

# **SKRIPSI**

**PERANCANGAN DAN PEMBUATAN INTERIOR MEJA TELEVISI  
MENGUNAKAN REMOTE CONTROL  
BERBASIS MIKROKONTROLLER ATMEL AT89S8252**



**Disusun Oleh :**

**INDRA GUSNIWAN**

**03.17.055**

**INSTITUT TEKNOLOGI NASIONAL MALANG  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK ELEKTRONIKA  
FEBRUARI 2010**

SECRET

SECRET AREA CONTAINS MATERIALS AND INFORMATION  
WHICH ARE UNCLASSIFIED  
EXCEPT WHERE SHOWN OTHERWISE

SECRET

UNCLASSIFIED

SECRET

SECRET AREA CONTAINS MATERIALS AND INFORMATION  
WHICH ARE UNCLASSIFIED  
EXCEPT WHERE SHOWN OTHERWISE  
SECRET

HALAMAN JUDUL

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**FAKULTAS TEKNOLOGI INDUSTRI**

**JURUSAN TEKNIK ELEKTRO S-1**

**KONSENTRASI TEKNIK ELEKTRONIKA**



**PERANCANGAN DAN PEMBUATAN INTERIOR MEJA TELEVISI  
MENGUNAKAN REMOTE CONTROL  
BERBASIS MIKROKONTROLLER ATMEL AT89S8252**

**SKRIPSI**

Disusun Oleh :  
**INDRA GUSNIAWAN**  
03.17.055

# LEMBAR PENGESAHAN



## PERANCANGAN DAN PEMBUATAN INTERIOR MEJA TELEVISI MENGUNAKAN REMOTE CONTROL BERBASIS MIKROKONTROLLER ATMEL AT89S8252

### SKRIPSI

*Disusun dan diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik  
Elektronika Strata Satu (S-1)*

Disusun Oleh :  
**INDRA GUSNIAWAN**  
03.17.055

Diperiksa dan disetujui :

Dosen Pembimbing I

  
( Irmalia Suryani F, ST, MT )  
NIP. 103 0000 365

Dosen Pembimbing II

  
( Ir. Eko Nurcahyo )  
NIP. 102 8700 172



Mengetahui :

~~Ketua Jurusan Teknik Elektro S-1~~

  
( Ir. F. Yudi Limpraptono, MT. )  
NIP. 1 039 500 274

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



INSTITUT TEKNOLOGI NASIONAL  
Jl. Bendungan Sigura-gura No. 2  
MALANG

**BERITA ACARA UJIAN SKRIPSI**  
**Fakultas Teknologi Industri**

Nama Mahasiswa : INDRA GUSNIAWAN  
NIM : 03 17 055  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan Dan Pembuatan Interior Meja Televisi  
Menggunakan Remote Control Berbasis  
Mikrokontroller Atmel AT89S8252

Dipertahankan dihadapan team penguji skripsi jenjang strata satu (S-1) pada

Hari : Selasa  
Tanggal : 9 Febuari 2010  
Dengan Nilai : 79,8 (B+) *BY*

**Panitia Penguji Skripsi**

Ketua Majelis Penguji

Ir. Sidik Noerdjahjono, MT.  
NIP. Y. 1028700163

Sekretaris Majelis Penguji

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

Penguji Pertama

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

Anggota Penguji

Penguji Kedua

Ir. TH Mimien Mustikawati, MT  
NIP. Y. 1030000352

## ABSTRAK

### Perencanaan Dan Pembuatan Interior Meja Televisi Menggunakan Remote Control Berbasis Mikrokontroler Atmel AT89S8252

Indra Gusniawan

03 17 055

Jurusan Teknik Elektronika S-1 – Institut Teknologi Nasional Malang

Jl .Raya Karanglo Km2 Malang

[Indra\\_gusniawan@yahoo.co.id](mailto:Indra_gusniawan@yahoo.co.id)

Dosen pembimbing : I. Irmalia Suryani F, ST,MT

II. Ir. Eko Nurcahyo

**Kata kunci** : Remote, lcd, Mikro AT89S8252, Handphone

Remote infra red sangat banyak digunakan dan sangat mudah mengoperasikan sehingga dapat difungsikan sebagai pengontrol interior. Untuk kemudahan mengontrol interior ditambahkan pula tombol manual agar dapat mengantisipasi kegagalan penggunaan remote. Sistem SMS pun dipasang menggunakan jalur GSM yang dapat menginformasikan dan mengontrol oleh pemilik jika tidak berada ditempat. Pada tulisan ini dibahas tentang sistem melalui handphone. Pengujian yang dilakukan meliputi : pengujian terhadap fungsi remote control, tombol manual, sms menggunakan AT Command, pengiriman SMS, data In/Out RS 232, dan sistem display pada LCD 16x2. Pengujian dilakukan untuk mengetahui apakah sistem yang dibuat bisa berjalan sesuai dengan perancangan. Hasil menunjukkan bahwa sistem dapat berjalan/berfungsi.

## KATA PENGANTAR

Dengan memanjatkan puji syukur kepa Tuhan Yang Maha Esa atas berkat, rahmat dan karunia-Nya sehingga dapat menyelesaikan skripsi yang berjudul perancangan dan pembuatan interior meja televisi menggunakan remote control berbasis mikrokontroler AT89S8252 dengan lancar. Skripsi ini merupakan persyaratan kelulusan studi di jurusan Teknik Elektro S1 konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar serjana.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan dari pihak. Untuk itu penyusunan meyampaikan terima kasih kepada:

1. Bapak Prof.Dr. Eng. Ir. Abraham Lomi, MSEE selaku rector ITN Malang.
2. Bapak Ir. Sidik Noertjahjono, MT sekalu Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S1
4. Ibu Irmalia Suryani F, ST, MT selaku dosen pembimbing I
5. Bapak Ir. Eko Nurcahyo selaku dosen pembimbing II
6. Orang tua serta saudara-saudara kami yang telah memberikan doa restunya dan motivasi semangat dan biaya.
7. Teman-teman instruktur Lab Perancangan Elektronika dan team robot KRI yang telah membantu
8. Semua pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini yang tidak dapat saya sebutkan satu-persatu.

Penyusun telah berusaha semaksimal mungkin dan meyakini sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini, untuk itu penyusun

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

Harapan penyusun semoga laporan ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan bagi pembaca

Malang, Februari 2010

Penyusun



## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN .....	ii
BERITA ACARA UJIAN SKRIPSI.....	iii
ABSTRAK.....	iv
KATA PENGANTAR .....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan masalah.....	2
1.3. Tujuan .....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian .....	3
1.6. Sistematika Penulisan.....	4
BAB II DASAR TEORI .....	6
2.1. Pendahuluan .....	6
2.2. Mikrokontroller .....	6
2.2.1. ATMEL AT89S8252.....	6
2.3. Remote Control .....	18
2.4. Swicth.....	21
2.4.1. <i>Push Botton</i> (Saklar Tekan) .....	21

2.4.2. Limit Swicth.....	22
2.5. LCD ( <i>Liquid Cristal Display</i> ) .....	23
2.6. Transistor.....	24
2.7. Resistor.....	25
2.8. RELAY .....	28
2.9. Motor DC .....	31
2.10. <i>Handphone</i> Siemens C55 .....	35
2.11. RTC ( <i>Real Time Clock</i> ).....	45
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>48</b>
3.1. Pendahuluan .....	48
3.2. Blok Diagram Alat .....	49
3.3. Prinsip Kerja Alat.....	51
3.4. Perancangan Perangkat Keras .....	53
3.4.1. Remote control .....	53
3.4.1.1. Remote pengirim .....	53
3.4.1.2. Penerima remote.....	54
3.4.2. Tombol Manual .....	55
3.4.3. Sensor .....	56
3.4.3.1. Sensor Buka Paksa .....	56
3.4.3.2. Sensor Sudut Putar .....	56
3.4.4. Driver Motor DC.....	57
3.4.5. Rangkaian Timer .....	58
3.4.6. Rangkaian LCD.....	59
3.4.7. Alarm .....	60
3.4.8. Interface serial .....	62

3.5. Perancangan Perangkat Lunak .....	63
3.5.1. Step Pertama.....	63
3.5.2. Step Kedua .....	64
3.5.3. Step Ketiga .....	65
<b>BAB IV PENGUJIAN DAN HASIL ANALISA.....</b>	<b>66</b>
4.1. Pendahuluan .....	66
4.2. Penguji Rangkaian Penerima Remote.....	66
4.2.1. Tujuan .....	66
4.2.2. Alat Yang Digunakan.....	66
4.2.3. Langkah Pengujian.....	66
4.2.4. Hasil Pengujian .....	67
4.3. Tombol Manual.....	68
4.3.1. Tujuan .....	68
4.3.2. Alat Yang Digunakan.....	68
4.3.3. Langkah Pengujian.....	68
4.3.4. Hasil Dan Analisa Pengujian .....	69
4.4. Sensor Sudut.....	69
4.4.1. Tujuan .....	69
4.4.2. Alat Yang Digunakan.....	70
4.4.3. Langkah Pengujian.....	70
4.4.4. Hasil Pengujian .....	70
4.5. Driver Motor DC.....	71
4.5.1. Tujuan .....	71
4.5.2. Alat Yang Digunakan.....	71
4.5.3. Langkah Pengujian.....	71

4.5.4. Hasil Pengujian .....	72
4.6. Rangkaian Timer .....	73
4.6.1. Tujuan .....	73
4.6.2. Alat Yang Digunakan.....	73
4.6.3. Langkah Pengujian.....	73
4.6.4. Hasil Pengujian .....	73
4.7. Display LCD .....	74
4.7.1. Tujuan .....	74
4.7.2. Alat Yang Digunakan.....	74
4.7.3. Langkah Pengujian.....	74
4.7.4. Hasil Pengujian .....	74
4.8. Alarm.....	75
4.8.1. Tujuan .....	75
4.8.2. Alat Yang Digunakan.....	75
4.8.3. Langkah Pengujian.....	75
4.8.4. Hasil Pengujian .....	76
4.9. Interface Serial .....	76
4.9.1. Tujuan .....	76
4.9.2. Alat Yang Digunakan.....	76
4.9.3. Langkah Pengujian.....	77
4.9.4. Hasil Pengujian .....	77
Bab V Kesimpulan dan Saran .....	78
<b>DAFTAR PUSTAKA .....</b>	<b>80</b>

## DAFTAR GAMBAR

GAMBAR 2. 1 DIAGRAM BLOK MIKROKONTROLER AT89S8252 .....	9
GAMBAR 2. 2 KONFIGURASI PENA-PENA AT89S8252 .....	10
GAMBAR 2. 3 OSILATOR EKSTERNAL AT89S8252 .....	12
GAMBAR 2. 4 BENTUK SINYAL REMOTE TV SONY .....	19
GAMBAR 2. 5 SIMBOL INFRA RED RECEIVER MODUL (IRM) .....	21
GAMBAR 2. 6 PUSH BUTTON .....	22
GAMBAR 2. 7 SYMBOL LIMIT SWITCH .....	22
GAMBAR 2. 8 BENTUK FISIK DARI LCD ( LIQUID CRISTAL DISPLAY ) .....	24
GAMBAR 2. 9 SUSUNAN TRANSISTOR DWI KUTUB .....	24
GAMBAR 2. 10 DASAR KERJA RELAY .....	29
GAMBAR 2. 11 GARIS-GARIS MEDAN MAGNET .....	31
GAMBAR 2. 12 KAIDAH TANGAN KANAN .....	32
GAMBAR 2. 13 CARA KERJA MOTOR DC .....	32
GAMBAR 2. 14 MOTOR DC .....	34
GAMBAR 2. 15 PENGENDALIAN ARAH PUTARAN MOTOR DC .....	34
GAMBAR 2. 16 HANDPHONE SIEMENS TIPE C55 .....	35
GAMBAR 2. 17 PIN OUT SIEMENS TIPE C55 .....	35
GAMBAR 2. 18 DS 1307 8-DIP PIN .....	45

<b>GAMBAR 3 1 BLOK DIAGRAM SISTEM KESELURUHAN .....</b>	<b>49</b>
<b>GAMBAR 3 2 IRM 8510.....</b>	<b>55</b>
<b>GAMBAR 3 3 RANGKAIAN KEYPAD.....</b>	<b>55</b>
<b>GAMBAR 3 4 LIMIT SWICTH.....</b>	<b>56</b>
<b>GAMBAR 3 5 RANGKAIAN SUDUT PUTAR.....</b>	<b>57</b>
<b>GAMBAR 3 6 DRIVER MOTOR DC .....</b>	<b>58</b>
<b>GAMBAR 3 7 RANGKAIAN RTC DS1307 .....</b>	<b>59</b>
<b>GAMBAR 3 8 RANGKAIAN LCD .....</b>	<b>60</b>
<b>GAMBAR 3 9 RANGKAIAN BUZZER .....</b>	<b>61</b>
<b>GAMBAR 3 10 RANGKAIAN INTERFACE SERIAL .....</b>	<b>62</b>
<b>GAMBAR 3 11 FLOW CHAR STEP PERTAMA .....</b>	<b>63</b>
<b>GAMBAR 3 12 FLOW CHAR STEP KEDUA .....</b>	<b>64</b>
<b>GAMBAR 3 13 FLOW CHAR STEP KETIGA.....</b>	<b>65</b>
<b>GAMBAR 4 12PENGUJIAN RANGKAIAN RM870.....</b>	<b>67</b>
<b>GAMBAR 4. 1 UJI KEYPAD .....</b>	<b>69</b>
<b>GAMBAR 4. 2 UJI SENSOR SUDUT.....</b>	<b>70</b>
<b>GAMBAR 4 3: PENGUJIAN RANGKAIAN DRIVER MOTOR DC .....</b>	<b>72</b>
<b>GAMBAR 4. 3 UJI RANKAIAN TIMER.....</b>	<b>73</b>
<b>GAMBAR 4. 4 TMPILAN PROFIL LCD.....</b>	<b>74</b>
<b>GAMBAR 4: 5 UJI RANGKAIAN BUZZER.....</b>	<b>75</b>

**GAMBAR 4. 6 HASIL PENGUJIAN RANGKAIAN BUZZER..... 76**  
**GAMBAR 4: 7 UJI KONEKSI SERIAL..... 77**

## DAFTAR TABEL

TABEL 2. 1 FUNGSI KHUSUS PADA PORT 3 .....	11
TABEL 2. 2 SPECIAL FUNCTION REGISTER .....	14
TABEL 2. 3 ALAMAT SUMBER INTERUPSI.....	16
TABEL 2. 4 DATA REMOTE SONY .....	18
TABEL 2. 5 TABEL GELANG WARNA RESISTOR.....	26
TABEL 2. 6 FUNGSI MASING-MASING PIN OUT.....	36
TABEL 2. 7 NOMOR SMS CENTER DENGAN CARA <i>NATIONAL CODE</i> .....	40
TABEL 2. 8 NOMOR SMS CENTER DENGAN CARA <i>INTERNATIONAL CODE</i> .....	40
TABEL 2. 9 DAFTAR PIN DS1307 .....	45



# BAB I PENDAHULUAN

## 1.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi pada saat ini semakin berkembang dengan pesat seiring dengan perkembangan zaman terutama dibidang elektronik. Salah satu perkembangan yang paling menonjol saat ini adalah perkembangan sistem control. Dengan adanya sistem ini dapat memudahkan kita mengontrol peralatan elektronik. Kenyamanan pengguna alat control menjadikan setiap aspek yang berada disekitar kita terutama interior-interior rumah kita dapat dikontrol.

Salah satu perangkat elektronik yang tidak lepas dari keseharian adalah televisi. Kenyamanan menonton televisi menjadikan aspek interior meja televisi yang dapat mendukung hal tersebut dituntut lebih. Pemilihan interior meja televisi merupakan hal yang penting pula agar dapat terlihat elegan. Penggunaan teknologi dengan sistem kontrol dan keamanan yang dapat dipercaya dipadukan dengan bentuk yang indah. Meja televisi sebagai tempat peletakan televisi juga bisa digunakan sebagai tempat penyimpanan televisi untuk dapat mengantisipasi pencurian televisi. Keamanan serta mudah disimpan menjadikan tantangan agar dapat menjadikan interior meja televisi yang ekonomis dan aman yang berpenampilan elegan

Melihat dari kebutuhan masyarakat akan kebutuhan interior meja televisi yang elegan dilengkapi dengan sistem kontrol yang efisien. Untuk itu penulis akan merancang dan membuat perancangan dan pembuatan interior meja televisi menggunakan remote control dilengkapi alarm berbasis mikrokontroler AT89S8252. Dimana interior televisi ini menggunakan remote control untuk mengangkat televisi dari dalam meja dan menghidupkan power tegangan dari listrik. Apabila terjadi pemaksaan meja dibuka akan mengakibatkan alarm berbunyi dan mengirim informasi melalui sms bawa adanya pembukaan secara paksa.

## **1.2. Rumusan masalah**

Berdasarkan latar belakang yang telah diuraikan pada bagian sebelumnya, maka permasalahannya adalah

- Bagaimana merancang dan membuat suatu sistem pengontrol untuk mengangkat televisi didalam meja
- Bagaimana merancang dan membuat suatu sistem pengontrol untuk membuka tutup meja
- Bagaimana merancang dan membuat suatu sistem pengontrol untuk memutar meja televisi dengan maksimal 60 derajat
- Bagaimana merancang dan membuat suatu sistem pengontrol untuk alarm apabila atau tutup meja televisi dibuka secara paksa
- Bagaimana merancang dan membuat suatu sistem pengontrol untuk mengirim informasi melalui sms apabila alarm berbunyi

### **1.3. Tujuan**

Tujuan pembuatan skripsi ini adalah untuk

- Mempermudah posisi sudut pandang menonton televisi
- Dapat mengontrol pemakaian jarak jauh
- Dapat menginformasikan pemakaian televisi kepada pemilik

### **1.4. Batasan Masalah**

Dalam menyusun tugas akhir ini diperlukan suatu batasan masalah agar tidak menyimpang dari ruang lingkup yang akan dibahas. Adapun batasan masalahnya:

1. Menggunakan Mikrokontroler Atmel 89S8252
2. Menggunakan remote infra red
3. Tidak membahas torsi motor penggerak secara detail
4. Remote televisi tidak sama dengan remote yang digunakan
5. Tidak membahas power supply dan power supply dianggap stabil
6. Alat yang dibuat ini berupa prototipe

### **1.5. Metodologi Penelitian**

Metodologi yang dipakai dalam pembuatan skripsi ini adalah:

#### **1.5.1. Literature**

Mencari referensi-referensi yang berhubungan dengan perencanaan dan pembuatan alat yang akan dibuat.

### **1.5.2. Penelitian lapangan**

Melakukan penelitian secara langsung mengenai objek-objek yang berhubungan langsung dengan perencanaan alat yang akan dibuat

### **1.5.3. Pengolahan Data**

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

## **1.6. Sistematika Penulisan**

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

### **BAB I Pendahuluan**

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

### **BAB II Landasan Teori**

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perencanaan dan pembuatan alat ini

### **BAB III Perencanaan dan Pembuatan Alat**

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

### **BAB IV Pengujian alat**

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

#### **BAB V Penutup**

Pada bab ini disampaikan kesimpulan dari perencanaan pembuatan sistem ini.

## BAB II DASAR TEORI

### 2.1. Pendahuluan

Pada bab ini akan dibahas mengenai teori penunjang dari peralatan yang direncanakan. Pokok pembahasan pada bab ini adalah:

- 2.1.1. Mikrokontroller atmel AT89S8252
- 2.1.2. *Remote Control*
- 2.1.3. Swicth
- 2.1.4. Photo diode
- 2.1.5. LCD (*Liquid Cristal Display*)
- 2.1.6. Transistor
- 2.1.7. Resistor
- 2.1.8. Relay
- 2.1.9. Motor DC
- 2.1.10. Handphone Siemens C55

### 2.2. Mikrokontroller

#### 2.2.1. ATMEL AT89S8252

##### 2.2.1.1. Pendahuluan

Perbedaan mendasar antara mikrokontroler dan mikroprocessor adalah mikrokontroler selain memiliki CPU juga dilengkapi memori dan input output yang merupakan kelengkapan sebagai system minimum mikrokomputer sehingga sebuah

mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal ( *Single Chip Mikrokomputer* ) yang dapat berdiri sendiri.

Mikrokontroler AT89S8252 adalah mikrokontroler ATMEL yang kompatibel penuh dengan mikrokontroler keluarga MCS-51, membutuhkan daya rendah, memiliki performance yang tinggi dan merupakan mikrokomputer 8 bit yang dilengkapi dengan 8 Kilobyte Flash Memori untuk Program, 2 Kilobyte EEPROM ( *Electrical Eraseable and Programable Read Only Memory* ) dan 256 Byte RAM internal. Program memori yang dapat diprogram ulang dalam system atau menggunakan programmer nonvolatile Memori Konvensional. Dalam system mikrokontroler terdapat dua hal mendasar, yaitu : perangkat lunak dan perangkat keras yang keduanya saling terkait dan mendukung.

#### **2.2.1.2. Perangkat Keras Mikrokontroler AT89S8252**

Secara umum Mikrokontroler AT89S8252 memiliki :

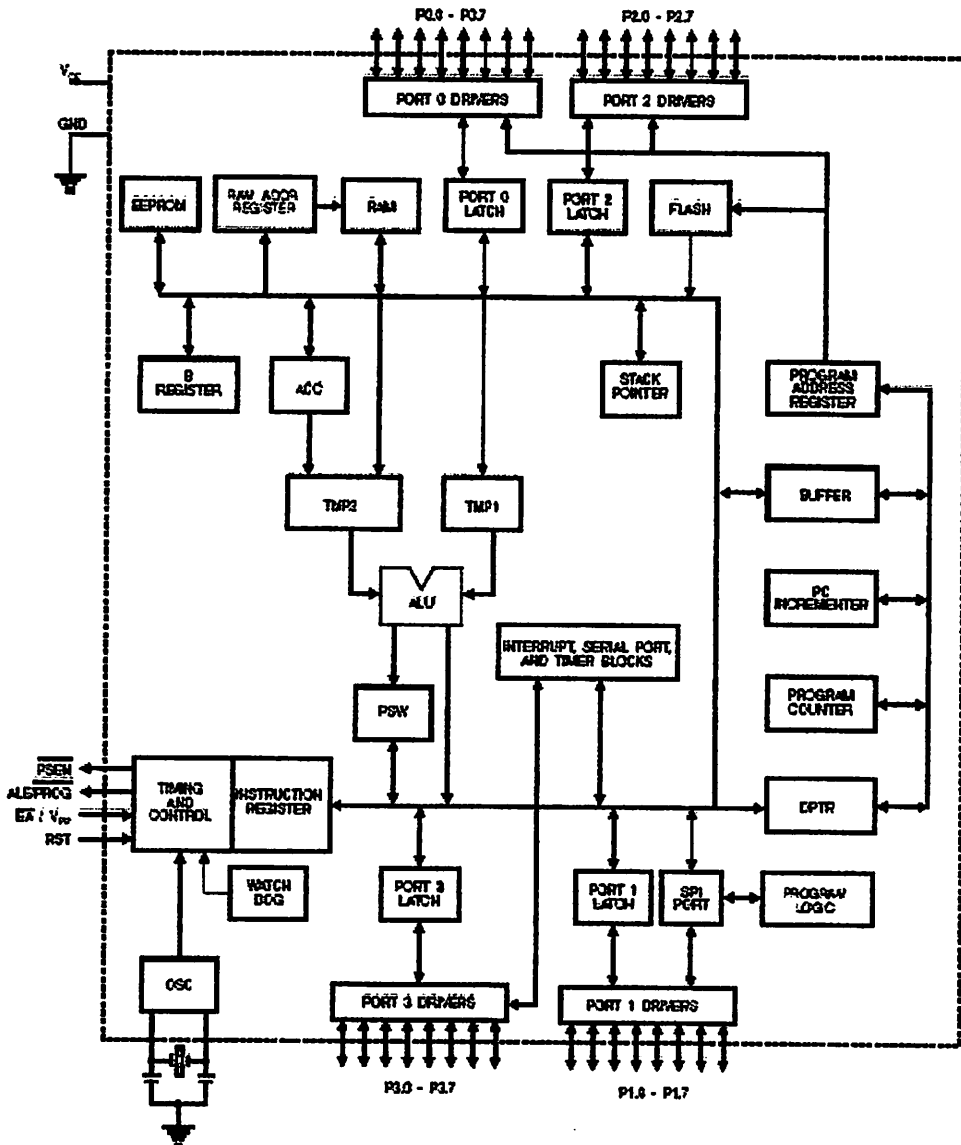
- CPU 8 bit termasuk keluarga MCS-51
- 8 Kbyte *Downloadable Flash Memory*
- 256 Byte Memori Internal
- 32 Port I/O, Masing-masing terdiri atas 8 jalur I/O
- 3 Timer/Counter 16 Bit
- 2 *Serial Port Full Duplex*
- Kecepatan pelaksanaan intruksi per siklus  $1\mu\text{S}$  pada frekuensi clock 12 MHz

- SPI Serial Interface
- 2 DPTR ( Data Pointer )
- *Programmable Watchdog Timer*
- *Fleksibel ISP Programming*

Terlihat bahwa mikrokontroler Atmel AT89S8252 memiliki banyak fitur yang menguntungkan. Dipakainya *Downloadable flash* memory memungkinkan mikrokontroler ini bekerja sendiri tanpa diperlukan tambahan chip lainnya. Sementara Flash memorinya mampu diprogram hingga seribu kali. Hal lain yang menguntungkan adalah sistem pemrograman menjadi lebih sederhana dan tidak memerlukan rangkaian yang rumit seperti rangkaian untuk memprogram produk Atmel lainnya yaitu AT89C51.

