi tahu bahwa ke delapan data seluruhnya sudah diterima dan pengiriman data dari mikrokontroller dibatalkan untuk melanjutkan proses selanjutnya. Tetapi apabila nilai data pada *register C* mempunyai kondisi *high* maka EEPROM memberi tahu bahwa telah terjadi kesalahan dalam pengiriman data. Hal ini akan dilanjutkan dengan proses hang atau looping pada kondisi tanpa operasi

secara terus-menerus. Kesalahan ini terjadi misalnya terjadi pengiriman data yang besarnya kurang dari delapan bit.

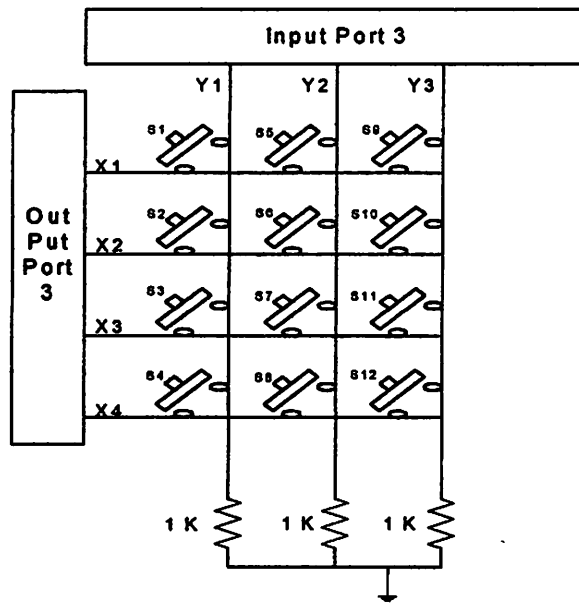
EEPROM 24C64 memerlukan pengalamatan 8 bit yang mengawali kondisi awal yang membuat EEPROM dapat dibaca maupun ditulis. Pada pengalamatan ini diperlukan urutan akses yaitu 1010 sebagai data awalan. Tiga bit data berikutnya data pengalamatan EEPROM, namun untuk tipe 24C64 tidak memerlukan pengalamatan *device*, bit-bit ini harus diberi logika nol. Bit ke delapan dari pengalamatan *device* menandakan operasi baca atau operasi tulis yang akan dilakukan. Untuk operasi pembacaan bit ini harus berlogika tinggi dan untuk operasi penulisan bit ini harus berlogika rendah. Pola pengalamatan memori dapat dilihat pada Bab 2

3.1.1.7 Papan Tombol (*Keypad*)

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port 3*.

Papan tombol tersebut mempunyai matrik 4 baris dan 3 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port 3* yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan *ground* (berlogika 0) dan *port 3* (P30-P32) yang difungsikan sebagai *input* mikrokontroler. Sedangkan deretan baris dihubungkan ke *port 3* (P34-P37) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri. Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. *Port*

ini difungsikan sebagai *output* dari mikrokontroller. Rangkaian papan tombol tersebut dapat dilihat dalam Gambar 3.11



Gambar 3.11 Koneksi *Keypad* dengan MCU

3.1.1.8 Optocoupler

Sensor yang digunakan adalah *laser diode* sebagai pemancar sinar yang diarahkan pada piringan alumunium dan *photo transistor* sebagai penerima pantulan sinar laser dari piringan alumunium. Dalam perencanaan rangkaian sensor putaran ini diharapkan apa bila sinar *laser diode* jatuh pada sisi atas bagian tepi dari piringan alumunium, maka mikrokontroller akan menghitung waktu antara saat sensor mengenai garis hitam dan saat sensor mengenai garis hitam lagi atau waktu 1 putaran piringan.

3.1.1.8.1 LD (Laser Diode)

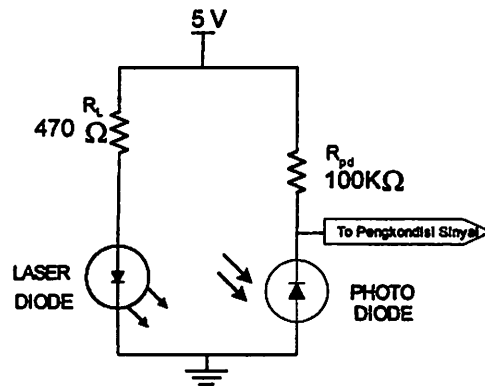
Laser diode merupakan diode yang mampu memancarkan sinar laser. Dalam proyek akhir ini laser diode yang digunakan adalah laser diode bekerja pada tegangan 4,5 Volt. Dari pengukuran arus yang mengalir pada laser diode tersebut sebesar 20mA. Sementara tegangan yang tersedia adalah sebesar 5 Volt sehingga untuk membuat tegangan pada laser diode tersebut menjadi 4,5 Volt dan arus yang mengalir sebesar 1mA maka diperlukan *resistor* sebesar :

$$\begin{aligned} R_L &= \frac{V_{CC} - V_F}{I_F} \dots\dots\dots (3.1) \\ &= \frac{5 - 4,5}{1 \times 10^{-3}} \\ &= 500 \text{ atau dibulatkan ke } 470 \text{ ohm.} \end{aligned}$$

3.1.1.8.2 Photo-Diode

Photo-diode adalah diode yang mampu merespon pancaran cahaya. Jika arus yang dialirkan pada *photo*-diode sebesar 0,04 mA dan tegangannya sebesar 0,7 volt maka diperlukan tahanan sebesar:

$$\begin{aligned} R_{pd} &= \frac{V_{CC} - V_d}{i_d} \dots\dots\dots (3.2) \\ &= \frac{5 - 0,7}{8 \cdot 10^{-5}} \\ &= 107500 \approx 100k \text{ ohm} \end{aligned}$$



Gambar 3.12 Rangkaian Sensor Putaran
(Sumber: Perancangan)

3.1.1.9 Perangkat lunak (software)

Mikrokontroler adalah perangkat yang membaca informasi dari masukan data dan memanipulasi informasi tersebut sesuai dengan program yang disimpan dalam memori yang kemudian menghasilkan informasi pada keluarannya.

Untuk melakukan proses pembacaan, manipulasi data yang kemudian menghasilkan informasi, diperlukan perangkat tambahan yaitu bahasa yang dimengerti oleh IC MCU tersebut. Dalam hal ini bahasa yang digunakan adalah bahasa assembler. Bahasa assembler merupakan tata cara untuk mewakili operasi CPU dalam bentuk format bahasa simbol yang disusun secara berurutan dalam pernyataan-pernyataan (statement). Masing-masing pernyataan menerjemahkan kedalam instruksi bahasa mesin atau operasi biner yang disebut Operation Code (OPCODE). Assembler adalah program komputer yang mengubah bahasa tersebut kedalam operasi kode sehingga dimengerti oleh mikroprocessor (CPU) dalam sistem komputer.

Dalam penulisan bahasa mesin untuk program assembler pada umumnya, terdapat berbagai macam kelompok instruksi, antara lain :

1) *Perpindajian Data*. Instruksi ini berguna untuk memindahkan data antara register-register, memori-memori, register-memori, antar muka register dan antar muka memori. Ada empat perintah yang digunakan dalam perpindahan data tersebut, yaitu :

- a) MOV → Perintah perpindahan data antar register, cara penulisan MOV REGTUJUAN, REGSUMBER dan perintah untuk pemasukan data secara langsung dengan suatu data, cara penulisan MOVREG, #DATA.
- b) MOVX → Perintah pembacaan atau pemasukan data dari luar, cara penulisan MOVX@REGISTER, DATA atau sebaliknya tergantung dari fungsi perintah tersebut.
- c) PUSH dan POP → Perintah untuk memasukkan data ke Stack. Cara penulisan PUSH ALAMAT kemudian diakhiri dengan POP ALAMAT.
- d) XCH → Perintah untuk pertukaran data antar register dan memori. Cara penulisan XCH REG,REG.

Dan terdapat empat mode pengalamatan yang digunakan untuk mengakses data :

- a) Immediate Addressing Mode.
- b) Register Addressing Mode.

- c) Direct Addressing Mode.
 - d) Indirect Addressing Mode.
- 2) **Operasi Aritmatika.** Instruksi ini berfungsi untuk melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, perkalian maupun pembagian.

Cara penulisan adalah :

- a) INC REGISTER : Data pada register bertambah satu.
 - b) DEC REGISTER : Data pada register berkurang satu.
 - c) MUL RegA RegB : Data antar register dikalikan.
 - d) DIV RegARegB : Data register A dibagi register B.
- 3) **Operasi Percabangan.** Instruksi suatu program untuk melaksanakan di lain tempat yang kita perlukan saat ini. Perintah ini dapat dibagi berdasarkan keadaan antara lain :
- a) Lompat berdasarkan keadaan bit. Perintah yang digunakan JB, JNB, JBC, JC, JNC.
 - b) Membandingkan byte dan lompat bila tidak sama. Perintah yang digunakan CJNE, DJNZ, JZ, JNZ.
 - c) Byte berkurang satu dan lompat bila tidak sama dengan 0. perintah yang digunakan JZ, JNZ.
 - d) Lompat tanpa syarat. Perintah yang digunakan JMP, AJMP, LJMP, SJMP.
 - e) CALL subroutine. Perintah yang digunakan CALL.

f) **RETURN** Kembali dari subroutine. Perintah yang digunakan **RETURN**.

4) *Operasi Logika*. Instruksi ini digunakan untuk operasi membandingkan isi suatu register atau variabel. Adapun tata penulisan operasi logika adalah :

OR/AND/XOR Operand1, Operand2

Syarat **Operand1** adalah variabel register ataupun lokasi memori, sedangkan syarat **Operand2** adalah register variabel, isi lokasi memori ataupun suatu angka. Hasil dari operasi logika tersebut akan selalu ditampung di **Operand1**.

BAB IV

PENGUJIAN DAN ANALISIS

4.1 Umum

Bab ini membahas tentang pengujian dan analisis alat yang telah dibuat. Secara umum, pengujian ini bertujuan untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pengujian piranti ini dilakukan dalam dua tahap. Pertama, dilakukan pengujian terhadap perangkat keras pada masing-masing blok rangkaian penyusun sistem antara lain rangkaian MCU, LCD. Pengujian kedua dilakukan pada sistem secara keseluruhan dengan mengintegrasikan perangkat keras dan perangkat lunak untuk mengetahui unjuk kerja sistem. Adapun tujuan pengujian yang dilakukan terhadap sistem adalah sebagai berikut:

- ◆ Mengetahui unjuk kerja rangkaian MCU
- ◆ Mengetahui unjuk kerja rangkaian Keypad
- ◆ Mengetahui unjuk kerja rangkaian LCD
- ◆ Mengetahui unjuk kerja rangkaian Optocoupler
- ◆ Mengetahui unjuk kerja sistem secara keseluruhan dengan mengintegrasikan perangkat keras dan perangkat lunak.

4.2 Pengujian sistem setiap blok

4.2.1 Pengujian Sistem Mikrokontroller

- Tujuan

Untuk mengetahui kondisi awal dari mikrokontroller apakah sudah sesuai dengan yang direncanakan.

- Peralatan yang dibutuhkan

1. Komputer PC AT
2. Writer + Emulator MCS51
3. Led Display

- Prosedur Pengujian

1. Membuat program yang digunakan dalam pengujian mikrokontroller.

Program yang digunakan dalam pengujian mikrokontroller ini merupakan program sederhana yang meletakkan 0FH dan F0H pada ACC secara bergantian kemudian memindahkannya pada *Port 1* 89S51.

Program yang dibuat adalah sebagai berikut

```
ORG 0000H

                JMP     START

START:  MOV     A,#0FH

                MOV     P1,A

                CALL    TUNDA

                MOV     A,#F0H

                MOV     P1,A

                JMP     START

TUNDA:  MOV     R3,#0FFH

TUNDA1: MOV     R2,#0FFH

                DJNZ    R2,$

                MOV     R1,#0FH

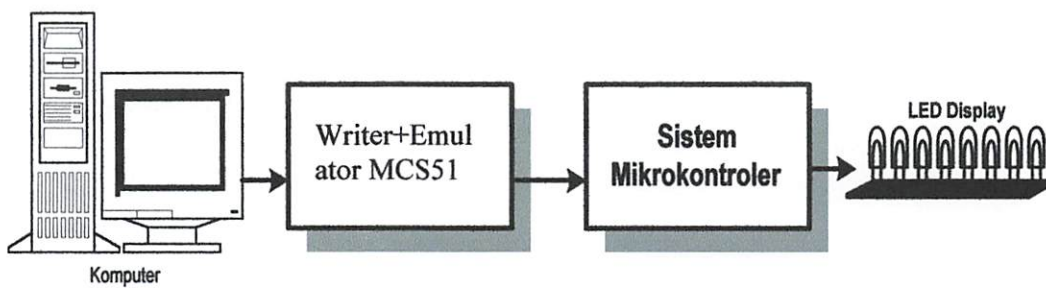
                DJNZ    R1,$
```

DJNZ R3,TUNDA1

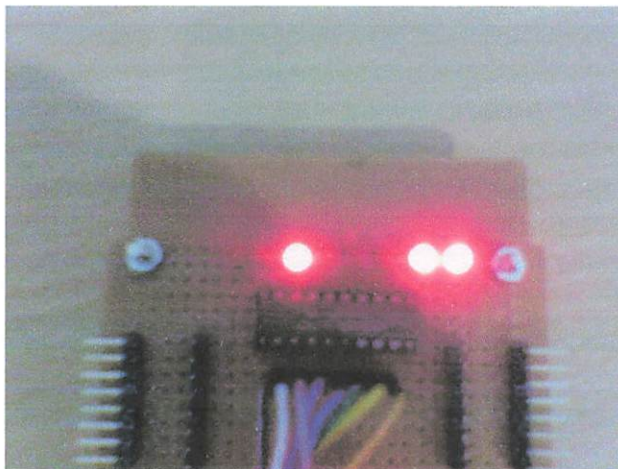
RET

END

2. Rangkaian dibuat seperti Gambar 4.1.
3. Memasang catu rangkaian sebesar 5 volt.
4. Download program diatas .
5. Mengamati keluaran pada LED Display .



Gambar 4.1 Diagram blok Pengujian Mikrokontroler



Gambar 4.2 LED Display Pengujian Mikrokontroler

- Hasil Pengujian

Hasil pengujian ditunjukkan dalam Tabel 4.1.

Tabel 4.1 Hasil Pengujian Sistem Mikrokontroller

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Satu	1	1	1	1	0	0	0	0
Dua	0	0	0	0	1	1	1	1

- Analisis Pengujian

Dari hasil pengujian dalam tabel 4.1 dapat dilihat bahwa *port 1* memberikan logika 0F_H dan F0_H secara bergantian sesuai dengan isi program.

4.2.2 Pengujian Rangkaian Tampilan

- Tujuan

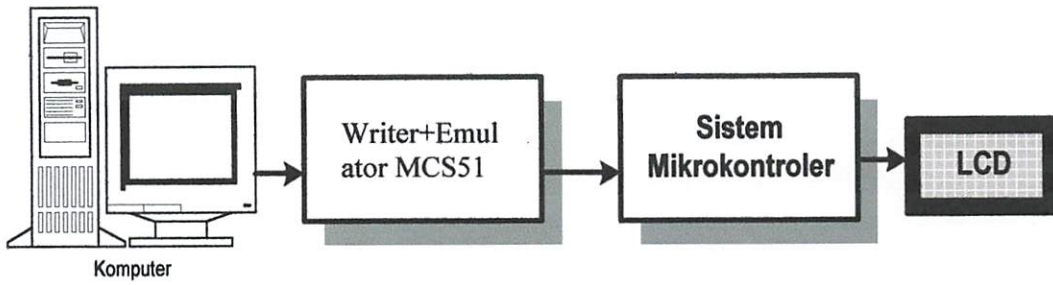
Untuk mengetahui kemampuan rangkaian tampilan yang sudah dibuat apakah dapat mendukung sistem yang direncanakan untuk memampikan data pada LCD.

- Peralatan yang dibutuhkan

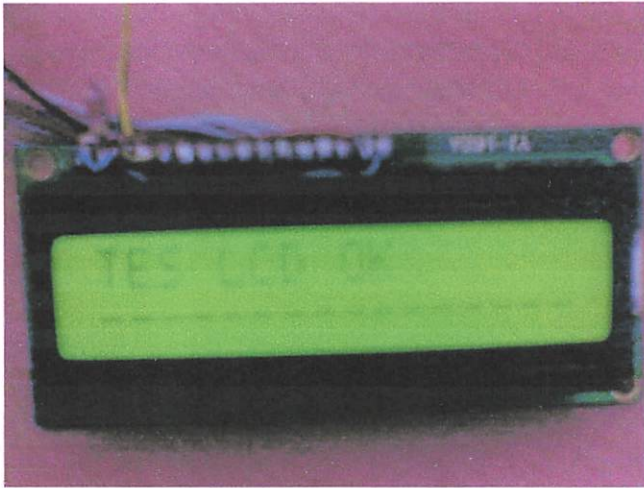
1. Komputer PC AT
2. Writer MCS51
3. Sistem Mikrokontroller dan LCD TM1632A

- Prosedur Pengujian

1. Menyusun rangkaian seperti dalam Gambar 4.3
2. Menjalankan program untuk menampilkan tulisan
3. Mengamati keluaran pada LCD



Gambar 4.3 Diagram blok Pengujian Rangkaian Tampilan



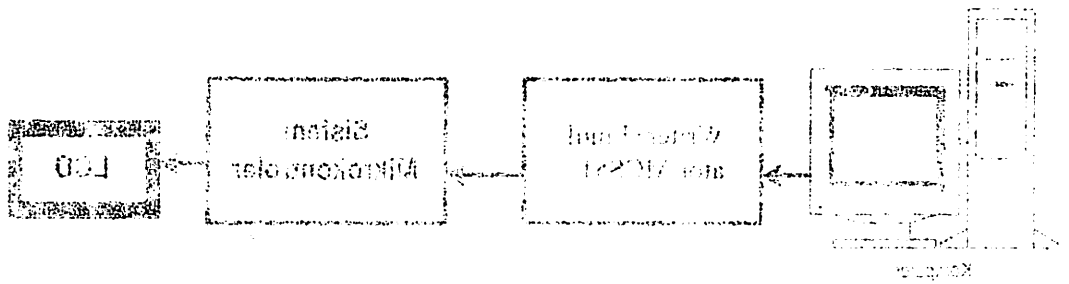
Gambar 4.4 Pengujian Rangkaian Tampilan

- Analisis pengujian

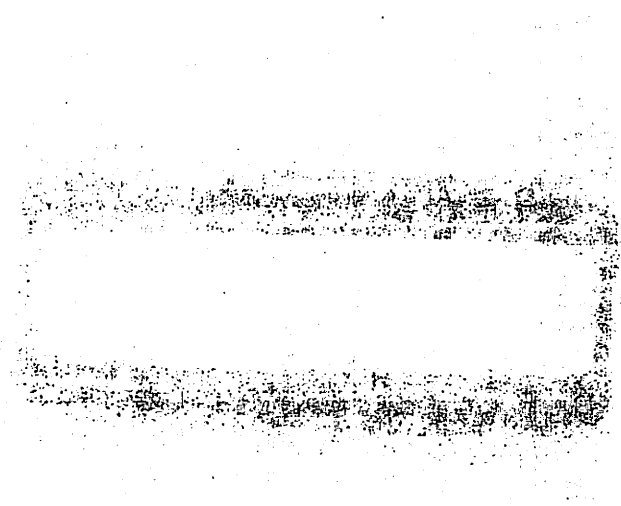
LCD menampilkan tulisan pada deret kolom maupun baris sesuai program yang dibuat

4.2.3 Pengujian Papan Tombol (*keypad*)

Pengujian ini bertujuan untuk mengetahui konfigurasi logika keluaran dari unit papan tombol saat tombol ditekan. Peralatan yang dipergunakan antara lain catu daya, unit papan tombol, dan *logic probe*. Dalam pengujian ini keluaran yang diamati adalah proses *scanning* yang terjadi pada lajur baris dan kolom. Lajur baris merupakan bagian output sedangkan lajur kolom merupakan bagian input.



Gambar 4.3 Diagram blok Program Aplikasi LCD



Gambar 4.4 Diagram Alir Program Aplikasi LCD

• Analisis perbandingan

LCD menggunakan tulisan pada layar untuk menampilkan hasil hasil program yang dibuat

4.1.3 Perbandingan Papan Tombol (Keyboard)

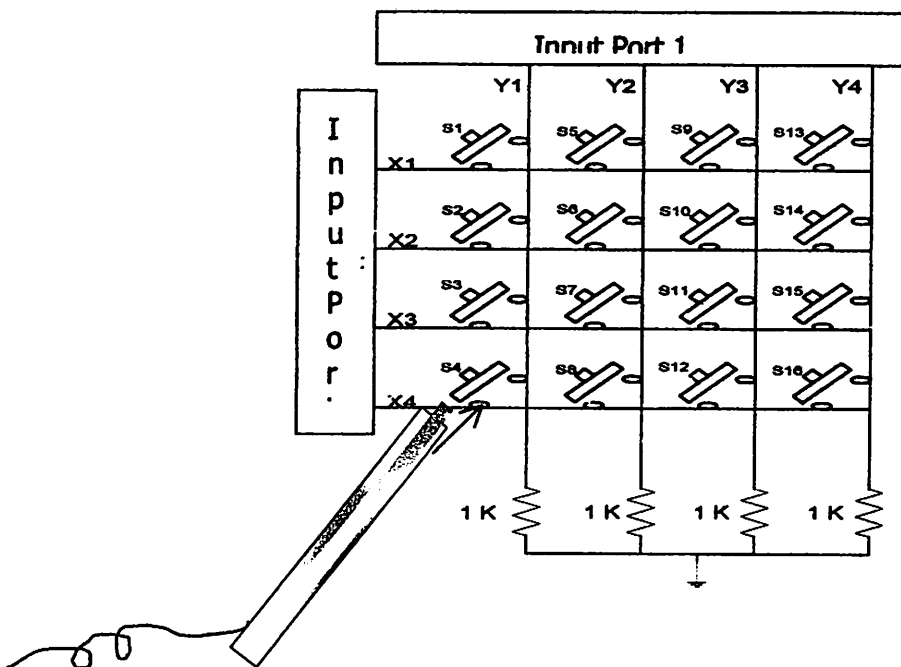
Perbedaan ini bertujuan untuk mengetahui kelebihan dan kekurangan dari unit papan tombol saat ini. Perbandingan yang digunakan antara lain cara kerja unit papan tombol dan keyboard. Pada perbandingan ini keyboard yang dimaksud adalah proses secara umum yang terjadi pada papan tombol. Papan tombol merupakan bagian output sedangkan papan tombol merupakan bagian input.

Untuk mengetahui kebenaran rangkaian keypad yang telah dibuat maka keluaran dari rangkaian keypad ini akan ditampilkan ke port 1 MCU 89S51.

Langkah-langkah pengujian papan tombol adalah sebagai berikut ;

- Menyiapkan catu daya dengan keluaran 5 V DC, rangkaian papan tombol seperti dalam Gambar 4.5 dan *logic probe*.
- Menyusun rangkaian pengujian papan tombol seperti terlihat dalam Gambar 4.1 serta memastikan bahwa hubungan antar pin pada masing-masing saklar telah benar.
- Jalankan program pengujian keypad, amati keluaran pin Port 1 bagian output. Langkah ini dilakukan dalam keadaan tidak ada penekanan tombol sama sekali.
- Tekan sembarang tombol kemudian amati keluaran pin Port 1 bagian output.

Hasil dari pengujian rangkaian papan tombol terdapat dalam Tabel 4.2



Gambar 4.5 Pengujian Rangkaian Papan Tombol

Tabel 4.2 Pengujian Keypad

Posisi Tombol	(V)	Logic Probe
Ditekan	0	Low
Tidak Ditekan	5	High

Tabel 4.3 Hasil Pengujian Papan Tombol.

Input		Output Port 1			
Tombol yang ditekan		D	C	B	A
Nomor tombol	Definisi tombol				
-	-	X	X	X	X
(S1)	1	0	0	0	1
(S2)	4	0	1	0	0
(S3)	7	0	1	1	1
(S4)	*	1	0	1	0
(S5)	2	0	0	1	0
(S6)	5	0	1	0	1
(S7)	8	1	0	0	0
(S8)	0	0	0	0	0
(S9)	3	0	0	1	1
(S10)	6	0	1	1	0
(S11)	9	1	0	0	1
(S12)	#	1	0	1	1

Dari hasil pengujian, didapatkan data seperti dalam Tabel 4.1 maka dapat diketahui bahwa saat tombol ditekan maka keluaran port 1 mikrokontroler 89S51 akan berlogika sesuai dengan tombol yang ditekan. Hasil pengujian dalam Tabel 4.1 terlihat bahwa rangkaian papan tombol yang telah direalisasikan sesuai dengan unjuk kerja perencanaan.

4.2.4 Pengujian Optocoupler

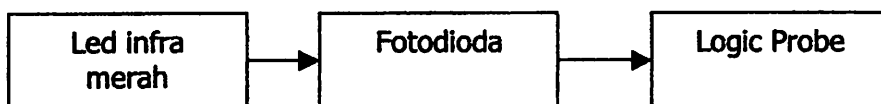
- Tujuan

Untuk mengetahui apakah optocoupler dapat bekerja dengan baik sesuai dengan perencanaan ataukah tidak.

- Peralatan yang digunakan

1. Laser dioda
2. Foto dioda
3. Logic Probe.
4. Catu daya 5 volt.

- Prosedur pengujian



Gambar 4.6 Rangkaian Pengujian Optocoupler

1. Merangkai peralatan yang digunakan sesuai Gambar 4.6.
2. Memberikan catu daya 5 volt pada rangkaian Optocoupler berupa led dan fotodioda.
3. Mengarahkan berkas sinar ke fotodioda pada jarak sekitar 2 cm.
4. Mengamati keluaran logic probe.