Timer/Counter juga bertambah satu dari standar 2 buah pada MCS-51. Selain itu frekuensi kerja yang lebar dan rancangan statik sangat membantu untuk proses *debugging*. Dengan adanya beberapa fitur tambahan itu, maka akan mengakibatkan bertambahnya SFR (Special Function Register). Adapun blok diagram dari mikrokontroler AT89S8252 adalah seperti yang terlihat pada gambar 2-1.



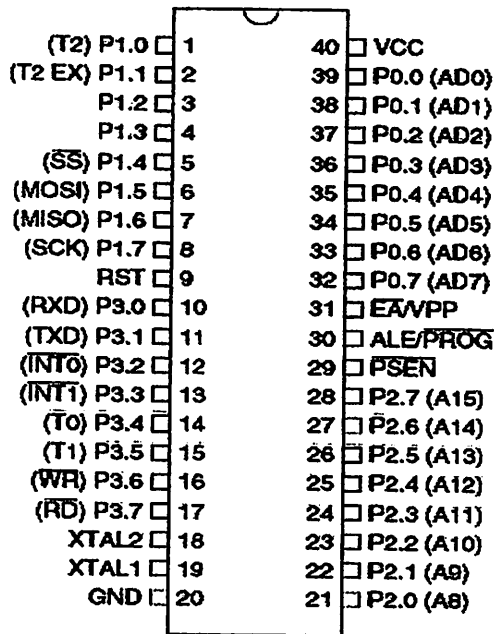


**Gambar 2. 1** Diagram Blok Mikrokontroler AT89S8252

( Sumber : Data Sheet Amel AT89S8252 )

### 2.2.1.3. Konfigurasi Pena-Pena Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 terdiri dari 40 pin dengan konfigurasi sebagai berikut seperti yang terlihat pada gambar 2-2 :



**Gambar 2. 2 Konfigurasi Pena-pena AT89S8252**

( Sumber : Data Sheet Atmel AT89S8252 )

Fungsi-fungsi tiap pinnya adalah sebagai berikut :

- VCC ( Supply Tegangan ) , pin 40
- GND ( Ground ) , pin 20
- Port 0, pin 32-39

Merupakan port input-output dua arah, tanpa *internal pull-up* dan konfigurasikan sebagai multipleks bus alamat rendah ( A<sub>0</sub>-A<sub>7</sub> ) dan data selain pengaksesan program memory dan data memory eksternal

- Port 1, pin 1-8

Merupakan port input-output dua arah dengan *internal pull-up*

- Port 2, pin 21-28

Merupakan port input-output dengan *internal pull-up*. Mengeluarkan alamat tinggi selama pengambilan program memori eksternal.

- Port 3, pin 10-17

Merupakan port input-output dengan *internal pull-up*, dimana Port 3 juga memiliki fungsi khusus dan dapat dilihat pada Tabel 2-1 berikut :

Tabel 2. 1 Fungsi Khusus Pada Port 3

Nama Penyemat	Fungsi Khusus
Port 3.0	RXD (Port Masukan Serial)
Port 3.1	TXD (Port Keluaran Serial)
Port 3.2	/INT0 (Masukan Interupsi Eksternal 0)
Port 3.3	/INT1 (Masukan Interupsi Eksternal 1)
Port 3.4	T0 (Masukan Pewaktu Eksternal 0)
Port 3.5	T1 (Masukan Pewaktu Eksternal 1)
Port 3.6	/WR (Sinyal Tulis Memori Data Eksternal)
Port 3.7	/RD (Sinyal Baca Memori Data Eksternal)

( Sumber : Data Sheet Atmel AT89S8252)

- RST ( Reset ), pin 9

Input Reset merupakan reset master untuk AT89S8252

- ALE/Prog ( *Addresses Lacth Enable* ), pin 30

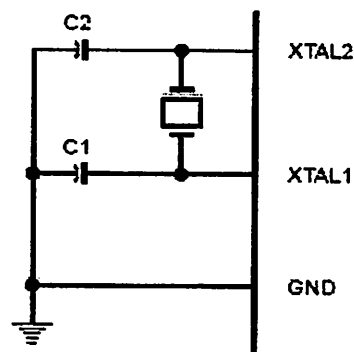
Digunakan untuk menahan alamat memori eksternal selama pelaksanaan intruksi.

- **PSEN ( *Program Store Enable* ), pin 29**

Merupakan sinyal pengontrol yang memperbolehkan program memori eksternal masuk ke dalam bus.
- **ES/VPP ( *Eksternal Access* ), pin 31**

Dapat diberikan logika rendah ( *Ground* ) atau logika tinggi ( + 5 Volt ). Jika diberikan logika tinggi maka mikrokontroler akan mengakses program dari ROM internal ( EEPROM/Flash Memori ), dan jika diberikan logika rendah maka mikrokontroler akan mengakses program dari memori eksternal.
- **X-TAL 1 dan X-TAL 2, pin 19,18**

Pin ini dihubungkan dengan kristal bila menggunakan osilator internal. X-TAL 1 merupakan masukan ke rangkaian osilator internal sedangkan X-TAL 2 keluaran dari rangkaian osilator internal. Untuk keperluan ini diperlukan kapasitor penstabil sebesar 30pF. Dan nilai dari X-TAL tersebut antara 4-24 MHz. Untuk lebih jelasnya dapat dilihat gambar 2-3 mengenai pemasangan X-TAL serta kapasitor yang digunakannya.



**Gambar 2. 3 Osilator Eksternal AT89S8252**

( *Sumber : Data Sheet Atmel AT89S8252* )

#### 2.2.1.4. Organisasi Memori

Organisasi memori pada mikrokontroler AT89S8252 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program dapat digunakan untuk menyimpan intruksi-intruksi yang akan dijalankan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat yang sedang diolah mikrokontroler.

Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler AT89S8252 dilengkapi dengan ROM internal, sehingga untuk menyimpan program tidak digunakan ROM eksternal yang terpisah dari mikrokontroler. Agar tidak menggunakan memori program eksternal, penyemat/EA dihubungkan dengan Vcc ( Logika 1 ).

Memori program mikrokontroler menggunakan alamat 16 bit mulai dari 0000H – FFFF<sub>H</sub> sehingga kapasitas penyimpanan program maksimal adalah 4Kbyte. Sinyal/PSEN ( *Program Store Enable* ) tidak digunakan jika digunakan memori internal.

Selain program mikrokontroler AT89S8252 juga memiliki data internal sebesar 128 Byte dan mampu mengakses memori data eksternal sebesar 64 Kbyte. Semua memori data internal dapat dialamati dengan data langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* adalah alamat register yang berisi alamat data yang akan diolah. Sebagian memori tersebut dapat dialamati dengan pealamatan register,

dan sebagian lagi dapat dialamati dengan memori satu bit. Untuk membaca data digunakan sinyal /RD sedangkan untuk menulis digunakan sinyal /WR.

#### 2.2.1.5. SFR ( *Special Function Register* )

Register fungsi khusus ( *Special Function Register* ) terletak pada 128 Byte bagian atas memori data internal dan berisi register-register untuk pelayanan *latch port, timer, program status word, control peripheral*, dan sebagainya. Alamat register fungsi khusus ditunjukkan pada Tabel 2-2

Tabel 2. 2 Special Function Register

<b>Simbol</b>	<b>Nama Register</b>	<b>Alamat</b>
ACC	Accumulator	E0 <sub>H</sub>
B	Register B	F0 <sub>H</sub>
PSW	Program Status Word	D0 <sub>H</sub>
SP	Stack Pointer	81 <sub>H</sub>
DPTR	Data pointer 2 Byte	82 <sub>H</sub>
DPL	Bit Rendah	83 <sub>H</sub>
DPH	Bit Tinggi	80 <sub>H</sub>
P0	Port0	90 <sub>H</sub>
P1	Port 1	A0 <sub>H</sub>
P2	Port 2	B0 <sub>H</sub>
P3	Port 3	D8 <sub>H</sub>
IP	Interupt Priority Control	A8 <sub>H</sub>
IE	Interupt Enable Control	89 <sub>H</sub>
TMOD	Timer/Counter Mode Control	88 <sub>H</sub>
TCON	Timer/Counter Control	8C <sub>H</sub>
TH0	Timer/Counter 0 High Control	8A <sub>H</sub>
TL0	Timer/Counter 0 Low Control	8D <sub>H</sub>
TH1	Timer/Counter 1 high Control	8B <sub>H</sub>

TL1	Timer/Counter 1 Low Control	98 <sub>H</sub>
SCON	Serial Control	99 <sub>H</sub>
SBUF	Serial Data Buffer	87 <sub>H</sub>
PCON	Power Control	

( Sumber : Data Sheet Atmel AT89S8252 )

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

- *Accumulator* (ACC) merupakan register untuk penambahan dan pengurangan. Perintah *Mnemonic* untuk mengakses akumulatur disederhanakan sebagai A.
- Register B merupakan register khusus yang berfungsi melayani operasi perkalian dan pembagian.
- *Stack Pointer* (SP) merupakan register 8 bit yang dapat diletakkan di alamat manapun pada RAM internal.
- 2 Data Pointer (DPTR) terdiri atas dua register, yaitu untuk byte tinggi (*Data Pointer High*, DPH) dan byte rendah (*Data Pointer Low*, DPL) yang berfungsi untuk mengunci alamat 16 bit.
- Port 0 sampai Port 3 merupakan register yang berfungsi untk membaca dan mengeluarkan data pada port 0,1,2,3. Masing-masing register ini dapat dialamati per-byte maupun per-bit.
- Control Register terdiri dari register yang mempunyai fungsi kontrol. Untuk mengontrol system interupsi, terdapat dua register khusus, yaitu register IP (*Interupt Priority*) dan register IR (*Interupt Enable*). Untuk mengontrol pelayanan timer/counter terdapat register khusus, yaitu register TCON

(*Timer/Counter Control*) serta pelayanan port serial menggunakan register *SCON (Serial Port Control)*.

#### 2.2.1.6. Sistem Interupsi

Mikrokontroler AT89S8252 mempunyai 5 buah sumber interupsi yang dapat membangkitkan permintaan interupsi, yaitu INT0, INT1, T1, T2, dan Port Serial.

Saat terjadinya interupsi mikrokontroler secara otomatis akan menuju ke subrutin pada alamat tersebut. Setelah interupsi selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Tiap-tiap sumber interupsi dapat *enable* atau *disable* secara software.

Tingkat prioritas semua sumber interrupt dapat diprogram sendiri-sendiri dengan set atau clear bit pada IP (*Interrupt Priority*). Jika dua permintaan interupsi dengan tingkat prioritas yang berbeda diterima secara bersamaan, permintaan interupsi dengan prioritas yang sama diterima bersamaan, akan dilakukan polling untuk menentukan mana yang akan dilayani.

Tabel 2-3 bawah ini akan menunjukkan daftar interupsi pada mikrokontroler AT89S8252 dan alamat awal vector interupsinya :

Tabel 2. 3 Alamat Sumber Interupsi

Sumber Interupsi	Alamat Awal
Interupt Luar 0 (INT0)	03 <sub>H</sub>
Pewaktu / Pencacah 0 (T0)	0B <sub>H</sub>



Interupt Luar 1 (INT1)	13 <sub>H</sub>
Pewaktu / Pencacah 0 (T0)	1B <sub>H</sub>
Port Serial	23 <sub>H</sub>

( Sumber : Data Sheet Atmel AT89S8252)

### 2.2.1.7. Metode Pengalamatan

1. Pengalamatan bit (*Direct Bit Addressing*)

Pengalamat bit tiap bit ini hanya dilakukan pada lokasi RAM internal yaitu 20H-2FH, dan sebagian SFR yaitu : port 0, port 1, port 2, port 3, TCON register, SCON register, IE register, PSW register, ACC dan B register.

2. Pengalamatan Tak Langsung (*Indirect Bit Addressing*)

Pada pengalamatan tak langsung, intruksi menunjukkan suatu register yang isinya adalah alamat dari operand, eksternal dan internal RAM dapat berupa R0 dan R1 yang digunakan untuk memilih angka register atau stack pointer. Register alamat untuk data, dengan lebar 16 bit digunakan Data Pointer (DPTR).

3. Pengalamatan Ber-indeks

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

#### 4. Konstanta Immediet

Pengalamatan langsung dilakukan dengan memberikan nilai ke register secara langsung, dilakukan dengan menggunakan tanda #, (Contoh : MOV A, #100)

### 2.3. Remote Control

#### 2.3.1. Remote Pengirim

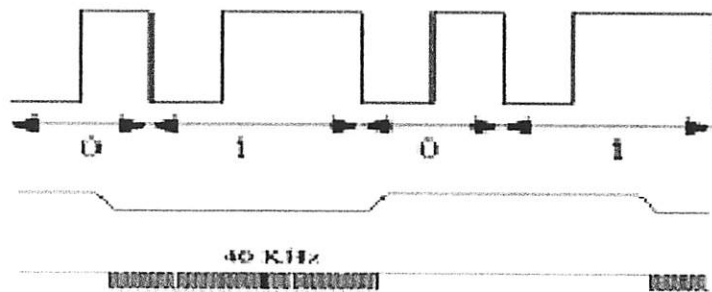
Berfungsi sebagai Sebagai pengontrol. dan Remote yang di gunakan Remote dengan format data Sony yang mempunyai panjang data sebanyak 12 bit. Sinyal control inilah yang nantinya akan menjadi masukan untuk unit pemroses AT89S51. Penerima sinyal Remote menggunakan IR reciver pabrikasi yang telah berfungsi untuk menerima sinyal dengan frekwensi sinyal Remote Tv Sony yaitu 40 KHz.

Tabel 2. 4 Data Remote Sony

Tombol	Hexa	Tombol	Hexa
1	#080	Vol-	#093
2	#081	Power(toggle)	#095
3	#082	PIC Mode	#096
4	#083	A/B	#097
5	#084	TV/Video	#0A5
6	#085	Sleep	#0B6
7	#086	+	#0F4
8	#087	-	#0F5
9	#088	Select	#0FC
0	#089		
Prog+	#090		
Prog-	#091		
Vol+	#092		

(sumber: [www.sbprojects.com](http://www.sbprojects.com))

Dimana data yang untuk di gunakan untuk menembak oleh modul ini adalah sama dengan penekanan tombol “1” pada remote sony. Pengguna dapat mencoba dengan menembakan tombol “1” pada sensor. Oleh karna itu data yang harus dimodulasikan oleh modul adalah data 80h. Data tersebut di kirim secara serial dalam bentuk pulse code modulation dimana logika 0 di wakili oleh logika 0 dan logika 1 yang pendek sedangkan logika 0 diwakili oleh logika 0 panjang dan logika 1 untuk yang pendek seperti pada gambar 2-3 berikut:



**Gambar 2. 4 Bentuk Sinyal Remote Tv Sony**

*(sumber: [www.sbprojects.com](http://www.sbprojects.com))*

Namun sinyal PCM ini tidk dapat di kirimkan langsung ke LED infrared melainkan terlebih dahulu melalui proses modulasi dengan sinyal carrier 40 KHz sehingga tampak seperti pada gambar di atas logika 0 dari hasil PCM akan diubah menjadi sinyal 40 KHz dan logika 1 diubah jadi logika 1 biasa.

### 2.3.2. IRM (Infra Red Receiver Module)

Inframerah Receiver Module yang lebih dikenal dengan modul infra merah, menerima gelombang infra merah dengan panjang gelombang  $10^{-3} - 10^{-6}$  m dan frekwensi sebesar  $10^{11} - 10^{14}$  Hz, dengan frekwensi carrier 38 – 40 Khz.

Proses penerimaan pada prinsipnya sama dengan proses fotodioda dan sudah terintegrasi bersama satu komponen IC, dimana pada fotodioda terdapat suatu jendela kecil yang memungkinkan cahaya luar dapat masuk dan mengenai PN junction. Pada keadaan normal fotodioda berlaku sebagai dioda biasa yang dapat menghantarkan arus listrik dari anoda ke katoda, namun mempunyai tahanan balik yang besar. Bila cahaya luar mengenai junction fotodioda, maka tahanan balik akan mengecil dan menimbulkan arus balik, sehingga fotodioda berlaku sebagai dioda yang balik atau dibias reverse.

Semakin besar intensitas cahaya yang diterima, maka semakin besar pula arus balik yang ditimbulkannya. Bila energi foton diserap dalam suatu semikonduktor, maka akan dihasilkan pasangan elektron-elektron dan hole-hole yang telah dibangkitkan oleh foton yang saling memisahkan diri karena pengaruh medan listrik. Dimana elektron-elektron akan menuju sisi N dan hole-hole menuju ke sisi P, sehingga dihasilkan arus dari katoda menuju anoda. Arus balik yang dihasilkan sebanding dengan sinar yang diserap. Karena pengaruh suhu junction yang lebih tinggi menciptakan

lebih banyak pasangan elektron-hole, sehingga mengakibatkan aliran arus balik yang melewati junction bertambah.



**Gambar 2. 5 Simbol Infra Red Receiver Modul (IRM)**

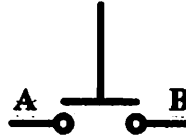
*(sumber: data sheet irm)*

## **2.4. Swicth**

### **2.4.1. *Push Botton* (Saklar Tekan)**

Saklar tekan dioperasikan dengan cara menekan sebuah tombol. Terdapat dua jenis saklar semacam ini. Kebanyakan di antaranya termasuk kedalam jenis *push to make* (tekan untuk menyambungkan). Dengan menekan tombol, kontak-kontak akan saling bersetuhan dan saklar menutup. Jenis lainnya adalah *push to break* (tekan untuk memutuskan). Kontak-kontaknya adalah normal tertutup, namun akan dipaksa membuka ketika tombol ditekan.

Masing-masing jenis saklar yang disebutkan diatas dapat bekerja untuk memutuskan atau menghubungkan sambungan selama sekejap. Tombol tersebut akan menghubungkan atau memutuskan selama tombol tersebut ditekan. Gambar dari push buttom dapat dilihat pada gambar 2-29 dibawah ini.



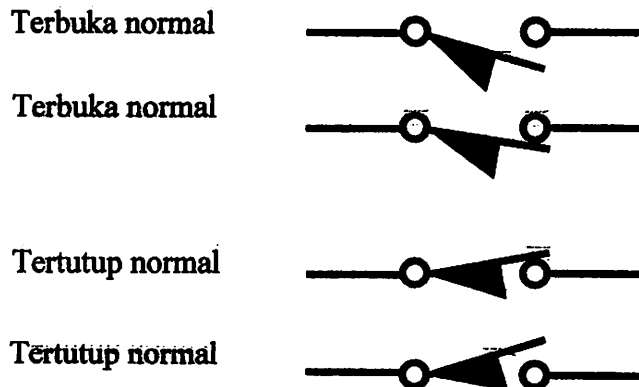
**Gambar 2. 6 Push Button**

*(sumber : 302 rangkaian elektronika)*

### 2.4.2. Limit Swicth

saklar yang dioperasikan secara mekanis adalah saklar yang dikontrol oleh factor-faktor secara otomatis seperti tekanan. Saklar limit adalah alat pengendali industri yang sangat umum. Saklar limit dirancang hanya untuk beroperasi apabila batas yang sudah ditentukan sebelumnya sudah dicapai, dan saklar-saklar tersebut biasanya diaktifkan kontak dengan obyek.

Dengan ukuran yang sangat kecil dan tuas pengoperasian yang bermacam-macam membuat saklar limit sangan banyak manfaatnya dalam dunia industri. Saklar ini dapat bekerja dengan tekanan yang kecil, berikut adalah symbol dari Limit switch.



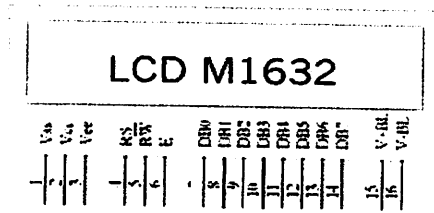
**Gambar 2. 7 Symbol Limit Switch**

## 2.5. LCD (*Liquid Cristal Display*)

*Liquid Cristal Display* adalah modul tampilan yang mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroler CMOS didalamnya. Kontroler tersebut berfungsi sebagai pembangkit ROM / RAM dan *display* data RAM. Semua fungsi tampilan dikontrol oleh suatu instruksi, modul LCD dapat dengan mudah diinterfacekan dengan MPU.

LCD yang digunakan dalam skripsi ini adalah LCD yang memiliki kemampuan sebagai berikut:

- Meliputi 32 karakter yang dibagi menjadi 2 baris dengan *display dot matrik* 5 x 7 ditambah *cursor*.
- Karakter generator ROM dengan 192 karakter.
- Karakter generator RAM dengan 8 tipe karakter.
- Dilengkapi fungsi tambahan yaitu *display clear*, *cursor home*, *display ON/OFF*, *cursor ON/OFF*, *display character blink*, *cursor shift* dan *display shift*.
- Internal data.
- 80 x 8 bit *display* data RAM.
- Dapat diinterfacekan dengan  $\mu$ C 8 atau 4 bit.
- Internal otomatis dan *reset* pada *power ON*.
- + 5 volt *power supply* tunggal.



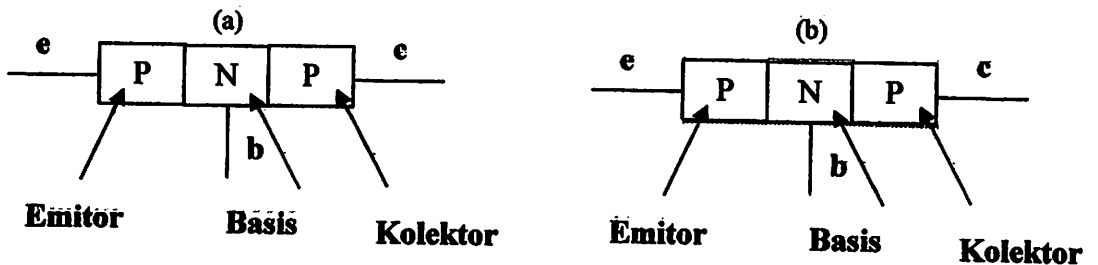
**Gambar 2. 8 Bentuk fisik dari LCD ( Liquid Cristal Display )**

(*sumber: data Sheet LCD*)

## 2.6. Transistor

Transistor adalah suatu komponen aktif dari bahan semi konduktor. Ada dua macam transistor, yaitu transistor dwi kutub (*bipolar*) dan transistor efek medan (*Field Effect Transistor*).

Transistor digunakan dalam rangkaian untuk memperkuat isyarat artinya isyarat lemah pada masukan diubah menjadi isyarat yang kuat pada keluaran. Pada masa kini transistor ada dalam setiap peralatan elektronika. Transistor dwi kutub dibuat dengan menggunakan semi konduktor ekstrinsik jenis P dan N.



**Gambar 2. 9 Susunan Transistor Dwi Kutub**

(*sumber: Malvino, Prinsip-prinsip elektronika*)

Ketiga bagian transistor disebut emitor, basis, dan kolektor. Masing-masing bagian transistor ini dihubungkan keluaran transistor dengan menggunakan konduktor



sebagai kaki transistor. Pada transistor dwi kutub sambungan p-n antara emitor dan basis diberi bias maju sehingga arus mengalir dari emitor ke basis. Bias adalah tegangan dan arus DC yang harus lebih dahulu dipasang agar rangkaian transistor bekerja. Agar mudah dibayangkan kita gunakan transistor PNP untuk mempelajari cara kerja transistor.

Mayoritas yang dikeluarkan oleh emitor bertumpu di basis, dan ditampung oleh kolektor. Sekarang jelaslah makna nama-nama bagian transistor. Emitor berasal dari bahasa inggris "Emitter" yang berarti pengeluar. Basis berasal dari bahasa inggris "Base" yang berarti tumpuan atau landasan. Dan kolektor berasal dari kata "Collector" yang berarti pengumpul.

Kerja transistor berdasarkan kepekaan arus yang dihasilkan oleh emitor akibat beda tegangan antara emitor dan basis (tumpuan). Jika tegangan emitor naik sedikit sehingga beda tegangan antara basis emitor naik sedikit, arus yang dikeluarkan oleh emitor akan berubah banyak. Arus ini dikumpulkan oleh kolektor yang diberi bias mundur oleh Vcc sehingga arus tidak dapat membalik dari kolektor ke basis.

## **2.7. Resistor**

Pada dasarnya semua bahan memiliki sifat resistif namun beberapa bahan seperti tembaga, perak, emas dan bahan metal umumnya memiliki resistansi yang sangat kecil. Bahan-bahan tersebut menghantar arus listrik dengan baik, sehingga dinamakan konduktor. Kebalikan dari bahan yang konduktif, bahan material seperti karet, gelas,

karbon memiliki resistansi yang lebih besar menahan aliran elektron dan disebut sebagai insulator.

Tabel 2. 5 Tabel Gelang Warna Resistor

Warna	Nilai	Faktor Pengali	Toleransi
Hitam	0	1	
Coklat	1	10	1%
Merah	2	100	2%
Jingga	3	1.000	
Kuning	4	10.000	
Hijau	5	100.000	
Biru	6	$10^6$	
Violet	7	$10^7$	
Abu-abu	8	$10^8$	
Putih	9	$10^9$	
Emas	-	0.1	5%
Perak	-	0.01	10%
Tanpa warna	-	-	20%

(sumber: 302 rangkaian elektronika)

Resistor adalah komponen dasar elektronika yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian. Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon. Dari hukum Ohm diketahui, resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Satuan resistansi dari suatu resistor disebut Ohm atau dilambangkan dengan simbol  $\Omega$  (Omega).

Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk memudahkan pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohm meter. Kode warna tersebut adalah standar manufaktur yang dikeluarkan oleh *EIA (Electronic Industries Association)* seperti yang ditunjukkan pada tabel 2-2.

Resistansi dibaca dari warna gelang yang paling depan ke arah gelang toleransi berwarna coklat, merah, emas atau perak. Biasanya warna gelang toleransi ini berada pada badan resistor yang paling pojok atau juga dengan lebar yang lebih menonjol, sedangkan warna gelang yang pertama agak sedikit ke dalam. Dengan demikian pemakai sudah langsung mengetahui berapa toleransi dari resistor tersebut. Kalau kita telah bisa menentukan mana gelang yang pertama selanjutnya adalah membaca nilai resistansinya.

Jumlah gelang yang melingkar pada resistor umumnya sesuai dengan besar toleransinya. Biasanya resistor dengan toleransi 5%, 10% atau 20% memiliki 3 gelang (tidak termasuk gelang toleransi). Tetapi resistor dengan toleransi 1% atau 2% (toleransi kecil) memiliki 4 gelang (tidak termasuk gelang toleransi). Gelang pertama dan seterusnya berturut-turut menunjukkan besar nilai satuan, dan gelang terakhir adalah faktor pengalinya.

Misalnya resistor dengan gelang kuning, violet, merah dan emas. Gelang berwarna emas adalah gelang toleransi. Dengan demikian urutan warna gelang resistor ini adalah, gelang pertama berwarna kuning, gelang kedua berwarna violet dan gelang ke tiga berwarna merah. Gelang ke empat tentu saja yang berwarna emas dan ini

adalah gelang toleransi. Dari tabel 2-2 diketahui jika gelang toleransi berwarna emas, berarti resistor ini memiliki toleransi 5%. Nilai resistansinya dihitung sesuai dengan urutan warnanya. Pertama yang dilakukan adalah menentukan nilai satuan dari resistor ini. Karena resistor ini resistor 5% (yang biasanya memiliki tiga gelang selain gelang toleransi), maka nilai satuannya ditentukan oleh gelang pertama dan gelang kedua. Masih dari tabel 2-2 diketahui gelang kuning nilainya = 4 dan gelang violet nilainya = 7. Jadi gelang pertama dan kedua atau kuning dan violet berurutan, nilai satuannya adalah 47. Gelang ketiga adalah faktor pengali, dan jika warna gelangnya merah berarti faktor pengalinya adalah 100. Sehingga dengan ini diketahui nilai resistansi resistor tersebut adalah nilai satuan dikalikan faktor pengali atau  $47 \times 100 = 4.7K \text{ Ohm}$  dan toleransinya adalah 5%, atau nilainya yang berada pada kisaran  $(4700 \pm 235) \Omega$ .

Spesifikasi lain yang perlu diperhatikan dalam memilih resistor pada suatu rancangan selain besar resistansi adalah besar watt-nya. Karena resistor bekerja dengan dialiri arus listrik, maka akan terjadi disipasi daya berupa panas sebesar  $W=I^2R$  watt. Semakin besar ukuran fisik suatu resistor bisa menunjukkan semakin besar kemampuan disipasi daya resistor tersebut.

Umumnya di pasar tersedia ukuran 1/8, 1/4, 1, 2, 5, 10 dan 20 watt. Resistor yang memiliki disipasi daya 5, 10 dan 20 watt umumnya berbentuk kubikal memanjang persegi empat berwarna putih dibalut keramik, namun ada juga yang berbentuk silinder. Tetapi biasanya untuk resistor ukuran jumbo ini nilai resistansi dicetak langsung dibadannya, misalnya 1W, 5W.

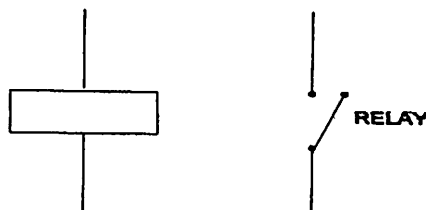
## 2.8. RELAY

*Relay* adalah suatu perangkat switch yang dioperasikan gaya elektromagnetik ( magnet permanen ) atau di sebut juga komponen elektromekanis.

*Relay* biasanya digunakan untuk pensaklaran khusus ( misalnya pemutusan daya dengan remote ). Keuntungan dari pemakaian relay umumnya terletak pada pengaturan switching daya / tegangan tinggi, dengan catu daya rendah sehingga terdapat isolasi antara catu daya relay yang relative rendah dengan catu beban yang tinggi yang akan di putus sambungkan. Kerugian relay umumnya terjadi efek 'bouncing' dan tanggapan waktu ( respon time ) saat on tau off yang relative lebih lambat, kontrol relay :

Semua relay elektromagnetik terdapat tiga bagian utama, yaitu :

- a. Koil magnet ( Kumbaran penggerak magnetisasi )
- b. Hubungan dari kumbaran transformasi keterminal keluaran ( out put )
- c. Perubahan kondisi saklar ( on atau off ), selama kumbaran berenergi.



**Gambar 2. 10 Dasar Kerja Relay**

(sumber: Malvino, Prinsip prinsip elektronika)

Bagian kontak relay dapat dipakai sebagai :

- a. Pole ( kutub utama )

- b. Throw ( Kutub Pelepasan )
- c. Posisi Normal ( Normally Open atau Normally Closed ).

Misalnya sebuah relay dengan parameter SPST ( Single Pole Singgel Throw ), NO, ( Normally Open ), DM ( Double Male ), artinya : relay dengan 1 induk 1 anak, pada keadaan normalnya terbuka ( OFF ), dengan kontak penyambungan ganda.

Relay di kelompokkan dalam tiga bagian menurut pemakaian dan kegunaannya, Yaitu :

1. Pemakaian umum : ( General Purpose ), contoh : Relay AC, relay DC
2. Pemakaian khusus : ( Specific Puspose ), contoh : DLL relay.
3. Pemakaian Terbatas : ( Define Purpose), contoh : reed relay, kontaktor.

Sedangkan menurut kemampuan yang spesifik, relay dapat di kelompokkan sebagai berikut:

- a. **Margina** : Kemampuan yang baik dalam mengambil atau melepaskan arus listrik ( Pick up and drop –out current ).
- b. **Timing** : Kemampuan dalam waktu pelepasan kontak ( release time), waktu perlambatan ( delay time ), dan waktu lainnya.
- c. **Sensitivity** : Kemampuan dalam sensitifitas ( kepekaan ) dalam mengambil atau melepas arus listrik pada operasi normal.

d. Latching : Kemampuan dalam urutan perioda switctingnya.

e. Sequencing : Kemampuan dalam urutan perioda switctingnya.

f. Frequency sensitivity : Kepekaan perioda frekwensikerja relay

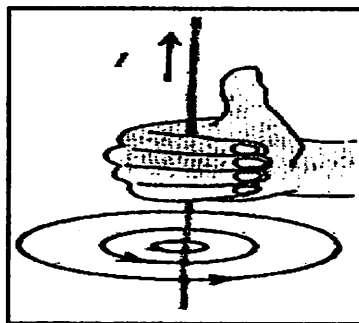
( Khususnya relay AC ).

g. Thermal Response : Kemampuan akan temperaturnya akan kerja dari

kumparan saat operasi.

## 2.9. Motor DC

Setiap arus yang mengalir melalui sebuah konduktor akan menimbulkan medan magnet. Arah medan magnet dapat ditentukan dengan kaidah tangan kanan. Ibu jari tangan menunjukkan arah aliran arus listrik sedangkan jari-jari yang lain menunjukkan arah medan magnet yang timbul, seperti yang ditunjukkan oleh gambar 2.20 berikut ini.



**Gambar 2. 11 Garis-garis Medan Magnet**

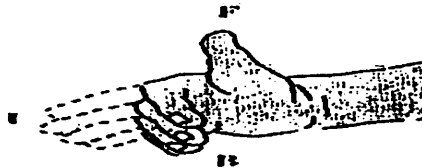
(sumber: Malvino, Prinsip prinsip elektronika)

Kaidah tangan kanan untuk motor menunjukkan arah arus yang mengalir didalam sebuah konduktor yang berada dalam medan magnet. Jari tengah menunjukkan arah arus yang mengalir pada konduktor, jari telunjuk menunjukkan arah medan magnet dan ibu jari menunjukkan arah gaya putar. Adapun besarnya gaya yang bekerja pada konduktor tersebut dapat dirumuskan dengan :

$$F = B.I.L \sin \theta \quad (\text{Newton})$$

Dimana :     B   = kerapatan *fluks* magnet (weber)  
               L   = panjang konduktor (meter)  
               I   = arus listrik ( ampere)

$\sin \theta$  = sudut antara antara arus dengan garis-garis medan.

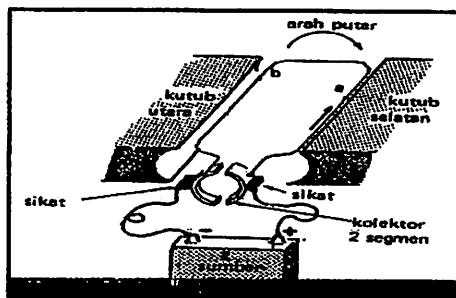


**Gambar 2. 12 Kaidah Tangan Kanan**

(sumber: Malvino, Prinsip prinsip elektronika)

### Cara Kerja Motor DC

Adapun cara kerja motor dc dapat dilihat pada gambar dibawah ini:



**Gambar 2. 13 Cara Kerja Motor DC**

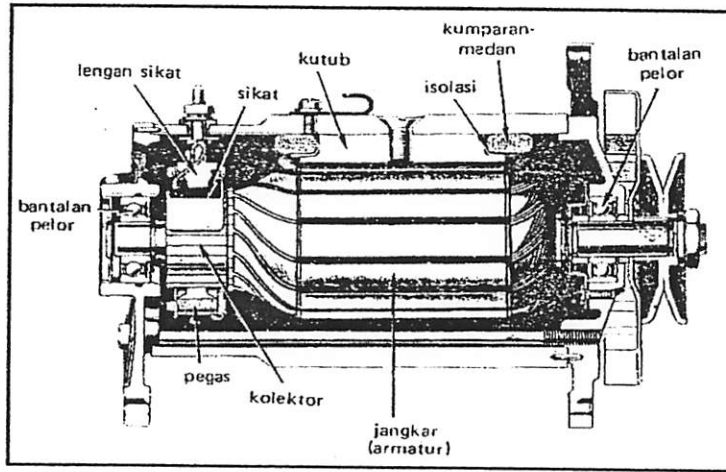
(sumber: Malvino, Prinsip prinsip elektronika)



Ada satu lilit kawat a – b berada di dalam medan magnet. Lilitan ini dapat berputar dengan bebas, lilitan ini bisasa disebut dengan jangkar (*armour*).

Pada jangkar dimasukkan arus yang berasal dari sumber (baterai) E. koneksi baterai dengan jangkar melalui sikat-sikat. Sikat-sikat ini terpasang pada sebuah cincin yang terbelah dua, yang disebut kolektir. Adapun tujuan dari kontruksi ini adalah agar lilitan kawat dapat berputar apabila ada arus listrik yang melewatinya.

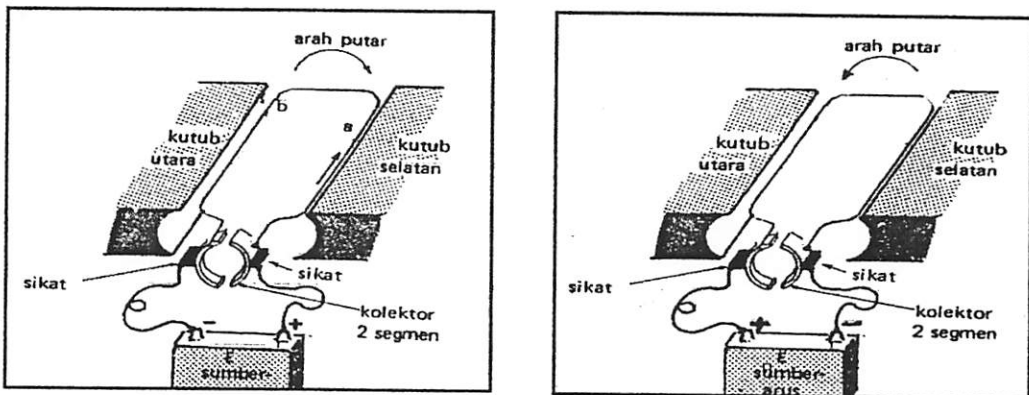
Pada kawat yang berada di kanan arus mengalir dari depan ke belakang . dalam kawat yang di kiri, arus mengalir dari belakang ke depan . kawat a dan b secara berganti-gantian berada di kiri dan kanan. Karena itu arah arus di a dan arah arus di b selalu membolak balik. Pembalikan arah arus itu terjadi pada saat lilitan kawat melintasi possisi vertikal. Disini kolektor berfungsi bagaikan penyearah mekanik. Flux magnet yang ditimbulkan magnet permanen disebut medan magnetnya motor. Dalam gambar arah fluk magnetik adalah dari kiri ke kanan. Adapun gaya yang bekerja pada penghantar b adalah ke atas, sementara gaya yang bekerja pada penghantar a adalah ke bawah . Gaya-gaya yang bekerja sama kuatnya, jadi ada kopel yang bekerja pada kawat sehingga lilitan pun dapat berputar. Setelah berputar  $90^0$  arah arus berbalik, pada saat itu penghantar a dan penghantar b bertukar tempat. Akibatnya arah gerak putaran tidak berubah.



**Gambar 2. 14 Motor DC**

*(sumber: Malvino, Prinsip prinsip elektronika)*

### Pengendalian Arah Putaran Motor DC



**Gambar 2. 15 Pengendalian Arah Putaran Motor DC**

*(sumber: Malvino, Prinsip prinsip elektronika)*

Dari gambar 2.15 diatas, agar arah putaran motor dc berubah, maka polaritas tegangan pada baterai harus dibalik.

## 2.10. Handphone Siemens C55

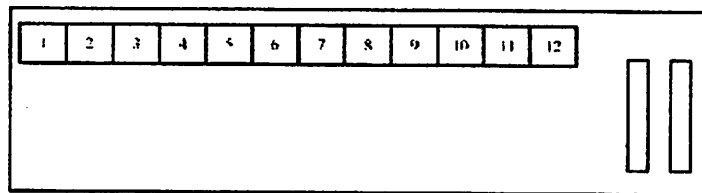
Pada saat ini, sistem komunikasi *Handphone* sudah berkembang pesat. Sistem *Handphone* merupakan bagian dari sistem komunikasi yang berfungsi sebagai media pengirim dan sekaligus sebagai media penerima, selain untuk komunikasi audio, pada handphone juga mampu untuk komunikasi data berupa teks yang sering disebut SMS (*Short Messages Service*) . gambar 2.17 menunjukkan handphone Siemens tipe C55.



**Gambar 2. 16 Handphone Siemens Tipe C55**

(Sumber : [www.elektronika-elektronika.blogspot.com](http://www.elektronika-elektronika.blogspot.com))

Untuk berkomunikasi dengan perangkat lain seperti PC (*Personal Computer*), ataupun Mikrokontroler handphone memiliki jalur data komunikasi serial yang terletak pada pin out-nya berikut merupakan konfigurasi dan fungsi masing–masing pin out dari handphone Siemens tipe C55. Gambar 2.17 menunjukkan pin out Siemens tipe C55. Sedangkan fungsi masing – masing pin out ditunjukkan pada tabel 2.6



**Gambar 2. 17 Pin Out Siemens Tipe C55**

(Sumber : [www.elektronika-elektronika.blogspot.com](http://www.elektronika-elektronika.blogspot.com))

Tabel 2. 6 Fungsi Masing-Masing Pin Out

Pin	Nama	Fungsi	In/Out
1	GND	Ground	
2	SELF SERVICE	Recognition / control battery charger	In/ Out
3	LOAD	Charging Voltage	In
4	BATTERY	Battery	Out
5	DATA OUT	Data Sent	Out
6	DATA IN	Data Received	In
7	Z_CLK	Recognition / control accessories	
8	Z_DATA	Recognition / control accessories	
9	MICG	Ground for microphone	In
10	MIC	Microphone input	
11	AUD	Loudspeaker	Out
12	AUDG	Ground for eksternal speaker	

(Sumber : [www.elektronika-elektronika.blogspot.com](http://www.elektronika-elektronika.blogspot.com))

### 2.10.1. Short Messages Service dan Protocol Data Unit

#### 2.10.1.1. Short Messages Service (SMS)

SMS adalah fasilitas yang dimiliki oleh jaringan GSM (*Global System for Mobile Communication*) yang memungkinkan pelanggan untuk mengirimkan dan menerima pesan-pesan singkat sepanjang 160 karakter. SMS ditangani oleh jaringan melalui pusat layanan atau *SMS Service Center (SMS SC)* yang berfungsi menyimpan dan meneruskan pesan dari sisi pengirim ke sisi penerima. Format SMS yang dipakai oleh produsen MS (*Mobile Station*) adalah *Protokol Data Unit (PDU)*. Format PDU akan mengubah septet kode ASCII (7 bit) menjadi bentuk Byte (8 bit) pada saat pengiriman data dan akan diubah kembali menjadi ASCII pada saat diterima oleh MS.

### **2.10.1.2. Sistem Kerja SMS**

Dibalik tampilan menu messages pada sebuah ponsel terdapat AT *Command* yang bertugas mengirim dan menerima data dari SMS *Center*. AT *Command* tiap-tiap SMS device bisa berbeda-beda, tetapi pada dasarnya sama. Perintah AT *Command* biasanya disediakan oleh vendor handphone.

Pada ponsel GSM terdapat fasilitas pengaksesan data melalui koneksi serial atau dengan antarmuka infra merah. Untuk mengakses data, diperlukan urutan instruksi pada antarmuka ponsel. ETSI (*European Telecommunication Standard Institute*) menstandarkan instruksi tersebut dalam spesifikasi teknik GSM pada dokumen GSM 07.07 dan GSM 07.05, dimana setiap ponsel harus mengacu pada instruksi tersebut, instruksi ponsel diawali dengan karakter AT dan diakhiri dengan enter atau 0Dh. Perintah yang diterima akan direspon dengan diterimanya data 'OK' atau 'Error'. Instruksi yang diterima oleh ponsel dan sedang diproses akan terinterupsi oleh instruksi lain sehingga setiap pengiriman instruksi harus menunggu datangnya respon dari ponsel.

Beberapa contoh AT Command yang penting untuk SMS adalah sebagai berikut :

AT+CMGS – Untuk mengirim SMS

AT+CMGL - Untuk memeriksa SMS

AT+CMGD – Untuk menghapus SMS

AT+CMGR – Untuk membaca SMS

### **2.10.1.3. Protocol Data Unit**

AT Command untuk SMS biasanya diikuti oleh data I/O yang diwakili oleh unit-unit PDU. Data yang mengalir ke dan dari SMS Center harus berbentuk PDU (*Protocol Data Unit*). PDU berisi bilangan – bilangan heksadesimal yang mencerminkan I/O. PDU terdiri dari beberapa header. Header untuk mengirim SMS center berbeda dengan SMS yang diterima SMS center.

Terdapat delapan header untuk mengirim SMS, header – header tersebut adalah :

#### **1. Nomor SMS center**

Header pertama ini terdiri dari atas tiga sub header, yaitu :

- a. Jumlah pasangan heksadesimal SMS center dalam bilangan heksa.
- b. National atau international kode  
Untuk national, kode sub-headernya adalah 81  
Untuk international, kode sub-headernya adalah 91.
- c. Nomor SMS center sendiri dalam pasangan heksa dibalik-balik.

Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan F didepannya.

Contoh nomor SMS Center Indosat-M3 dapat ditulis dengan dua cara

- Cara 1 (National)

0855000000 diubah menjadi :

- 06 : ada 6 pasang.
- 81 : 1 pasang.
- 80-55-00-00-00 : 5 pasang.

Digabung menjadi : 06818055000000

- Cara 2 (International)

62855000000 diubah menjadi :

- 07 : ada 7 pasang.
- 91 : 1 pasang .
- 26-58-05-00-00-F0

Digabung menjadi : 07912658050000F0

Berikut beberapa nomor SMS- Center operator seluler di Indonesia.

Tabel 2.7 menunjukkan nomor sms center dengan cara *national code*, sedangkan pada tabel 2.3 menunjukan nomor sms center dengan cara *international code*.

Tabel 2. 7 Nomor SMS Center Dengan Cara *National Code*

No	Operator Seluler	SMS Center	Kode PDU
1	Telkomsel	0811000000	06818011000000
2	Satelindo	0816124	0681806121F4
3	Excelcom	0818445009	06818081440590
4	Indosat - M3	0855000000	06818055000000

(Sumber : [www.elektronika-elektronika.blogspot.com](http://www.elektronika-elektronika.blogspot.com))

Tabel 2. 8 Nomor SMS Center Dengan Cara *International Code*

No	Operator Seluler	SMS Center	Kode PDU
1	Telkomsel	0811000000	07912618010000F0
2	Satelindo	0816124	059126181642
3	Excelcom	0818445009	07912618485400F9
4	Indosat - M3	0855000000	07912658050000F0

(Sumber : [www.elektronika-elektronika.blogspot.com](http://www.elektronika-elektronika.blogspot.com))

## 2. Tipe SMS

Tipe send, tipe SMS = 1. Jadi bilangan heksanya adalah 01.