5. Berkas sinar ditutup dengan suatu penghalang.
6. Mengamati keluaran logic probe.

- Hasil Pengujian

Hasil pengujian Optocoupler ditunjukkan dalam Tabel 4.3 berikut ini :

Tabel 4.4 Hasil Pengujian Optocoupler

Ada/Tidak Halangan	Keluar Logic Probe	Volt
Ada	Low	0
Tidak	High	5.15



Gambar 4.7 Pengujian pada Optocoupler saat tidak terhalang



Gambar 4.8 Pengujian pada Optocoupler pada Saat terhalang

- Analisis Hasil Pengujian

Berdasarkan hasil pengujian di atas, terlihat bahwa Optocoupler tersebut dapat bekerja sesuai dengan perencanaan.

4.3 Pengujian Sistem Keseluruhan

Setelah melakukan pengujian sistem setiap blok secara terpisah langkah selanjutnya adalah melakukan pengujian sistem secara keseluruhan.

- Tujuan

Mengetahui unjuk kerja sistem secara keseluruhan apakah sesuai dengan rancangan yang dibuat atau tidak.

- Peralatan yang digunakan

1. Rangkaian adaptor tegangan yang diset dengan output 12 V.
2. Sistem yang diuji **SISTEM PENGAMAN MOBIL DENGAN SISTEM *PASSWORD* MENGGUNAKAN *CARD READER* BERBASIS MIKROKONTROLLER AT89S51.**

- Prosedur pengujian

1. Merangkai setiap blok sistem menjadi sistem yang terpadu untuk menjalankan fungsi sesuai rancangan.
2. Mengaktifkan adaptor dengan output tegangan yang telah diset 12V.
3. Menghubungkan sistem dengan adaptor.
4. Mengkondisikan kontak pada posisi ON.
5. Memasang *card* pada slot yang tersedia pada sistem.
6. Menjalankan sistem sesuai Flowchart perangkat lunak.

7. Melakukan pengamatan dan pengujian sistem untuk kemudian dilakukan analisa lebih lanjut.

- Analisis Hasil Pengujian

Setelah dilakukan pengamatan maka diperoleh data sebagai berikut :

- Pengujian Tegangan Input

Tabel 4.5 Pengujian tegangan input

Tegangan Input DC (Volt)	Kondisi sistem
12	Bekerja dengan baik
9	Bekerja dengan baik
7,5	Bekerja dengan baik
6	Bekerja dengan baik, tampilan terganggu
4,5	Bekerja dengan baik, tampilan redup dan terganggu
Dibawah 4,5 Volt	Sistem tidak dapat bekerja

Berdasarkan Tabel 4.4 diatas dapat disimpulkan bahwa alat ini mampu bekerja optimal pada range tegangan antara 4,5 Volt hingga 12 Volt. Input tegangan maksimal 12 Volt disesuaikan dengan tegangan baterai yang pada umumnya digunakan pada kendaraan bermotor.

- Pengujian Penggunaan Card

Card hanya diperlukan apabila prosedur pengamanan diaktifkan, apabila prosedur tersebut dinonaktifkan maka card tidak berfungsi. Setiap prosedur yang

berhubungan dengan password selalu memerlukan card, kondisi yang dimaksud adalah pada saat :

1. Mengaktifkan starter
2. Mengganti PIN

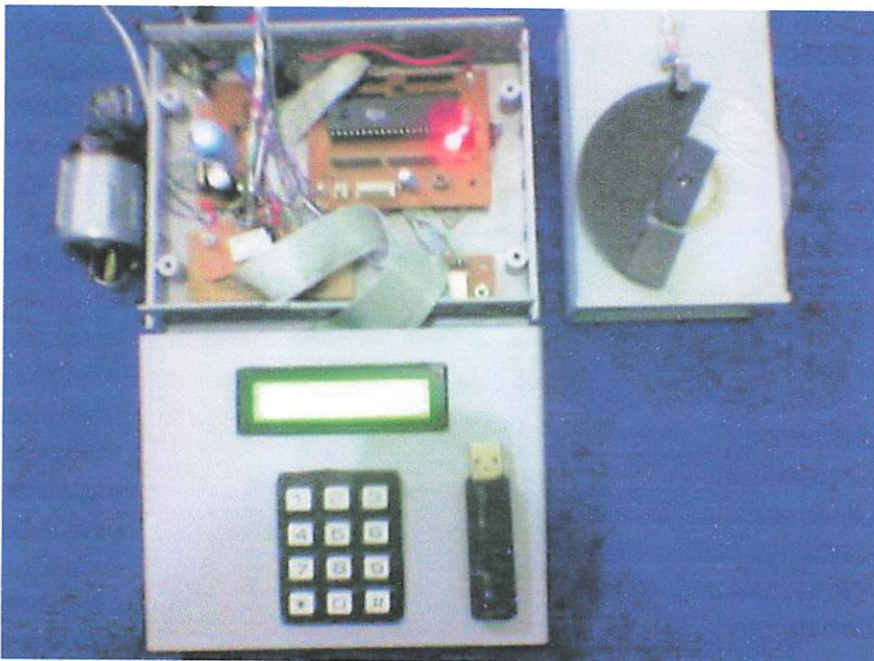
Hal tersebut dikarenakan PIN yang berlaku tersimpan dalam card, sehingga tanpa card prosedur pengamanan tidak dapat dilalui.

- Berdasarkan pengamatan diatas maka disimpulkan alat dapat beroperasi sesuai rencana yang dibuat.

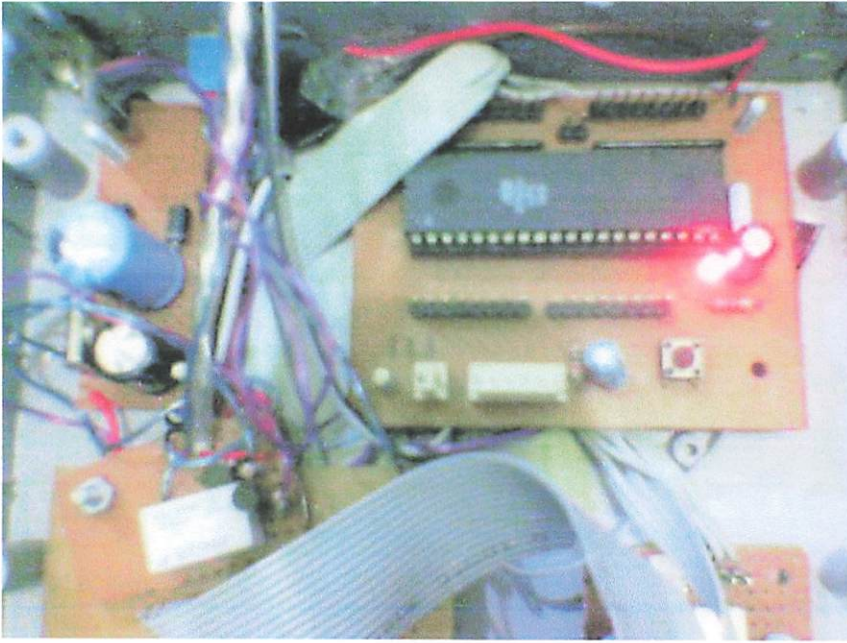
Foto-foto Alat



Gambar 4.9 Alat Nampak Luar



Gambar 4.10 Rangkaian Elektronik sistem



Gambar 4.11 Rangkaian Pemroses Elektronik

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil perancangan dan pembuatan alat dapat disimpulkan sebagai berikut:

1. Alat ini hanya mampu bekerja dengan baik dalam kondisi tegangan *supply* yang stabil yaitu 7,5 Volt hingga 12 Volt.
2. Alat ini menggunakan PIN dengan format angka sebanyak 4 digit, dengan tujuan memperaman sistem *password* dengan banyaknya kombinasi yang dapat dibuat dari 4 angka dan mempermudah pengingatan bagi operator.
3. *Card* memiliki fungsi menyimpan PIN yang digunakan, tanpa *card* prosedur password tidak bisa berfungsi, kecuali digunakan sakelar pengaman yang berfungsi menonaktifkan prosedur pengaman.

3.2 Saran-saran

1. Penggunaan alat ini sebaiknya didukung dengan *powersupply* tambahan yang stabil sehingga sistem dapat bekerja dengan optimal dalam berbagai kondisi, terutama pada saat dilakukan langkah starter yang pada umumnya mempengaruhi kelistrikan.
2. sistem ini dapat digunakan secara terpadu dengan sistem pengaman lain yang pada saat ini telah tersedia, sehingga penggunaanya dapat mengoptimalkan sistem pengaman yang telah ada.

2. sistem ini dapat digunakan secara terpadu dengan sistem pengaman lain yang pada saat ini telah tersedia, sehingga penggunaanya dapat mengoptimalkan sistem pengaman yang telah ada.

DAFTAR PUSTAKA

1. Barry B. Brey, *Mikroprocessor Intel*, ERLANGGA, Jakarta, 2002.
2. Frans Gunterus, *Sistem Pengendalian Proses*, PT. Elex Media Komputindo, Jakarta, 1994.
3. Fendy dkk, *Belajar Mikroprosesor dan Mikrokontroler melalui PC*, PT. Elex Media Komputindo, Jakarta, 1994.
4. Hasan Hasril, *Kamus Komputer Untuk Orang Awam*, Maxikom, Palembang, 2004.
5. <http://www.atmel.com/literature>, AT89S51 Data Sheets, 2005.
6. I Erwien, *Dasar Elektronika*, Laporan Praktikum, Malang, 2003.
7. Kelompok Gramedia, *Data Praktis Elektronika*, PT. Elex Media Komputindo, Jakarta, 1995,
8. Malvino, Albert Paul PHD, *Prinsip-prinsip Elektronika*, edisi ketiga alih bahasa: Hanapi Gunawan, Penerbit Erlangga, Jakarta, 1996.
9. Roger L. Tokheim, *Elektronika Digital*, ERLANGGA, Jakarta, 1995.
10. Tomkins J.G, *Interfacing Sensor to The IBM PC*, Prentice Hall International Inc., London, 1988.

DAFTAR PUSTAKA

1. Barry B. Bryer, Mikroprosesor dan HIRVA/OGA, Jakarta 2002.
2. Frans Gunawan, Sistem Pengolahan Teks, PT. Elz Media Komputindo, Jakarta, 1994.
3. Randy Eric, Belajar Mikroprosesor dan Mikrokomputer melalui PC, PT. Elz Media Komputindo, Jakarta, 1994.
4. Hasan Hauri, Kawan Komputer Untuk Orang Maziqon Palembang, 2004.
5. [http://www.smbc.com/learn/AT8251/DataSheet](http://www.smbc.com/learn/AT8251>DataSheet), 2002.
6. I Fawzan, Dasar Elektronika Laporan Praktikum, Malang, 2003.
7. Kelompok Ormedia, Data Teknik Elektronika, PT. Elz Media Komputindo, Jakarta, 1995.
8. Marino, Albert Paul PhD, Prinsip-prinsip Elektronika edisi ketiga jilid bahasa Inggris, Gunung Cendekia, Jakarta, 1996.
9. Roger L. Tokheim, Elektronika Digital, ERI/ANDGA, Jakarta, 1995.
10. Tomkins J.G. Inventing Saver to The IBM PC, Prentice Hall International Inc, London, 1988.

LAMPIRAN



Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Elektro S-1
Konsentrasi Elektronika

LEMBAR BIMBINGAN SKRIPSI

Nama : R. Ervien Indiarso
NIM : 0117163
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perencanaan dan Pembuatan Sistem Pengaman
Mobil dengan Sistem Password Menggunakan
Card Reader Berbasis Mikrokontroler AT89S51.
Tanggal Pengajuan Skripsi : 29 Juli 2005
Selesai Penulisan Skripsi : 10 Maret 2006
Telah Dievaluasi dengan nilai : 88 (A)

Mengetahui
Ketua Jurusan Teknik Elektro S-1













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

Diperiksa dan Disetujui
Dosen Pembimbing

Ir. Eko Nurcahyo
NIP.Y. 1028700172

FORMULIR BIMBINGAN SKRIPSI

Nama : Ervien Indiarto
 Nim : 0117163
 Masa Bimbingan : 15-September-2005 s/d 18-Maret-2006
 Judul Skripsi : Perancangan dan Pembuatan Sistem Pengaman Starter Mobil dengan Sistem Password Menggunakan Card Reader Berbasis Mikrokontroler AT89S51

NO	Tanggal	Uraian	Paraf Pembimbing
1	18-9-2005	ACC BAB I PENDAHULUAN	
2	18-9-2005	ACC BAB II LANDASAN TEORI	
3	10-11-2005	KONSULTASI BAB III PRENCANAAN & PEMBUATAN ALAT	
4	12-11-2005	REVISI PERHITUNGAN OPTOCOUPLER	
5	25-11-2005	REVISI GAMBAR DIUBAH KE FOTO	
6	3-12-2005	ACC BAB III	
7	5-12-2005	KONSULTASI IV PENGUJIAN DAN ANALISIS	
8	6-12-2005	REVISI FOTO, PENGUKURAN SETIAP BAGIAN	
9	10-12-2005	ACC BAB IV DAN BAB V	
8	3-1-2006	ACC SEMINAR HASIL	
9	5-3-2006	REVISI FLOWCHART,GAMBAR, PENGUKURAN, KESIMPULAN	
10	10-3-2006	ACC MAJU UJIAN SKRIPSI	

Malang, 18 Maret 2006
 Dosen Pembimbing


 Ir Eko Nurcahyo



Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Elektro S-1
Konsentrasi Elektronika

FORMULIR PERBAIKAN SKRIPSI

Nama : R. Erwien Indiarto
NIM : 0117163
Judul : Perencanaan dan Pembuatan Sistem Pengaman Mobil dengan Sistem Password Menggunakan Card Reader Berbasis Mikrokontroler AT89S51.

No	Tanggal	Uraian	Paraf
1	22-03-2006	• Memperbaiki Diagram Blok sensor	
2	22-03-2006	• Merubah keyboard menjadi keypad	
3	22-03-2006	• Menambah gambar rangkaian lengkap dengan Buzzer	
4	22-03-2006	• Merangkai Gambar	
5	22-03-2006	• Panah FlowChart	

Mengetahui

Dosen Pembimbing

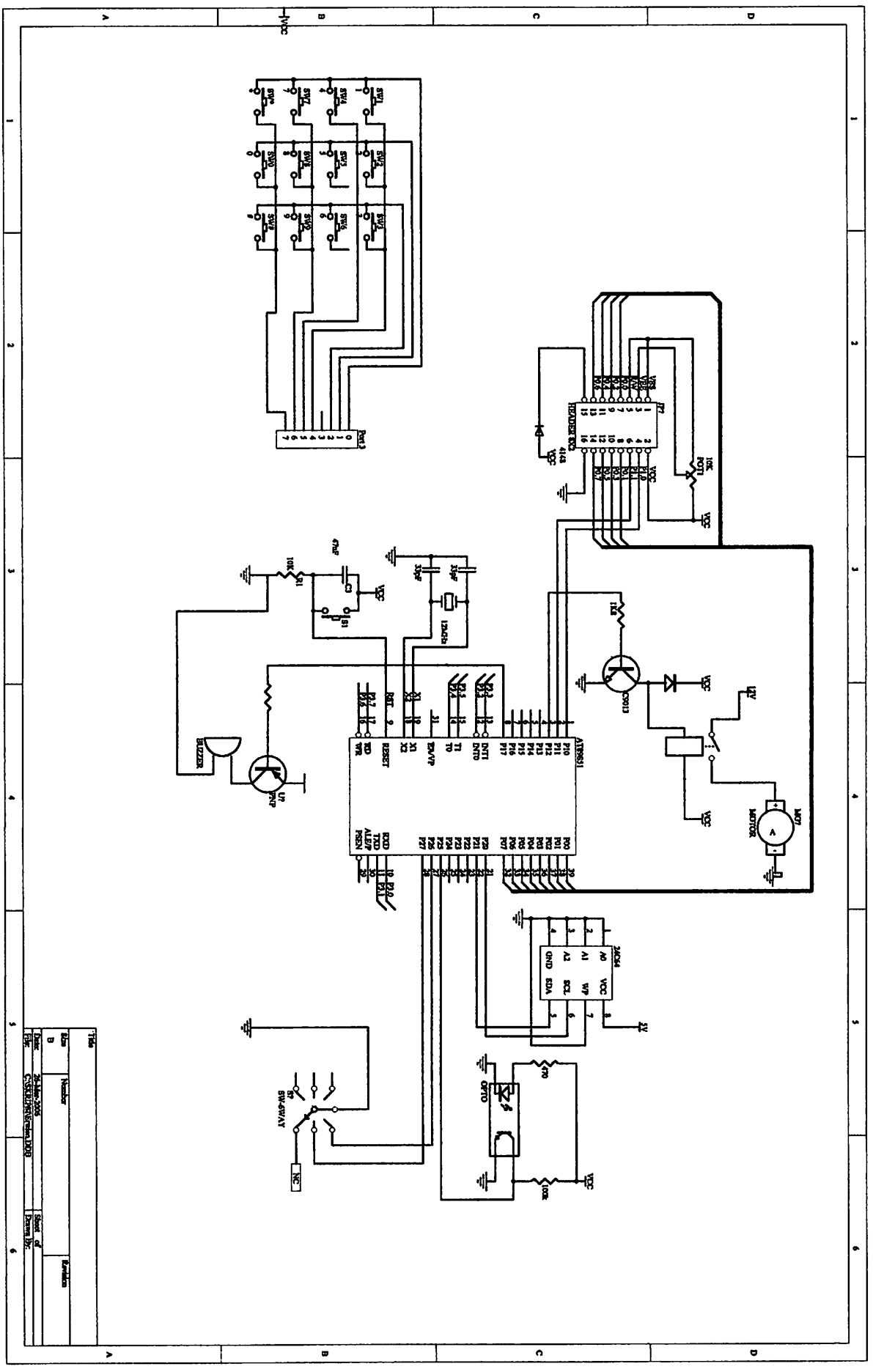
(Ir. Eko Nurcahyo)
NIP.Y 1028700172

Penguji I

(Ir. Usman Djuanda, MM)
NIP.Y 1018700143

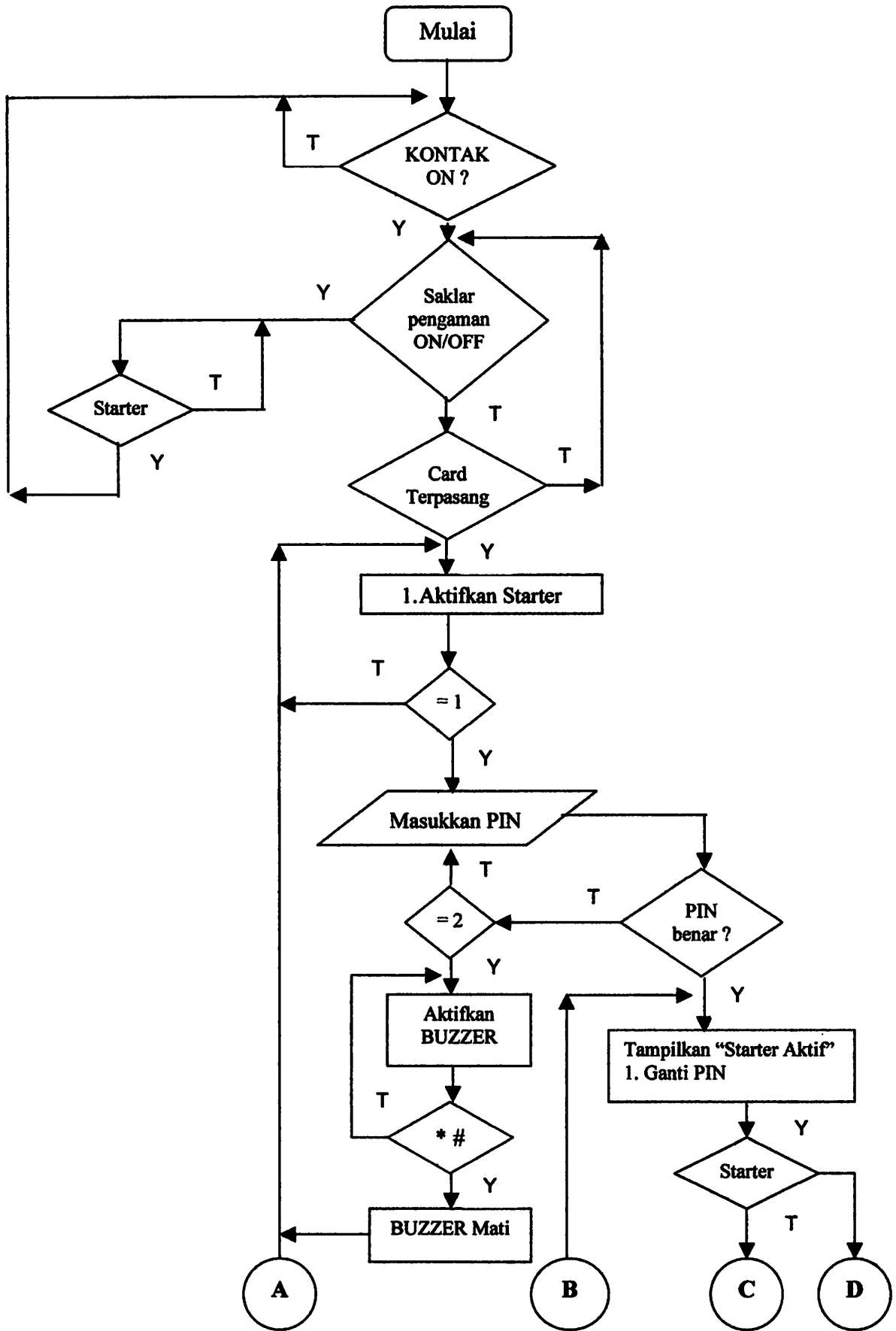
Penguji II

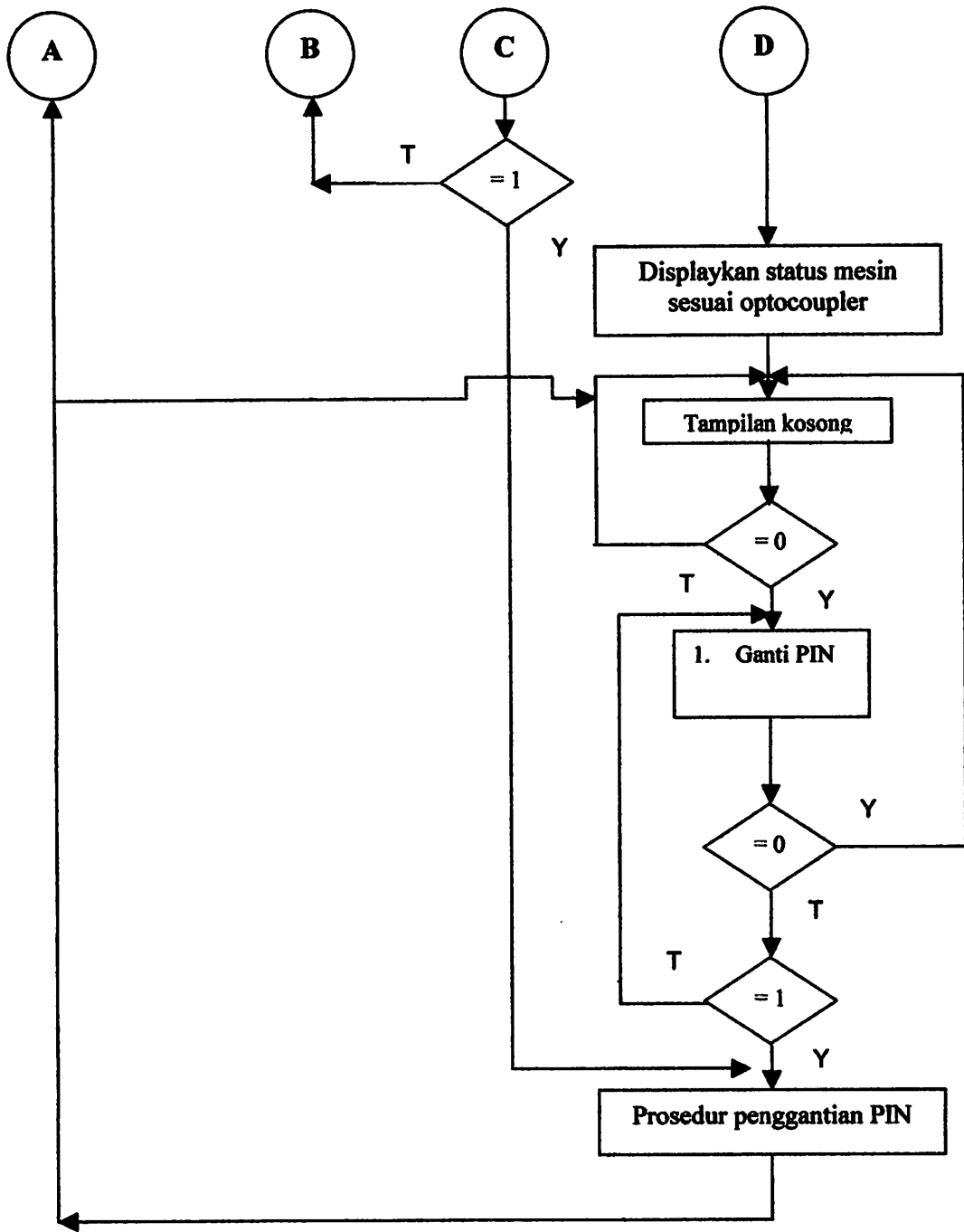
(Ir. Mimin Mustikawati)
NIP.Y.1030000352