## 3. Nomor referensi SMS

Nomor referensi dibiarkan 0. Jadi bilangan heksanya 00.

Selanjutnya akan diberikan sebuah nomor referensi otomatis oleh ponsel atau alat SMS – *gateway*.

## 4. Nomor Ponsel Penerima

Sama seperti menulis PDU header untuk SMS center, header ini juga terbagi atas tiga bagian, yaitu :



- a. Jumlah bilangan desimal nomor ponsel yang dituju dalam bilangan heksa.
- b. *National code* atau *International code*.
- c. Untuk *national code* sub-headernya : 81.
- d. untuk *international code*, kode sub-headernya : 91.
- e. Nomor ponsel yang dituju dalam pasangan heksa dibalik – balik.

Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut dipasangkan dengan F di depannya.

Contoh :

Nomor ponsel yang dituju : 628129573337 dapat ditulis dengan dua cara sebagai berikut :

Cara 1 : 0812957337 diubah menjadi :

a. 0B : ada 11 angka

b. 81

c. 80-21-59-37-33-F7

Digabung : 0B818021593733F7

Cara 2 : 628129573337 diubah menjadi :

a. 0C ada 12 angka

b. 91

c. 26-81-92-75-33-73

Digabung menjadi : 0C91261892753373

5. Bentuk SMS

0 : 00 >> Dikirim sebagai SMS

1 : 01 >> Dikirim sebagai telex

2 : 02 >> Dikirim sebagai fax

Dalam hal ini, pengiriman dalam bentuk SMS sehingga memakai bilangan heksa 00

6. Skema *Encoding Data I/O*

Ada dua skema *encoding data* yaitu :

- a. Skema 7bit >> Ditandai dengan angka 0 : 00 heksa
- b. Skema 8bit >> Ditandai dengan angka lebih besar dari 0 yang diubah ke heksa.

Pada umumnya SMS *gateway* yang tersedia pada Service Center menggunakan Skema 7bit sehingga menggunakan kode 00 heksa

7. Jangka waktu Sebelum SMS *expired*

Jika bagian ini tidak diisi, maka tidak ada batasan waktu berlakunya SMS, tetapi jika diisi dengan bilangan integer yang

diubah ke dalam pasangan heksa tertentu, bilangan yang diberikan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

## 8. Isi SMS

Header ini terdiri atas dua sub-header, yaitu :

a. Panjang isi atau jumlah huruf dari isi

Misalnya, untuk kata "hello" terdapat 5 huruf : 05 heksa.

b. Isi berupa pasangan bilangan heksa

Ponsel atau SMS *gateway* berskema *encoding 7bit* berarti jika kita mengetikkan suatu huruf dari *keypad* – nya, kita telah membuat & angka I/O berturutan.

Ada dua langkah yang harus dilakukan untuk mengkonversikan isi sms, yaitu :

1. Mengubah isi SMS tersebut menjadi kode 7bit.
2. Mengubah kode 7bit menjadi 8bit, yang diwakili oleh pasangan heksa.

Contoh untuk kata "hello"

• Langkah pertama :

Bit	7	1
h	110	1000
e	110	0101
l	110	1100

l 110 1100  
o 110 1111

- Langkah kedua :

E 8  
h 1 110 1000  
3 2  
e 00 11 0010 1  
9 B  
l 100 1 1011 00  
F D  
l 1111 1101 100  
0 6  
o 0000 0 110 1111

Oleh karena total 7bit x 5 huruf = 35 bit, sedangkan yang diperlukan adalah 8 x 5 bit = 40 bit, maka diperlukan 5 bit dummy yang diisi dengan bilangan 0. Dengan demikian, kata “hello” hasil konversinya adalah E8329BFD06.

## 9. Menggabungkan Delapan Header

Masing – masing header maupun sub-header harus digabungkan menjadi sebuah PDU yang lengkap.

Contoh, akan dikirimkan kata “hello” ke ponsel nomor 62812957337 lewat SMS center Indosat-M3, tanpa validitas waktu, maka PDU lengkapnya adalah:

07912658050000F001000C91261892753373000005E8329B

FD06

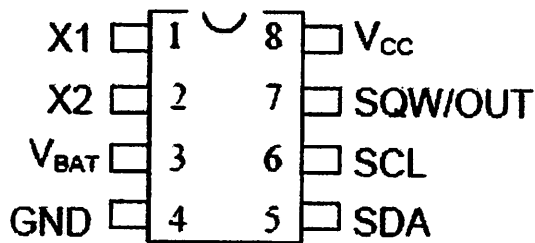
### 2.11. RTC (*Real Time Clock*)

DS1307 merupakan Real-time clock (RTC) dengan jalur data parallel yang memiliki Antarmuka serial *Two-wire* (I2C), Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*), Deteksi otomatis kegagalan-daya (*powerfail*) dan rangkaian *switch*, Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional osilator. Tersedia fitur industri dengan ketahanan suhu: -40°C hingga +85°C. Tersedia dalam kemasan 8-pin DIP atau SOIC

Tabel 2. 9 Daftar pin DS1307

VCC	Primary Power Supply
X1, X2	32.768kHz Crystal Connection
VBAT	+3V Battery Input
GND	Ground
SDA	Serial Data
SCL	Serial Clock
SQW/OUT	Square Wave/Output Driver

(sumber: data sheet RTC 1307)



**Gambar 2. 18 DS 1307 8-DIP PIN**

*(sumber: data sheet RTC 1307)*

Untuk masing-masing pin akan di jelaskan sebagai berikut:

1. **X1**  
Merupakan pin yang digunakan untuk dihubungkan dengan
2. **X2**  
Berfungsi sebagai keluaran/ output dari crystal yang digunakan. Terhubung juga dengan X1
3. **VBAT**  
Merupakan backup supply untuk serial RTC dalam menjalankan fungsi waktu dan tanggal. Besarnya adalah 3V dengan menggunakan jenis Lithium Cell atau sumber energy lain. Jika pin ini tidak di gunakan maka harus terhubung dengan Ground. Sumber tegangan dengan 48mAH atau lebih besar dapat digunakan sebagai cadangan energy sampai lebih dar 10 tahun, namun dengan persyaratan untuk pengoprasian dalam suhu 25°C.
4. **GND**  
Berfungsi sebagai Ground
5. **SDA**

Berfungsi sebagai masukan / keluaran (I/O) untuk I2C serial interface. Pin ini bersifat open drain, oleh sebab itu membutuhkan eksternal pull up resistor.

6. SCL

Berfungsi sebagai clock untuk input ke I2C dan digunakan untuk mensinkronisasipergerakan data dalam serial interface. bersifat open drain, oleh sebab itu membutuhkan eksternal pull up resistor.

7. SWQ/OUT

Sebagai square wave / Output Driver . jika di aktifkan, maka akan menjadi 4 frekuensi gelombang kotak yaitu 1 Hz, 4kHz, 8kHz, 32kHz sifat dari pin ini sama dengan sifat pin SDA dan SCL sehingga membutuhkan eksternal pull up resistor. Dapat dioprasikan dengan VCC maupun dengan VBAT.

8. VCC

Merupakan sumber tegangan utama. Jika sumber tegangan terhubung dengan baik, maka pengaksesan data dan pembacaan data dapat dilakukan dengan baik. Namun jika backup supply terhubung juga dengan VCC, namun besar VCC di bawah VTP, maka pengaksesan data tidak dapat dilakukan.

## **BAB III**

### **PERANCANGAN DAN PEMBUATAN ALAT**

#### **3.1. Pendahuluan**

Dalam bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan juga fungsi masing-masing blok rangkaian tersebut.

Secara garis besar terdapat dua bagian perangkat yang ada yaitu :

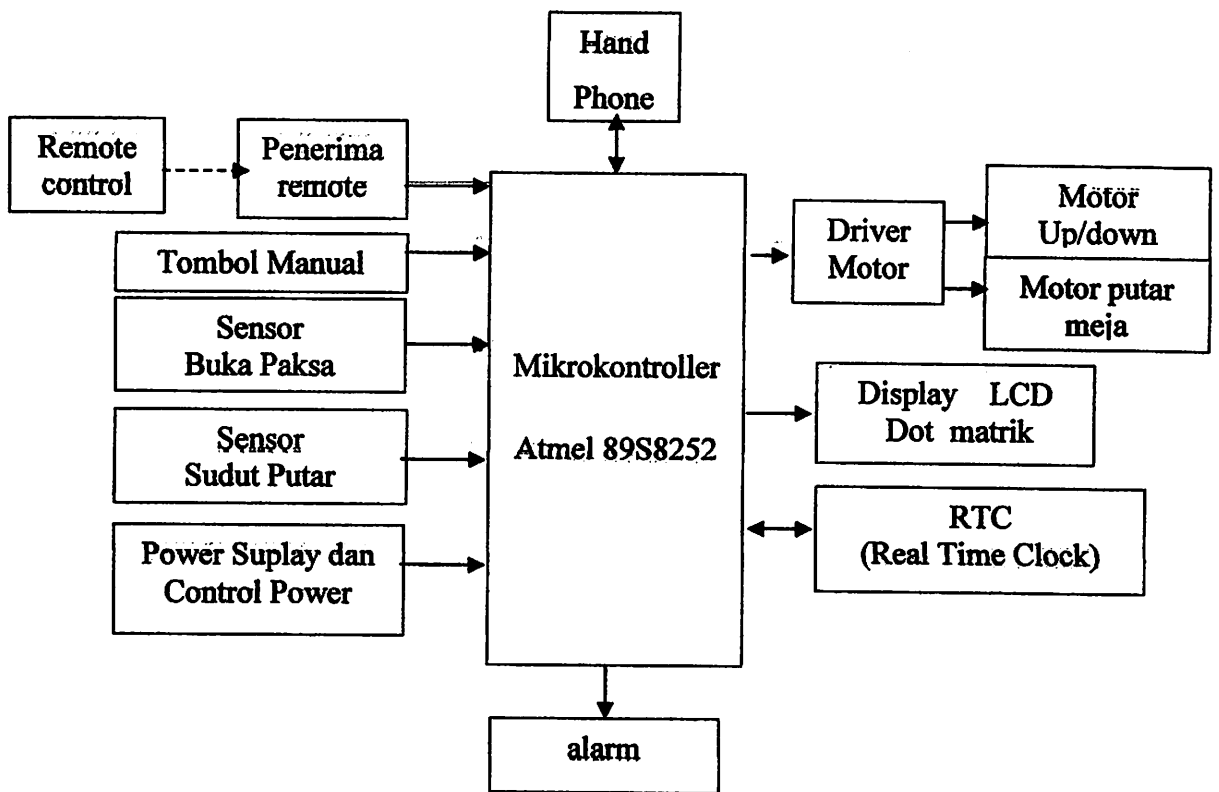
- Perancangan perangkat keras (*Hardware*).
- Perancangan perangkat lunak (*Software*).

Pada perancangan perangkat keras akan meliputi *peripheral-peripheral* yang digunakan pada sistem ini. Sedangkan pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum. Akan tetapi kedua perangkat ini dalam kerjanya saling menunjang satu sama lain.



### 3.2. Blok Diagram Alat

Adapun prinsip kerja dari interior meja televisi dapat dilihat pada blok diagram dibawah ini :



**Gambar 3 1 Blok diagram sistem keseluruhan**

Fungsi dari tiap – tiap blok diagram dijelaskan sebagai berikut :

**Remote Control** : Remote control memberikan inputan kepada penerima untuk membuka (mengangkat televisi dari dalam meja) dan menutup serta menghidupkan power suplay televisi.

- Penerima Remote** : Menerima inputan dari remote agar dapat membedakan intruksi dari tiap-tiap tombol kemudian di teruskan ke mikrokontroller.
- Tombol Manual** : Digunakan untuk menggantikan remote agar dapat mengantisipasi kegagalan remote.
- Sensor Buka Paksa** : Sensor ini menggunakan limit switch yang dipasang di tutup meja, cara kerja limit switch ini adalah memberikan logika on dan off pada mikrokontroller.
- Sensor Posisi** : Sensor ini untuk menentukan sudut putaran meja atas
- Sensor sudut** : Sensor ini untuk menentukan sudut putaran meja atas dan agar pada saat televisi ditutup arah televisi kembali seperti semula dibuka.
- Power control** : Unit untuk mengantisipasi apabila tegangan dari sumber listrik tegangan ac mati maka akan dialihkan ke tegangan cadangan dari batrai
- Alarm** : Alarm akan berbunyi apabila tempat penyimpanan televisi dibuka secara paksa tanpa menggunakan remote atau tombol manual.
- Hand phone** : Handphone berfungsi untuk memberikan informasi melalui sms apabila televisi diangkat dari atas meja atau tutup meja televisi dibuka secara paksa difungsikan.

- Driver Motor** : Digunakan untuk menggerakkan 2 bush motor DC yang mengangkat/menurunkan televisi dan membuka/menutup atas meja
- Motor DC** : Motor digunakan sebagai penggerak, diantaranya untuk mengangkat (motor Up/Down), memutar bagian atas meja  
Motor menggunakan motor DC 12 Volt
- Mikrokontroller** : Mikrokontroller digunakan untuk mengontrol sistem dan mengatur proses pengiriman data menuntukan arah pergerakan motor.
- Display LCD** : Menggunakan display Lcd dot matrik 16 karakter dan 2 baris.
- RTC** : RTC (Real Time Clock) digunakan sebagai timer untuk menghitung waktu mulai dari detik, menit sampai dengan tahun. Walaupun sistem dimatikan akan tetapi RTC tetap menyala dikarenakan memiliki power suplay sendiri. Dan dapat juga difungsikan sebagai timer sleep

### **3.3. Prinsip Kerja Alat**

Untuk menggunakan interior meja televisi ini, pemilik harus menggunakan remote control. Dalam remote control terdapat beberapa tombol yang dapat digunakan. Untuk mengngukan atau menghidupkan televisi untuk pertama kalinya, pemilik harus mengangkat televisi ke atas meja serta membuka tutup meja dan sekaligus mematikan sensor alarm.

Dengan menggunakan remote control pertama-tama pemilik menekan tombol power guna menghidupkan power listrik televisi dan meja televisi ini. Kemudian sistem meminta untuk mengisikan password yang ditampilkan pada display dot matrik. Setelah password dimasukan dengan benar, meja tersebut secara otomatis akan mengangkat televisi dari dalam meja bersamaan dengan penutup dibuka dan mematikan fungsi sensor. Kemudian fungsi tombol lainnya dapat digunakan.

Salah satu fungsi tombolnya adalah dapat memutar televisi sebesar 90 derajat sehingga memudahkan pemilik dapat menentukan posisi yang nyaman untuk menonton televisi. Dengan remote control ini dapat juga digunakan untuk mematikan alarm yang sedang berbunyi. Dengan menetakan tombol off alarm maka sistem akan meminta untuk memasukan password. Setelah password yang dimasukan dengan benar, maka sistem akan mematikan alarm.

Untuk mengakhiri penggunaan televisi maka pemilik menekan tombol power dan ditahan selama 30 detik kemudian sistem akan mengatur sudut televisi kembali seperti semula dan menurunkan televisi dan penutup meja akan menutup kemudian sensor buka paksa akan aktif kembali.

Apabila alarm berbunyi dikarenakan adanya penutup meja dibuka secara paksa maka sistem secara bersamaan mengirimkan informasi kepada pemilik melalui pesan singkat (sms). Alarm tersebut dapat juga dimatikan melalui sms dengan format yang sudah diatur oleh pemilik.

### 3.4. Perancangan Perangkat Keras

Perancangan perangkat keras yang direncanakan meliputi: pembuatan mekanik, blok diagram keseluruhan dan prinsip kerja alat, pembuatan skema seluruh rangkaian yang direncanakan, penghitungan nilai komponen yang digunakan, pembuatan PCB, dan perakitan seluruh komponen.

#### 3.4.1. Remote control

##### 3.4.1.1. Remote pengirim

Remote yang digunakan adalah remote sony RM 870 dan tombol yang digunakan memiliki fungsi masing-masing. Tombol dari remote ini berjumlah 27 buah dengan model standart buatan pabrik . Fungsi dari masing masing tombol dapat dilihat di tabel 3.1

Tabel 3 1 Fungsi Tombol Remote

Nama Tombol	Fungsi Step1	Fungsi step 2	Fungsi Strep 3
Power	Mengaktifkan sistem dan meminta kata sandi	Membatalkan kata sandi	Mematikan sistem
Mutting	Informasi tanggal		
Display	Informasi Hp		
Text			
Video			
TV			
1		Angka 1	Posisi 10° kiri
2		Angka 2	Posisi 20° kiri
3		Angka 3	Posisi 30° kiri
4		Angka 4	
5		Angka 5	Posisi center
6		Angka 6	
7		Angka 7	Posisi 10° kanan
8		Angka 8	Posisi 20° kanan
9		Angka 9	Posisi 30° kanan
0		Angka 0	Posisi center

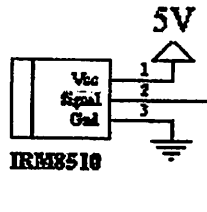
-/--		<i>Backspase</i>	
Text Clr		Hapus layar	
+■			
SELECT		Ok konfirmasi	
A/B			
-			
PIC MODE			
Wake Up			
Sleep			
Vol +			Putar kanan
Vol -			Putar kiri
Prog +			Putar kanan
Prog -			Putar kiri

ket: yang kosong tidak difungsikan

Fungsi step 1 saat awal sistem hidup karena tv masih didalam maka sebulum menekan tombol power maka sistem belum aktif . setelah menekan tombol power maka akan berlanjut kefungsi step2 dimana sistem akan meminta kata sandi. Setelah kata sandi benar maka dilanjutkan ke fungsi step 3

#### 3.4.1.2. Penerima remote

Untuk perancangan IRM digunakan IRM 8510 karena IRM dapat menerima panjang gelombang infra merah 10 – 10 m dan frekwensi sebesar 10 – 10 Hz, dengan frekwensi carrier 38 – 40 KHz. Karena proses penerimaannya pada prinsipnya sama dengan proses fotodiode yang terintegrasi bersama satu komponen Ic. Perencanaan IRM dapat dilihat pada gambar 3-7 berikut :

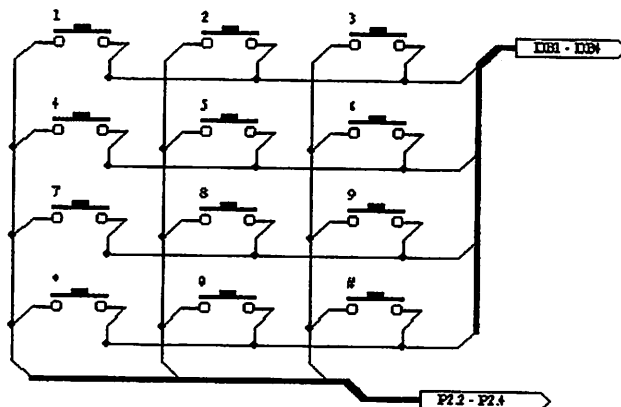


**Gambar 3 2** IRM 8510

### **3.4.2. Tombol Manual**

Keypad digunakan sebagai masukkan data dan mengubah data yang diinginkan. Data tersebut dapat dirubah dalam bentuk kode biner oleh rangkaian keypad.

Keypad yang dipakai yaitu berukuran 3 x 4 ( 3 bagian kolom dan 4 bagian baris ) yang terdiri dari saklar angka 0 sampai 9 sebagai masukkan data dan karakter Pagar (#) yang digunakan sebagai data untuk proses setting ( *enter* ) suhu dan Waktu serta Bintang (\*) sebagai pembatalan ( *cancel* ).

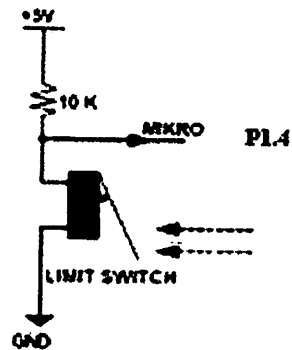


**Gambar 3 3** Rangkaian Keypad

### 3.4.3. Sensor

#### 3.4.3.1. Sensor Buka Paksa

Sensor ini menggunakan saklar limit switch yang memiliki besaran fisik yang kecil. Prinsip kerjanya adalah on/off sebagai fungsi untuk mengetahui bahwa meja terbuka atau tidak sehingga memudahkan mikro untuk menentukan bahwa meja dibuka secara paksa tanpa dibuka dengan sistem program yang semestinya



Gambar 3 4 limit swieth

#### 3.4.3.2. Sensor Sudut Putar

Sensor sudut menggunakan modifikasi saklar rotary encoder dengan sistem saklar geser yang berputar pada poros dan akan terbagi menjadi 7 sudut yang akan menjadi batasan. Pasisi saklar juga dapat membantu program untuk mengetahui posisi meja atas.

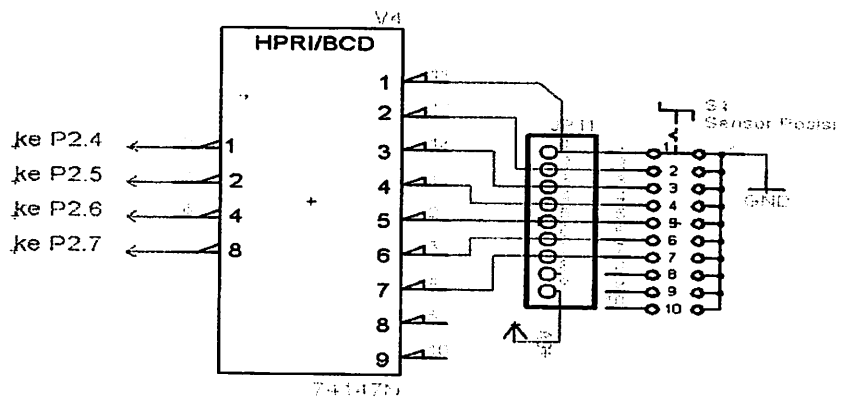
Dengan 7 inputan dari sensor dimasukan ke encoder decimal to bcd menggunakan IC 74ls147. Sehingga inputan yang masuk ke mikrokontroller



hanya 3 bit bcd. Komsumsi daya yang diserap hanya menggunakan 5 volt dc

Tabel 3 2 Sudut Posisi Meja

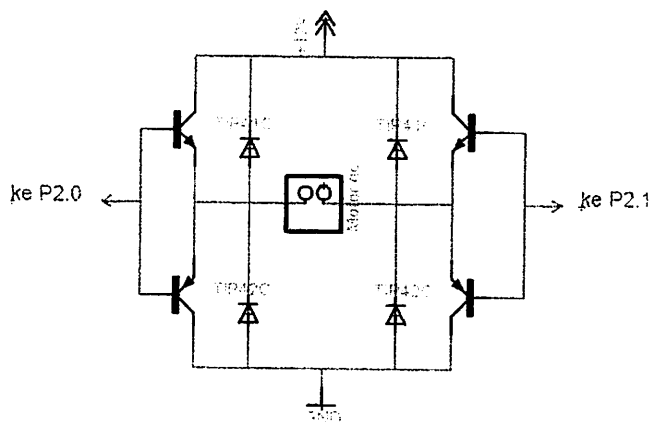
Posisi	1	2	3	4	5	6	7
Sudut	30°	20°	10°	Center	10°	20°	30°
Arah	kiri	kiri	kiri	Tengah	Kanan	Kanan	Kanan



Gambar 3 5 Rangkaian Sudut Putar

#### 3.4.4. Driver Motor DC

Motor disini menggunakan motor DC dengan catu daya 12 Volt DC dan diantara kutub-kutub motor terdapat 4 buah transistor. Transistor-transistor diatas merupakan rangkaian BTL ( Bridge Transformer Less ). Yang mana berfungsi sebagai merubah polaritas dari motor DC tersebut.



**Gambar 3 6 Driver motor DC**

Pada gambar diatas dapat kita lihat jika  $Tr_1$  dan  $Tr_2$  diberi masukan negative 0 ( Low ) maka  $Tr_1$  akan OFF dan  $Tr_2$  akan ON. Untuk  $Tr_3$  dan  $Tr_4$  diberi masukan positif 1 ( Hight ) maka  $Tr_3$  akan ON dan  $Tr_4$  akan OFF. Begitu pula sebaliknya.

**Tabel 3 3 Tabel Kebenaran Driver Motor**

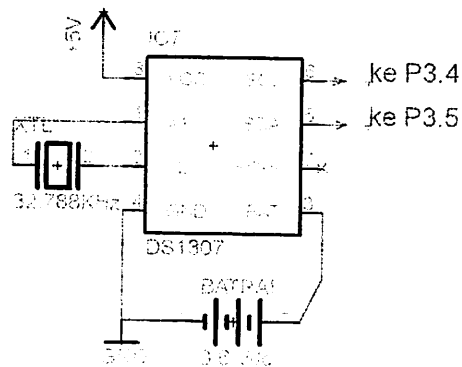
Kondisi	P2.0	P2.1	Motor
1	0	0	Berhenti
2	0	1	Berputar kekiri
3	1	0	Berputar kekanan
4	1	1	Berhenti cepat

### 3.4.5. Rangkaian Timer

Rangkaian timer dihubungkan ke mikrikontroller menggunakan 2 kabel dengan menggunakan sistem komunikasi I2C. IC RTC yang digunakan adalah IC produksi Dalas Instrumen yaitu DS1307 yang memiliki 8 pin lebih menghemat pin dibandingkan dengan rangkaian lainnya. Untuk

menjaga kestabilan penghitungan, maka ditambahkan baterai sebagai suplai perhitungan waktu. Maksud dari penambahan ini adalah untuk mengantisipasi kemungkinan sistem akan mati yang akan menghentikan perhitungan.

Antara mikrokontroller dan RTC interfacenya menggunakan SDA dan SCL. Pin SCL RTC yang berada di kaki 6 dihubungkan ke P3.4 mikrokontroller dan SDA sebagai data yang berada pada kaki 5 dihubungkan ke P3.5 mikrokontroller.



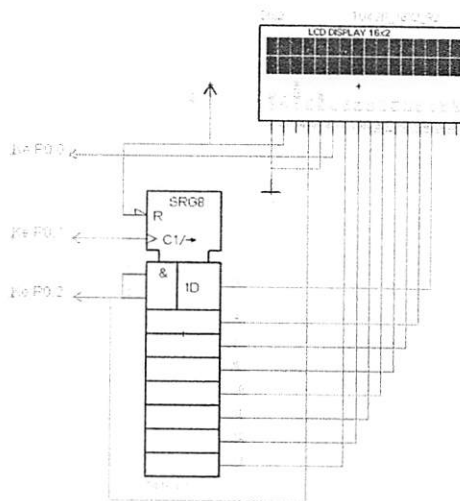
**Gambar 3 7 Rangkaian RTC DS1307**

### 3.4.6. Rangkaian LCD

Untuk berhubungan dengan mikrokontroller, pemakai LCD M1632 dilengkapi dengan 8 jalur data (DB0..DB7) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan E, R/W\* dan RS seperti layaknya komponen yang kompatibel dengan mikroprosesor.

Untuk menghemat pemakaian pin maka ditambahkan ic shif register 74ls164 yang dihubungkan ke jalur data. Ic ini akan menerima data serial kemudian di keluarkan secara paralel setiap bit dicontrol melalu clock yang dibuat oleh program mikrokontroller. Sedangkan untuk pengiriman ke mikro kita menontrol RS dan E saja. Sehingga jalur yang digunakan hanya 3 bit dapat menghemat 7 bit dari 10 bit yang biasa digunakan apabila dihubungkan langsung.

RS, singkatan dari Register Select, dipakai untuk membedakan jenis data yang dikirim ke M1632, apabila RS=0 data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau RS=1 data yang dikirim adalah kode ASCII yang ditampilkan.

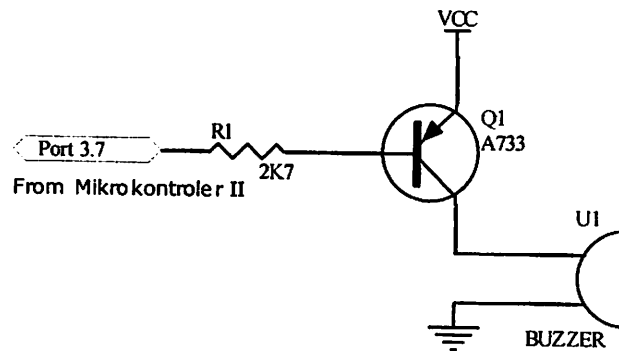


**Gambar 3 8 Rangkaian LCD**

### 3.4.7. Alarm

Alarm yang digunakan menggunakan buzzer. Untuk menghidupkan buzzer diperlukan suatu *driver*. Rangkaian *driver* buzzer ini

menggunakan transistor jenis PNP A733 yang dihubungkan dengan port 3.7 pada mikrokontroler II.



**Gambar 3 9 Rangkaian Buzzer**

Sumber : Perancangan

Diket :

$$H_{fe} \text{ Transistor A733} = \beta_{dc} = 90$$

$$I_{\text{buzzer}} = 2,04 \text{ mA saat } V = 5 \text{ V}$$

$$I_{b \text{ max}} = I_{OL} = 1,6 \text{ mA}$$

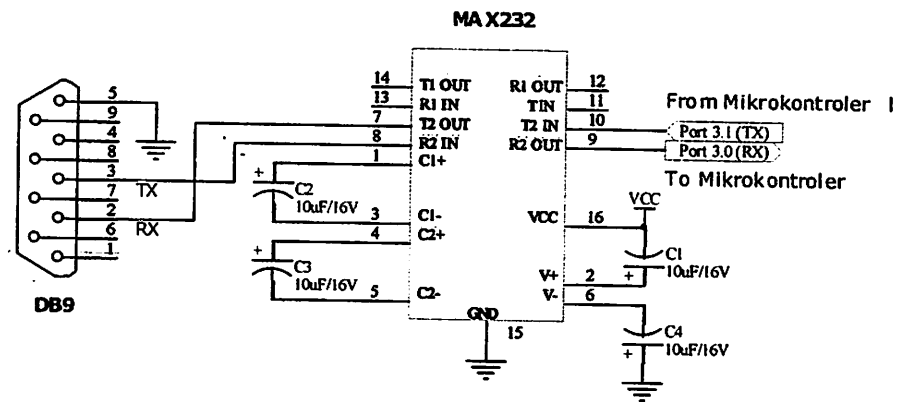
Sehingga besar  $R_B$  adalah :

$$\begin{aligned} R_b &= \frac{V_{CC} - V_{be} - V_{OL}}{I_b} \\ &= \frac{5 - 0,7 - 0,45}{1,6 \text{ mA}} \\ &= 2,40625 \text{ K}\Omega \end{aligned}$$

Karena resistor sebesar  $2,40625 \text{ K}\Omega$  tidak ada maka digunakan resistor sebesar  $2,7 \text{ K}\Omega$ .

### 3.4.8. Interface serial

Pada perancangan interface serial ini menggunakan IC MAX 232 ini, di butuhkan penyambungan pin-pin. Adapun pin-pin yang digunakan dalam rangkaian IC MAX 232 adalah pin 1 dihubungkan ke pin 3, pin 4 dihubungkan ke pin 5, dan pin 2 dihubungkan ke pin 6, dan masing-masing dipasang kapasitor 22 $\mu$ /16. Pin 7 dan pin 8 dihubungkan ke handphone. Pin 9 dihubungkan ke port 3.0, yang digunakan sebagai modem untuk memberi tahu bahwa data sudah di terima. Pin 10 dihubungkan ke port 3.1, yang digunakan sebagai modem untuk memberi tahu bahwa data sudah di kirim. Pin 16 (Vcc) dihubungkan dengan tegangan supply +5 volt. Pin 15 (GND) dihubungkan dengan tegangan supply ground.



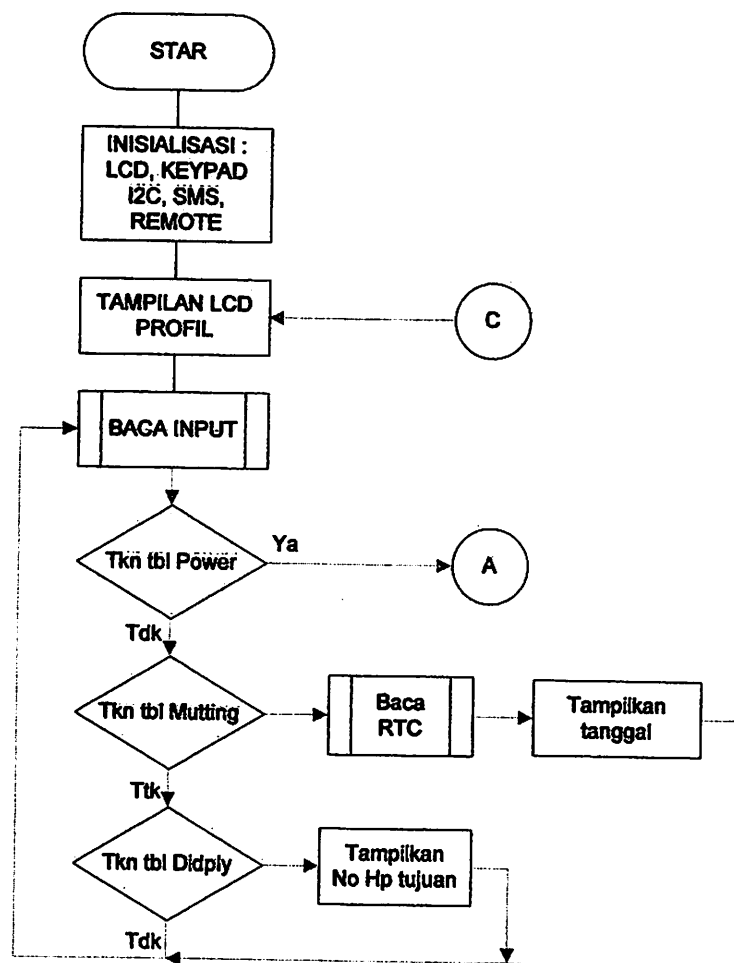
Gambar 3 10 Rangkaian Interface Serial

### 3.5. Perancangan Perangkat Lunak

Perancangan perangkat lunak (software) didasarkan pada perancangan perangkat keras yang telah dibuat sebelumnya, guna mendapatkan sistem kerja yang diharapkan. Pembuatan program menggunakan M-IDE dengan bahasa pemrograman C :

#### 3.5.1. Step Pertama

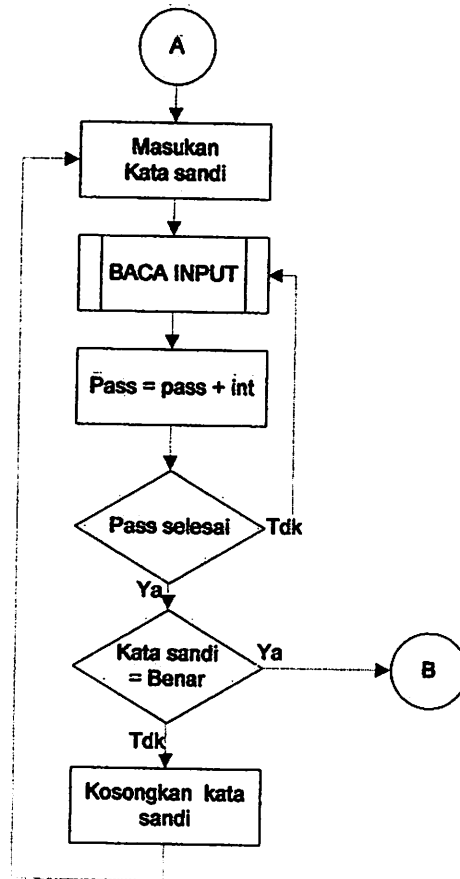
Step pertama ini merupakan awal sistem dihidupkan



Gambar 3 11 Flow Char Step Pertama

### 3.5.2. Step Kedua

Step Kedua merupakan pemasukan kata sandi serta mencocokkan kata sandi yang diinputkan

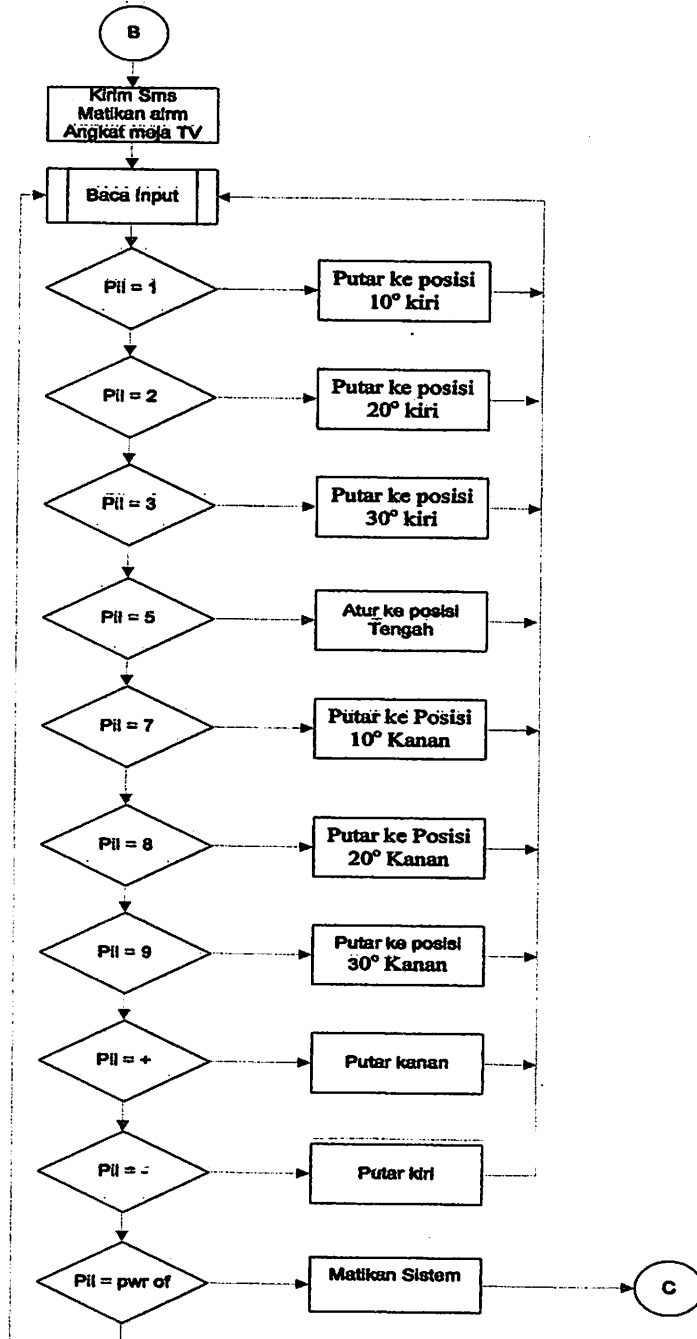


Gambar 3 12 Flow Char Step Kedua



### 3.5.3. Step Ketiga

Pada step ini merupakan proses penggunaan untuk pengaturan posisi televisi



Gambar 3 13 Flow Char Step Ketiga

## **BAB IV**

### **PENGUJIAN DAN HASIL ANALISA**

#### **4.1. Pendahuluan**

Pada bab ini membahas cara pengujian dan analisa dari alat yang dirancang, sehingga dapat diketahui apakah alat tersebut dapat bekerja sesuai dengan yang telah direncanakan. Dalam rangka pengujian alat tersebut diuraikan percobaan yang dilakukan untuk mengetahui respon dari keseluruhan alat yang telah dirancang.

Untuk mengetahui kemampuan alat dan sistem kerja sesuai dengan program yang telah dibuat maka dilakukan pengujian pada alat dan sistem kerja alat.

#### **4.2. Penguji Rangkaian Penerima Remote**

##### **4.2.1. Tujuan**

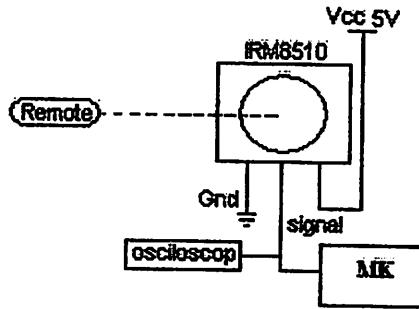
Tujuan dari pengujian remote ini adalah untuk mengetahui kerja keseluruhan tombol pada remote yang digunakan.

##### **4.2.2. Alat Yang Digunakan**

1. Rangkaian Driver IRM (Infra red receiver modul ).
2. Catu Daya 5 Volt DC.
3. Remote Tv Sony.
4. Osciloscop.

##### **4.2.3. Langkah Pengujian**

Merangkai rangkaian penerima intruksi Remote Tv Sony RM870 seperti pada gambar berikut:



**Gambar 4 12** Pengujian Rangkaian RM870


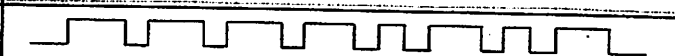
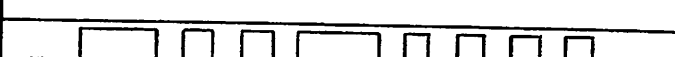
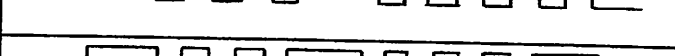
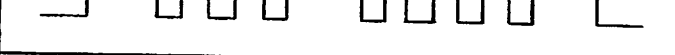

Penekanan tombol pada remote kemudian hasil dibaca pada oscilloscope hasilnya digambarkan pada tabel.

Kode yang dikembangkan oleh perusahaan SONY ini menggunakan frekuensi *sub-carrier* 40 kHz dan selang waktu minimum antara 2 data = 25 ms.

#### 4.2.4. Hasil Pengujian

Tabel 4. 1 Hasil pengujian penerima remote

Tombol	Hexa	Biner	Bentuk sinyal
1	#080	1000 0000	
2	#081	1000 0001	
3	#082	1000 0010	
4	#083	1000 0011	
5	#084	1000 0100	
6	#085	1000 0101	
7	#086	1000 0110	
8	#087	1000 0111	
9	#088	1000 1000	
0	#089	1000 1001	

1-	#0F4	1111 0100	
2-	#0F5	1111 0101	
+ Prog	#090	1001 0000	
- Prog	#091	1001 0001	
+ Vol	#092	1001 0010	
- Vol	#093	1001 0011	

### 4.3. Tombol Manual

#### 4.3.1. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui jalur pada kaki keypad 4x3 berfungsi dengan baik

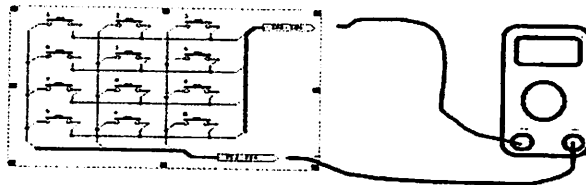
#### 4.3.2. Alat Yang Digunakan

1. Keypad 4x3
2. Multimeter

#### 4.3.3. Langkah Pengujian

1. Gunakan multimeter dan set pada posisi buzzer ( apabila terhubung akan mengeluarkan bunyi)
2. Hubungkan kabel merah (+) multimeter ke colom pertama dan kabel hitam (-) ke baris 1,2,3,dan 4
3. Tekan tombol pada keypad 1,4,7,\*
4. Hubungkan kabel merah (+) multimeter ke colom kedua dan kabel hitam (-) ke baris 1,2,3,dan 4
5. Tekan tombol pada keypad 2,5,8,dan 0

6. Hubungkan kabel merah (+) multimeter ke kolom ketiga dan kabel hitam (-) ke baris 1,2,3,dan 4
7. Tekan tombol pada keypad 3,6,9 dan #
8. Catat hasil pengujian



**Gambar 4. 1 uji keypad**

#### 4.3.4. Hasil Dan Analisa Pengujian

**Tabel 4. 2 Hasil Pengujian Keypad**

Tekan Tombol	Colom			Baris			
	1	2	3	1	2	3	4
1	1	0	0	1	0	0	0
2	0	1	0	1	0	0	0
3	0	0	1	1	0	0	0
4	1	0	0	0	1	0	0
5	0	1	0	0	1	0	0
6	0	0	1	0	1	0	0
7	1	0	0	0	0	1	0
8	0	1	0	0	0	1	0
9	0	0	1	0	0	1	0
0	1	0	0	0	0	0	1
*	0	1	0	0	0	0	1
#	0	0	1	0	0	0	1

Ket: logic 0 tidak terhubung  
Logic 1 berhubungan

#### 4.4. Sensor Sudut

##### 4.4.1. Tujuan

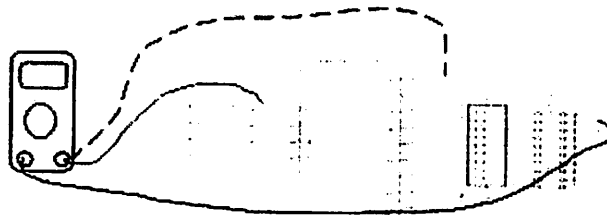
Tujuan dari pengujian ini adalah untuk mengetahui data yang dikeluarkan ke mikro oleh rangkaian sesuai dengan sudut putaran

#### 4.4.2. Alat Yang Digunakan

1. Rangkaian Sensor Sudut putar
2. Indikator logic
3. Power suplay +5 Vdc

#### 4.4.3. Langkah Pengujian

1. Hubungkan rangkaian dengan power suplay
2. Tentukan sudut putaran menjadi 7 bagian
3. Putar meja dengan sudut yang telah ditentukan
4. Catat hasil yang dikeluarkan



Gambar 4. 2 uji sensor sudut

#### 4.4.4. Hasil Pengujian

Tabel 4. 3 Hasil pengamata ouput data Sensor Sudut

Sudut meja	Out data	Out BCD ke MK
Kiri 30°	0111111	0001
Kiri 20°	1011111	0010
Kiri 10°	1101111	0011
Center 0°	1110111	0100

Kanan 10°	1111011	0101
Kanan 20°	1111101	0110
Kanan 30°	1111110	0111

## **4.5. Driver Motor DC**

### **4.5.1. Tujuan**

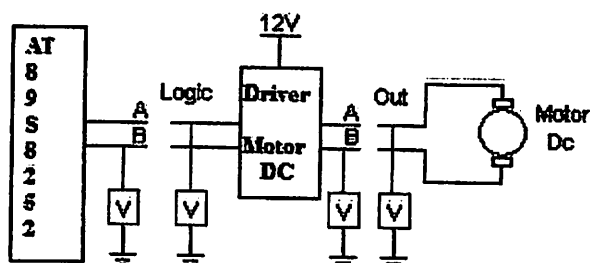
Pengujian ini bertujuan untuk mengetahui fungsi dari derive motor DC yang diputar kekanan dan sebaliknya. Diriver pertama difungdikan untuk memutar kekanan penggerak untuk mengangkat tv dari dalam meja. Driver kedua untuk memutar sudut televisi untuk memudahkan menonton.

### **4.5.2. Alat Yang Digunakan**

1. Multimeter SUNWA YX-360TRD
2. Rangkaian driver motor Dc yang akan diuji.
3. Catu Daya 12 Volt DC dan 5 Volt.
4. Motor Dc.

### **4.5.3. Langkah Pengujian**

3. Merangkai rangkaian driver motor Dc seperti pada gambar di bawah ini:



**Gambar 4 3. Pengujian Rangkaian Driver Motor Dc**

4. Menghubungkan rangkaian driver motor Dc dengan tegangan DC 12 Volt untuk catu daya pada Driver Motor untuk suplai tegangan 5 Volt untuk penyedia tegangan dan Vss.
5. Menghubungkan motor Dc dengan driver motor. Untuk pin control pengangkat TV pada P2.0 dan P2.1 dan pin control Pemutar sudut TV pada P2.2 dan P2.3 Mengamati hasil pengukuran tegangan output rangkaian driver motor Dc.
5. Melakukan pengukuran output dari rangkaian driver motor Dc.