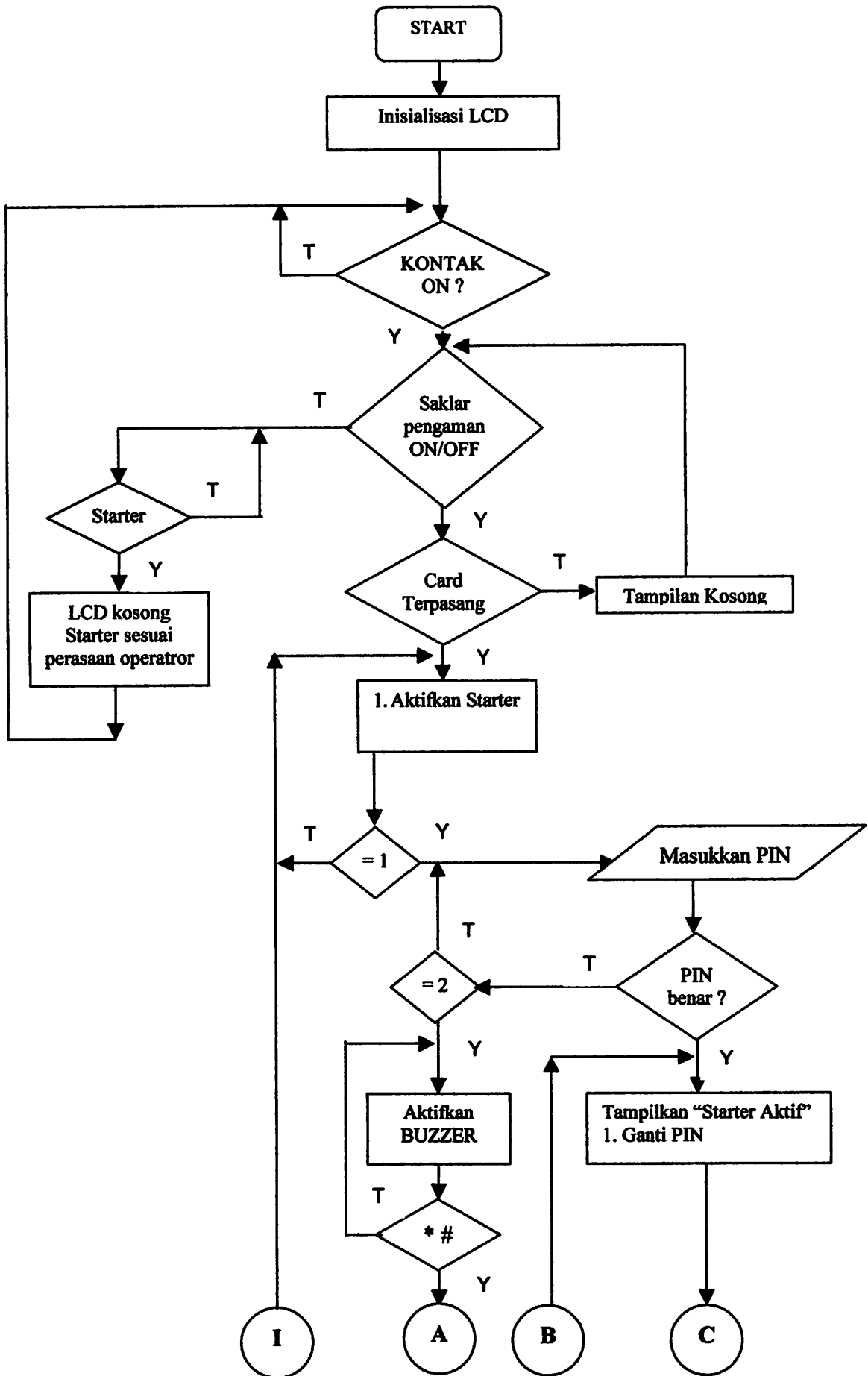
Title		Number		Revision	
B		20		1	
Name		Author		Date	
Circuit		Circuit		Date	

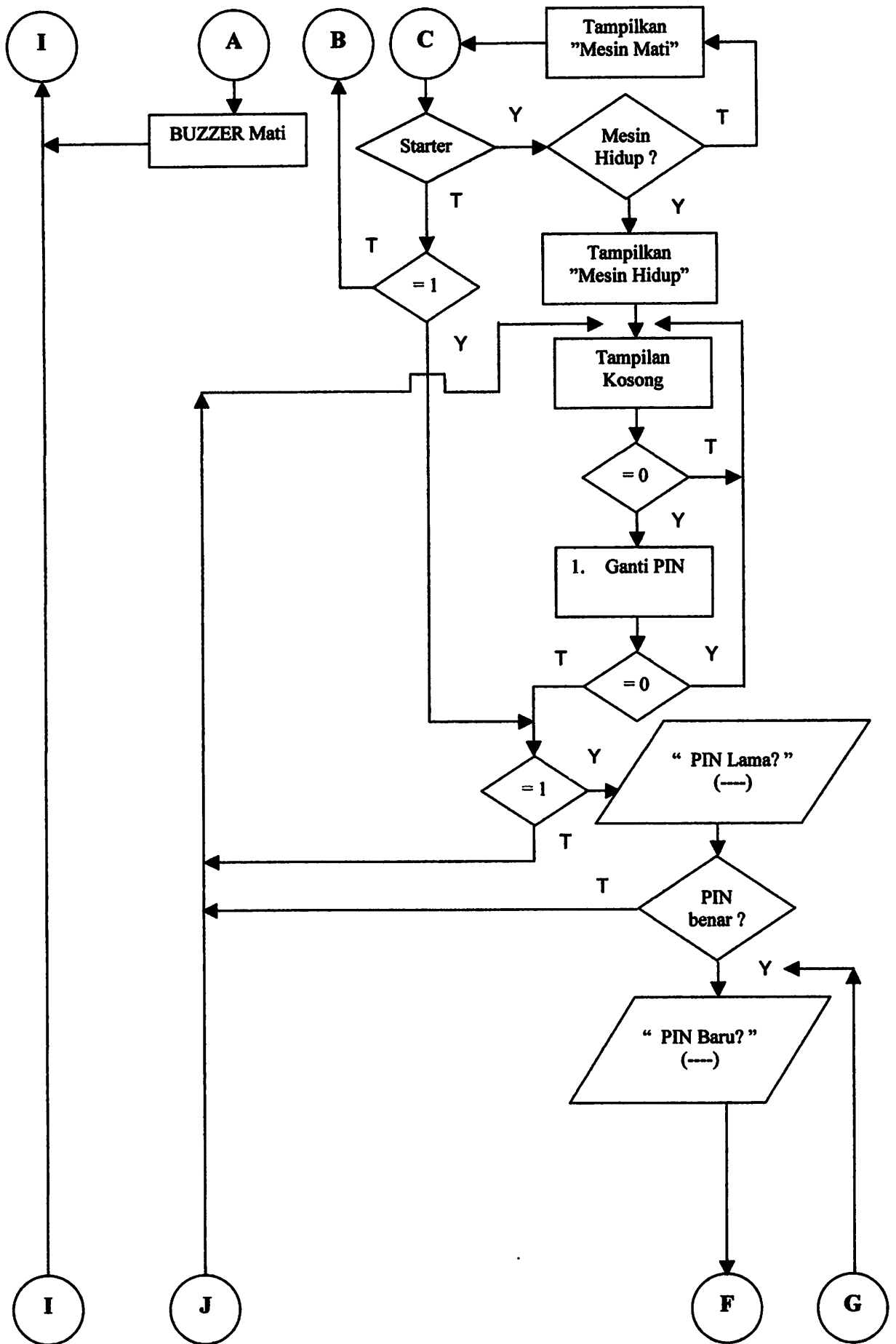
Flowchart Operasi sistem

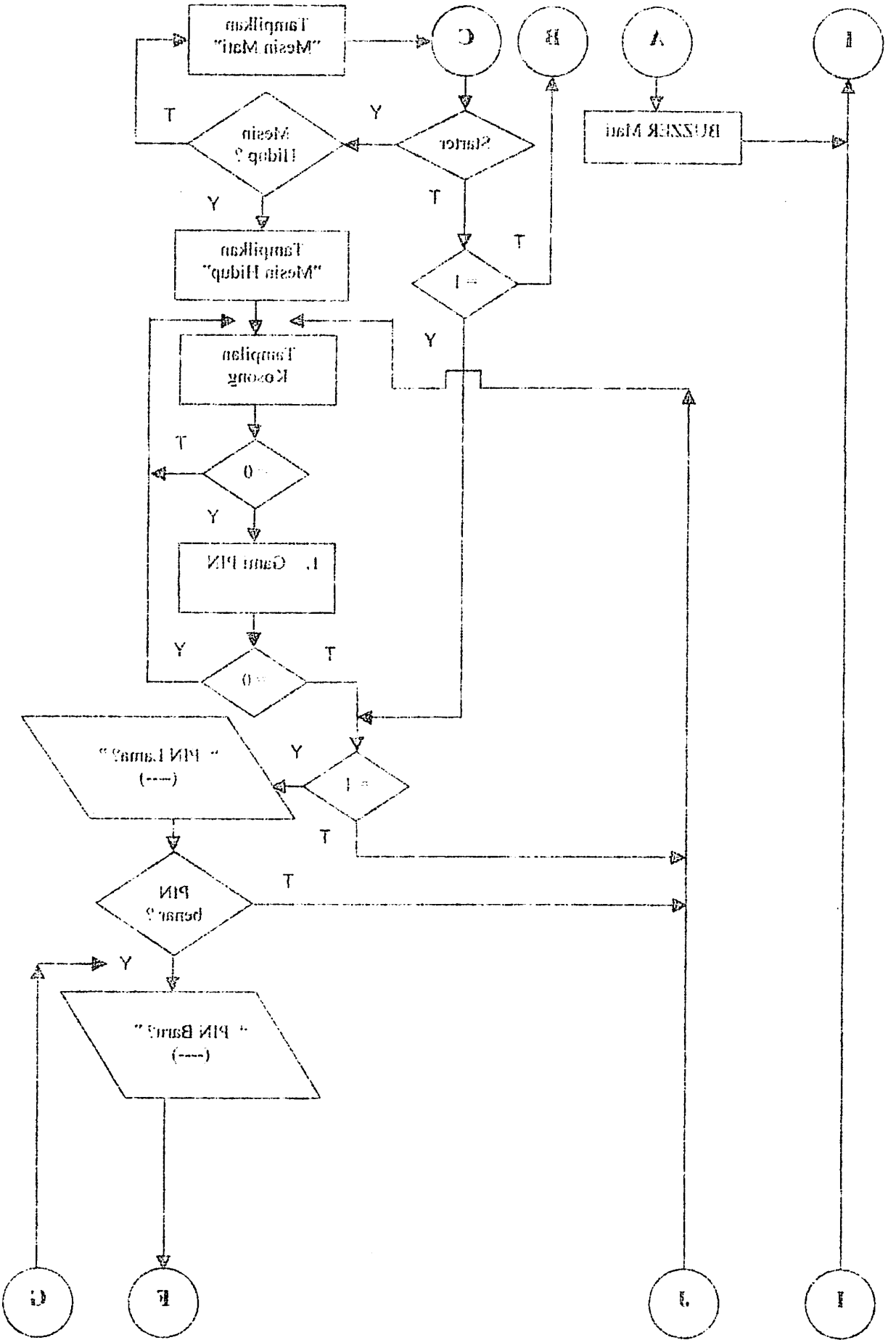


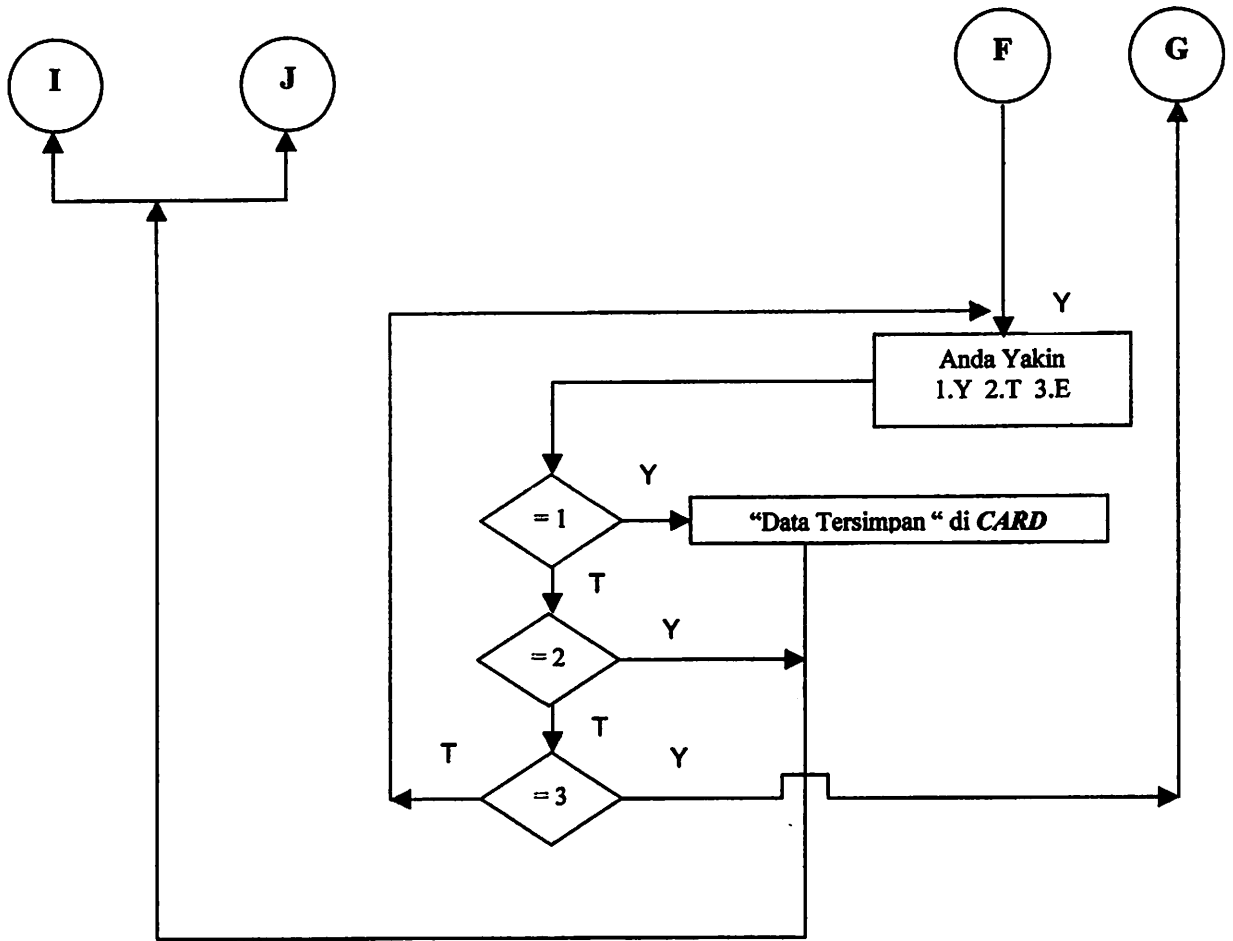


Flowchart Perangkat Lunak









```

kolom1      Bit      P3.4      ; kiri (1,4,7,redial)
kolom2      Bit      P3.5      ;      (2,5,8,0)
kolom3      Bit      P3.6      ;      (3,3,9,#
kolom4      Bit      P3.7

baris1      Bit      P3.0      ; atas (1,2,3)
baris2      Bit      P3.1      ;      (4,5,6)
baris3      Bit      P3.2      ;      (7,8,9)
baris4      Bit      P3.3      ;      (*,0,#)
;
tmp_kesalahan Data    33h
keyport     Data    P3
keydata     Data    15h
FlagKey     Data    16h
FlagKey_matrix Data  17h
CntPulse    Data    30h
cntdata     Data    31h
cntdown     Data    32h

Slave_AddrSEE Data    1
DataSEE     Data    1
SDA         Bit      P2.0      ; pin 5
SCL         Bit      P2.1      ; pin 6

Sw_rahasia  Bit      P1.0
Kunci       Bit      P1.1
starter     Bit      P1.2

sensor_P    Bit      P1.3
aktif_sistem Bit    P1.4

Buzer       Bit      P1.5
relayMTR    Bit      P1.6

Pin1 Equ    0100h

    Mov      SP,#40h
    Call     LCD_Inisialisasi
    Clr      EA
    Clr      relaymtr

start:
    Call     LCD_Clear
    Call     lcd_line_1
    Mov      DPTR,#txt_blank
    Call     LCD_TampilKata
    Call     LCD_Line_2
    Mov      DPTR,#txt_blank
    Call     LCD_TampilKata
    Call     Delay_Fix_1s

start1:
    Clr      relayMTR
    Jb       Sw_rahasia,Cek_Sistem

```

```

Call    Delay_Fix_100ms
Jb      Kunci,start
Call    Delay_Fix_100ms
Jb      aktif_sistem,start
Call    Delay_Fix_100ms
Mov     tmp_kesalahan,#2

mulai:
Call    LCD_Clear
Call    lcd_line_1
Mov     DPTR,#txt_menu_utama
Call    LCD_TampilKata
Call    LCD_Line_2
Mov     DPTR,#txt_menu_utama1
Call    LCD_TampilKata
;      Sjmp     $
Pilih:
Jb      aktif_sistem,start
Jb      Kunci,start
Mov     FlagKey_matrix,#0
Call    key_matrix
Cjne   A,#'1',pilih1
Jmp     Buka_Pintu

Pilih1:
Cjne   A,#'2',Pilih
Jmp     Ganti_Pin
Cek_sistem:
Jmp     Sistem_nonAktif
;=====
; Menu Buka
;=====
Buka_Pintu:
Call    LCD_Clear
Call    lcd_line_1
Mov     DPTR,#txt_menu_adm
Call    LCD_TampilKata
Call    LCD_Line_2
Mov     DPTR,#txt_menu_adm1
Call    LCD_TampilKata
Call    Delay_Fix_100ms

Masuk_Psw1:

Mov     R0,#25h
Call    LCD_Cursor_Position

;Mov     R1,#'*'      ; karakter password = '*'
Mov     FlagKey,#1      ; Mode tampilan karakter.
Call    Keypad_KeyIn
Mov     20h,A          ; 0 - 7fh

Mov     FlagKey,#1
Call    Keypad_KeyIn

```

```

Mov     21h,A

Mov     FlagKey,#1
Call   Keypad_KeyIn
Mov     22h,A

Mov     FlagKey,#1
Call   Keypad_KeyIn
Mov     23h,A

Call   Delay_Fix_100ms

```

Banding_mem1:

```

Mov     DPTR,#Pin1
Call   EEPROM_Read
Cjne   A,20h,baca_error

Inc     DPTR
Call   EEPROM_Read
Cjne   A,21h,baca_error

Inc     DPTR
Call   EEPROM_Read
Cjne   A,22h,baca_error

Inc     DPTR
Call   EEPROM_Read
Cjne   A,23h,baca_error

Jump   Pin_Benar

```

Baca_Error:

```

Djnz   tmp_kesalahan,Baca_error1
Mov     cntdown,#3

```

Baca_Error1a:

```

Clr     Buzer
Call   LCD_Clear
Call   lcd_line_1
Mov     DPTR,#txt_error
Call   LCD_TampilKata
Call   Delay_Fix_1s
Call   Delay_Fix_1s
Call   Keypad_KeyIn
Cjne   A,'#*',Baca_Error1a
Call   Delay_Fix_100ms
Call   Delay_Fix_100ms
Call   key_matrix
Cjne   A,'#',Baca_Error1a
Setb   Buzer
Jump   start

```

Baca_error1:

```

    Call    LCD_Clear
    Call    lcd_line_1
    Mov     DPTR,#txt_error
    Call    LCD_TampilKata
    Call    Delay_Fix_1s
    Call    Delay_Fix_1s
    Jb     Kunci,dddd
    Jmp     Buka_Pintu
dddd:
    Jmp     start
;=====
Pin_Benar:
    Call    LCD_Clear
    Call    lcd_line_1
    Mov     DPTR,#txt_benar1
    Call    LCD_TampilKata
    Call    LCD_Line_2
    Mov     DPTR,#txt_starter
    Call    LCD_TampilKata
    Call    Delay_Fix_100ms
    Call    Delay_Fix_1s
    Call    Delay_Fix_1s
    ;Call   Delay_Fix_1s
Cek_Kunci:
    Jb     Sw_rahasia,start_jauh
    Jb     Kunci,start_jauh
    Call    Delay_Fix_100ms
    Jb     starter,Cek_Kunci

    Setb   relayMTR
    Call    Delay_Fix_100ms

    Call    Cek_Mesin

    Mov     A,cntdata
    Cjne   A,#0Fh,Cek_kuncil
    ;motor hidup
    Call    LCD_Clear
    Call    lcd_line_1
    Mov     DPTR,#txt_motor_on
    Call    LCD_TampilKata
    Call    Delay_Fix_1s
    Jmp     Cek_Menu

Cek_kuncil:
    Cjne   A,#00h,Cek_Kunci
    ;motor mati
    Call    LCD_Clear
    Call    lcd_line_1
    Mov     DPTR,#txt_motor_off
    Call    LCD_TampilKata
    Call    Delay_Fix_1s

```

```

        Jmp      Cek_Kunci
start_jauh:
        Jmp      start
Cek_menu:
        Jmp      Ubah_Pin1

cek_mesin_off:
        Call     Cek_Mesin
        Mov      A,cntdata
        Cjne     A,#00h,Cek_menu
        ;Mov     30h,#55h
        Jmp      Cek_Kunci

Sistem_nonAktif:
        Jnb     Sw_rahasia,start_jauh

        Call     LCD_Clear

        Jb      Kunci,start_jauh
        Call     Delay_Fix_100ms
        Jb      starter,Sistem_nonAktif

        Setb    relayMTR
        Call     Delay_Fix_100ms
        Jmp     Sistem_nonAktif

Ubah_Pin1:
        Call     LCD_Clear
Ubah:
        Jb      Kunci,start_jauh
        Mov     FlagKey_matrix,#2
        Call     key_matrix
        Cjne     A,'#0',Ubah_Pin1

        ;Jmp     Ubah_Pin1x
MasukMenu:
        Call     LCD_Clear

        Call     lcd_line_1
        Mov     DPTR,#txt_menu_adm2
        Call     LCD_TampilKata
        Call     LCD_Line_2
        Mov     DPTR,#txt_menu_adm3
        Call     LCD_TampilKata
        Call     Delay_Fix_100ms
Ubah_Pin1x:
        Jb      Kunci,start_jauh
        Mov     FlagKey,#2
        Call     Keypad_KeyIn
Pilih2:
        Cjne     A,'#1',Pilih3
        Jmp     Masukkan_Pin1

```



```

Pilih3:
    Cjne    A,#'2',pilih4
    Jmp     Lihat_Pin

pilih4:
    Cjne    A,#'0',Pilih5

Pilih5:
    Jmp     Ubah_Pin1

;=====admin ubah pin1=====
Lihat_Pin:
    Call    LCD_Clear
    Call    lcd_line_1
    Mov     DPTR,#txt_lihat
    Call    LCD_TampilKata
    Call    LCD_Line_2
    Mov     DPTR,#txt_menu_adm1
    Call    LCD_TampilKata
    Call    Delay_Fix_100ms

    Mov     R0,#25h
    Call    LCD_Cursor_Position

    Mov     DPTR,#Pin1
    Call    EEPROM_Read
    Call    LCD_Tulis_Data

    Inc     DPTR
    Call    EEPROM_Read
    Call    LCD_Tulis_Data

    Inc     DPTR
    Call    EEPROM_Read
    Call    LCD_Tulis_Data

    Inc     DPTR
    Call    EEPROM_Read
    Call    LCD_Tulis_Data

    Call    Delay_Fix_1s
    Call    Delay_Fix_1s

    Jmp     Cek_menu

Masukkan_Pin1:
    ;Call    LCD_Initialization
    Call    LCD_Clear

    Call    lcd_line_1
    Mov     DPTR,#txt_menu_adm
    Call    LCD_TampilKata

```

```

Call    LCD_Line_2
Mov     DPTR,#txt_menu_adm1
Call    LCD_TampilKata
Call    Delay_Fix_100ms

Mov     R0,#25h
Call    LCD_Cursor_Position
ambil1:
;Mov    R1,'#*'      ; karakter password = '*'
Mov     FlagKey,#1   ; Mode tampilan karakter.
Call    Keypad_KeyIn
Mov     20h,A        ; 0 - 7fh

Mov     FlagKey,#1
Call    Keypad_KeyIn
Mov     21h,A

Mov     FlagKey,#1
Call    Keypad_KeyIn
Mov     22h,A

Mov     FlagKey,#1
Call    Keypad_KeyIn
Mov     23h,A

Call    LCD_Clear

Call    lcd_line_1
Mov     DPTR,#txt_keputusan
Call    LCD_TampilKata
Call    LCD_Line_2
Mov     DPTR,#txt_keputusan1
Call    LCD_TampilKata

Pilih2a:
Mov     FlagKey,#2
Call    Keypad_KeyIn
Cjne   A,'#1',Pilih3a
Jump   simpan_mem1

Pilih3a:
Cjne   A,'#2',Pilih4a
Jump   Ubah_Pin1

Pilih4a:
Cjne   A,'#3',Pilih5a
Jump   Masukkan_Pin1

Pilih5a:
Jump   Pilih2a

Ganti_Pin:
Call    LCD_Clear

Call    lcd_line_1
Mov     DPTR,#txt_menu_adm

```

```

    Call    LCD_TampilKata
    Call    LCD_Line_2
    Mov     DPTR,#txt_menu_adm1
    Call    LCD_TampilKata
    Call    Delay_Fix_100ms

    Mov     R0,#25h
    Call    LCD_Cursor_Position
gantil:
    ;Mov     R1,'#*'      ; karakter password = '*'
    Mov     FlagKey,#1      ; Mode tampilan karakter.
    Call    Keypad_KeyIn
    Mov     20h,A          ; 0 - 7fh

    Mov     FlagKey,#1
    Call    Keypad_KeyIn
    Mov     21h,A

    Mov     FlagKey,#1
    Call    Keypad_KeyIn
    Mov     22h,A

    Mov     FlagKey,#1
    Call    Keypad_KeyIn
    Mov     23h,A
    Call    LCD_Clear

    Call    lcd_line_1
    Mov     DPTR,#txt_keputusan
    Call    LCD_TampilKata
    Call    LCD_Line_2
    Mov     DPTR,#txt_keputusan1
    Call    LCD_TampilKata
Pilih2ax:
    Mov     FlagKey,#2
    Call    Keypad_KeyIn
    Cjne   A,'#1',Pilih3ax
    Jmp     simpan_mem2
Pilih3ax:
    Cjne   A,'#2',Pilih4ax
    Jmp     start
Pilih4ax:
    Cjne   A,'#3',Pilih5ax
    Jmp     Ganti_Pin
Pilih5ax:
    Jmp     Pilih2ax

simpan_mem1:
    Mov     DPTR,#Pin1
    Mov     A,20h
    Call    EEPROM_Save

```

```

Inc    DPTR
Mov    A,21h
Call   EEPROM_Save

```

```

Inc    DPTR
Mov    A,22h
Call   EEPROM_Save

```

```

Inc    DPTR
Mov    A,23h
Call   EEPROM_Save

```

```

Call   LCD_Clear
Call   lcd_line_1
Mov    DPTR,#txt_data
Call   LCD_TampilKata
Call   Delay_Fix_1s

```

```

Jump   Ubah_Pin1

```

simpan_mem2:

```

Mov    DPTR,#Pin1
Mov    A,20h
Call   EEPROM_Save

```

```

Inc    DPTR
Mov    A,21h
Call   EEPROM_Save

```

```

Inc    DPTR
Mov    A,22h
Call   EEPROM_Save

```

```

Inc    DPTR
Mov    A,23h
Call   EEPROM_Save

```

```

Call   LCD_Clear
Call   lcd_line_1
Mov    DPTR,#txt_data
Call   LCD_TampilKata
Call   Delay_Fix_1s

```

```

Jump   start

```

Subroutine Cek_Mesin:

```

Mov    cntdown,#10
Mov    CntPulse,#00h

```

CekWaktulx:

```

Jnb    sensor_P,CekWaktul
Call   Delay_Fix_10ms

```

CekWaktu2x:

```

    Jb      sensor_P,CekWaktul
    Nop
    Inc     CntPulse
    ;Mov    A,CntPulse
    ;Call   LCD_Hexa
    ;Sjmp   $

```

CekWaktul:

```

    Mov     A,CntPulse
    Cjne   A,#01h,BandingLagi
    Mov     cntdata,#0Fh
    Jmp    CekWaktulQuit

```

BandingLagi:

```

    Jc     CekWaktuKecil
    Mov     cntdata,#0Fh
    Jmp    CekWaktulQuit

```

CekWaktuKecil:

```

    Mov     cntdata,#00h
    ; Ret

```

CekWaktulQuit:

```

    Djnz   cntdown,CekWaktulx
    Ret

```

CekWaktu2:

```

;    Inc     CntPulse
    Call   Delay_Fix_100ms
    Mov     A,CntPulse
    Cjne   A,#03h,BandingLagi2
    Mov     cntdata,#0Fh
    Jmp    CekWaktu2Quit

```

BandingLagi2:

```

    Jc     CekWaktuKecil2
    Mov     cntdata,#0Fh
    Jmp    CekWaktu2Quit

```

CekWaktuKecil2:

```

    Mov     cntdata,#00h
    ; Ret

```

CekWaktu2Quit:

```

    Djnz   cntdown,CekWaktu2x
    Ret

```

EndSub

Subroutine EEPROM_Save:

```

    Push   DPH
    Push   DPL

    Mov     DataSEE,A
    Call   TulisDPTRSEE16b
    Mov     A,DataSEE

```

```

        Pop        DPL
        Pop        DPH
EndSub
Subroutine EEPROM_Read:
    Push        DPH
    Push        DPL

    Mov        R7, #0A0h
    Mov        B, DPH
    Mov        R6, DPL
    Call       Baca_SEE16b

    Pop        DPL
    Pop        DPH
EndSub
Subroutine Tulis_SEE16b:
    Call       Siapkan16bAlamatSEE
    Jc        Wrong_Write
    Mov        A, DataSEE
    Call       KirimDataSEE
    Jc        Wrong_Write
    Call       Buat_StopBit
    Ret
Wrong_Write:
    Call       Buat_StopBit
    Clr        C
    Jmp       Tulis_SEE16b
EndSub
Subroutine Tulis_SEE8b:
    Call       Siapkan8bAlamatSEE
    Jc        Wrong_Write8b
    Mov        A, DataSEE
    Call       KirimDataSEE
    Jc        Wrong_Write8b
    Call       Buat_StopBit
    Ret
Wrong_Write8b:
    Call       Buat_StopBit
    Clr        C
    Jmp       Tulis_SEE8b
EndSub
Subroutine Baca_SEE16b:
    Call       Siapkan16bAlamatSEE
    Jc        Wrong_Read
    Call       Buat_StartBit                ;Kirim Device Address de
ngan
    Call       ModeBacaSEE
    Jc        Wrong_Read
    Call       BacaDataSEE
    Ret
Wrong_Read:
    Call       Buat_StopBit

```

```

        Clr      C
        Ajmp    Baca_SEE16b
EndSub
Subroutine Ambil_Ack:

        Setb   SDA
        Setb   SCL
        Mov    C, SDA
        Clr    SCL
EndSub
Subroutine KirimDeviceAddress:
        Call   Buat_StartBit
        Push   ACC
        Call   KirimDataSEE      ;
        Pop    ACC                ;
EndSub
Subroutine Kirim1WordAddress:
        Push   ACC                ;
        Mov    A, B                ;First Word Address
        Call   KirimDataSEE      ;
        Pop    ACC                ;
EndSub
Subroutine Kirim2WordAddress:
        Push   ACC
        Call   KirimDataSEE      ;
        Pop    ACC
EndSub
Subroutine BacaDataSEE:
        Push   B
        Mov    B, #08h
        Clr    A
LoopBacaSEE16b:
        Push   B
        Rl     A
        Setb   SDA
        Setb   SCL
        Mov    ACC.0, C
        Clr    SCL
        Pop    B
        Djnz   B, LoopBacaSEE16b
        Call   Ambil_Ack
        Call   Ambil_Ack
        Call   Buat_StopBit
        Pop    B
EndSub
Subroutine KirimDataSEE:
        Push   B
        Mov    B, #8
Send8_bitloop:
        Mov    SDA, C
        Call   Pulse_SEE
        Djnz   B, Send8_bitloop
        Pop    B

```

```

        Clr      C
        Call     Ambil_Ack
EndSub
Subroutine Pulse_SEE:
    Push      B
    Setb     SCL
    Clr      SCL
    Pop      B
EndSub
Subroutine Buat_StartBit:
    Setb     SDA
    Setb     SCL
    Clr     SDA
    Clr     SCL
EndSub
Subroutine Buat_StopBit:
    Clr     SDA
    Setb    SCL
    Setb    SDA
EndSub
LoopPageSEE16bWrite:
    Mov     A,DataSEE
    Call    KirimDataSEE
    Jc     Loop2PageSEE16bWrite
    Djnz   R5,LoopPageSEE16bWrite
    Pop    05h
    Call   Buat_StopBit
    Ret
WrongPWrite:
    Pop    05h
    Call   Buat_StopBit
    Jmp    PageSEE8bWrite
EndSub
Subroutine Siapkan16bAlamatSEE:
    Call   KirimDeviceAddress
    Jc     SalahTulisAlamat
    Call   Kirim1WordAddress
    Jc     SalahTulisAlamat
    Call   Kirim2WordAddress
SalahTulisAlamat:
    Ret
EndSub
Subroutine ModeBacaSEE:
    Push   ACC          ;

    Setb   ACC.0
    Mov    Slave_AddrSEE,A
    Pop    ACC
    Mov    A,Slave_AddrSEE
    Call   KirimDataSEE
EndSub
Subroutine DPTRSEE16bit:
    Mov    R7,#0A0h

```



```

        Mov     B,DPH
        Mov     R6,DPL
        Call    Baca_SEE16b
EndSub
Subroutine TulisDPTRSEE16b:
        Mov     R7,#0A0h
        Mov     B,DPH
        Mov     R6,DPL
        Call    Tulis_SEE16b
EndSub

txt_tampil:           Db      ;'0123456789abcdef'
txt_tampil1:          Db      'selamat datang',0
txt_tampil2:          Db      'di mobil canggih',0

txt_menu_utama:       Db      '1.Aktifkan Mobil',0
txt_menu_utama1:      Db      '2.Ganti Pin',0
txt_menu_adm:         Db      '  Masukkan Pin',0
txt_menu_adm1:        Db      '    (----)',0
txt_menu_adm2:        Db      '1.Ubah Pin',0
txt_menu_adm3:        Db      '2.Lihat Pin',0
txt_ubah_adm4:        Db      '0.Kembali',0
txt_data:             Db      ' Data Tersimpan',0

txt_lihat:            Db      '  Pin Anda',0

txt_kembali:          Db      '    Tekan 0 ',0
txt_kembali1:         Db      ' untuk ke menu ',0

txt_keputusan:        Db      'Anda Yakin?',0
txt_keputusan1:       Db      '1.Y  2.T  3.E',0

blank:                Db      '                                ',0

txt_benar1:           Db      '  Pin Benar',0

txt_starter:          Db      'Starter Aktif',0
txt_error:            Db      '  Pin Salah',0

txt_motor_off:        Db      '  Mesin Mati',0
txt_motor_on:         Db      '  Mesin Hidup',0

;=====
Subroutine key_matrix:
        Push    B                                ; amankan register B
KeyInGetmatrix1:
        Call    Keypad3x4                        ;scan keypad
        Mov     A,keydata                        ;isi keydata = data d
i rutin keypad
        Cjne    A,#0FFh,KeyInGetmatrix0         ;jk isi a # 0ff
h ==>lompat
        Jmp     KeyInOutmatrix1
KeyInGetmatrix0:
        Mov     B,A                              ;simpan isi a to b

```

```

KeyInGetmatrix:
    Call    Keypad3x4
    Mov     A, keydata
    Cjne   A, B, KeyInOutmatrix           ;jk isi A # B l
ompat
    Jmp    keyIngetmatrix
KeyInOutmatrix:
    Mov     A, FlagKey_matrix
    Cjne   A, #0, KeyInOutmatrix1
    Mov     A, B
    Call   LCD_Tulis_Data
KeyInOutmatrix1:
    Mov     A, B
    Pop     B                             ;idem
EndSub

;=====
Subroutine Keypad_KeyIn:
    Push    B                             ; amankan register B
KeyInGet1:
    Call    Keypad3x4                     ;scan keypad
    Mov     A, keydata                    ;isi keydata = data d
i rutin keypad
    Cjne   A, #0FFh, KeyInGet0           ;jk isi a # 0ffh ==>l
ompat
    Jmp    keyInGet1
KeyInGet0:
    Mov     B, A                           ;simpan isi a to b
KeyInGet:
    Call    Keypad3x4
    Mov     A, keydata
    Cjne   A, B, KeyInOut                 ;jk isi A # B lompat
    Jmp    keyInget
KeyInOut:
    Mov     A, FlagKey
    Cjne   A, #0, KeyInOut1a
    Mov     A, B
    Call   LCD_Tulis_Data
    Jmp    keyinout1
Keyinout1a:
    Cjne   A, #1, KeyInOut1
    Mov     A, #'*'
    Call   LCD_Tulis_Data

KeyInOut1:
    Mov     A, B
    Pop     B                             ;idem
EndSub

;=====
; routine u/ baca keypad 3x4
; output pd keydata (0-9, E=redial, F=#)
;=====
Subroutine Keypad3x4:

```

```

        Mov    keyport,#0FFh
        Clr    kolom1                ;
;-----
ull:
        Jb    baris1,key1
        Mov    keydata,#'1'
        Ret

key1:

        Jb    baris2,key2
        Mov    keydata,#'4'
        Ret

key2:

        Jb    baris3,key3
        Mov    keydata,#'7'
        Ret

key3:

        Jb    baris4,key4
        Mov    keydata,#'*'
        Ret
;-----
key4:
        Setb   kolom1
        Clr    kolom2
;-----
        Jb    baris1,key5
        Mov    keydata,#'2'
        Ret

key5:

        Jb    baris2,key6
        Mov    keydata,#'5'
        Ret

key6:

        Jb    baris3,key7
        Mov    keydata,#'8'
        Ret

key7:

        Jb    baris4,key8
        Mov    keydata,#'0'
        Ret
;-----
key8:
        Setb   kolom2
        Clr    kolom3
;-----
        Jb    baris1,key9
        Mov    keydata,#'3'
        Ret

key9:

        Jb    baris2,key10
        Mov    keydata,#'6'
        Ret

```

```

key10:      Jb      baris3,key11
            Mov     keydata,#'9'
            Ret

key11:      Jb      baris4,key12
            Mov     keydata,#'#'
            Ret

key12:      Mov     keydata,#0FFh

EndSub

Subroutine LCD_Clear:
    Call     lcd_line_1
    Mov     DPTR,#txt_blank
    Call     LCD_TampilKata
    Call     LCD_Line_2
    Mov     DPTR,#txt_blank
    Call     LCD_TampilKata
EndSub
txt_blank:  Db      '          ',0

Subroutine LCD_Cleara:
    Mov     A,#01h
    Call     LCD_Tulis_Inst
EndSub
Subroutine LCD_Blink_Off:
    Push    ACC
    Mov     A,#00001100b ;
    Lcall   LCD_Tulis_Inst
    Pop     ACC
EndSub
Subroutine LCD_Blink_On:
    Push    ACC
    Mov     A,#00001101b ;
    Lcall   LCD_Tulis_Inst
    Pop     ACC
EndSub
Subroutine lcd_line_1:
    Mov     A,#80h
    Call     LCD_Tulis_Inst
EndSub
Subroutine LCD_Line_2:
    Mov     A,#0C0h
    Call     LCD_Tulis_Inst
EndSub
Subroutine LCD_Cursor_Position:
    Push    ACC
    Mov     A,R0      ;15      26
    Anl    A,#0F0h   ;10      20
    Cjne   A,#10h,lcd_cursor_position1
    Mov     A,R0      ;15

```

```

    Anl    A,#0Fh  ;05
    Orl    A,#80h  ;85
    Call   LCD_Tulis_Inst
    Jmp    lcd_cursor_position_end
lcd_cursor_position1:
    Cjne   A,#20h,lcd_cursor_position_end
    Mov    A,R0      ; 26
    Anl    A,#0Fh    ;06
    Orl    A,#0C0h   ; c6
    Call   LCD_Tulis_Inst
lcd_cursor_position_end:
    Pop    ACC
EndSub
Subroutine LCD_TampilKata:
    Push  ACC
    Push  DPL
    Push  DPH
TampilKata1:
    Clr   A
    Movc  A,@A+DPTR
    Cjne  A,#0,TampilKata2           ;intinya
    Jmp   out
TampilKata2:
    Inc   DPTR
    Call  LCD_Tulis_Data
    Jmp   TampilKata1
out:
    Pop   DPH
    Pop   DPL
    Pop   ACC
EndSub

Subroutine LCD_Inisialisasi:
    ;Call  delayL
    Mov    A,#3Fh
    Call   LCD_Tulis_Inst
    Mov    A,#0Ch
    Call   LCD_Tulis_Inst
    Mov    A,#06h
    Call   LCD_Tulis_Inst
    Mov    A,#1Ch
    Call   LCD_Tulis_Inst
    Mov    A,#01h
    Call   LCD_Tulis_Inst
EndSub
Subroutine LCD_Tulis_Inst:
    Clr    P2.7    ;e
    Mov    P0,A
    Setb   P2.6    ;rs
    Clr    P2.6
    Call   delayL
EndSub
Subroutine LCD_Tulis_Data:

```

```

    Setb    P2.7
    Mov     P0,A
    Setb    P2.6
    Clr     P2.6
    Call    delayL
EndSub
Subroutine LCD_Hexa:
    Push    07h
    Push    ACC

    Mov     7,A
    Anl     A,#0F0h
    Swap    A
    Orl     A,#30h
    Lcall   tes_huruf_
    Call    LCD_Tulis_Data
    Mov     A,7
    Anl     A,#0Fh
    Orl     A,#30h
    Lcall   tes_huruf_
    Lcall   LCD_Tulis_Data

    Pop     ACC
    Pop     07h
    Ret

tes_huruf_ :
    Cjne    A,#3Ah,tes_huruf_1
    Mov     A,#'a'
    Ret
tes_huruf_1:
    Cjne    A,#3Bh,tes_huruf_2
    Mov     A,#'b'
    Ret
tes_huruf_2:
    Cjne    A,#3Ch,tes_huruf_3
    Mov     A,#'c'
    Ret
tes_huruf_3:
    Cjne    A,#3Dh,tes_huruf_4
    Mov     A,#'d'
    Ret
tes_huruf_4:
    Cjne    A,#3Eh,tes_huruf_5
    Mov     A,#'e'
    Ret
tes_huruf_5:
    Cjne    A,#3Fh,tes_huruf_6
    Mov     A,#'f'
    Ret
tes_huruf_6:
    Ret
EndSub

```

Subroutine delayL:

```

    Push    0
    Push    1
    Push    2

    Mov     2,#80
tunda1qqqa:
    Mov     0,#10
    Djnz   0,$;tunda1qqqa
    Mov     1,#30
    Djnz   1,$
    Mov     1,#1
    Djnz   1,$
    Djnz   2,tunda1qqqa

    Pop     2
    Pop     1
    Pop     0
EndSub

```

Subroutine tunda:

```

    Push    0
    Push    1
    Push    2

    Mov     R0,#100
tunda1:
    Mov     R1,#100
tunda2:
    Mov     R2,#100
    Djnz   R2,$
    Djnz   R1,tunda2
    Djnz   R0,tunda1

    Pop     2
    Pop     1
    Pop     0
EndSub

```

```

;===== DELAY =====
Subroutine Delay_Var_1ms:
    Call    Delay_Fix_1ms
    Djnz   R0,Delay_Var_1ms
EndSub
Subroutine Delay_Var_10ms:
    Call    Delay_Fix_10ms
    Djnz   R0,Delay_Var_10ms
EndSub
Subroutine Delay_Var_100ms:
    Call    Delay_Fix_1ms

```

```

        Djnz    R0,Delay_Var_100ms
EndSub
Subroutine Delay_Var_1s:
    Call    Delay_Fix_1s
    Djnz    R0,Delay_Var_1s
EndSub
Subroutine Delay_Var_10s:
    Call    Delay_Fix_10s
    Djnz    R0,Delay_Var_10s
EndSub
Subroutine Delay_Var_10us:
    Call    Delay_Fix_10us
    Djnz    R0,Delay_Var_10us
EndSub
Subroutine Delay_Fix_10us:
    Push    1
    Mov     1,#20
    Djnz    1,$
    Pop     1
EndSub
Subroutine Delay_Fix_10s:
    Push    1
    Mov     1,#100
delay_fix_10s_1:
    Call    Delay_Fix_100ms
    Djnz    1,delay_fix_10s_1
    Pop     1
EndSub
Subroutine Delay_Fix_1s:
    Push    1
    Mov     1,#100
delay_fix_1000ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_1000ms_1
    Pop     1
EndSub
Subroutine Delay_Fix_100ms:
    Push    1
    Mov     1,#10
delay_fix_100ms_1:
    Call    Delay_Fix_10ms
    Djnz    1,delay_fix_100ms_1
    Pop     1
EndSub
Subroutine Delay_Fix_10ms:
    Mov     TMOD,#00000001b    ; Timer 1 bekerja pada mode
1
    Mov     TL0,#3Dh    ; siapkan waktu tunda 50 mili-deti
k
    Mov     TH0,#0B0h
    Clr     TFO    ; me-nol-kan bit limpahan
    Setb    TR0    ; timer mulai bekerja
    Jnb     TFO,$    ; tunggu di sini sampai mel

```



```
impah      Clr      TR0          ; timer berhenti kerja
           Ret
EndSub
Subroutine Delay_Fix_1ms:
  Mov      TMOD,#00000001b    ; Timer 1 bekerja pada mode
1
  Mov      TLO,#0EDh        ; siapkan waktu tunda 50 mili-det
ik
  Mov      TH0,#78h
  Clr      TFO              ; me-nol-kan bit limpahan
  Setb     TR0              ; timer mulai bekerja
  Jnb     TFO,$            ; tunggu di sini sampai mel
impah
  Clr      TR0              ; timer berhenti kerja
  Ret
EndSub
```

Features

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

Description

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

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



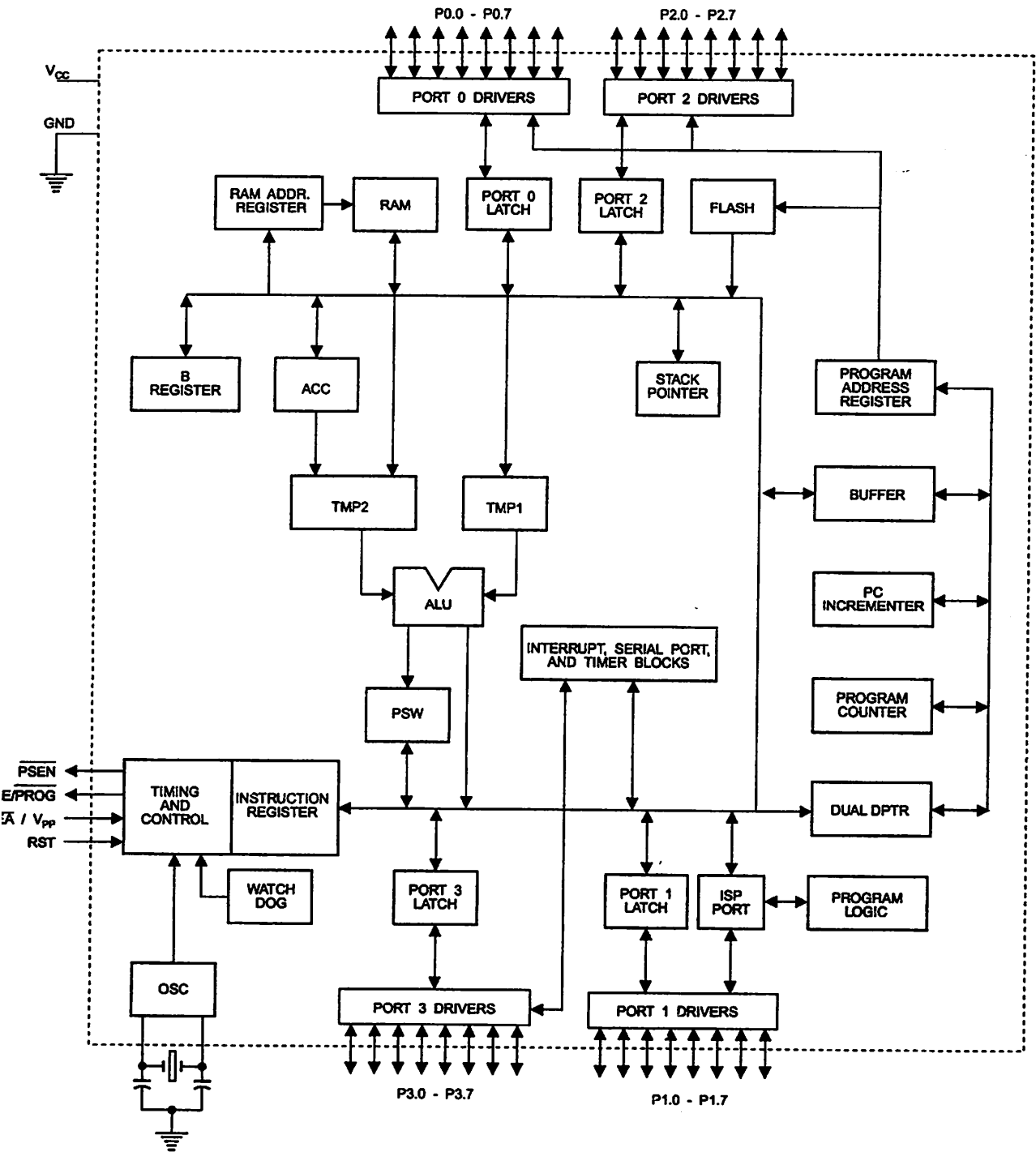
**8-bit
Microcontroller
with 4K Bytes
In-System
Programmable
Flash**

AT89S51

2487B-MICRO-12/03



Block Diagram





1 Description

C

Supply voltage (all packages except 42-PDIP).

ID

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.

Port 0

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

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

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

Port 1

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

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

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

Port 2

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

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

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

Port 3

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

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

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

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

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.

\overline{ALE} /PROG

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

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

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

\overline{PSEN}

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

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

\overline{VPP}

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

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

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

AL1

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

AL2

Output from the inverting oscillator amplifier



Special Function Registers

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

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

Table 1. AT89S51 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0			WDTRST 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 0XX00000	87H

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

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

Table 2. AUXR: Auxiliary Register

AUXR		Address = 8EH						Reset Value = XXX00XX0B
Not Bit Addressable								
	-	-	-	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						

Memory Organization

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

Program Memory

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

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

Data Memory

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

Watchdog Timer (enabled with reset-out)

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

Using the WDT

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

AT89S51

WDT During Power-down and Idle

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

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

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

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

UART

The UART in the AT89S51 operates the same way as the UART in the AT89C51. For further information on the UART operation, refer to the Atmel Web site (<http://www.atmel.com>). From the home page, select "Products", then "Microcontrollers", then "8051-Architecture", then "Documentation", and "Other Documents". Open the Adobe® Acrobat® file "AT89 Series Hardware Description".

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. For further information on the timers' operation, refer to the Atmel Web site (<http://www.atmel.com>). From the home page, select "Products", then "Microcontrollers", then "8051-Architecture", then "Documentation", and "Other Documents". Open the Adobe Acrobat file "AT89 Series Hardware Description".

Interrupts

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

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

Note that Table 4 shows that bit positions IE.6 and IE.5 are unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

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



Table 4. Interrupt Enable (IE) Register

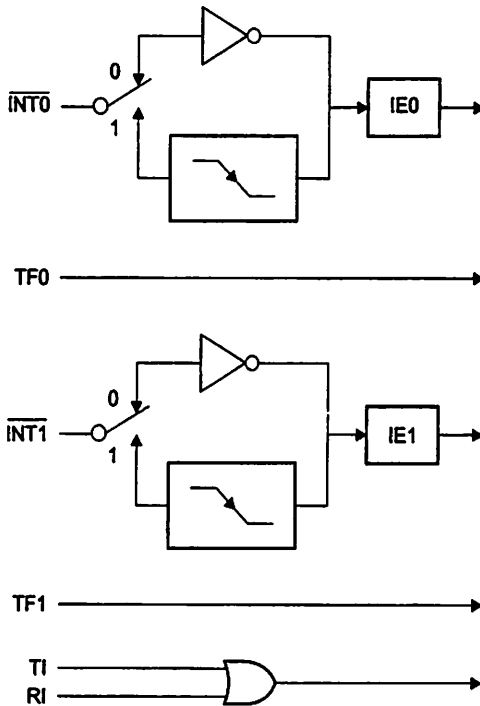
(MSB)				(LSB)			
EA	-	-	ES	ET1	EX1	ET0	EX0

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.