#### 4.5.4. Hasil Pengujian

**Tabel 4. 4 Hasil Pengukuran Arah Putaran Motor DC**

logic		V logic		V out		Ket
A	B	A	B	A	B	
0	0	0,04	0,03	0,06	0,05	Stop
0	1	0,03	4,70	0,05	11,55	Kiri
1	0	4,98	0,03	11,75	0,05	Kanan
1	1	4,98	4,70	11,75	11,55	Break



## 4.6. Rangkaian Timer

### 4.6.1. Tujuan

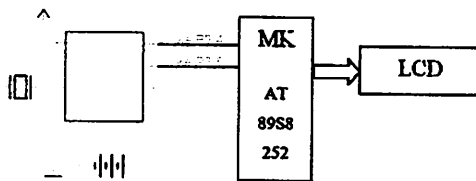
Tujuan dari pengujian timer ini adalah untuk menyesuaikan waktu yang ditampilkan dengan waktu yang sebenarnya

### 4.6.2. Alat Yang Digunakan

1. Rangkaian Timer
2. Minimum sistem
3. Rangkaian LCD
4. Power suplay 5 Volt

### 4.6.3. Langkah Pengujian

1. Hubungkan rangkaian seperti gambar dibawah ini



**Gambar 4. 3 uji rangkaian timer**

2. Jalankan program perhitungan waktu
3. Amati perubahan waktu setiap 5 menit.
4. Catat hasil pehitungan yang ditampilkan di LCD

### 4.6.4. Hasil Pengujian

Tabel 4. 5 pengujian rangkaian RTC

No	Jam Komputer		Jam Pengukuran	
	Jam	Menit	Jam	Menit
1	07	53	07	53
2	07	58	07	58
3	08	03	08	03
4	08	08	08	08
5	08	13	08	13

## 4.7. Display LCD

### 4.7.1. Tujuan

Tujuan dari pengujian ini adalah untuk menampilkan tulisan informasi profil yaitu nama dan nim dari penulis

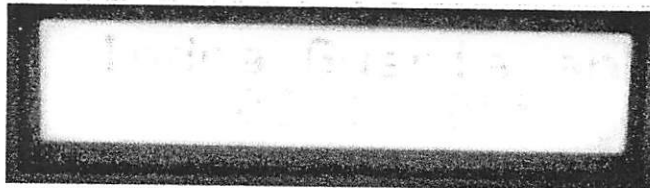
### 4.7.2. Alat Yang Digunakan

1. Minimum Sistem AT89S8252
2. Rangkaian LCD 3 Wire
3. Power Suplay +5 Vdc

### 4.7.3. Langkah Pengujian

1. Hubungkan rangkaian minimum sistem sengan rangkaian LCD
2. Jalankan Program LCD profil.c

### 4.7.4. Hasil Pengujian



Gambar 4. 4 Tmpilan profil LCD

Tabel 4. 6 Hasil tampilan LCD

Baris LCD	Ouput Tulisan
Pertama (0x80)	Indra Gusniawan
Kedua (0xC0)	0317055

## 4.8. Alarm

### 4.8.1. Tujuan

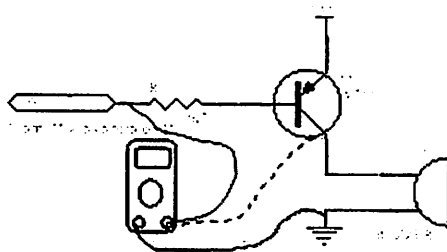
Tujuan dari pengujian ini adalah untuk mengetahui berfungsi apa tidaknya dari rangkaian alarm.

### 4.8.2. Alat Yang Digunakan

1. Rangkaian alarm
2. Multimeter Digital Sunwa
3. Power suplay + 5 VDC

### 4.8.3. Langkah Pengujian

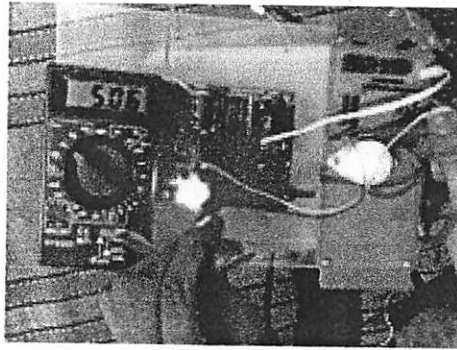
1. Hubungkan rangkaian dengan power suplay
2. Inputan alarm diberikan Logika 1 dan 0



Gambar 4. 5 Uji Rangkaian Buzzer

3. Amati hasil tegangan yang dioutputkan ke buzzer

#### 4.8.4. Hasil Pengujian



Gambar 4. 6 hasil pengujian rangkaian buzzer

Tabel 4. 7 Hasil pengujian rangkaian Buzer

Input Logic	Input tegangan	Out teg Buzzer	Reaksi Buzzer
0	0.142 V	4.524 V	Bunyi
1	4.652 V	0.223 V	Tidak Berbunyi

### 4.9. Interface Serial

#### 4.9.1. Tujuan

Fungsi dari pengujian ini adalah untuk mengetahui bahwa data dari komunikasi serial dapat disalurkan dengan baik oleh HP baik mengirim maupun menerima

#### 4.9.2. Alat Yang Digunakan

1. Rangkaian Interface Serial
2. Komputer P4 OS XP
3. Aplikasi Hyper Terminal

#### 4. HP siemens C55

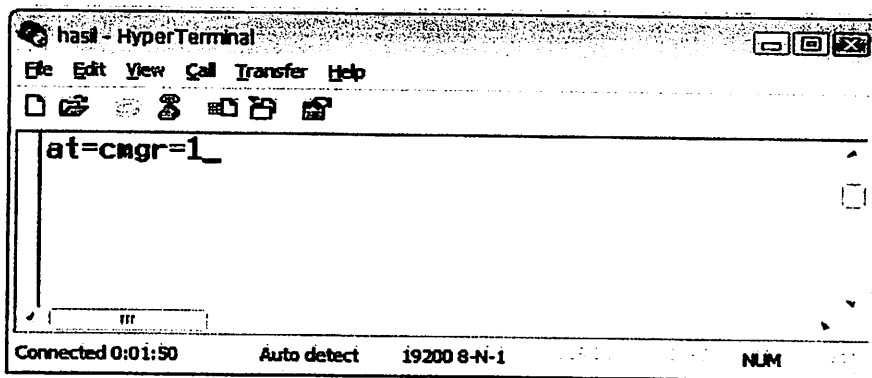
### 4.9.3. Langkah Pengujian

1. Hubungkan Rangkaian ke komputer melalui Port Com1 (DB9)
2. Jalankan komputer dan buka aplikasi Hyper Terminal
3. Seting komunikasi serial dengan baudrate 19200
4. Ketik perintah AT+CMGR=1

### 4.9.4. Hasil Pengujian

Tabel 4. 8 Pengujian komunikasi serial

Input dari Keyboard PC	Out HT
A	A
T	T
+	+
C	C
M	M
G	G
R	R
=	=
I	I



Gambar 4. 7 uji koneksi serial

## **Bab V**

### **Kesimpulan dan Saran**

#### **1. Kesimpulan**

1. Penggunaan remote dapat memudahkan pengguna untuk memutar sudut pandang televisi
2. Sudut dari televisi harus ditentukan terlebih dahulu  $10^{\circ}$  hingga  $30^{\circ}$  guna memudahkan pemrograman
3. Pada pengujian rangkaian jam menampilkan jam dan menit tidak terjadi error
4. Semakin besar LCD yang digunakan semakin mudah tulisannya untuk dilihat
5. Pengiriman informasi melalui sms dapat dipengaruhi oleh profaider penyedia layanan sms

#### **2. Saran**

Tujuan utama dari penulisan adalah cara membuat alat untuk mempermudah menonton tv dan dapat mengontrol jarak jauh, maka dari itu penulis menginginkan alat pengontrol interior lebih fleksibel dan sederhana dalam pemakaian. Saran-saran yang bisa digunakan untuk pengembangan alat ini secara lanjut antara lain:

1. Dalam perancangan mekanik harus diperhatikan dalam pemotongan bahan sehingga nilai toleransi ataupun kemiringan yang akan mengakibatkan gesekan dapat dikurangi.
2. Pada pengerjaan rangkaian elektronika sebaiknya mengurangi kabel penghubung dan memberikan konektor header sebagai titik pengukur.

3. Perancangan pembuatan software disarankan menggunakan bahasa c. karena lebih mudah dipahami.
4. Penggunaan remote yang lebih efisien sehingga tidak terkesan pemborosan.

## DAFTAR PUSTAKA

- Totok Budioko, Belajar dengan mudah dan cepat pemrograman bahasa C, Gaya Media, 2005
- Malvino, Prinsip-Prinsip Elektronika, Erlangga, 2001
- Widodo Budiharto dan Gamayel Rizal, 12 Proyek Mikrokontroler untuk Pemula, ElexMedia Koputindo, 2007
- 302 Rangkaian Elektronika, P. Pratomo, Elex Media Koputindo, 1993
- Data Sheet Book 1, Warsito S, Elex Media Koputindo, 1995
- Display dot matrik ([www.innovativeelectronics.com](http://www.innovativeelectronics.com))
- Send message siemens C35 ([www.pdu.edu](http://www.pdu.edu))
- Control motor dc ([www.indra-ea.blogspot.com](http://www.indra-ea.blogspot.com))
- Format pengiriman dan membaca sms  
(<http://www.globalkomputer.com/Bahasan/Komunikasi-Data/Topik/PDU/>)





INSTITUT TEKNOLOGI NASIONAL  
Jl. Bendungan Sigura-gura No. 2  
MALANG

### FORMULIR PERBAIKAN SKRIPSI

Nama Mahasiswa : INDRA GUSNIAWAN  
NIM : 03 17 055  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan Dan Pembuatan Interior Meja  
Televisi Menggunakan Remote Control  
Berdasarkan Mikrokontroler Atmel AT89S8252  
Hari/tanggal ujian Skripsi : Selasa 9 Februari 2010

No	Uraian	Paraf
1	Kata pengantar no 5 direvisi	
2	Tujuan direvisi	
3	Pada bab II gambar dan table diberi sumber pustaka	
4	Kesimpulan direvisi	
5	Gambar lengkap direvisi	

Disetujui oleh  
Penguji II

Ir. Th Mimien Mustikawati, MT.  
Nip. Y. 1030000352

Mengatahui,

Dosen Pembimbing I

Dosen Pembimbing II

( Irmalia Suryani F, ST, MT )  
NIP. 103 0000 365

( Ir. Eko Nurcahyo )  
NIP. 102 8700 172



INSTITUT TEKNOLOGI NASIONAL  
Jl. Bendungan Sigura-gura No. 2  
MALANG

### FORMULIR PERBAIKAN SKRIPSI

Nama Mahasiswa : INDRA GUSNIAWAN  
NIM : 03 17 055  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan Dan Pembuatan Interior Meja  
Televisi Menggunakan Remote Control  
Berbasis Mikrokontroler Atmel AT89S8252  
Hari/tanggal ujian Skripsi : Selasa 9 Februari 2010

No	Uraian	Paraf
1	Bersyarat, 2 minggu ujian lagi !!!!	

Disetujui oleh  
Penguji I

Ir. F Yudi Linpraptono, MT.  
Nip. Y. 1039500274

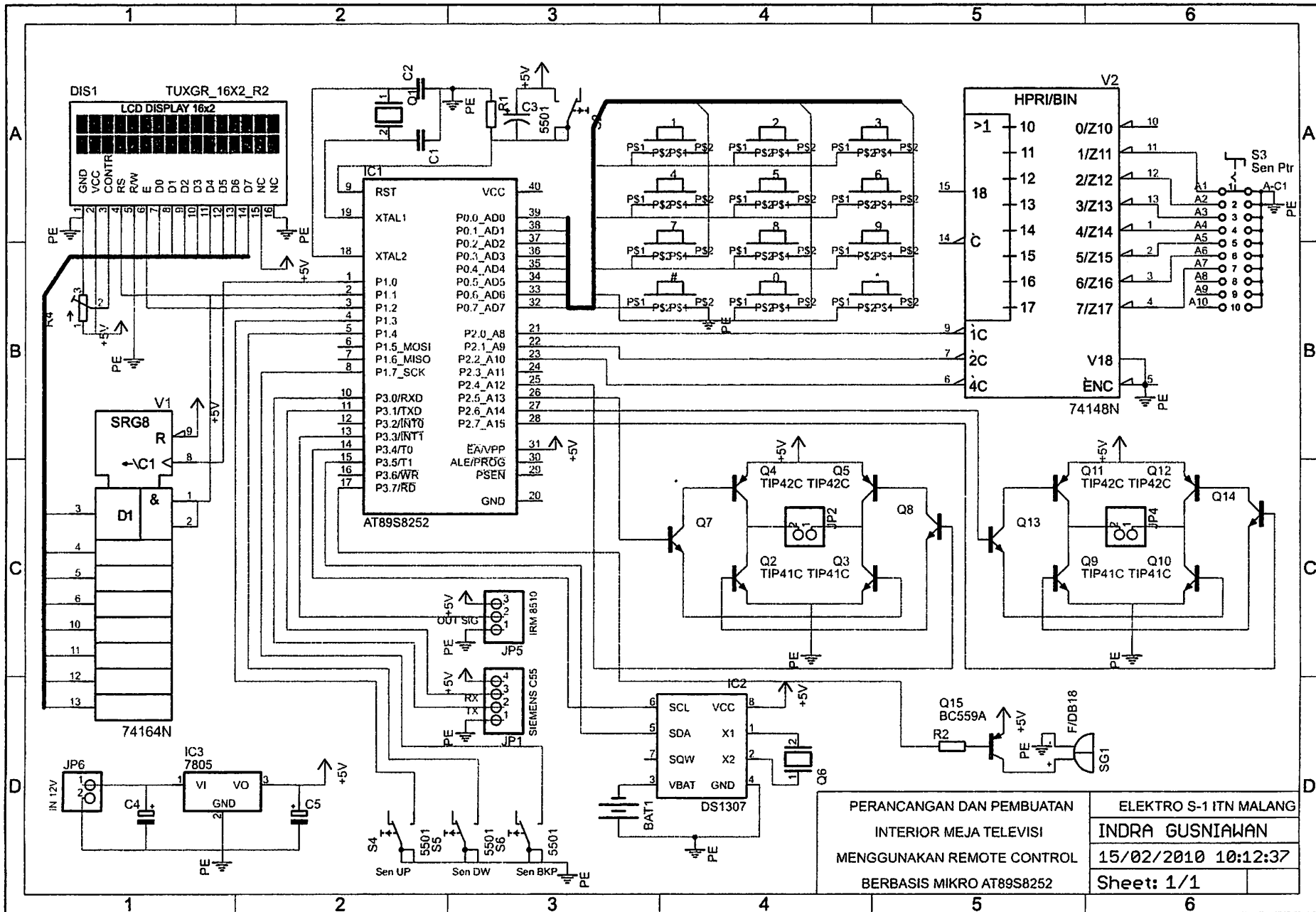
Mengatahui,

Dosen Pembimbing I

Dosen Pembimbing II

( Irmalia Suryani F. ST, MT )  
NIP. 103 0000 365

( Ir. Eko Nurcahyo )  
NIP. 102 8700 172



PERANCANGAN DAN PEMBUATAN INTERIOR MEJA TELEVISI MENGGUNAKAN REMOTE CONTROL BERBASIS MIKRO AT89S8252	ELEKTRO S-1 ITN MALANG
	INDRA GUSNIAWAN
	15/02/2010 10:12:37
	Sheet: 1/1

```

1: #include <8051.h>
2:
3: void key();
4: void step1();
5: void step2();
6: void step3();
7: void stop();
8: void kiri();
9: void kanan();
10: void putar();
11: void naik();
12: void turun();
13: void backl();
14: void mati();
15: void star();
16: void cekpas();
17: void input();
18:
19: void kirim (char dataku);
20: void init();
21: void ngirim (char *teks);
22: void init_serial();
23:
24: void tvon();
25: void tvoff();
26: void alarmon();
27: void alarmoff();
28:
29: void Init_interupt (void);
30: void ExternalInterrupt_1 (void) ;
31: void remo (void);
32: void scan ();
33:
34: void profil();
35:
36: void cmd(char c, char dat);
37: void tulis (char a, char* dat);
38: void tulispas (char a, char* dat);
39:
40: unsigned char pil, kunci,pass,fi,sen,in,emot,awal;
41: unsigned int Dataremote ;
42: //unsigned char pil ;
43:
44: /*----- delay -----*/
45: void delay (long tunggu)
46:     { while (tunggu--);}
47: /*=====*/
48: void Init_interupt(void)
49: {
50:     TCON = 0;

```

```

51:  PX1    = 1;    // Priority interupt
52:  IT1    = 1;    // set ext. int.1 to down
53:  TMOD = 0x29; // Timer 0 sebagai 16 bit counter
54:  EX1    = 1;    // Enable External interupt 1
55:  EA     = 1;    // Aktipkan Sistem interupsi MCS51
56: )
57:
58: /*****\
59:  Rutin Layanan Interupsi untuk External Timer 1, Bila ada
60:  sinyal IR remote control maka akan masuk ke rutin ini.
61: \*****/
62: void ExternalInterrupt_1(void)
63: interrupt IE1_VECTOR
64: {
65:  TR0 = 0;
66:  awal++;
67:  if (awal == 1) {}
68:  else
69:  {
70:    if (awal == 2)
71:    {
72:      if ((TH0 == 0x0B) | (TH0 == 0x0C)) {} // Start bit cocok
73:      else
74:      { // start bit tidak cocok
75:        awal = 0;
76:        Dataremote = 0;
77:      }
78:    }
79:    else // berarti awal > 1 yg menandakan start bit sdh cocok
80:    {
81:      if ((TH0 == 0x04) | (TH0 == 0x05))
82:      {
83:        Dataremote = Dataremote >> 1;
84:        Dataremote = Dataremote & 0x07FF;
85:      }
86:      if ((TH0 == 0x06) | (TH0 == 0x07))
87:      {
88:        Dataremote = Dataremote >> 1;
89:        Dataremote = Dataremote | 0x0800;
90:      }
91:    }
92:  }
93:  if (awal == 14)
94:  {
95:    Dataremote = Dataremote >> 1;
96:    emot = Dataremote & 0x00FF;
97:    scan();
98:    awal = 0;
99:  }
100: TH0 = 0;

```

```

101:  TL0  = 0;
102:  TR0  = 1;
103: }
104:
105:
106: /*****\
107: *
108: \*****/
109: void remo (void)
110: {
111:     awal = 0;
112:     Dataremote = 0;
113:     TH0  = 0;
114:     TL0  = 0;
115:
116:     Init_interupt ();
117: }
118: void scan ()
119:     { emot= 0 ;
120:         if (emot ==0x80 )           //angka 1
121:             {
122:                 pil =1;
123:             }
124:         else if (emot == 0x81 )     // angka 2
125:             {
126:                 pil = 2;
127:             }
128:         else if (emot == 0x82 )     // angka 3
129:             {
130:                 pil = 3;
131:             }
132:         else if (emot == 0x83 )     // angka 4
133:             {
134:                 pil =4 ;
135:             }
136:         else if (emot == 0x84 )     // angka 5
137:             {
138:                 pil = 5;
139:             }
140:         else if (emot == 0x85 )     // angka 6
141:             {
142:                 pil = 6;
143:             }
144:         else if (emot == 0x86 )     // angka 7
145:             {
146:                 pil = 7;
147:             }
148:         else if (emot == 0x87 )     // angka 8
149:             {
150:                 pil = 8;

```

```

151:         }
152:     else if (emot == 0x88 )           // angka 9
153:     {
154:         pil = 9;
155:     }
156:     else if (emot == 0x89 )           // angka 0
157:     {
158:         P2 =0x00;
159:     }
160:     else if (emot == 0x95 )           // power
161:     {
162:         pil = 14;
163:     }
164:
165:     else if (emot == 0xA6 )           // clr
166:     {
167:         pil = 13;
168:     }
169:     else if (emot == 0x92 )           // vol +
170:     {
171:         P2 =0xf1;
172:     }
173:     else if (emot == 0x93 )           // vol -
174:     {
175:         P2 =0xf0;
176:     }
177:     else if (emot == 0x90 )           // prog +
178:     {
179:         P2 =0xf0;
180:         Tulis_LCD(0x80," TV Berputar ");
181:         Tulis_LCD(0xC0,"ke kanan ==> bos");
182:     }
183:     else if (emot == 0x91 )           // prog -
184:     {
185:         P2 =0xf0;
186:         Tulis_LCD(0x80," TV Berputar ");
187:         Tulis_LCD(0xC0," <== ke kiri bos");
188:     }
189:     else if (emot == 0x9D )           // -/--
190:     {
191:         P2 =0xf2;
192:     }
193: }
194: /*----- lcd init -----*/
195: void cmd (char c, char dat)
196: {
197:     char i;
198:     for(i=0;i<8;i++)
199:     {     P1_1 = (dat & 0x01);
200:         dat >>=1;

```

```

201:         P1_0 = 0; P1_0 = 1;
202:         delay (1);
203:     }
204:     P1_2 = 1;
205:     P1_1 = c;
206:     P1_2 = 0;
207: }
208: void tulis (char a, char* dat)
209: {
210:     char i = 0; cmd(0,a);
211:     while(dat[i] != 0)
212:     {
213:         cmd(1,dat[i]); i++;
214:     }
215: }
216:
217: void tulispas (char a, char* dat)
218: {
219:     char b;
220:
221:     char i = 0; b=0xc5+a;cmd(0,b);
222:     while(dat[i] != 0)
223:     {
224:         cmd(1,dat[i]); i++;
225:     }
226: }
227: void profil()
228: {
229:     cmd(0,0x3F); cmd(0,0x0D); cmd(0,0x06);
230:     cmd(0,0x01); cmd(0,0x0C);
231:     tulis (0x80," Indra Gusniawan");
232:     tulis (0xc0," 03 17 055 ");
233:     delay (10000);
234: }
235:
236: /*----- serial hp -----*/
237: void init()
238: {
239:     TMOD = 0x20;
240:     PCON = 0x80;
241:     SM0 = 0;
242:     SM1 = 1 ;
243:     REN = 1 ;
244:     TH1 = 0xFD ;
245:     TL1 = 0xFD ;
246:     TR1 = 1;
247: }
248: void kirim (char dataku)
249: {
250:     SBUF = dataku ;

```



```

51:  while (! TI)
52:      {;}
53:  TI = 1;
54:  }
55:
56:  void ngirim (char *teks)
57:  {
58:      char i =0 ;
59:      while (teks[i] !=0)
60:          { delay(200);
61:            kirim (teks[i]) ;
62:              i++;
63:          }
64:
65:  }
66:
67:  void tvon()
68:  {
69:      char *teks ={"at+cmgs=90"};
70:      char *nosms={"0691261801000011000D91265832840366F80000FF"};
71:      char
72:      *hidup={"4F542B089926D7E120E17B3E0729926E7A599E7ECB41CDB23A0CA296D9657B7A9E5608F3A0A49B2C0F83
73: 8EF5B93B1DBE87DD8A22BBBCA6CBDF0692B064A529DA066981D769F01"};
74:      init () ;
75:      ngirim (teks);
76:      delay (2000);
77:      kirim(13);
78:      ngirim(nosms);
79:      ngirim (hidup);
80:      kirim (26);
81:  }
82:
83:  void tvoff()
84:  {
85:      char *teks ={"at+cmgs=90"};
86:      char *nosms={"0691261801000011000D91265832840366F80000FF"};
87:      char
88:      *mati={"4E542BA819A6A741C2F77C0E5224DDF4B23CFD96839A657518442DB3CBF6F43CAD10E6414937591E061DE
89: B73773A7C0FBB154576794D97BF41D3560C94A43A41CD303BEC3E03"};
90:      init () ;
91:      ngirim (teks);
92:      delay (2000);
93:      kirim(13);
94:      ngirim(nosms);
95:      ngirim (mati);
96:      kirim (26);
97:  }
98:  void alarmon()
99:  {
00:      char *teks ={"at+cmgs=90"};