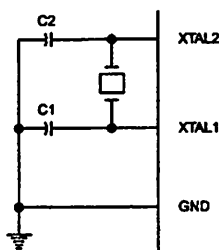
Figure 1. Interrupt Sources



Oscillator Characteristics

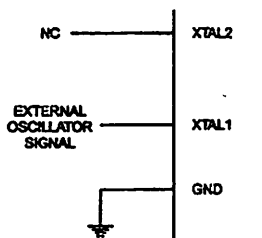
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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 (INT0 or INT1). Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.



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

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

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

Table 6. Lock Bit Protection Modes

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

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

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

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

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash Programming 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 $\overline{ALE}/\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 μ s. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

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

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

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

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

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

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

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

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

Programming the Flash – Serial Mode

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

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

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

Serial Programming Algorithm

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

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





Power-off sequence (if needed):

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

Set RST to "L".

Turn V_{CC} power off.

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

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

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

Figure 4. Programming the Flash Memory (Parallel Mode)

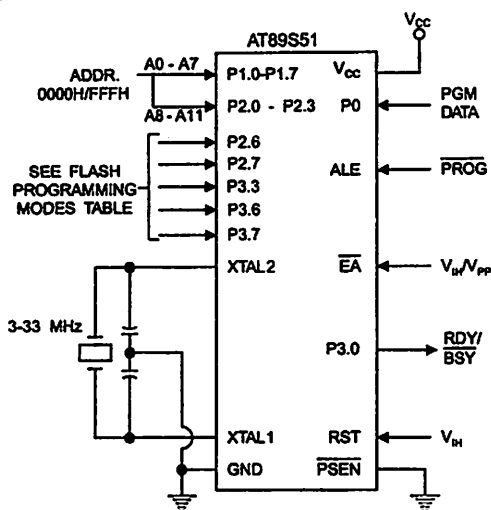
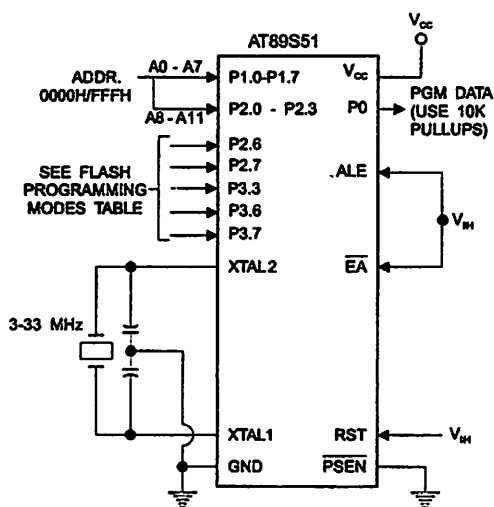


Figure 5. Verifying the Flash Memory (Parallel Mode)





Flash Programming and Verification Characteristics (Parallel Mode)

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

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 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	ENABLE Low to Data Valid		$48t_{CLCL}$	
t_{EHOZ}	Data Float After ENABLE	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		50	μs

Figure 6. Flash Programming and Verification Waveforms – Parallel Mode

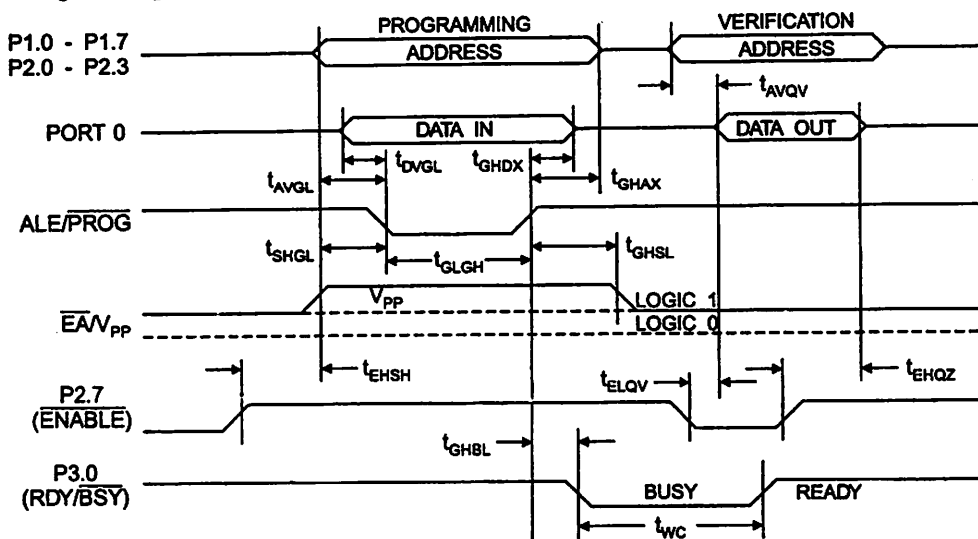
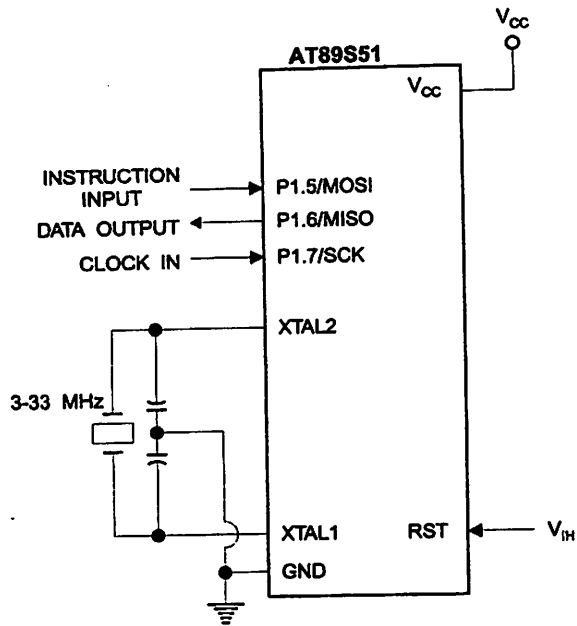


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms

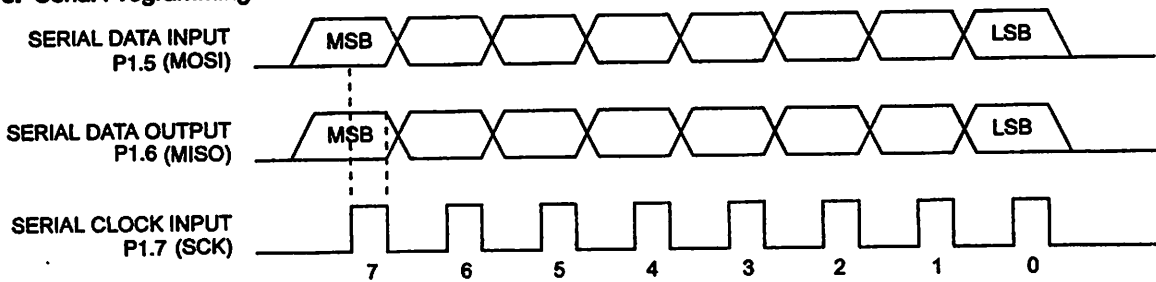




Table 8. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits ⁽¹⁾	1010 1100	1110 00B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx B2 B1 B0 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

Figure 9. Serial Programming Timing

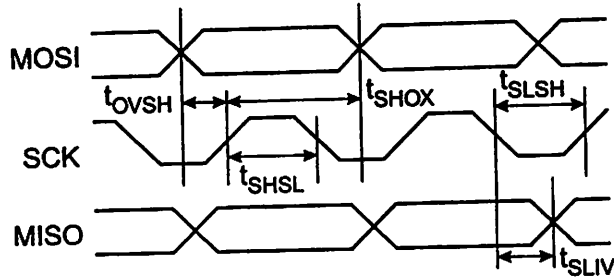


Table 9. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	3		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	$8 t_{CLCL}$			ns
t_{SLSH}	SCK Pulse Width Low	$8 t_{CLCL}$			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	$2 t_{CLCL}$			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			$64 t_{CLCL} + 400$	μs





Absolute Maximum Ratings*

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

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

C Characteristics

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

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage (\overline{EA})		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{L1}	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 5.5\text{V}$		50	μA

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

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port:

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

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

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

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

Characteristics

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

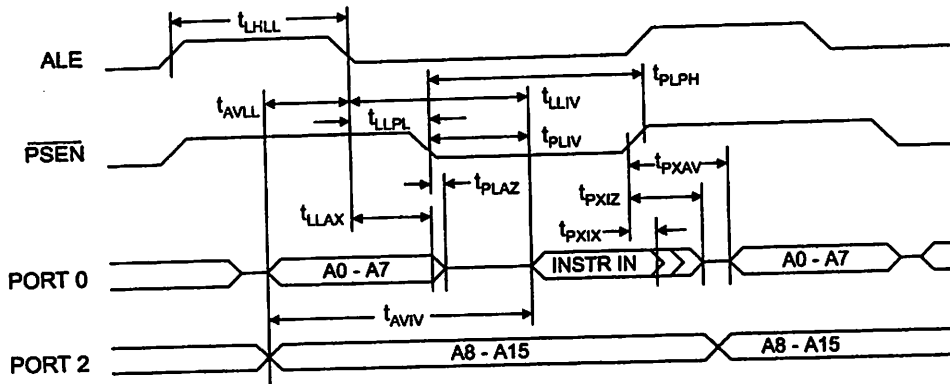
Internal Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{CLCL}	Oscillator Frequency			0	33	MHz
t_{HLL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{VLL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-25$		ns
t_{LAX}	Address Hold After ALE Low	48		$t_{\text{CLCL}}-25$		ns
t_{LV}	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
t_{LPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-25$		ns
t_{LPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-45$		ns
t_{LV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-60$	ns
t_{XIX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
t_{XIZ}	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-25$	ns
t_{XAV}	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
t_{VIV}	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-80$	ns
t_{LAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{LRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{LWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{LDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
t_{RDX}	Data Hold After $\overline{\text{RD}}$	0		0		ns
t_{RDZ}	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
t_{LDV}	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
t_{VDV}	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
t_{LWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-30$		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-130$		ns
t_{WHQX}	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-25$		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-25$	$t_{\text{CLCL}}+25$	ns

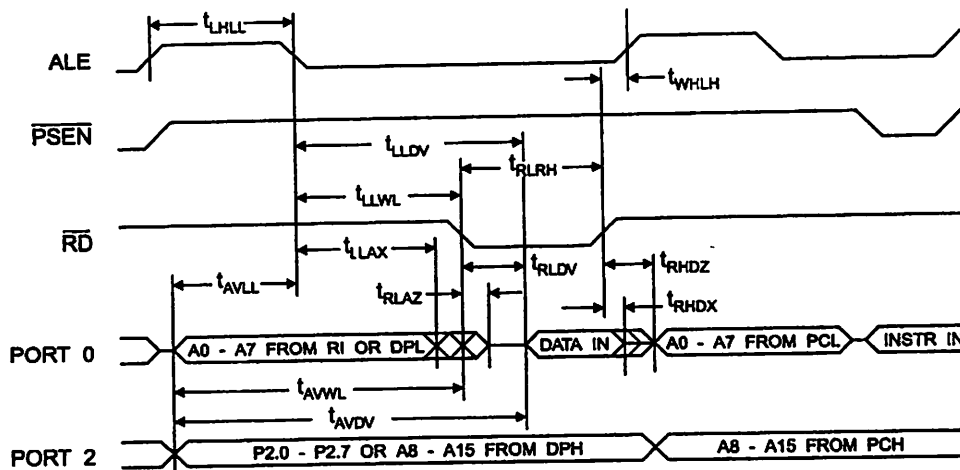




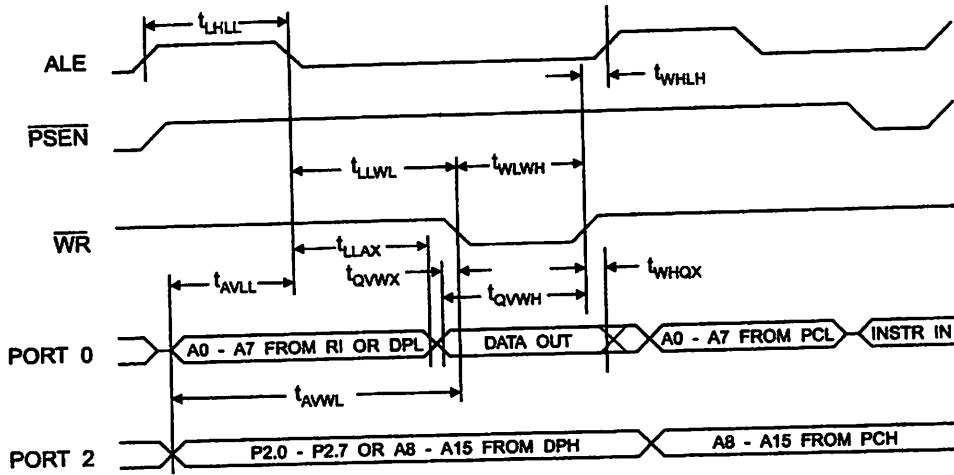
Internal Program Memory Read Cycle



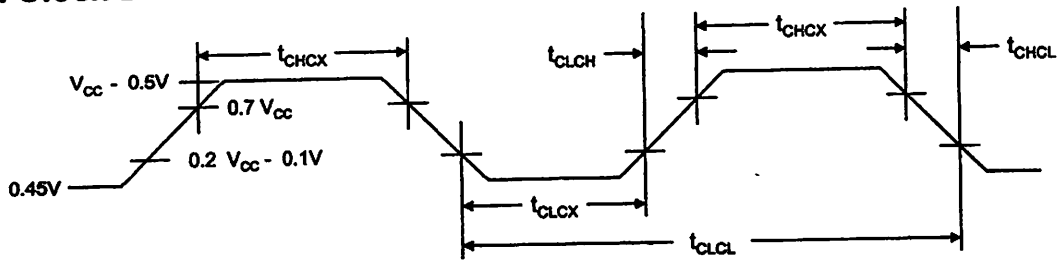
Internal Data Memory Read Cycle



Internal Data Memory Write Cycle



Internal Clock Drive Waveforms



Internal Clock Drive

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



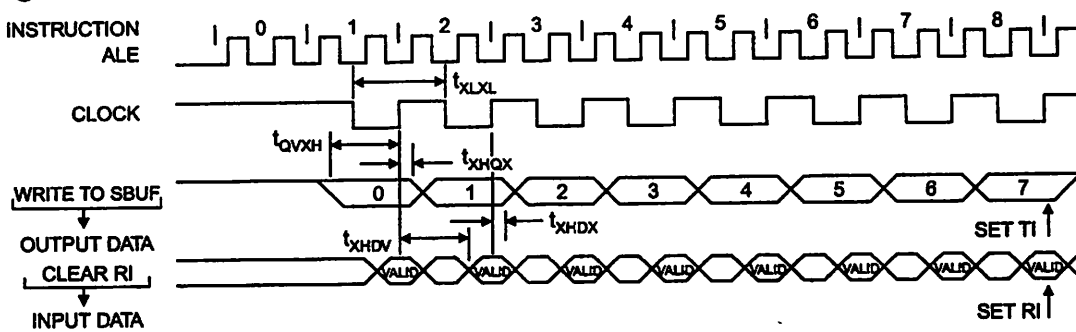


Serial Port Timing: Shift Register Mode Test Conditions

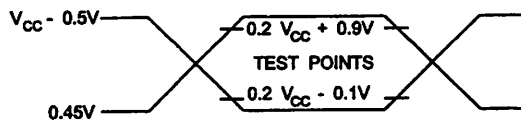
Values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = $80 pF$.

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

Shift Register Mode Timing Waveforms

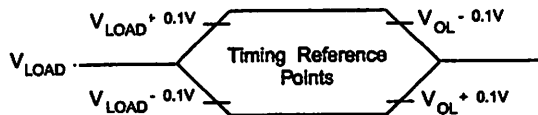


Testing Input/Output Waveforms⁽¹⁾



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

Load Waveforms⁽¹⁾



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

Ordering Information

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

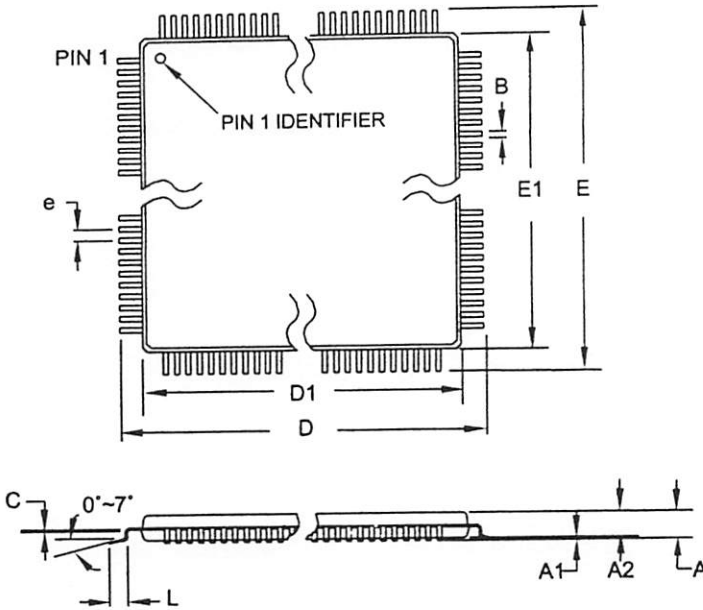
Package Type	
4A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
4J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
0P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
2PS6	42-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)





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.

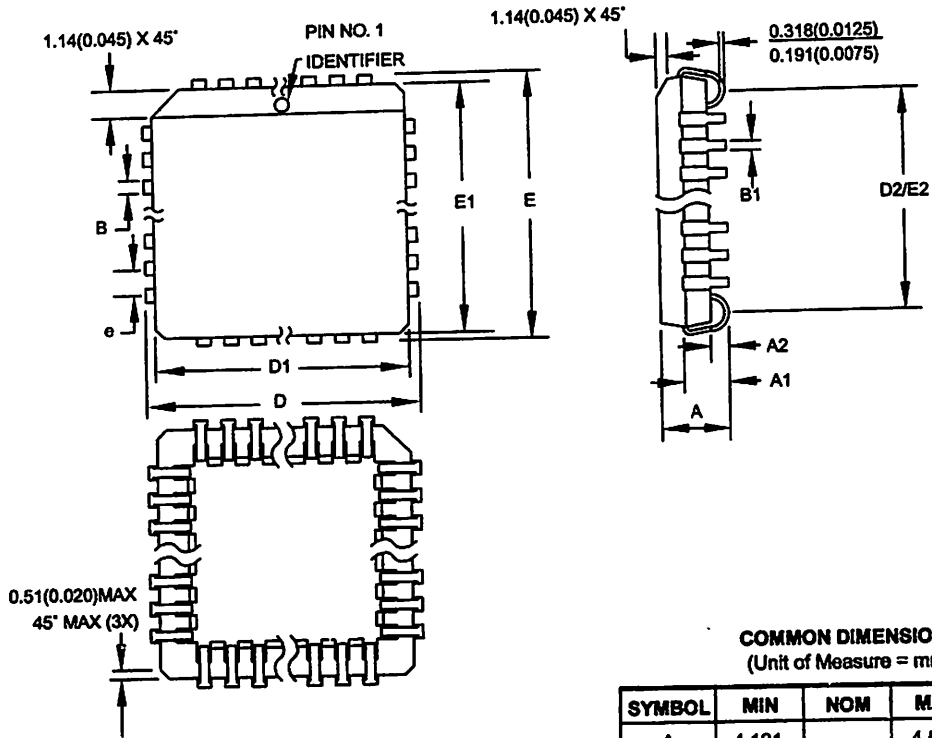
10/5/2001

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.	REV.
		44A	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.

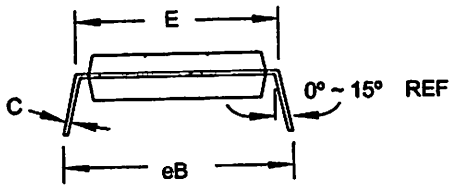
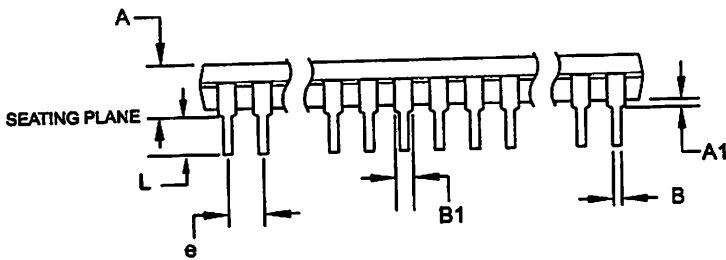
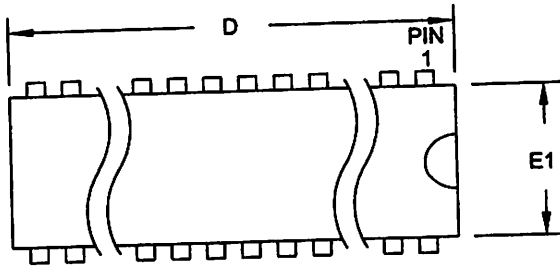
10/04/01

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





40P6 - PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

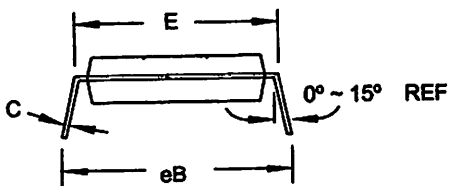
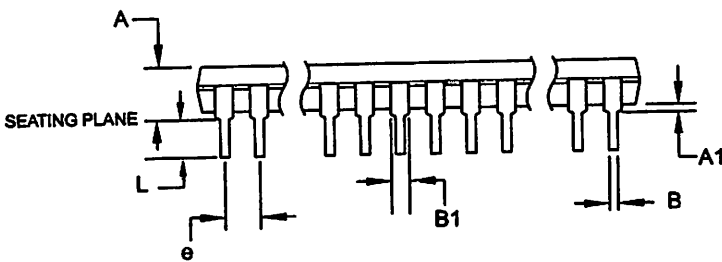
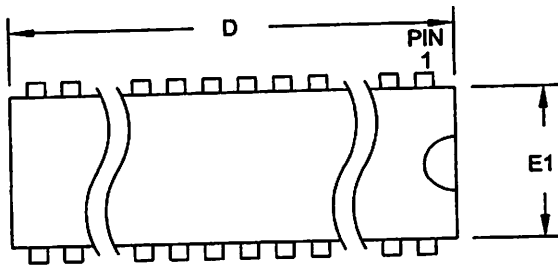
SYMBOL	MIN	NOM	MAX	NOTE
A	-	-	4.826	
A1	0.381	-	-	
D	52.070	-	52.578	Note 2
E	15.240	-	15.875	
E1	13.462	-	13.970	Note 2
B	0.356	-	0.559	
B1	1.041	-	1.651	
L	3.048	-	3.556	
C	0.203	-	0.381	
eB	15.494	-	17.526	
e	2.540 TYP			

- 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

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

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

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





Atmel Corporation

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

Regional Headquarters

Europe

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

Asia

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

Japan

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

Atmel Operations

Memory

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

Microcontrollers

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

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

ASIC/ASSP/Smart Cards

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

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

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

RF/Automotive

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

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

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

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

Literature Requests
www.atmel.com/literature

Disclaimer: 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

Features

- Low-voltage and Standard-voltage Operation
 - 2.7 ($V_{CC} = 2.7$ to $5.5V$)
 - 1.8 ($V_{CC} = 1.8$ to $5.5V$)
- Low-power Devices ($I_{SB} = 6 \mu A$ at $5.5V$) Available
- Internally Organized 8192×8
- 2-Wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bi-directional Data Transfer Protocol
- 100 kHz Clock Rate
- Write Protect Pin for Hardware Data Protection
- 2-Byte Page Write Mode (Partial Page Writes Allowed)
- Self-Timed Write Cycle (5 ms max)
- High Reliability
 - Endurance: 1 Million Write Cycles
 - Data Retention: 100 Years
- Lead-free/Halogen-free Devices Available
- 8-lead PDIP, 8-lead JEDEC SOIC and 8-lead TSSOP Packages

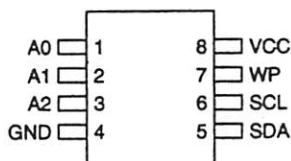
Description

The AT24C64B provides 65,536 bits of serial electrically erasable and programmable non-volatile memory (EEPROM) organized as 8192 words of 8 bits each. 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 AT24C64B is available in space saving 8-lead PDIP, 8-lead JEDEC SOIC and 8-lead TSSOP packages and is accessed via a 2-wire serial interface. In addition, the entire family is available in 2.7V (2.7 to 5.5V) and 1.8V (1.8 to 5.5V) versions.

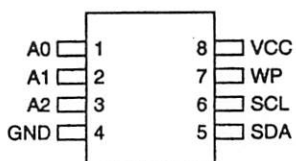
Pin Configurations

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

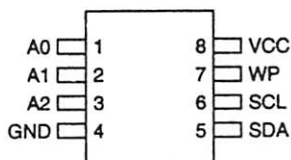
8-lead SOIC



8-lead PDIP



8-lead TSSOP



3350C-SEEPR-5/04



2-Wire Serial EEPROM

64K (8192 x 8)

AT24C64B

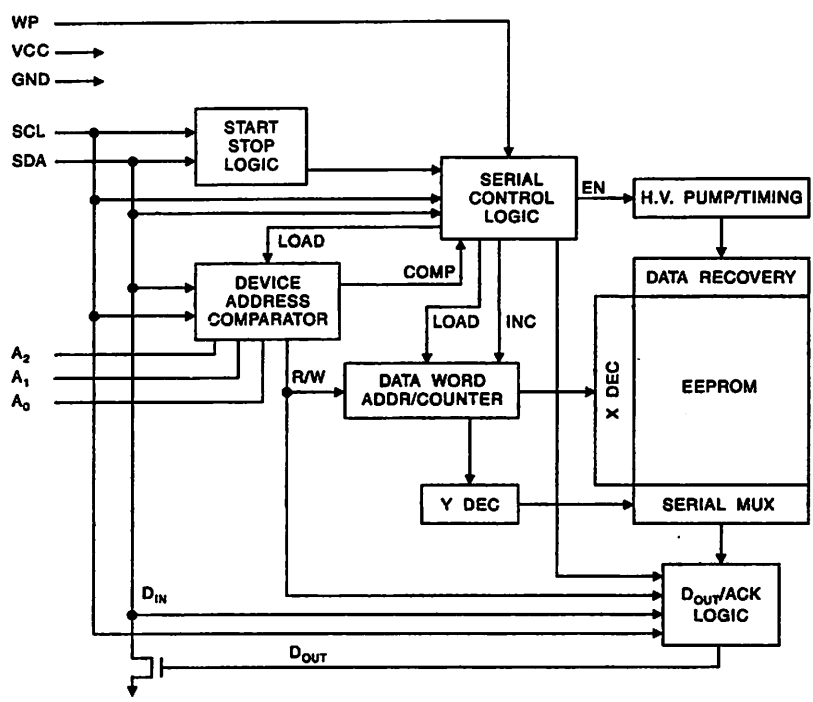


Absolute Maximum Ratings*

Operating Temperature	-55 to +125°C
Storage Temperature	-65 to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 to +7.0V
Maximum Operating Voltage	6.25V
IO 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



In 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/ADDRESSES (A2, A1, A0): The A2, A1 and A0 pins are device address inputs that are hard wired or left not connected for hardware compatibility with other AT24CXX devices. When the pins are hardwired, as many as eight 64K devices may be addressed on a single bus system (device addressing is discussed in detail under the Device Addressing section). If the pins are left floating, the A2, A1 and A0 pins will be internally pulled down to GND if the capacitive coupling to the circuit board V_{CC} plane is $<3pF$. If coupling is $>3pF$, Atmel recommends connecting the address pins to GND.

WRITE PROTECT (WP): The write protect input, when connected to GND, allows normal write operations. When WP is connected high to V_{CC} , all write operations to the upper quadrant (16K bits) of memory are inhibited. If the pin is left floating, the WP pin will be internally pulled down to GND if the capacitive coupling to the circuit board V_{CC} plane is $<3pF$. If coupling is $>3pF$, Atmel recommends connecting the pin to GND.

Memory Organization

AT24C64B, 64K SERIAL EEPROM: The 64K is internally organized as 256 pages of 32 bytes each. Random word addressing requires a 13 bit data word address.

Input Capacitance⁽¹⁾

Applicable 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
C_{IO}	Input/Output Capacitance (SDA)	8	pF	$V_{IO} = 0\text{V}$
C_{IN}	Input Capacitance ($A_0, A_1, A_2, \text{SCL}$)	6	pF	$V_{IN} = 0\text{V}$

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

DC Characteristics

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

Symbol	Parameter	Test Condition		Min	Typ	Max	Units
V_{CC1}	Supply Voltage			1.8		5.5	V
V_{CC2}	Supply Voltage			2.7		5.5	V
V_{CC3}	Supply Voltage			4.5		5.5	V
I_{CC1}	Supply Current	$V_{CC} = 5.0\text{V}$	READ at 400 kHz		0.4	1.0	mA
I_{CC2}	Supply Current	$V_{CC} = 5.0\text{V}$	WRITE at 400 kHz		2.0	3.0	mA
I_{S1}	Standby Current (1.8V option)	$V_{CC} = 1.8\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}			1.0	μA
I_{S2}	Standby Current (2.7V option)	$V_{CC} = 2.7\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}			2.0	μA
I_{S3}	Standby Current (5V option)	$V_{CC} = 4.5 - 5.5\text{V}$	$V_{IN} = V_{CC}$ or V_{SS}			6.0	μA
I_{IL}	Input Leakage Current	$V_{IN} = V_{CC}$ or V_{SS}			0.10	3.0	μA
I_{OL}	Output Leakage Current	$V_{OUT} = V_{CC}$ or V_{SS}			0.05	3.0	μA
V_{IL}	Input Low Level ⁽¹⁾			-0.6		$V_{CC} \times 0.3$	V
V_{IH}	Input High Level ⁽¹⁾			$V_{CC} \times 0.7$		$V_{CC} + 0.5$	V
V_{OL2}	Output Low Level	$V_{CC} = 3.0\text{V}$	$I_{OL} = 2.1\text{ mA}$			0.4	V
V_{OL1}	Output Low Level	$V_{CC} = 1.8\text{V}$	$I_{OL} = 0.15\text{ mA}$			0.2	V

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



Characteristics

Applicable over recommended operating range from $T_{AI} = -40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$, $V_{CC} = +1.8\text{V}$ to $+5.5\text{V}$, $C_L = 1$ TTL Gate and 10 pF (unless otherwise noted).

Symbol	Parameter	AT24C64B				Units
		1.8V – 3.6V		5.0V		
		Min	Max	Min	Max	
CL	Clock Frequency, SCL		400		400	kHz
CW	Clock Pulse Width Low	1.3		1.2		μs
CH	Clock Pulse Width High	0.6		0.6		μs
	Noise Suppression Time ⁽¹⁾		100		50	ns
LA	Clock Low to Data Out Valid	0.2	0.9	0.1	0.9	μs
UF	Time the bus must be free before a new transmission can start ⁽²⁾	1.3		1.2		μs
SDSTA	Start Hold Time	0.6		0.6		μs
SUSTA	Start Set-up Time	0.6		0.6		μs
DDAT	Data In Hold Time	0		0		μs
UDAT	Data In Set-up Time	100		100		ns
	Inputs Rise Time ⁽²⁾		0.3		0.3	μs
	Inputs Fall Time ⁽²⁾		300		300	ns
SDSTO	Stop Set-up Time	0.6		0.6		μs
DOH	Data Out Hold Time	200		50		ns
WR	Write Cycle Time		5		5	ms
Endurance ⁽¹⁾	5.0V, 25°C, Page Mode	1M		1M		Write Cycles

- Notes: 1. This parameter is characterized and is not 100% tested ($T_A = 25^{\circ}\text{C}$)
 2. This parameter is characterized and is not 100% tested.

Device Operation

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

START CONDITION: A high-to-low transition of SDA with SCL high is a start condition which must precede any other command (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 stop command 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 AT24C64B 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.

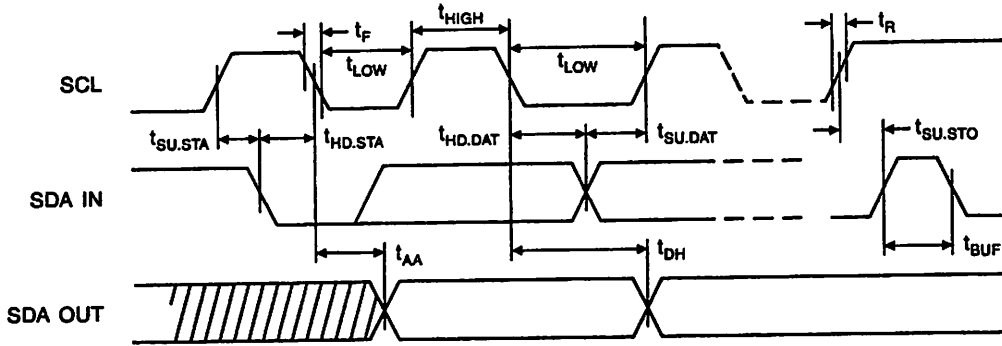
MEMORY RESET: After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

(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.



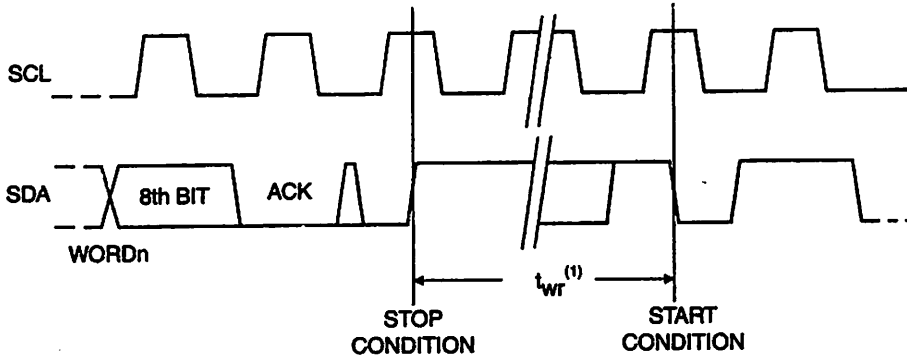
Timing

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



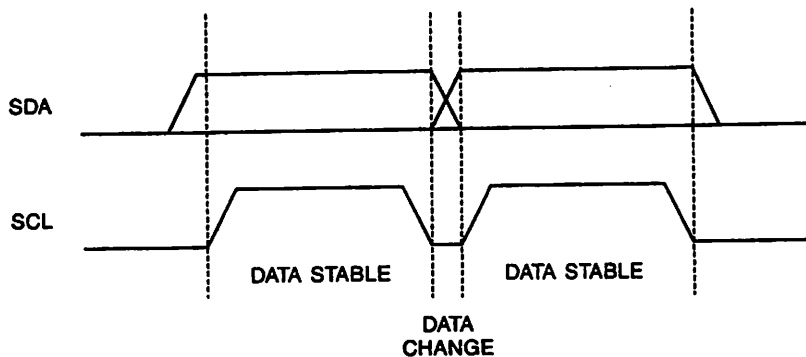
Write Cycle Timing

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

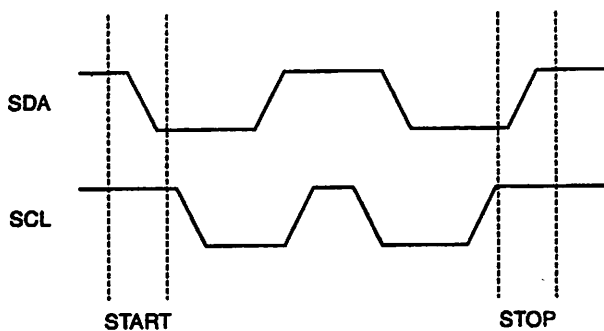


Note: 1. The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

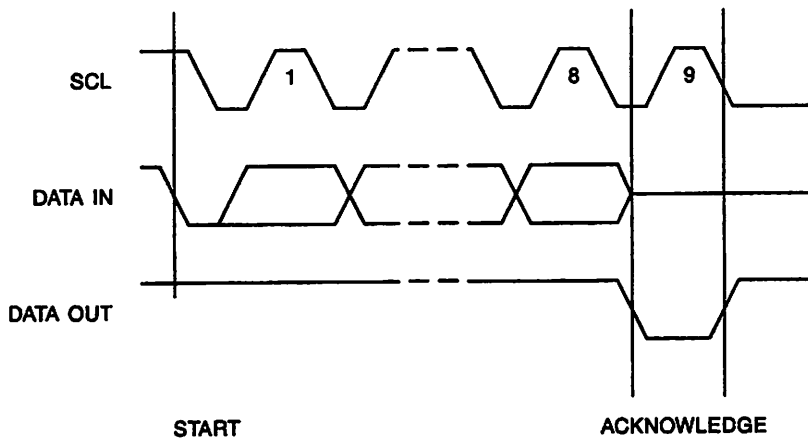
Data Validity



Start and Stop Definition



Output Acknowledge



Device Addressing

The 64K EEPROM requires an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 1). The device address word consists of a mandatory one, zero sequence for the first four most significant bits as shown. This is common to all 2-wire EEPROM devices.

The 64K uses the three device address bits A2, A1, A0 to allow as many as eight devices on the same bus. These bits 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.

Upon 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 SDA and SCL pins prevent small noise spikes from activating the device. A low- V_{CC} detector (5-volt option) resets the device to prevent data corruption in a noisy environment.

DATA SECURITY: The AT24C64B has a hardware data protection scheme that allows the user to write protect the upper quadrant (16K bits) of memory when the WP pin is at V_{CC} .

Write Operations

BYTE WRITE: A write operation requires two 8-bit data word 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 device, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle, t_{WR} , 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 64K EEPROM is capable of 32-byte 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 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 read is from the last byte of the last memory page, to the first byte of the first page. The address "roll over" during 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 acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will "roll over" and the sequential read will continue. 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

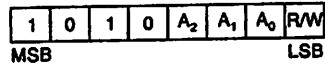


Figure 2. Byte Write

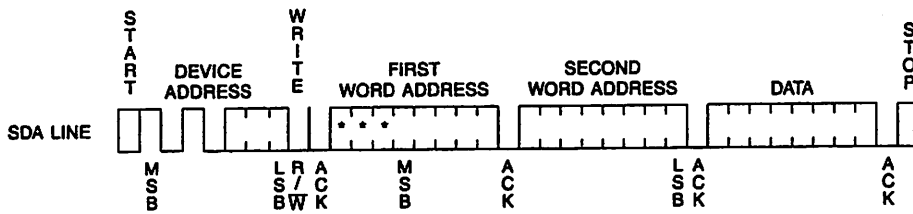
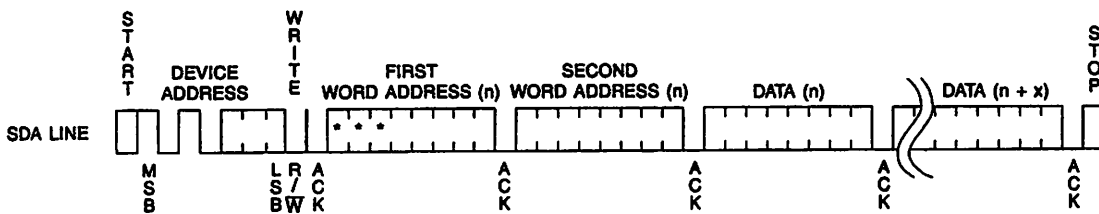


Figure 3. Page Write



1. * = DON'T CARE bits

Figure 4. Current Address Read

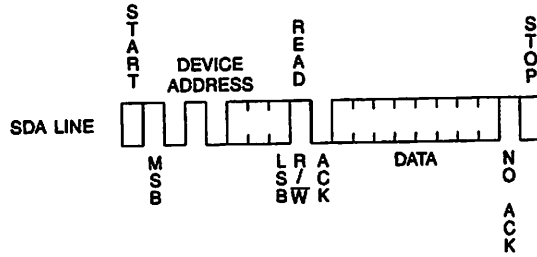
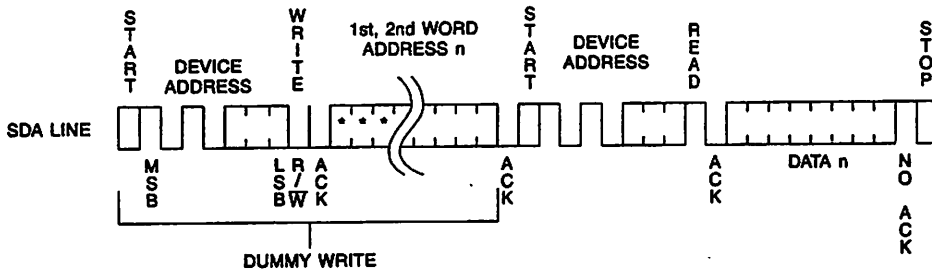
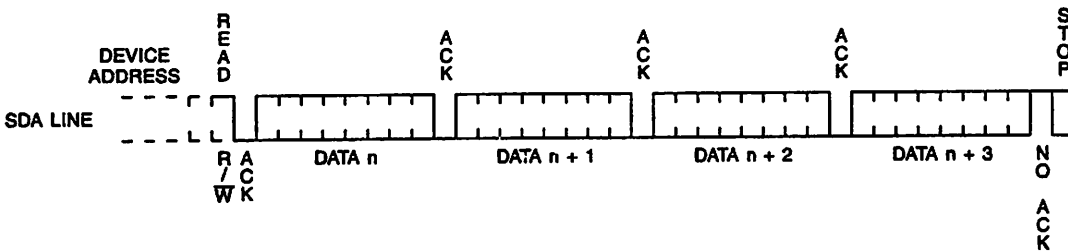


Figure 5. Random Read



1. * = DON'T CARE bits

Figure 6. Sequential Read



24C64B Ordering Information

Ordering Code	Package	Operation Range
24C64B-10PI-2.7 24C64BN-10SI-2.7 24C64B-10TI-2.7	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
24C64B-10PI-1.8 24C64BN-10SI-1.8 24C64B-10TI-1.8	8P3 8S1 8A2	Industrial Temperature (-40°C to 85°C)
24C64BN-10SU-2.7 24C64BN-10SU-1.8 24C64B-10TU-2.7 24C64B-10TU-1.8	8S1 8S1 8A2 8A2	Lead-free/Halogen-free Industrial Temperature (-40°C to 85°C)

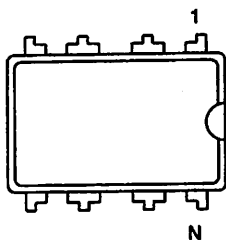
: For 2.7V devices used in the 4.5V to 5.5V range, please refer to performance values in the AC and DC characteristics tables.

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, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP)
Options	
7	Low Voltage (2.7V to 5.5V)
8	Low Voltage (1.8V to 5.5V)

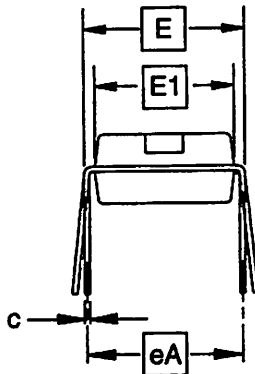


Packaging Information

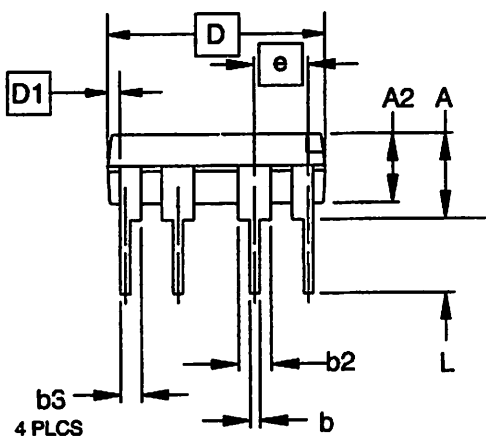
- PDIP



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = inches)

SYMBOL	MIN	NOM	MAX	NOTE
A			0.210	2
A2	0.115	0.130	0.195	
b	0.014	0.018	0.022	5
b2	0.045	0.060	0.070	6
b3	0.030	0.039	0.045	6
c	0.008	0.010	0.014	
D	0.355	0.365	0.400	3
D1	0.005			3
E	0.300	0.310	0.325	4
E1	0.240	0.250	0.280	3
e	0.100 BSC			
eA	0.300 BSC			4
L	0.115	0.130	0.150	2

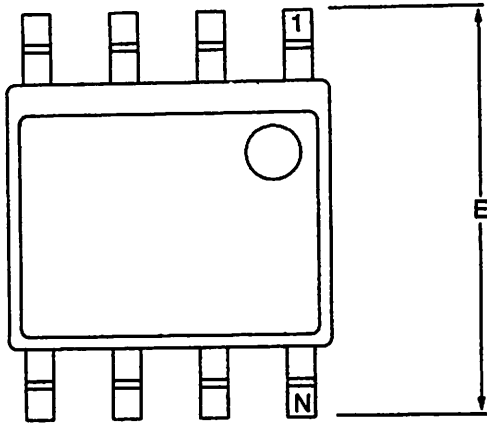
- Notes:
1. This drawing is for general information only; refer to JEDEC Drawing MS-001, Variation BA for additional information.
 2. Dimensions A and L are measured with the package seated in JEDEC seating plane Gauge GS-3.
 3. D, D1 and E1 dimensions do not include mold Flash or protrusions. Mold Flash or protrusions shall not exceed 0.010 inch.
 4. E and eA measured with the leads constrained to be perpendicular to datum.
 5. Pointed or rounded lead tips are preferred to ease insertion.
 6. b2 and b3 maximum dimensions do not include Dambar protrusions. Dambar protrusions shall not exceed 0.010 (0.25 mm).

01/09/02

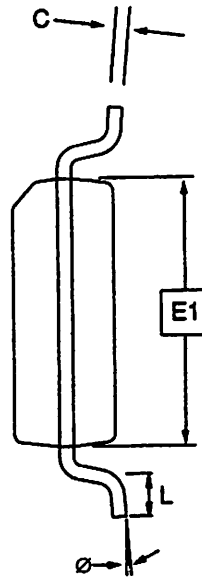
<p>2325 Orchard Parkway San Jose, CA 95131</p>	<p>TITLE 8P3, 8-lead, 0.300" Wide Body, Plastic Dual In-line Package (PDIP)</p>	<p>DRAWING NO. 8P3</p>	<p>REV. B</p>
--	--	-----------------------------------	--------------------------



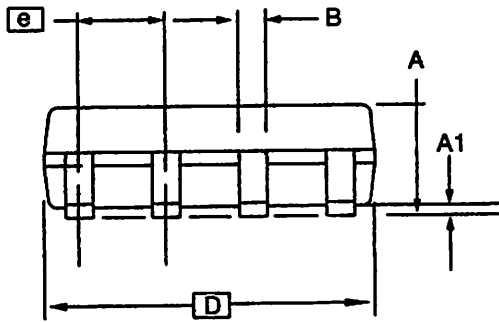
1 - JEDEC SOIC



Top View



End View



Side View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	1.35	-	1.75	
A1	0.10	-	0.25	
b	0.31	-	0.51	
C	0.17	-	0.25	
D	4.80	-	5.00	
E1	3.81	-	3.99	
E	5.79	-	6.20	
e	1.27 BSC			
L	0.40	-	1.27	
Ø	0	-	8	

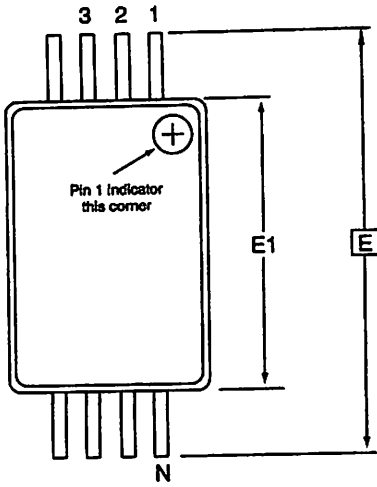
Note: These drawings are for general information only. Refer to JEDEC Drawing MS-012, Variation AA for proper dimensions, tolerances, datums, etc.

10/7/03

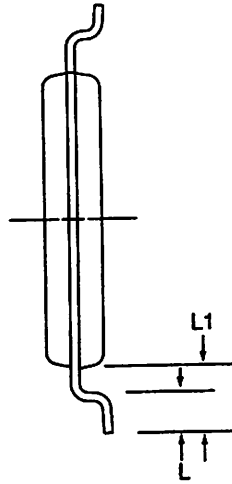
<p>1150 E. Cheyenne Mtn. Blvd. Colorado Springs, CO 80906</p>	<p>TITLE 8S1, 8-lead (0.150" Wide Body), Plastic Gull Wing Small Outline (JEDEC SOIC)</p>	<p>DRAWING NO. 8S1</p>	<p>REV. B</p>
---	--	-----------------------------------	--------------------------



2 - TSSOP



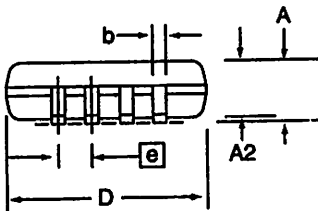
Top View



End View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
D	2.90	3.00	3.10	2, 5
E	6.40 BSC			
E1	4.30	4.40	4.50	3, 5
A	-	-	1.20	
A2	0.80	1.00	1.05	
b	0.19	-	0.30	4
e	0.65 BSC			
L	0.45	0.60	0.75	
L1	1.00 REF			



Side View

- Notes:
1. This drawing is for general information only. Refer to JEDEC Drawing MO-153, Variation AA, for proper dimensions, tolerances, datums, etc.
 2. Dimension D does not include mold Flash, protrusions or gate burrs. Mold Flash, protrusions and gate burrs shall not exceed 0.15 mm (0.006 in) per side.
 3. Dimension E1 does not include inter-lead Flash or protrusions. Inter-lead Flash and protrusions shall not exceed 0.25 mm (0.010 in) per side.
 4. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall be 0.08 mm total in excess of the b dimension at maximum material condition. Dambar cannot be located on the lower radius of the foot. Minimum space between protrusion and adjacent lead is 0.07 mm.
 5. Dimension D and E1 to be determined at Datum Plane H.

5/30/02

<p>2325 Orchard Parkway San Jose, CA 95131</p>	<p>TITLE 8A2, 8-lead, 4.4 mm Body, Plastic Thin Shrink Small Outline Package (TSSOP)</p>	<p>DRAWING NO. 8A2</p>	<p>REV. B</p>
--	---	-----------------------------------	--------------------------





Atmel Corporation

325 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
Shinachem Golden Plaza
7 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

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

Atmel Operations

Memory

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

Microcontrollers

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

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

ASIC/ASSP/Smart Cards

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

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 578-3300
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 Hellbronn, 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) 578-3300
Fax: 1(719) 540-1759

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

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

Literature Requests
www.atmel.com/literature

Disclaimer: 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 in critical components in life support devices or systems.

Atmel Corporation 2004. All rights reserved. Atmel® and combinations thereof, are the registered trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be the trademarks of others.



Printed on recycled paper.

3350C-SEEPR-5/04

xM

LIQUID CRYSTAL DISPLAY MODULE

M 1 6 3 2

USER MANUAL

Seiko Instruments Inc.