```

```

01: char *nosms={"0691261801000011000D91265832840366F80000FF"};
02: char
03: *alarm={"55417658DE0651AD20749A5C878384EFF91CA448BAE96579FA2D0735CBEA30885A6697EDE9795A21CC83
04: 926EB23COC3AD6E7EE74F81E762B8AECF29A2E7F83A6AD18284975829A6176D87D06"};
05: init () ;
06: ngirim (teks);
07: P3_7 = 0;
08: delay (2000);
09: kirim(13);
10: ngirim(nosms);
11: ngirim (alarm);
12: kirim (26);
13: }
14: void alarmof()
15: {
16: char *teks={"at+cmgs=90"};
17: char *nosms={"0691261801000011000D91265832840366F80000FF"};
18: char
19: *alarmmati={"54417658DE0651ADA076989E0609DFF339489174D3CBF2F45B0E6A96D56110B5CC2EDBD3F3B44298
20: 0725DD64791874ACCFDDE9F03DEC5614D9E5355DFE064D5B315092EA0435C3ECB0FB0C"};
21: init () ;
22: ngirim (teks);
23: P3_7 = 1;
24: delay (2000);
25: kirim(13);
26: ngirim(nosms);
27: ngirim (alarmmati);
28: kirim (26);
29: }
30:
31: /*=====*/
32: /*----- keypad 3x4 -----*/
33: void key ()
34: {
35: PO = 0xff;
36: if ((PO_4 == 1)&&(PO_5 == 1)&&(PO_6 == 1))
37: {
38: pil = 77;
39: }
40:
41: PO = 0xFD; // 7,8,9
42: if ( PO ==0xbd ) // 7
43: {delay(15000);
44: pil= 7;}
45: if ( PO ==0xdD ) // 8
46: { delay(15000);
47: pil = 8;}
48: if ( PO ==0xeD ) // 9
49: { delay(15000);
50: pil = 9;}

```

```

351:     P0 = 0xFB;                //4,5,6
352:         if ( P0 ==0xB )      // 4
353:             { delay(15000);
354:               pil =4;}
355:         if ( P0 ==0xD )      // 5
356:             { delay(15000);
357:               pil =5;}
358:         if ( P0 ==0xE )      // 6
359:             { delay(15000);
360:               pil =6; }
361:     P0 = 0xF7;                // 1,2,3
362:         if ( P0 ==0xB7 )     // 1
363:             {delay(15000);
364:               if (P3_6 =1)
365:                 { pil =1;}}
366:         if ( P0 ==0xD7 )     // 2
367:             { delay(15000);
368:               pil =2;}
369:         if ( P0 ==0xE7 )     // 3
370:             { delay(15000);
371:               pil =3;}
372:     P0 = 0xFE;
373:         if ( P0 ==0xBE )     // # kiri
374:             { //delay(25000);
375:               pil =13;}
376:         if ( P0 ==0xDE )     // 0
377:             { delay(15000);
378:               pil =0;}
379:         if ( P0 ==0xEE )     // * kanan
380:             { delay(15000);
381:               pil =14;}
382:     //P0 =0xFF ;
383:         if (P0 == 0xEF)
384:             {P2 = 0xAA;}
385:     }
386:
387: void kanan()
388: { tulis(0xc3," kanan ");
389:   P2_6 = 0;
390:   P2_7 = 1;
391: }
392: void kiri ()
393: { tulis(0xc3," kiri ");
394:   P2_6 = 1;
395:   P2_7 = 0;
396: }
397: void stop()
398: {   tulis(0xc3," stop ");
399:   P2_6 = 0;
400:   P2_7 = 0;

```

```

401:  }
402:
403:
404: void back1()
405: {
406:     while (1)
407:     {
408:         sen = (P2 & 7);
409:
410:         while (sen >= pil )
411:             { sen = (P2 & 7);
412:               if ( sen == 4)
413:                   {
414:                       tulis(0xc0," pass ah ");
415:                       delay (10000);
416:                       stop();
417:                       turun();           tvoff();           star();
418:                   }
419:               kiri();
420:             }
421:
422:         while ( sen <= pil)
423:             { sen = (P2 & 7);
424:               if ( sen == 4)
425:                   { tulis(0xc7," pass uh ");
426:                     delay (10000);
427:                     stop();turun();tvoff(); star();
428:                   }
429:               kanan();
430:             }
431:     }
432: }
433:
434: void putar()
435: {
436:     while (1)
437:     {
438:         sen = (P2 & 7);
439:         P0 =in;
440:         while (sen >= pil )
441:             {
442:                 sen = (P2 & 7);
443:
444:                 if ( sen == in)
445:                     { tulis(0xc0," pass deh ");
446:                       delay (10000);
447:                       stop();
448:                       step3();
449:                     }
450:                 kiri();

```

```

451:         }
452:     while ( sen <= pil)
453:     {   sen = (P2 & 7);
454:         if ( sen == in)
455:             { tulis(0xc7," pass deh ");
456:                 delay (10000);
457:                 stop(); step3();
458:             }
459:         kanan();
460:     }
461: }}
462:
463: void mati()
464: {
465:     while (1)
466:     {
467:         key();
468:         if (pil == 13)
469:         {
470:             pil = 4;
471:             in = 4;
472:             tulis(0x80," god bye bro ");
473:             back1();
474:
475:             delay (10000);
476:             step1 ();           }
477:         if (pil == 14)
478:         {
479:             tulis(0x80," gak jadi mati ");
480:             delay (10000);
481:             step3();
482:         }
483:     }
484: }
485: /*----- step 3 ----- on pakai -----*/
486:
487: /*=====*/
488: void step3 ()
489: {
490:     cmd (0,0x01);
491:     tulis (0xc0," tekan tombol");
492:     pil = 77;
493:     while (1)
494:     {
495:
496:         sen = (P2 & 7);
497:         key();
498:         if (P0_3 =0)
499:         {
500:             alrmon();

```

```
501:         }
502:     if (pil == 1)
503:     {
504:         pil = 0;
505:         tulis (0x80," putar kiri 30 ");
506:         in = 0;
507:
508:         putar();
509:     }
510:     if (pil == 2)
511:     {pil = 1;
512:         tulis (0x80," putar kiri 20 ");
513:         in = 1;
514:         putar();
515:     }
516:     if (pil == 3)
517:     {pil = 2;
518:         tulis (0x80," putar kiri 10 ");
519:         in = 2;
520:         putar();
521:     }
522:     if (pil == 5)
523:     {
524:         pil = 3;
525:         tulis (0x80," putar tengah 0 ");
526:         in = 3;
527:         putar();
528:     }
529:     if (pil == 7)
530:     {
531:         pil = 4;
532:         tulis (0x80," putar kanan 10 ");
533:         in = 4;
534:         putar();
535:     }
536:     if (pil == 8)
537:     {
538:         pil = 5;
539:         tulis (0x80," putar kanan 20 ");
540:         in = 5;
541:         putar();
542:     }
543:     if (pil == 9)
544:     {
545:         pil = 6;
546:
547:         tulis (0x80," putar kanan 30 ");
548:         in = 6;
549:         putar();
550:     }
```

```

551:         if (pil == 4)                // <<<
552:         {
553:             tulis (0x80," <<<<< kiri ");
554:
555:             if (sen == 0)
556:                 {stop();}
557:             kiri();
558:             delay(10000);
559:             stop();
560:         }
561:         if (pil == 6)                // >>>>
562:         {
563:             tulis (0x80," kanan >>>>> ");
564:             if (sen == 6)
565:                 {stop();}
566:             kanan();
567:             delay(10000);
568:             stop();
569:         }
570:         if (pil == 14)
571:         {
572:             tulis (0x80," matikan system ");
573:             delay(10000);
574:             mati();
575:         }
576:     }
577: }
578: /*----- cek pas -----*/
579: void cekpas()
580: {
581:     while (1)
582:     {
583:         if ( pass == kunci )        // pass ok
584:         { tulis(0x80," pass: ok punya ");
585:           delay (50000);
586:           naik();
587:           tvon();
588:           step3();
589:         }
590:         else                          // pass salah
591:         { tulis(0x80," pass: salah bos");
592:           delay (100000);
593:           step2();
594:         }
595:     }
596: }
597: /*----- step2 -----*/
598: void step2()
599: { kunci = 5;
600:   // char fi;

```

```

601:     fi =0xc4;
602:     cmd(0,0x01);
603:     tulis(0x80," Password ");
604:     pil = 77;
605:     tulis(0xc0,"");
606:     while (1)
607:     {
608:         key ();
609:         awal:
610:         if ((pil != 77)&&(pil !=13)&&(pil !=14))
611:         {
612:             pass = pil ;
613:             tulis( fi ,"*");
614:             fi = fi + 1;
615:         }
616:
617:         if (pil == 13)           // enter cek pas
618:         {
619:             cekpas();
620:         }
621:
622:     }
623:
624: }
625: /*----- proses step1 -----*/
626: void step1()
627: {
628:     if (P0_3 =0)
629:     {
630:         alarmon();
631:     }
632:     if (pil==1)
633:     {
634:         cmd(0,0x01);
635:         tulis(0x80," masuk ke step2");
636:         tulis(0xc0,"");
637:         pil = 77;
638:
639:         step2 ();
640:
641:     }
642:     if (pil ==2)
643:     {
644:         cmd(0,0x01);
645:         tulis(0x80," sekarang jam");
646:         tulis(0xc0,"");
647:
648:     }
649:     if (pil ==3)
650:     {

```



```

651:         cmd(0,0x01);
652:         tulis(0x80," no hp tujuan ");
653:         tulis(0xc0," 085 234 830 668");
654:     }
655:
656: }
657: /*=====*/
658:
659: /*=====*/
660:
661: void input ()
662: {
663:     key();
664: }
665:
666:
667: /*=====
668: ----- motor control -----*/
669: void naik()
670: { while (1)
671: { tulis(0xc3," naik ");
672:     P2_4 =0;
673:     P2_5 =1;
674:     if (P1_4 == 0)
675:     {
676:         P2_4 = 0;
677:         P2_5 = 0;
678:         step3();
679:     }
680: }}
681:
682: void turun()
683: {
684:     while(1)
685:     { tulis(0xc3," turun ");
686:     P2_4 =1;
687:     P2_5 =0;
688:     if (P1_5 ==0)
689:     {
690:         P2_4 =0;
691:         P2_5 =0;
692:         star();
693:     }
694: }}
695: /*=====*/
696: void star()
697: {
698:     profil();
699:     while (1)
700:     {

```

```
701:     key ();
702:     //     P2 =pil;
703:     step1();
704:     if (P0_3 =0)
705:     {
706:         alrmon();
707:     }
708: }
709: }
710:
711: /*=====*/
712: /*----- main program -----*/
713: void main()
714: {
715:     P2 =0x07 ;
716:     while(1)
717:     {star();}
718: }
719:
720:
721:
```

## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery From Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

## Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. 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 downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 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 interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



## 8-bit Microcontroller with 8K Bytes Flash

**AT89S8252**

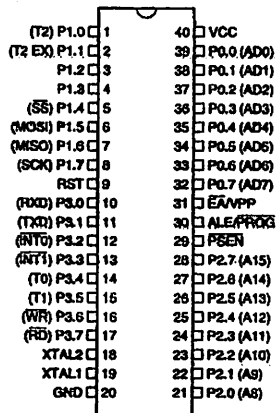
Rev. 0401E-02/00



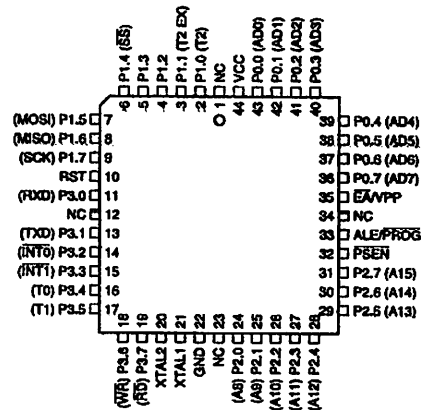


## Pin Configurations

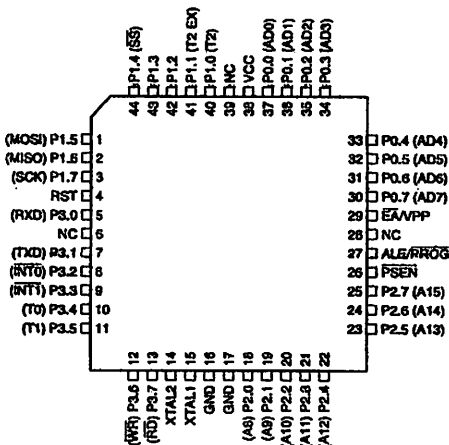
PDIP



PLCC



PQFP/TQFP



## Pin Description

### VCC

Supply voltage.

### GND

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

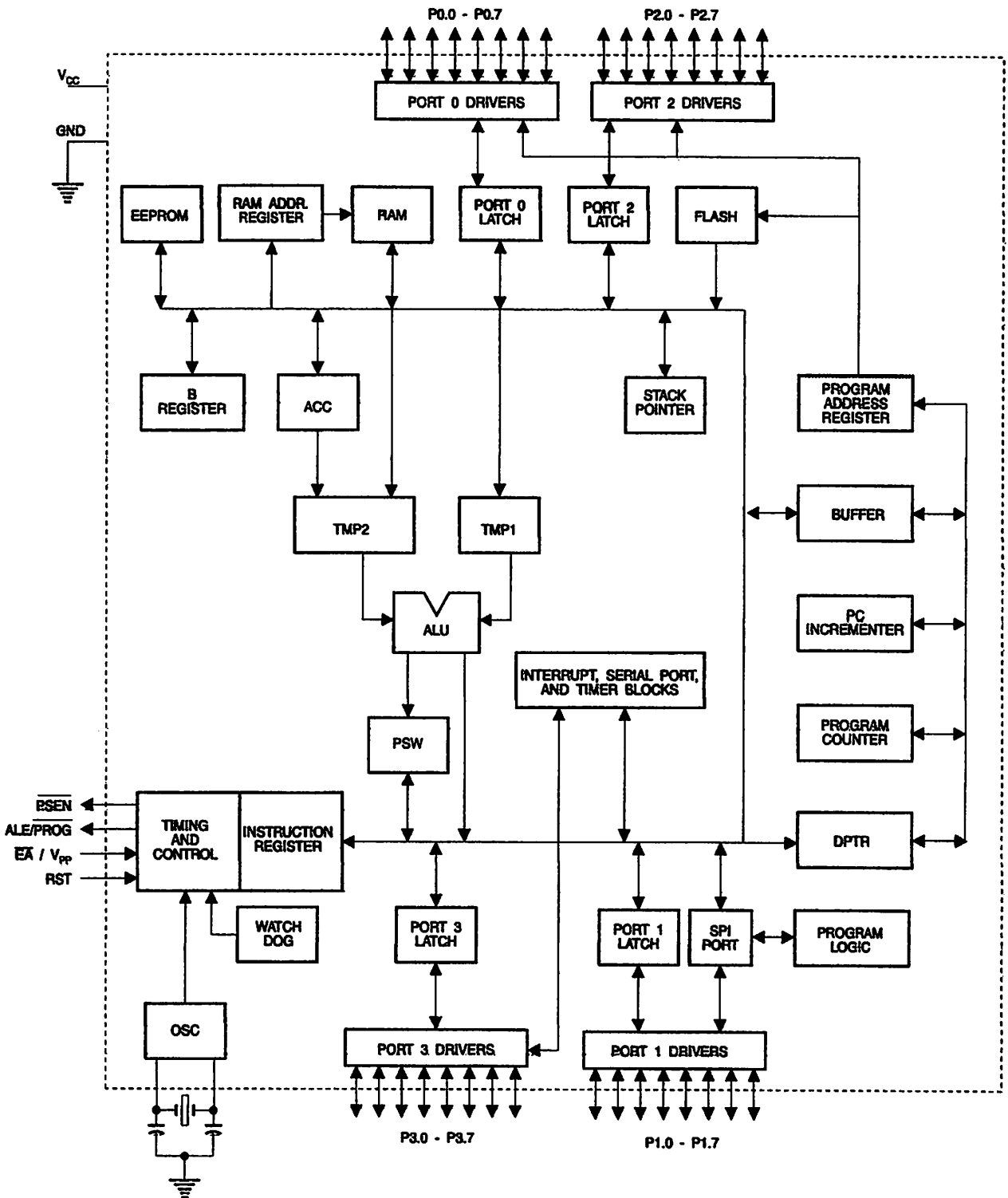
program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. 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 pullups 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 pullups.

## Block Diagram





Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

## Pin Description

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

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

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. 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 pullups 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 pullups.

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 pullups 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 pullups. 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 pullups 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 pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

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

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)

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

### ALE/ $\overline{PROG}$

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

When the AT89S8252 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.

### $\overline{EA}/VPP$

External Access Enable.  $\overline{EA}$  must be strapped to GND in order to enable the device to fetch code from external pro-

# AT89S8252

gram 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 when 12-volt programming is selected.

## XTAL1

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

## XTAL2

Output from the inverting oscillator amplifier.

**Table 1. AT89S8252 SFR Map and Reset Values**

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H





## 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.

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.

**Timer 2 Registers Control and status bits** are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

**Table 2. T2CON—Timer/Counter 2 Control Register**

T2CON Address = 0C8H								Reset Value = 0000 000B
Bit Addressable								
	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
Bit	7	6	5	4	3	2	1	0
Symbol	Function							
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.							
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).							
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.							
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.							
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.							
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.							
C/T2	Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered).							
CP/RL2	Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.							



**Watchdog and Memory Control Register** The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used

to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

**Table 3. WMCON—Watchdog and Memory Control Register**

WMCON Address = 96H					Reset Value = 0000 0010B			
	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1.
WDTRST RDY/BSY	Watchdog Timer Reset and EEPROM Ready/Busy Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/BSY flag in a Read-Only mode during EEPROM write. RDY/BSY = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/BSY bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

**SPI Registers** Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

**Interrupt Registers** The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

**Dual Data Pointer Registers** To facilitate accessing both internal EEPROM 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 WMCON 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 reset under software control and is not affected by RESET.





**Table 4. SPCR—SPI Control Register**

SPCR Address = D5H					Reset Value = 0000 01XXB			
Bit	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
7	6	5	4	3	2	1	0	

Symbol	Function
SPIE	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
SPE	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$ , MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
DORD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
MSTR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
CPOL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.
CPHA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.
SPR0 SPR1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{osc}$ , is as follows: SPR1SPR0 SCK = $F_{osc}$ divided by 0 0 4 0 1 16 1 0 64 1 1 128

**Table 5. SPSR – SPI Status Register**

SPSR Address = AAH					Reset Value = 00XX XXXXB			
Bit	SPIF	WCOL	–	–	–	–	–	–
7	6	5	4	3	2	1	0	

Symbol	Function
SPIF	SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register.
WCOL	Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register.

**Table 6. SPDR – SPI Data Register**

SPDR Address = 86H					Reset Value = unchanged			
Bit	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
7	6	5	4	3	2	1	0	

## Data Memory – EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

## Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the

actual timer periods (at  $V_{CC} = 5V$ ) are within  $\pm 30\%$  of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			Period (nominal)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

## Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which



the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

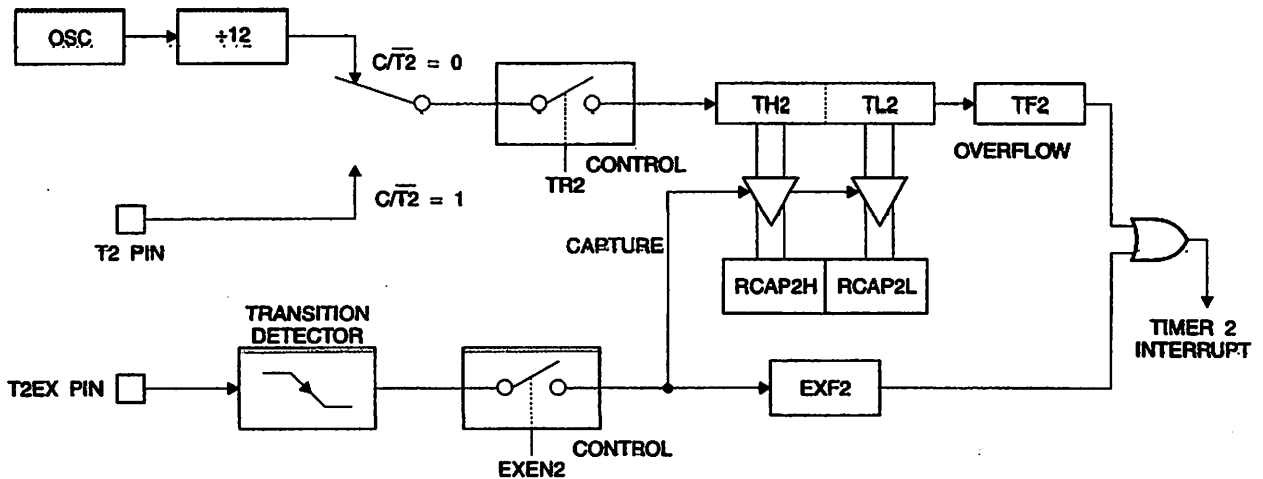
**Table 8. Timer 2 Operating Modes**

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

### Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

**Figure 1. Timer 2 in Capture Mode**



## Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or

by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 in Auto Reload Mode (DCEN = 0)

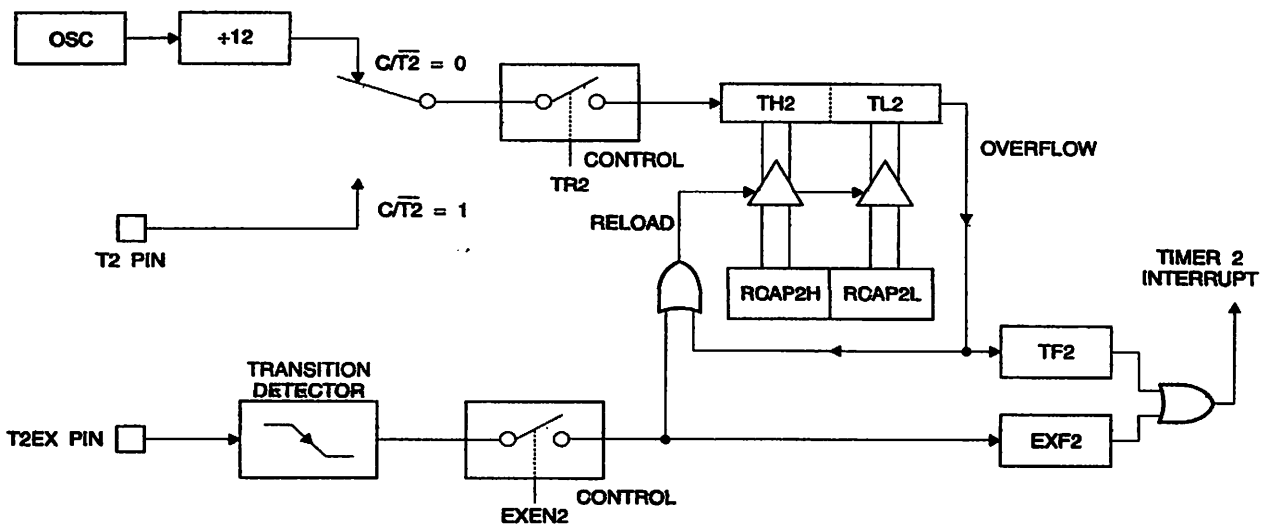
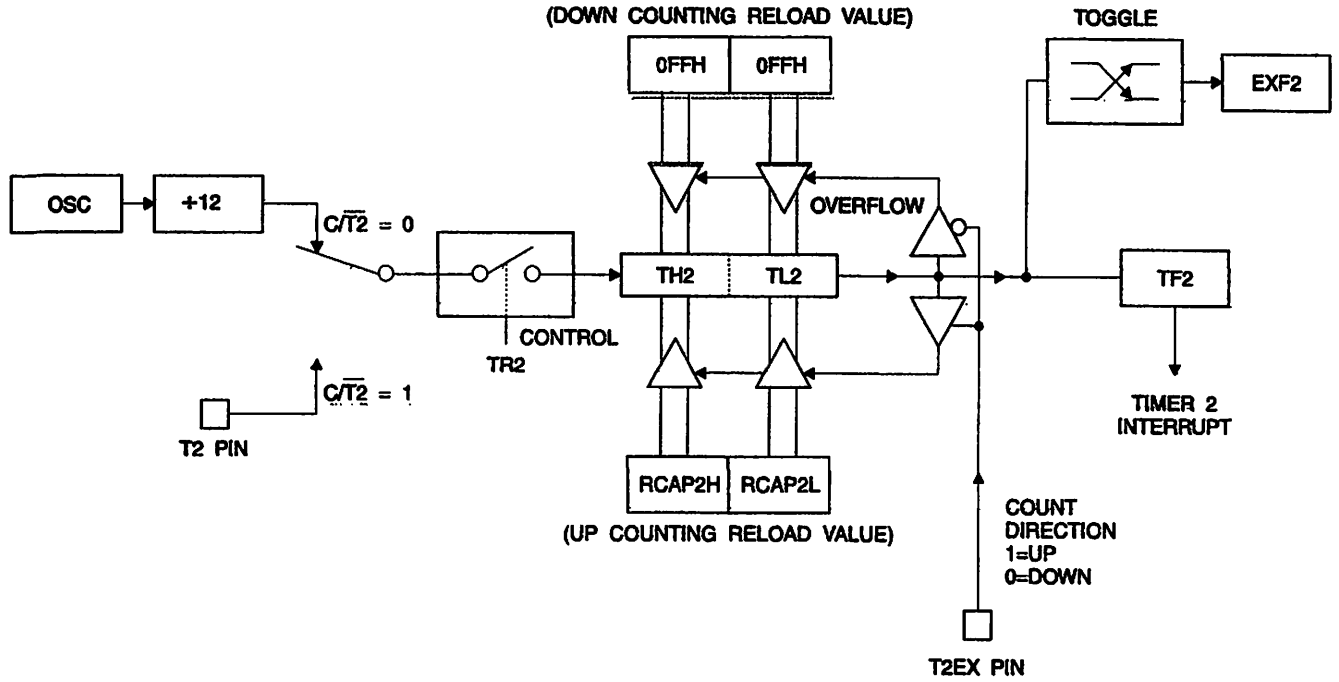