PREFACE

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

Revision Record

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

© Seiko Instruments Inc. 1987

Printed in Japan

GENERAL

1.1 General

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

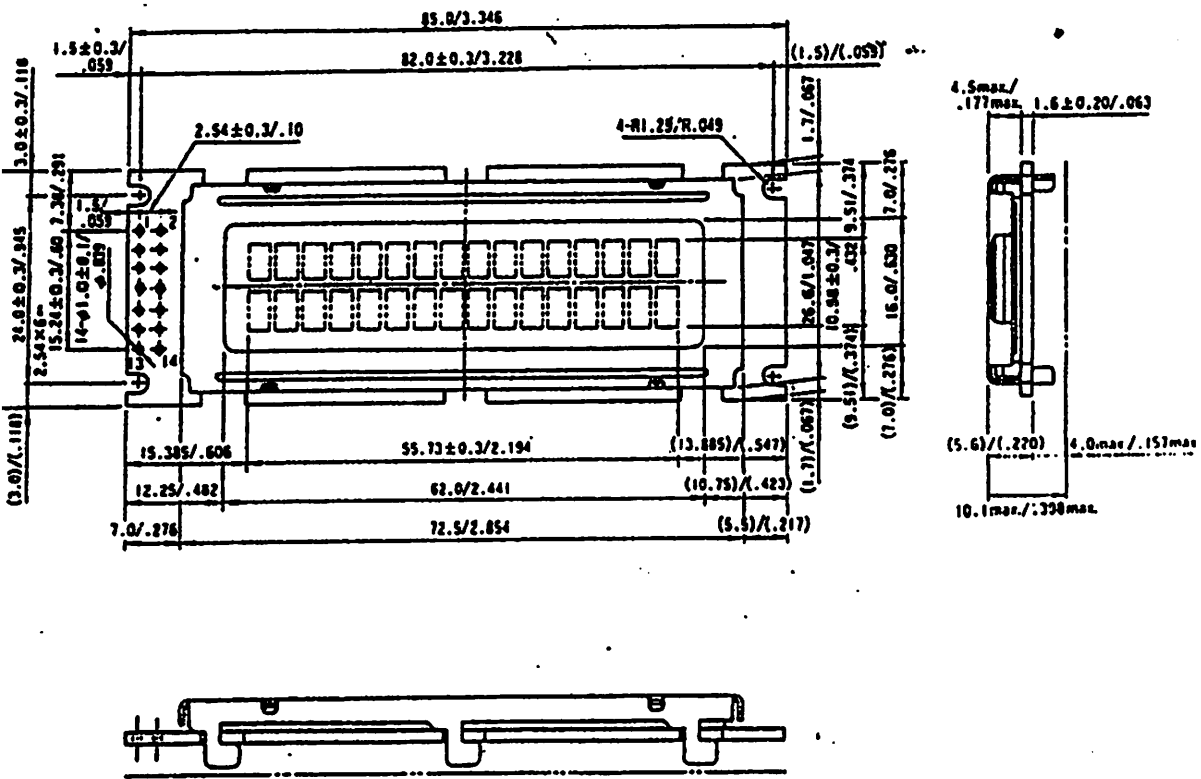
1.2 Features

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

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

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

3 Dimensions Diagram



Unit : mm/inch
General tolerance : ± 0.5 mm

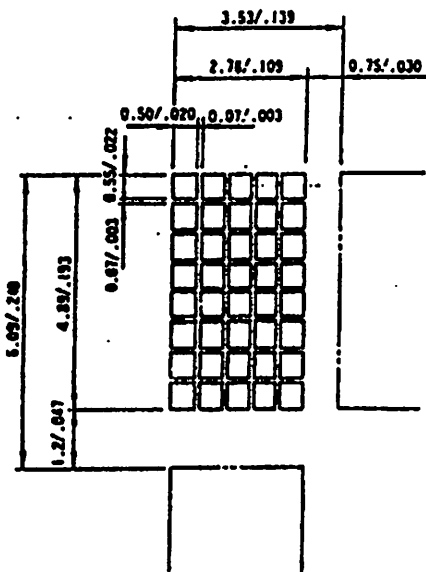
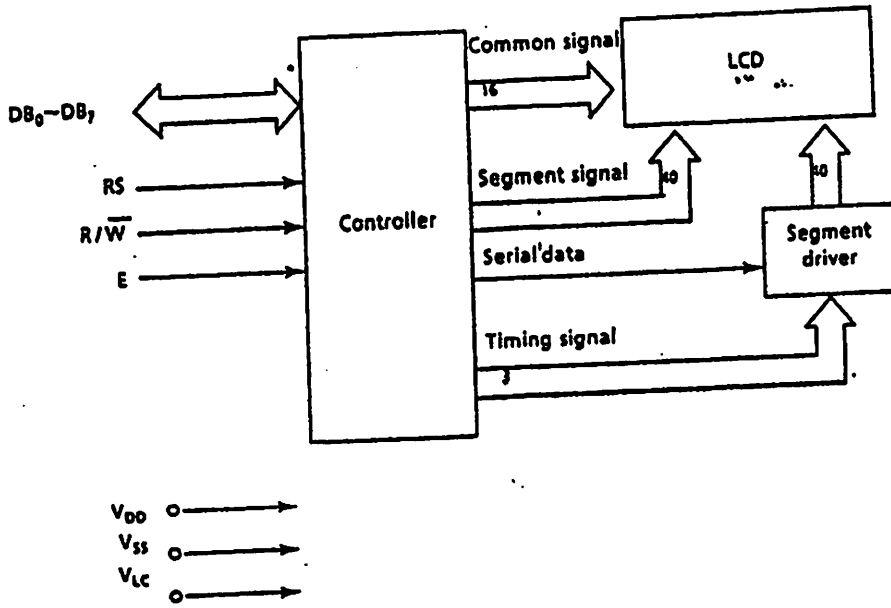


Figure 1 Dimensions diagram

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

Block Diagram



5 Absolute Maximum Ratings

$V_{SS} = 0V$

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

6 Electrical Characteristics

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

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

1.7 Optical Characteristics

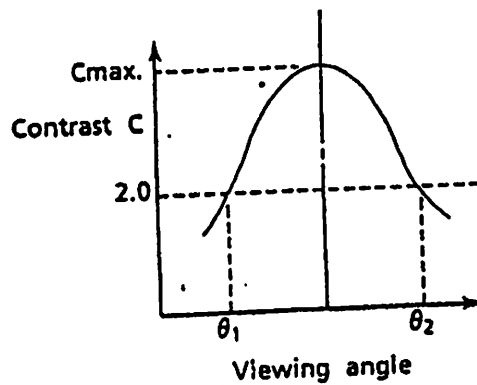
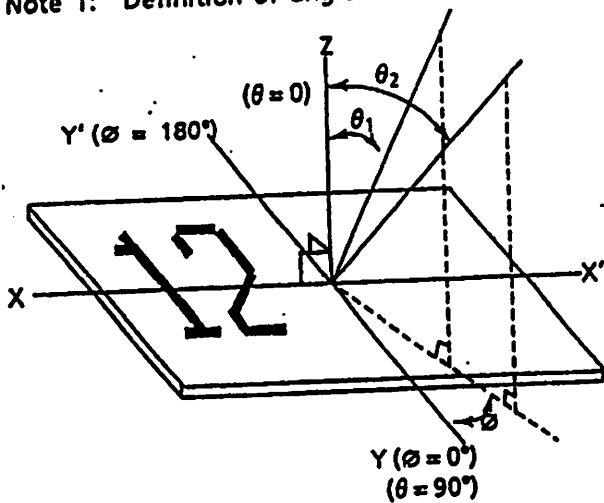
1.7.1 Optical characteristics

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

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

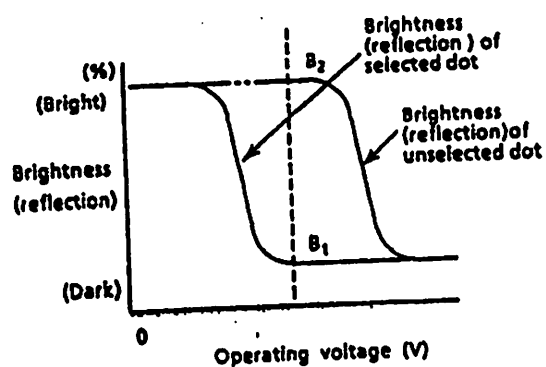
Note 1: Definition of angles \varnothing and θ

Note 2: Definition of viewing angles θ_1 and θ_2

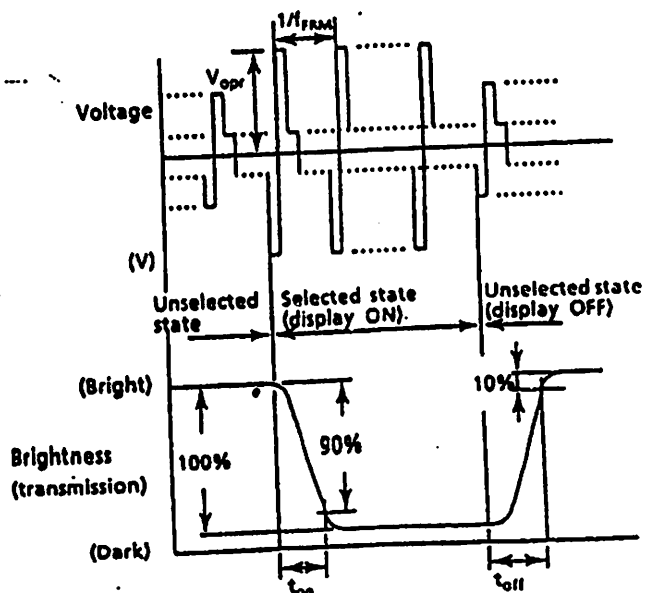


Note 3: Definition of contrast C

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



Note 4: Definition of response time



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

7.2 Recommended operating voltage

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

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

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

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

OPERATING INSTRUCTIONS

2.1 Terminal Functions

Table 1 Terminal functions

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

2 Basic Operations

2.2.1 Registers

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

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

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

Table 2 Register selection

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

2.2.2 Busy flag (BF)

The flag indicates whether the module is ready to accept the next instruction. As shown in Table 2, the signal is output to DB₇ if RS = 0 and $R/\overline{W} = 1$. If the value is 1, the module is working internally and the instruction cannot be accepted. If the value is 0, the next instruction can be written. Therefore, the flag status needs to be checked before executing an instruction. If an instruction is executed without checking the flag status, wait for more than the execution time shown by 2.4 Instruction Outline.

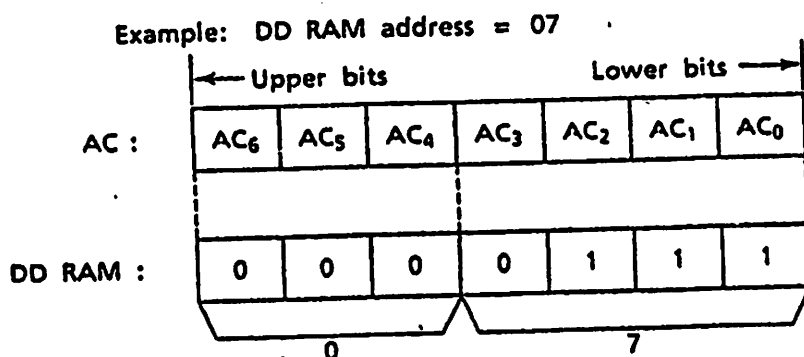
2.2.3 Address counter (AC)

The counter specifies an address when data is written into DD RAM or CG RAM and the data stored in DD RAM or CG RAM is read out. If an Address Set instruction (for DD RAM or CG RAM) is written in the IR, the address information is transferred from the IR to the AC. When display data is written into or read from DD RAM or CG RAM, the AC is automatically incremented or decremented by one according to the Entry Mode Set. The contents of the AC are output to DB₀ to DB₆ as shown in Table 2 if RS = 0 and $R/\overline{W} = 1$.

2.2.4 Display data RAM (DD RAM)

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

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



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

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

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

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

*Left shift

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

*Right shift

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

5 Character generator ROM (CG ROM)

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

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

6 Character generator RAM (CG RAM)

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

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

Table 3 Correspondence between character codes and character patterns

per bit 4 bit bit	0	2	3	7	8	C	7	1010	1011	1100	1101	1110	1111
x x 0000	CG RAM (1)		0	a	P	.	•	-	•	•	•	•	•
x x 0001	(2)	!	1	A	Q	a	•	•	•	•	•	•	•
x x 0010	(3)	"	2	E	R	b	•	•	•	•	•	•	•
x x 0011	(4)	#	3	O	S	c	•	•	•	•	•	•	•
x x 0100	(5)	\$	4	D	T	a	t	.	•	•	•	•	•
x x 0101	(6)	%	5	E	L	e	u	.	•	•	•	•	•
x x 0110	(7)	&	6	F	V	•	•	•	•	•	•	•	•
x x 0111	(8)	'	7	G	W	g	•	•	•	•	•	•	•
x x x 1000	(1)	(8	H	X	h	x	.	•	•	•	•	•
x x x 1001	(2))	9	I	Y	i	y	.	•	•	•	•	•
x x x 1010	(3)	*	:	J	Z	j	z	.	•	•	•	•	•
x x x 1011	(4)	+	;	K	[k	[.	•	•	•	•	•
x x x 1100	(5)	,	<	L]	l]	.	•	•	•	•	•
x x x 1101	(6)	-	=	M	^	m	^	.	•	•	•	•	•
x x x 1110	(7)	.	>	N	_	n	_	.	•	•	•	•	•
x x x 1111	(8)	/	?	O	~	o	~	.	•	•	•	•	•

Table 4 Relationships between CG RAM addresses and character codes (DD RAM) and character patterns (CG RAM data)

Character code (DD RAM data)		CG RAM address				Character pattern (CG RAM data)									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Upper bit → Lower bit →		← Upper bit		Lower bit →		← Upper bit		Lower bit →							
0 0 0 0 * 0 0 0 0		0 0 0		0 0 0		0 0 0		* * *				Example of character pattern (R)		← Cursor position	
0 0 0 0 * 0 0 0 1		0 0 1		0 1 1		0 1 1		* * *				Example of character pattern (Y)			
0 0 0 0 * 1 1 1 1		1 1 1		1 0 0		1 0 0		* * *							

Notes: • In CG RAM data, 1 corresponds to Selection and 0 to Non-selection on the display.

- Character code bits 0 to 2 and CG RAM address bits 3 to 5 correspond with each other (three bits, eight types).
- CG RAM address bits 0 to 2 specify a line position for a character pattern. Line 8 of a character pattern is the cursor position where the logical sum of the cursor and CG RAM data is displayed. Set the data of line 8 to 0 to display the cursor. If the data is changed to 1, one bit lights, regardless of the cursor.

The character pattern column positions correspond to CG RAM data bits 0 to 4 and bit 4 comes to the left end. CG RAM data bits 5 to 7 are not displayed but can be used as general data RAM.

When reading a character pattern from CG RAM, set to 0 all of character code bits 4 to 7. Bits 0 to 2 determine which pattern will be read out. Since bit 3 is not valid, 00H and 08H select the same character.

Timing Characteristics

1 Write timing characteristics

$$V_{DD} = 5.0V \pm 5\%, V_{SS} = 0V, T_A = 0^\circ\text{C to } 50^\circ\text{C}$$

Item	Symbol	Standard		Unit	
		Min.	Max.		
Enable cycle time	t_{cycE}	1000	-	ns	
Enable pulse width	High level	PW_{EH}	450	-	ns
Enable rise and fall time	t_{Er}, t_{Ef}	-	25	ns	
Setup time	t_{AS}	140	-	ns	
Address hold time	t_{AH}	10	-	ns	
Data setup time	t_{DSW}	195	-	ns	
Data hold time	t_H	10	-	ns	

Write operation

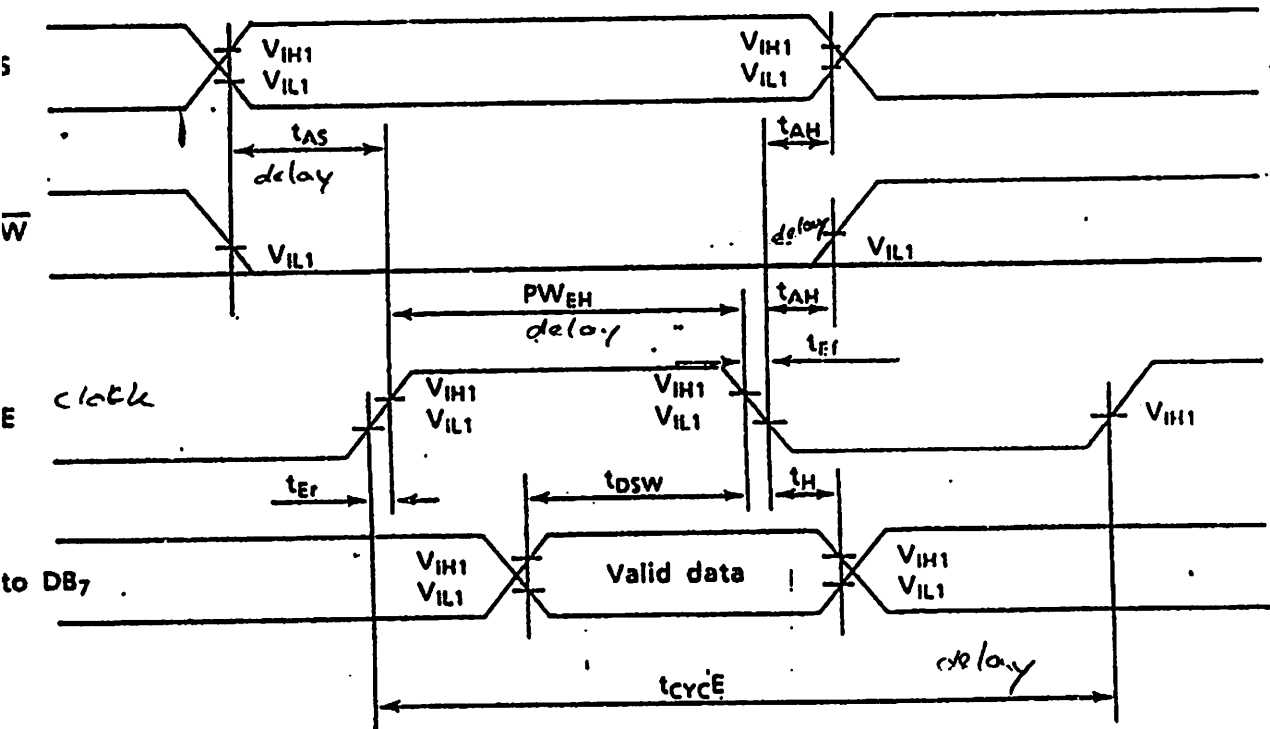


Figure 3 Data write from MPU to module

2 Read timing characteristics

$V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$; $T_A = 0^\circ C$ to $50^\circ C$

Item	Symbol	Standard		Unit	
		Min.	Max.		
Enable cycle time	t_{cycE}	1000	-	ns	
Enable pulse width	High level	PW_{EH}	450	-	ns
Enable rise and fall time	t_{Er}, t_{Ef}	-	25	ns	
Setup time	$RS, \overline{R/W}-E$	t_{AS}	140	-	ns
Address hold time	t_{AH}	10	-	ns	
Data delay time	t_{DDR}	-	320	ns	
Data hold time	t_{Hl}	20	-	ns	

Read operation

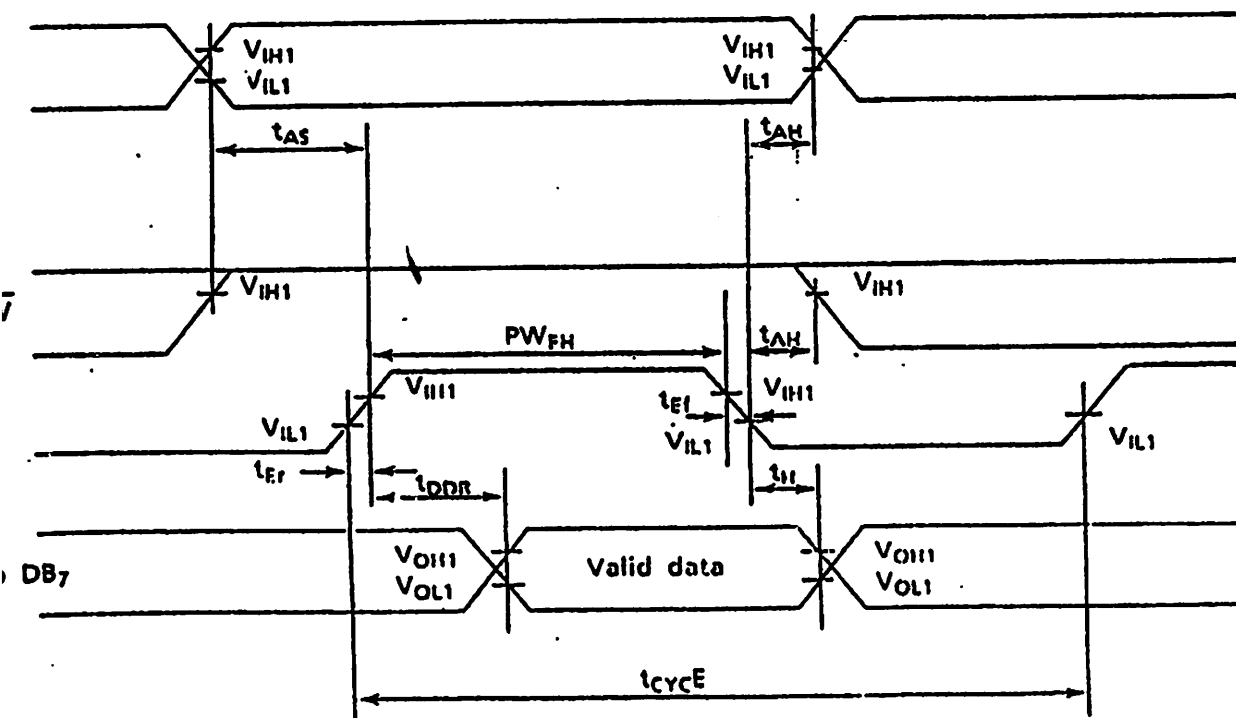


Figure 4 Data read from module to MPU

Instruction Outline

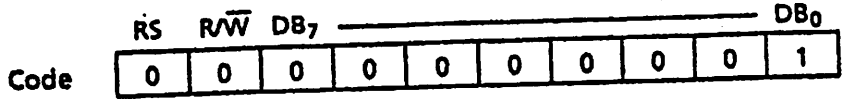
Table 5 List of instructions

Instruction	Code											Function	Execution time
	AS	AV	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Display clear ✓	0	0	0	0	0	0	0	0	0	0	1	Clears all display and returns cursor to home position (address 0)	1.64 ms
Return Home ✓	0	0	0	0	0	0	0	0	0	1	0	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
Mode Set ✓	0	0	0	0	0	0	0	0	1	WD	0	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 μs
Display ON/OFF Control	0	0	0	0	0	0	0	1	DL	C	B	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 μs
Cursor/Display Shift	0	0	0	0	0	0	1	S/C	RL	0	0	Moves cursor and shifts display without changing DD RAM contents	40 μs
Interface Set ✓	0	0	0	0	1	DL	1	0	0	0	0	Sets interface data length (DL)	40 μs
CG RAM Address	0	0	0	1	Acc						Sets CG RAM address to start transmitting or receiving CG RAM data	40 μs	
DD RAM Address	0	0	1	Add						Sets DD RAM address to start transmitting or receiving DD RAM data	40 μs		
Internal Address Read	0	1	BF	AC						Reads BF indicating module in internal operation and AC contents (used for both CG RAM and DD RAM)	0 μs		
Data Write to CG RAM or DD RAM	1	0	Write Data						Writes data into DD RAM or CG RAM	40 μs			
Data Read from CG RAM or DD RAM	1	1	Read Data						Reads data from DD RAM or CG RAM	40 μs			

- Valid bit
- CG RAM address
- DD RAM address
- S = 1 : Display shift
- S = 0 : No display shift
- D = 1 : Display ON
- D = 0 : Display OFF
- WD = 1 : Increment
- WD = 0 : Decrement
- S/C = 1 : Display shift
- S/C = 0 : Cursor movement
- C = 1 : Cursor ON
- C = 0 : Cursor OFF
- B = 1 : Blink ON
- B = 0 : Blink OFF
- RL = 1 : Right shift
- RL = 0 : Left shift
- DL = 1 : 8 bits
- DL = 0 : 4 bits
- BF = 1 : Internal operation in progress
- BF = 0 : Instruction can be accepted

Instruction Details

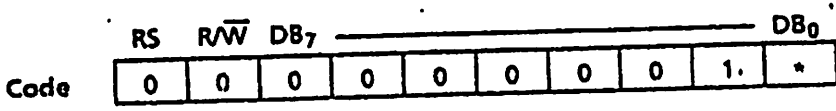
Display Clear



Display Clear clears all display and returns cursor to home position (address 0). Space code 20 (hexadecimal) is written into all the addresses of DD RAM, and DD RAM address 0 is set to the AC. If shifted, the display returns to the original position. After execution of the Display Clear instruction, the entry mode is incremented.

Note: When executing the Display Clear instruction, follow the restrictions listed in Table 6.

Cursor Home



* : Invalid bit

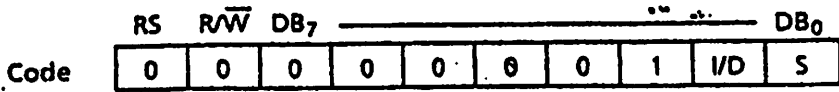
Cursor Home returns cursor to home position (address 0). DD RAM address 0 is set to the AC. The cursor returns to the home position. If shifted, the display returns to the original position. The DD RAM contents do not change. If the cursor or blinking is ON, it returns to the left side.

Note: When executing the Cursor Home instruction, follow the restrictions listed in Table 6.

Table 6 Restrictions on execution of Display Clear and Cursor Home instructions

Conditions of use	Restrictions
When executing the Display Clear or Cursor Home instruction when the display is shifted (after execution of Display Shift instruction)	The Cursor Home instruction should be executed again immediately after the Display Clear or Cursor Home instruction is executed. Do not leave an interval of a multiple of $400/f_{osc}$ second after the first execution. Example: 1.5 ms, 3 ms, 4.5 ms for $f_{osc} = 270$ kHz * f_{osc} : Oscillation frequency
When 23 ₁₁ , 77 ₁₁ , 63 ₁₁ , or 67 ₁₁ is used as a DD RAM address to execute Cursor Home instruction	Before executing the Cursor Home instruction, the data of the four DD RAM addresses given at the left should be read and saved. After execution, write the data again in DD RAM. (This restriction is necessary to prevent the contents of the DD RAM addresses from being destroyed after the Cursor Home instruction has been executed.)

Entry Mode Set



Entry Mode Set sets the direction of cursor movement and whether display will be shifted.

I/D : The DD RAM address is incremented or decremented by one when a character code is written into or read from DD RAM. This is also true for writing into or reading from CG RAM.

When I/D = 1, the address is incremented by one and the cursor or blink moves to the right.

When I/D = 0, the address is decremented by one and the cursor or blink moves to the left.

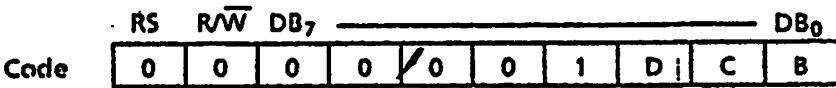
S : If S = 1, the entire display is shifted either to the right or left for writing into DD RAM. The cursor position does not change, only the display moves. There is no display shift for reading from DD RAM.

When S = 1 and I/D = 1, the display shifts to the left.

When S = 1 and I/D = 0, the display shifts to the right.

If S = 0, the display does not shift.

Display ON/OFF Control .



Display ON/OFF Control turns the total display and the cursor ON and OFF, and makes the cursor position start blinking. Cursor ON/OFF and blinking is done at the column indicated by the specified DD RAM address by the AC.

D : When D = 1, the display is turned ON.

When D = 0, the display is turned OFF.

If D = 0 is used, display data remains in DD RAM. Change 0 to 1 to display data.

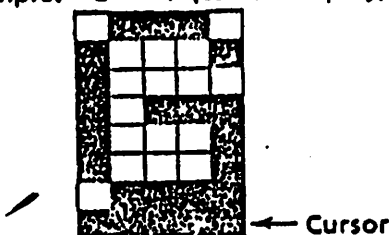
C : When $C = 1$, the cursor is displayed.
 When $C = 0$, the cursor is not displayed.

The cursor is displayed in the dot line below the 5 x 7 dot-matrix character fonts. If the cursor is OFF, display data is written into DD RAM in the order specified by I/D.

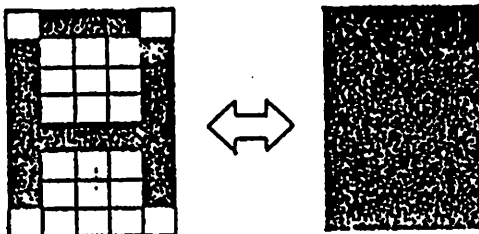
B : When $B = 1$, the character at the cursor position starts blinking.
 When $B = 0$, it does not blink.

For blinking, all-black dots and the character are switched about every 0.4 seconds. The cursor and blinking can be set at the same time.

Example: $C = 1$ (cursor display)



$B = 1$ (blinking)



Cursor/Display Shift



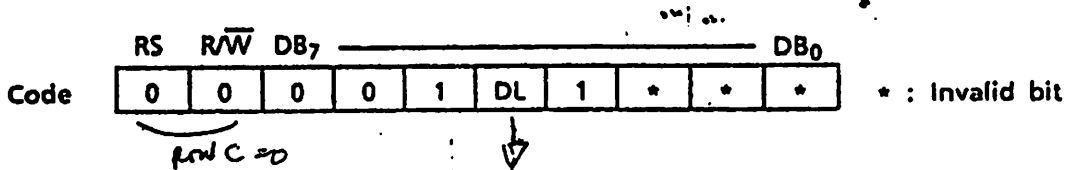
Cursor/Display Shift moves the cursor and shifts the display without changing the DD RAM contents.

The cursor position and the AC contents match. This instruction is available for display correction and retrieval because the cursor position or display can be shifted without writing or reading display data. Since the DD RAM capacity is 40-character and two lines, the cursor is shifted from digit 40 of line 1 to digit 1 of line 2. Displays of lines 1 and 2 are shifted at the same time. Therefore, the display pattern of line 2 is not shifted to line 1.

S/C	R/L	Operation
0	0	The cursor position is shifted to the left (the AC decrements one).
0	1	The cursor position is shifted to the right (the AC increments one).
1	0	The entire display is shifted to the left with the cursor.
1	1	The entire display is shifted to the right with the cursor.

Note: If only display shift is done, the AC contents do not change.

Function Set



Function Set sets the interface data length.

DL : Interface data length

When DL = 1, the data length is set at eight bits (DB₇ to DB₀).

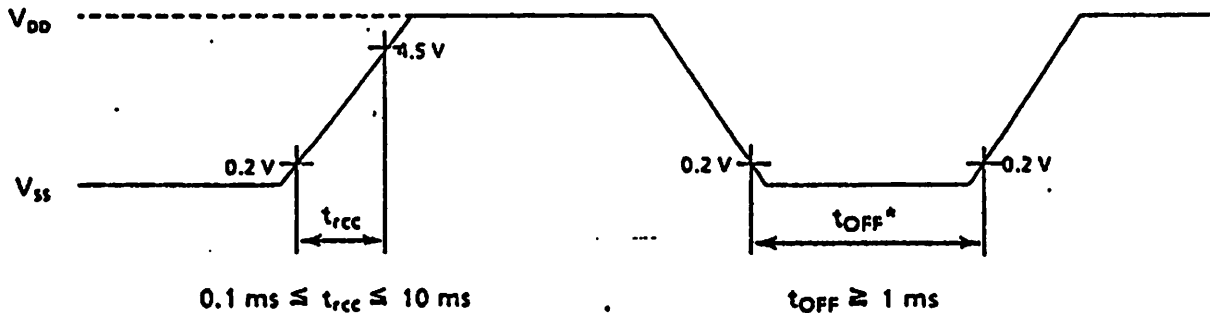
When DL = 0, the data length is set at four bits (DB₇ to DB₄).

The upper four bits are transferred first, then the lower four bits follow.

The Function Set instruction must be executed prior to all other instructions except for Busy Flag/Address Read. If another instruction is executed first, no function instruction except changing the interface data length can be executed.

Remarks: Initialization

The system is automatically initialized at power-on if the following power supply conditions are satisfied.



* t_{off} : Time when power supply is OFF if cut instantaneously or turned ON and OFF repeatedly

following instructions are executed for initialization.

5 x 7 dot-matrix character font: 1/8 duty

Display clear

Function Set

DL = 1: Interface data length: 8 bits

Display ON/OFF Control

D = 0: Display OFF

C = 0: Cursor OFF

B = 0: Blink OFF

Entry mode

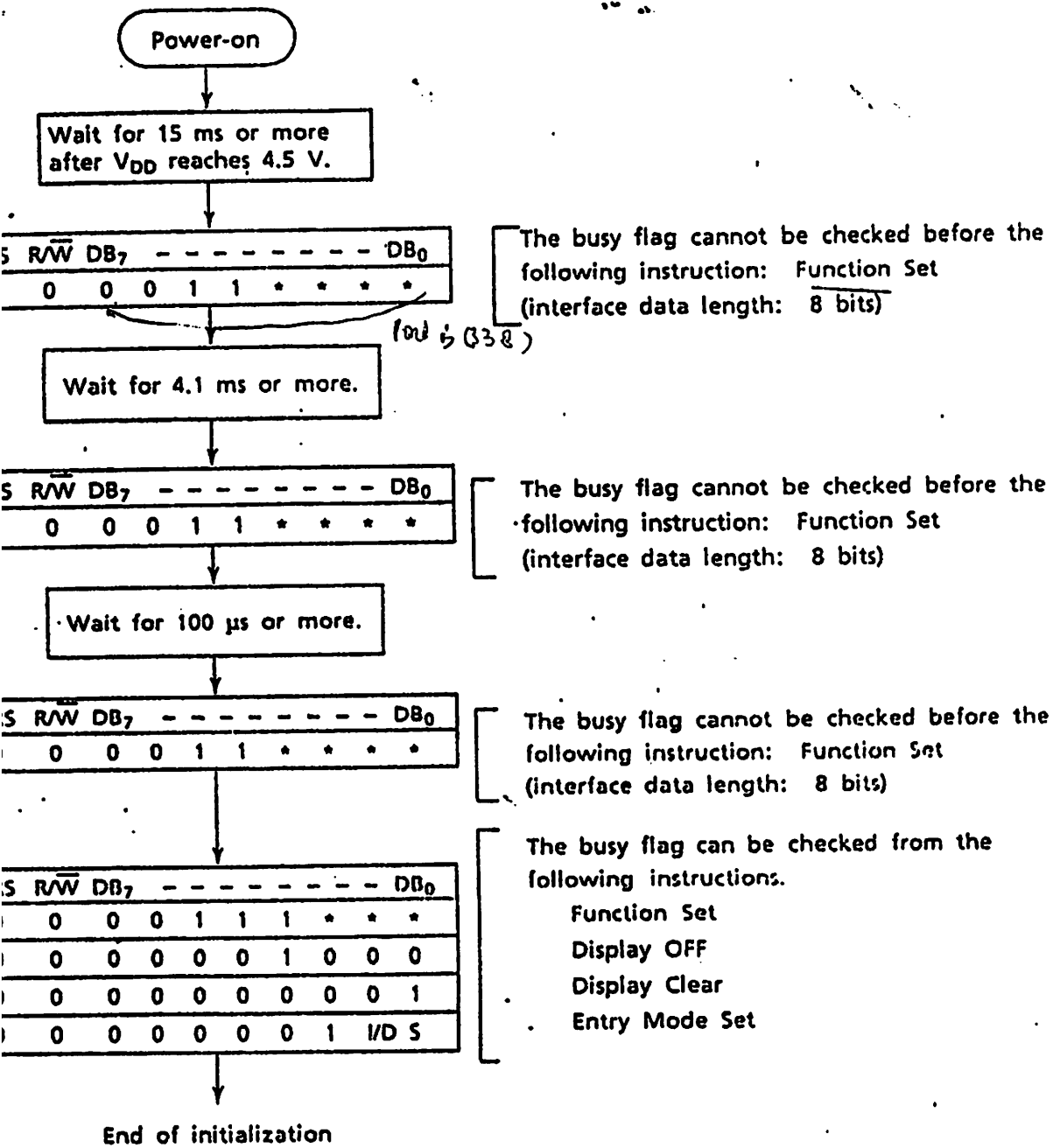
I/O = 1: Increment

S = 0: No display shift

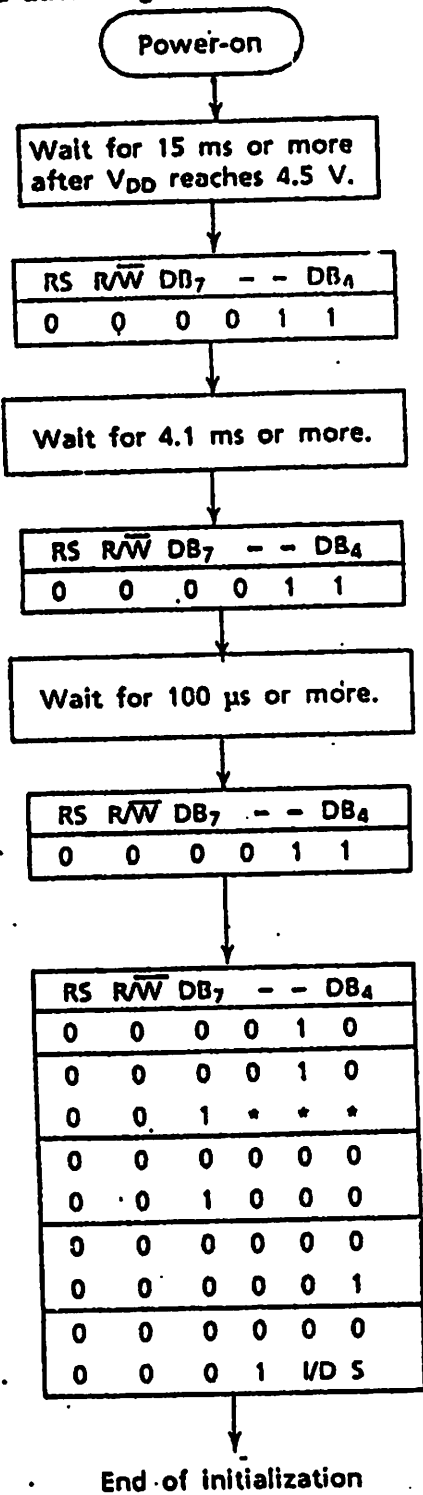
Since the condition is not suitable for the M1632, further function setting is necessary.

If automatic initialization is not executed because the above power supply conditions are not satisfied, use the instruction from next page on.

Interface data length : Eight bits



Interface data length: Four bits



The busy flag cannot be checked before the following instruction: Function Set (interface data length: 8 bits)

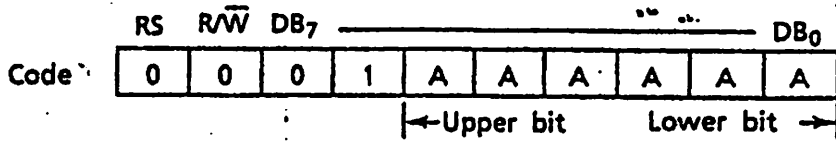
The busy flag cannot be checked before the following instruction: Function Set (interface data length: 8 bits)

The busy flag cannot be checked before the following instruction: Function Set (interface data length: 8 bits)

The busy flag can be checked from the following instructions.

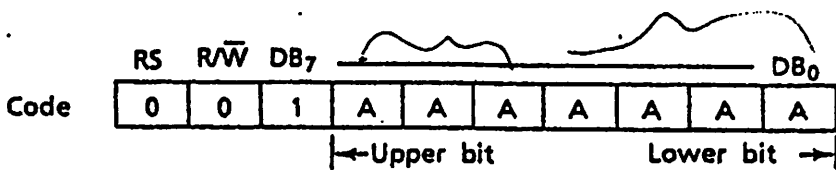
- Function Set (interface data length: 4 bits)
- Function Set
- Display OFF
- Display Clear
- Entry Mode Set

CG RAM Address Set



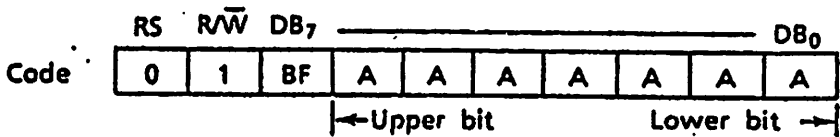
CG RAM addresses expressed as binary AAAAAA are set to the AC. Then data in CG RAM is written from or read to the MPU.

DD RAM Address Set



DD RAM addresses expressed as binary AAAAAAA are set to the AC. Then data in DD RAM is written from or read to the MPU. The addresses used for display in line 1 (AAAAAAA) are 00H to 27H and those for line 2 (AAAAAAA) are 40H to 67H.

Busy Flag/Address Read



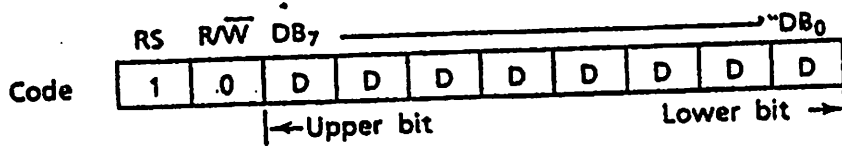
The BF signal is read out, indicating that the module is working internally because of the previous instruction.

When BF = 1, the module is working internally and the next instruction cannot be accepted until the BF value becomes 0.

When BF = 0, the next instruction can be accepted.

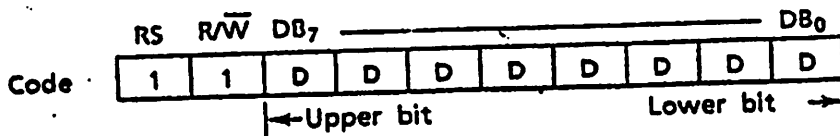
Therefore, make sure that BF = 0 before writing the next instruction. The AC values of binary AAAAAAA are read out at the same time as reading the busy flag. The AC addresses are used for both CG RAM and DD RAM but the address set before execution of the instruction determines which address is to be used.

Data Write to CG RAM or DD RAM



Binary eight-bit data DDDDDDDDD is written into CG RAM or DD RAM. The CG RAM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) before this instruction selects either RAM. After the write operation, the address and display shift are determined by the entry mode setting.

Data Read from CG RAM or DD RAM



Binary eight-bit data DDDDDDDDD is read from CG RAM or DD RAM. The CG RAM Address Set instruction of (7) or the DD RAM Address Set instruction of (8) before this instruction selects either RAM. In addition, either instruction (7) or (8) must be executed immediately before this instruction. If no address set instruction is executed before a read instruction, the first data read becomes invalid. If read instructions are executed consecutively, data is normally read from the second time. However, if the cursor is shifted by the Cursor Shift instruction when reading DD RAM, there is no need to execute an address set instruction because the Cursor Shift instruction does this.

After the read operation, the address is automatically incremented or decremented by one according to the entry mode, but the display is not shifted.

Note : The AC is automatically incremented or decremented by one according to the entry mode after a write instruction is executed to write data in CG RAM or DD RAM. However, the data of the RAM selected by the AC are not read out even if a read instruction is executed immediately afterwards.

Correct data is read out under the following conditions.

- An address set instruction is executed immediately before readout.
- For DD RAM, the Cursor Shift instruction is executed immediately before readout.
- The second, or later, instruction is executed in consecutive execution of read instructions.

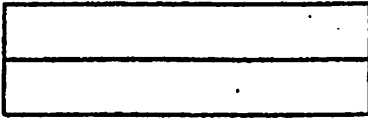
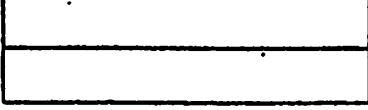
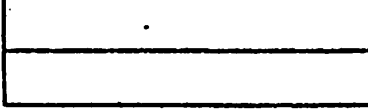


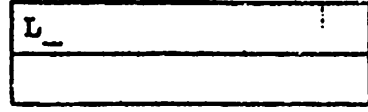
Examples of Instruction Use

Interface data length: Eight bits

Instruction	Display	Operation										
<p>Power-on</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>/</td> <td>/</td> <td colspan="3">/</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	/	/	/				The built-in reset circuit initializes the module.
RS	R/W	DB ₇	—	DB ₀								
/	/	/										
<p>Function Set ✓</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 1 1</td> <td>*</td> <td>*</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 1 1 1	*	*		The interface data length is set to 8 bits. The character format becomes 5 x 7 dot-matrix at 1/16 duty cycle.
RS	R/W	DB ₇	—	DB ₀								
0	0	0 0 1 1 1	*	*								
<p>Display ON/OFF Control</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 1 1</td> <td>1</td> <td>1 0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 1 1	1	1 0		The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB ₇	—	DB ₀								
0	0	0 0 0 0 1 1	1	1 0								
<p>Entry Mode Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 0 1</td> <td>1</td> <td>1 0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 0 1	1	1 0		The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
RS	R/W	DB ₇	—	DB ₀								
0	0	0 0 0 0 0 1	1	1 0								
<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 1 1</td> <td>0</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 1 1	0	0		L is written. The AC is incremented by one and the cursor shifts to the right.
RS	R/W	DB ₇	—	DB ₀								
1	0	0 1 0 0 1 1	0	0								
<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 0 0</td> <td>1</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 0 0	1	1		C is written.
RS	R/W	DB ₇	—	DB ₀								
1	0	0 1 0 0 0 0	1	1								
⋮	⋮											
<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 0 1 1 0 0</td> <td>1</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 0 1 1 0 0	1	0		2 is written in digit 16. Cursor disappears.
RS	R/W	DB ₇	—	DB ₀								
1	0	0 0 1 1 0 0	1	0								

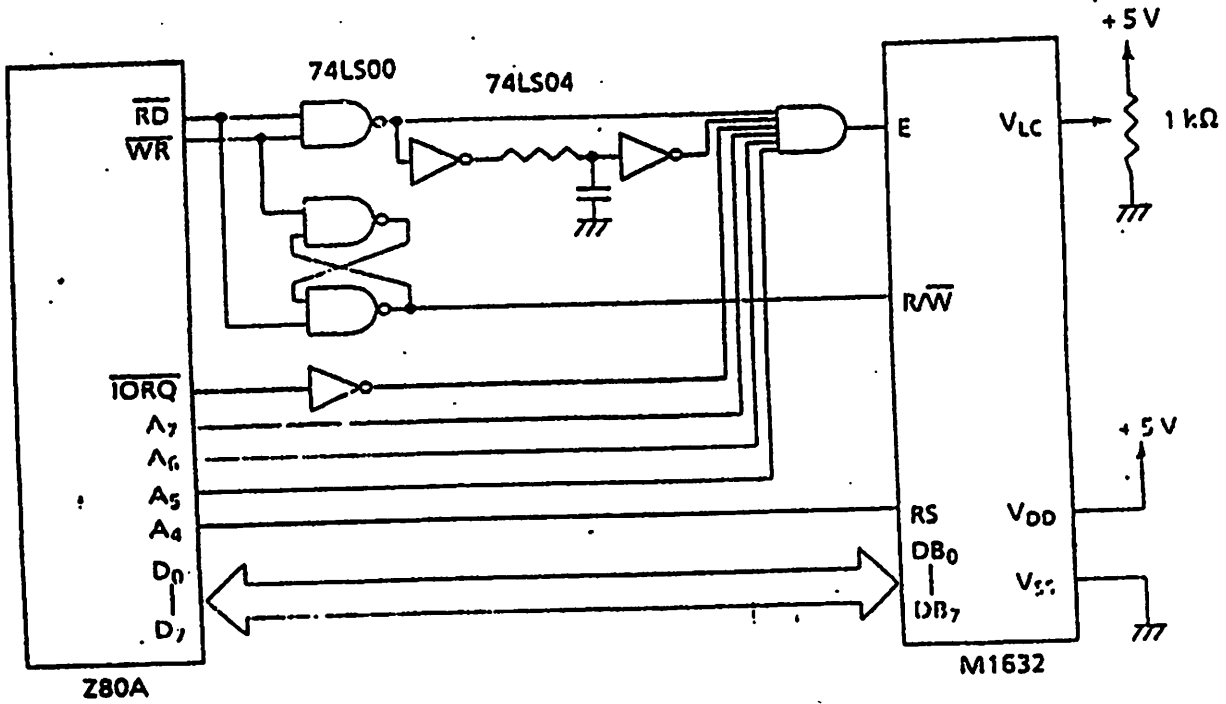
Instruction	Display	Operation										
DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0 0 0 0 0 0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	1	1	0 0 0 0 0 0	LCD MODULE M1632 —	The DD RAM address is set so that the cursor appears at digit 1 of line 2.
RS	R/W	DB ₇	—	DB ₀								
0	0	1	1	0 0 0 0 0 0								
Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1 1 0 0 0 1</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0	0	1 1 0 0 0 1	LCD MODULE M1632 1_	1 is written.
RS	R/W	DB ₇	—	DB ₀								
1	0	0	0	1 1 0 0 0 1								
Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1 1 0 1 1 0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0	0	1 1 0 1 1 0	LCD MODULE M1632 16_	6 is written.
RS	R/W	DB ₇	—	DB ₀								
1	0	0	0	1 1 0 1 1 0								
⋮	⋮											
Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0 1 0 0 1 1</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0	1	0 1 0 0 1 1	LCD MODULE M1632 16DIGITS, 2LINES	5 is written.
RS	R/W	DB ₇	—	DB ₀								
1	0	0	1	0 1 0 0 1 1								
DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0 0 0 0 0 0</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	1	0	0 0 0 0 0 0	LCD MODULE M1632 16DIGITS, 2LINES	The cursor returns to the home position.
RS	R/W	DB ₇	—	DB ₀								
0	0	1	0	0 0 0 0 0 0								
Display clear <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0 0 0 0 0 0 0 1</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0	0	0 0 0 0 0 0 0 1	—	All the display disappears and the cursor remains at the home position.
RS	R/W	DB ₇	—	DB ₀								
0	0	0	0	0 0 0 0 0 0 0 1								
⋮	⋮											

Interface data length: Four bits

Instruction	Display	Operation									
<p>Power-on</p> <table border="1" data-bbox="28 414 399 567"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	/	/	/		<p>The built-in reset circuit initializes the module.</p>			
RS	R/W	DB ₇ — DB ₄									
/	/	/									
<p>Function Set</p> <table border="1" data-bbox="28 644 399 797"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	/	/	/		<p>Four-bit operation mode is set. *Eight-bit operation mode is set by initialization, and the instruction is executed only once.</p>
RS	R/W	DB ₇ — DB ₄									
0	0	0 0 1 0									
/	/	/									
<p>Function Set</p> <table border="1" data-bbox="28 873 399 1026"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 * * *</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	0	0	1 * * *		<p>The 4-bit operation mode, 1/16 duty cycle, and 5 x 7 dot-matrix character format are selected. Then 4-bit operation mode starts.</p>
RS	R/W	DB ₇ — DB ₄									
0	0	0 0 1 0									
0	0	1 * * *									
<p>Display ON/OFF Control</p> <table border="1" data-bbox="28 1102 399 1255"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	1 1 1 0		<p>The display and cursor are turned ON, but nothing is displayed.</p>
RS	R/W	DB ₇ — DB ₄									
0	0	0 0 0 0									
0	0	1 1 1 0									
<p>Entry Mode Set</p> <table border="1" data-bbox="28 1321 399 1474"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	0 1 1 0		<p>The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.</p>
RS	R/W	DB ₇ — DB ₄									
0	0	0 0 0 0									
0	0	0 1 1 0									
<p>Write to CG RAM or DD RAM.</p> <table border="1" data-bbox="28 1550 399 1703"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1 1 0 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	1	0	0 1 0 0	1	0	1 1 0 0		<p>L is written. the AC is incremented by one and the cursor shifts to the right.</p>
RS	R/W	DB ₇ — DB ₄									
1	0	0 1 0 0									
1	0	1 1 0 0									

PU Connection Diagrams

Z80A



7.2 Z80A and 8255A