Table 9. T2MOD – Timer 2 Mode Control Register

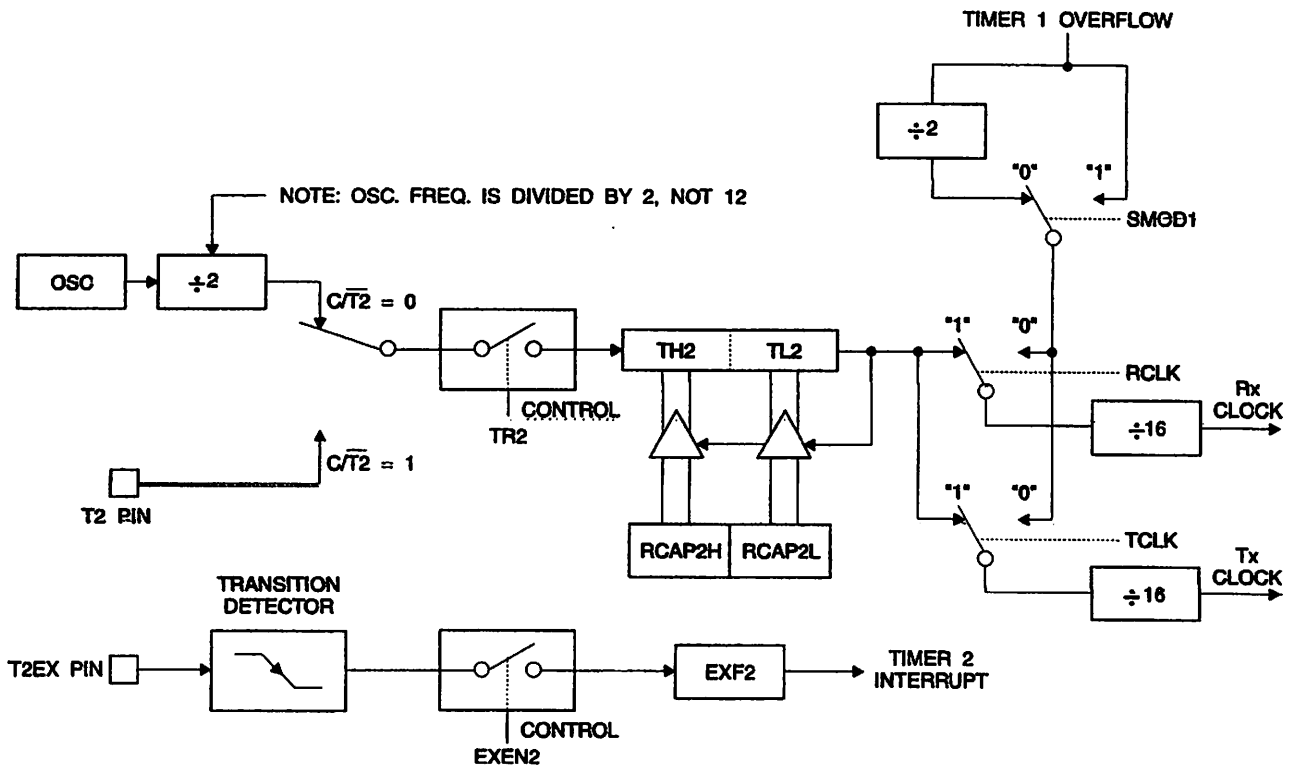
T2MOD Address = 0C9H							Reset Value = XXXX XX00B	
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN

Symbol	Function
–	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

**Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)**



**Figure 4. Timer 2 in Baud Rate Generator Mode**



## Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer

2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

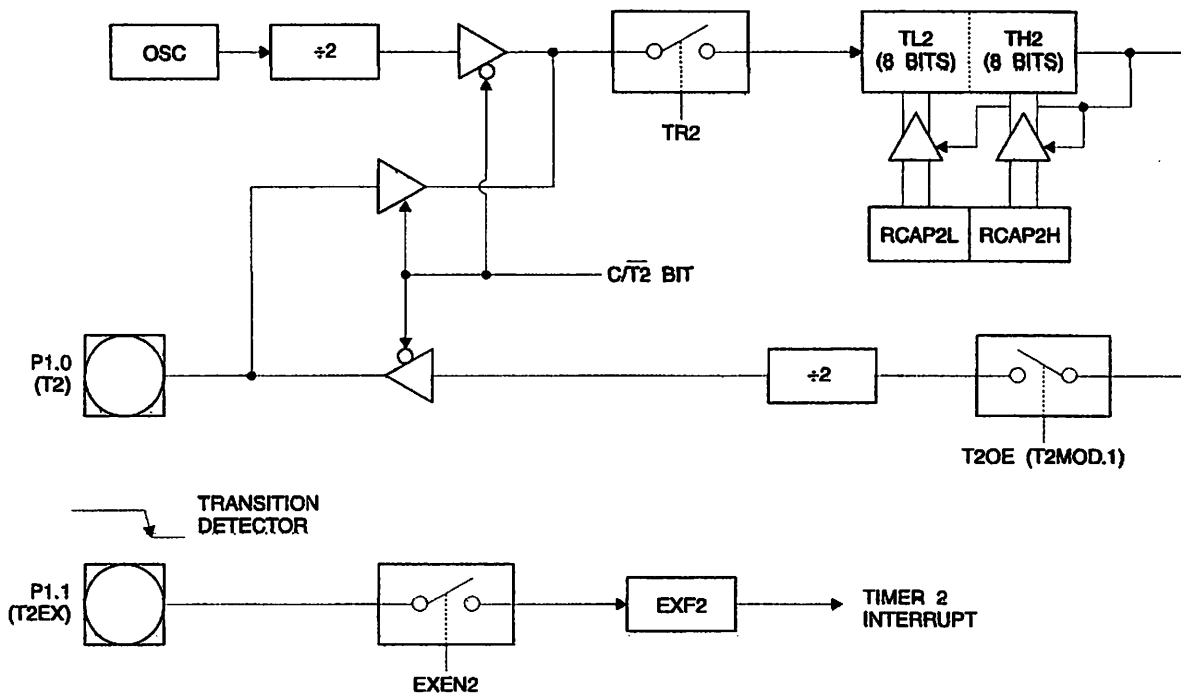
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

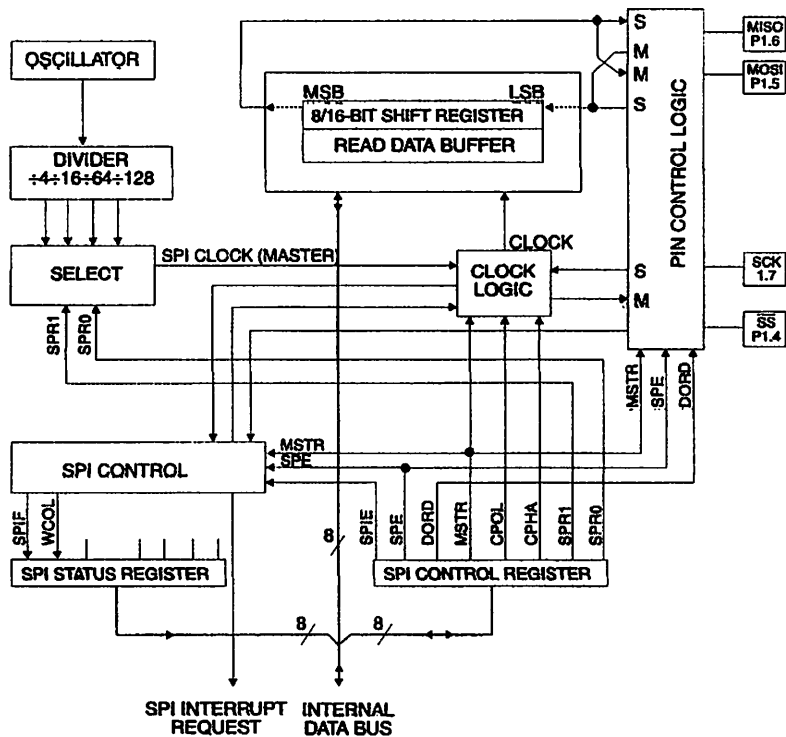
$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

**Figure 5. Timer 2 in Clock-out Mode**



**Figure 6. SPI Block Diagram**





## UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

## Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag

- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input,  $\overline{SS}/P1.4$ , is set low to select an individual SPI device as a slave. When  $\overline{SS}/P1.4$  is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 8 and Figure 9.

Figure 7. SPI Master-slave Interconnection

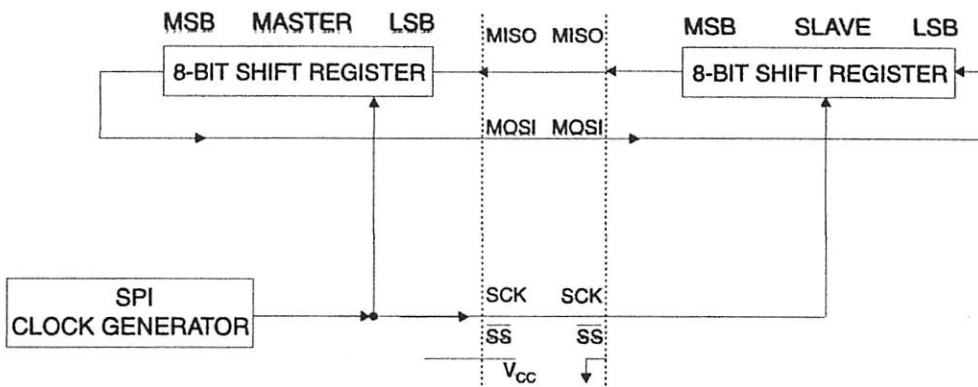
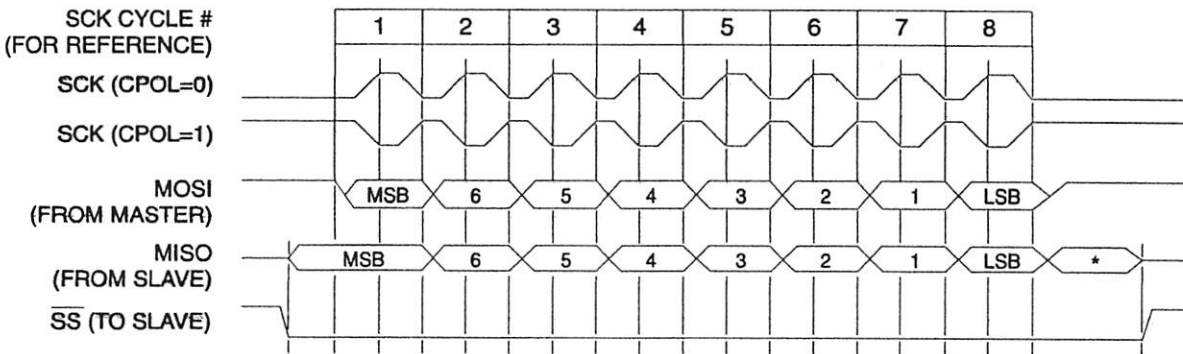


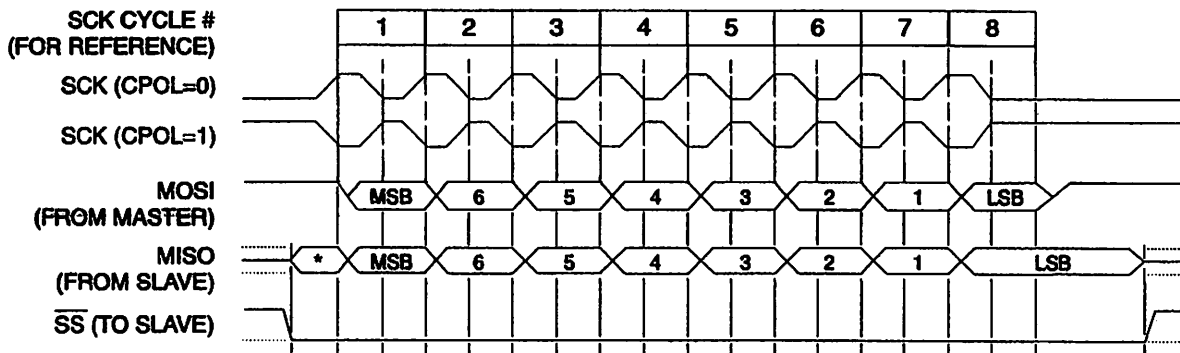
Figure 8. SPI transfer Format with CPHA = 0



\*Not defined but normally MSB of character just received



Figure 9. SPI Transfer Format with CPHA = 1



\*Not defined but normally LSB of previously transmitted character

## Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

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

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

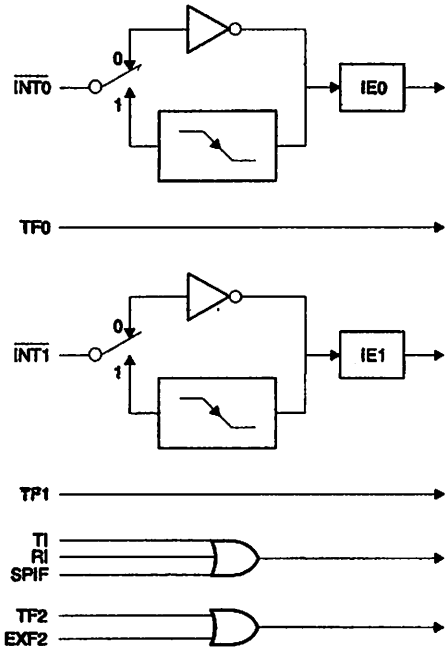
Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

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. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

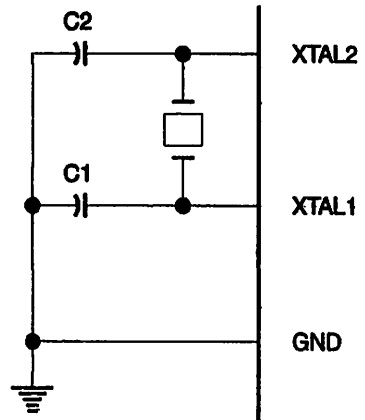
Table 10. Interrupt Enable (IE) Register

(MSB)(LSB)							
EA	-	ET2	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.					
ET2	IE.5	Timer 2 interrupt enable bit.					
ES	IE.4	SPI and UART 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 unimplemented bits, because they may be used in future AT89 products.							

**Figure 10. Interrupt Sources**

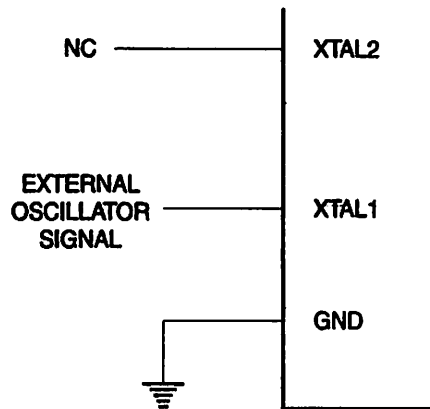


**Figure 11. Oscillator Connections**



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

**Figure 12. External Clock Drive Configuration**



## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. 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 12. 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.



## 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 functions 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.

## 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

## 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 can be initiated either by a hardware reset or by an enabled external interrupt. 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.

To exit power-down via an interrupt, the external interrupt must be enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

## Program Memory Lock Bits

The AT89S8252 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.

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.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

## Lock Bit Protection Modes<sup>(1)(2)</sup>

Program Lock Bits				Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
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 (parallel or serial mode) is disabled.
3	P	P	U	Same as Mode 2, but parallel or serial verify are also disabled.
4	P	P	P	Same as Mode 3, but external execution is also disabled.

Notes: 1. U = Unprogrammed  
2. P = Programmed

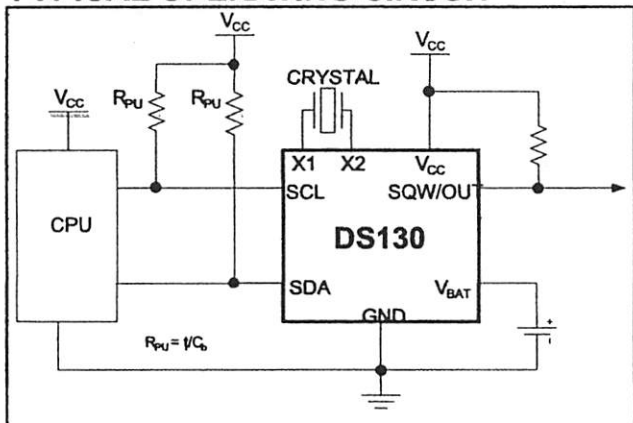
# DS1307

## 64 x 8, Serial, I<sup>2</sup>C Real-Time Clock

### GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I<sup>2</sup>C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

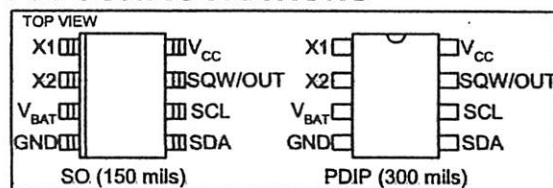
### TYPICAL OPERATING CIRCUIT



### FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I<sup>2</sup>C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

### PIN CONFIGURATIONS



### ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.

\*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

**ABSOLUTE MAXIMUM RATINGS**

Voltage Range on Any Pin Relative to Ground .....	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial.....	0°C to +70°C
Industrial .....	-40°C to +85°C
Storage Temperature Range .....	-55°C to +125°C
Soldering Temperature (DIP, leads).....	+260°C for 10 seconds
Soldering Temperature (surface mount).....	Refer to the JPC/JEDEC J-STD-020 Specification.

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

**RECOMMENDED DC OPERATING CONDITIONS**

( $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	$V_{CC}$		4.5	5.0	5.5	V
Logic 1 Input	$V_{IH}$		2.2		$V_{CC} + 0.3$	V
Logic 0 Input	$V_{IL}$		-0.3		+0.8	V
$V_{BAT}$ Battery Voltage	$V_{BAT}$		2.0	3	3.5	V

**DC ELECTRICAL CHARACTERISTICS**

( $V_{CC} = 4.5\text{V}$  to  $5.5\text{V}$ ;  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	$I_{LI}$		-1		1	$\mu\text{A}$
I/O Leakage (SDA, SQW/OUT)	$I_{LO}$		-1		1	$\mu\text{A}$
Logic 0 Output ( $I_{OL} = 5\text{mA}$ )	$V_{OL}$				0.4	V
Active Supply Current ( $f_{SCL} = 100\text{kHz}$ )	$I_{CCA}$				1.5	mA
Standby Current	$I_{CCS}$	(Note 3)			200	$\mu\text{A}$
$V_{BAT}$ Leakage Current	$I_{BATLKG}$			5	50	nA
Power-Fail Voltage ( $V_{BAT} = 3.0\text{V}$ )	$V_{PF}$		$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$	V

**DC ELECTRICAL CHARACTERISTICS**

( $V_{CC} = 0\text{V}$ ,  $V_{BAT} = 3.0\text{V}$ ;  $T_A = 0^\circ\text{C}$  to  $+70^\circ\text{C}$ ,  $T_A = -40^\circ\text{C}$  to  $+85^\circ\text{C}$ .) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
$V_{BAT}$ Current (OSC ON); SQW/OUT OFF	$I_{BAT1}$			300	500	nA
$V_{BAT}$ Current (OSC ON); SQW/OUT ON (32kHz)	$I_{BAT2}$			480	800	nA
$V_{BAT}$ Data-Retention Current (Oscillator Off)	$I_{BATDR}$			10	100	nA

**WARNING:** Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

**AC ELECTRICAL CHARACTERISTICS**(V<sub>CC</sub> = 4.5V to 5.5V; T<sub>A</sub> = 0°C to +70°C, T<sub>A</sub> = -40°C to +85°C.)

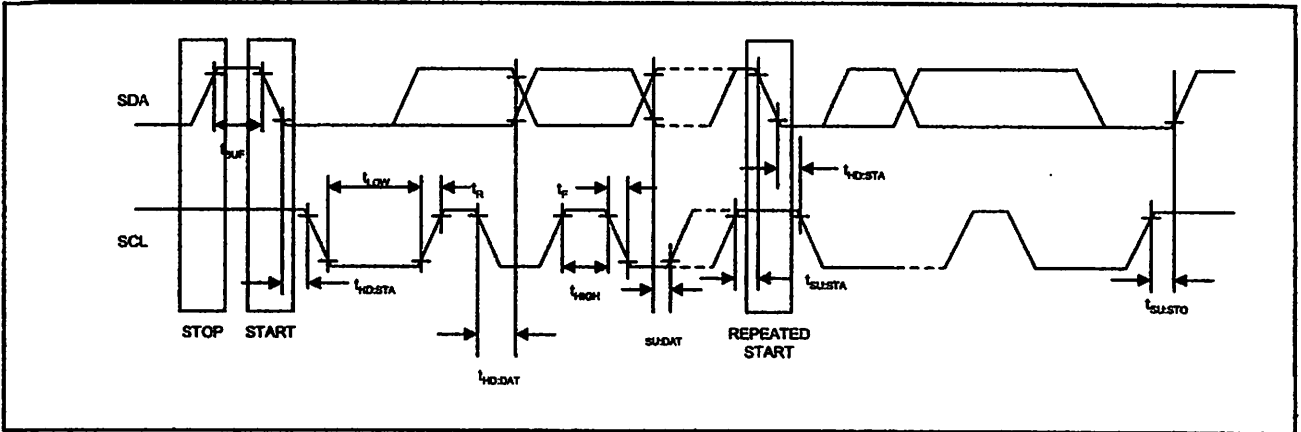
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f <sub>SCL</sub>		0		100	kHz
Bus Free Time Between a STOP and START Condition	t <sub>BUF</sub>		4.7			μs
Hold Time (Repeated) START Condition	t <sub>HD:STA</sub>	(Note 4)	4.0			μs
LOW Period of SCL Clock	t <sub>LOW</sub>		4.7			μs
HIGH Period of SCL Clock	t <sub>HIGH</sub>		4.0			μs
Setup Time for a Repeated START Condition	t <sub>SU:STA</sub>		4.7			μs
Data Hold Time	t <sub>HD:DAT</sub>		0			μs
Data Setup Time	t <sub>SU:DAT</sub>	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t <sub>R</sub>				1000	ns
Fall Time of Both SDA and SCL Signals	t <sub>F</sub>				300	ns
Setup Time for STOP Condition	t <sub>SU:STO</sub>		4.7			μs

**CAPACITANCE**(T<sub>A</sub> = +25°C)

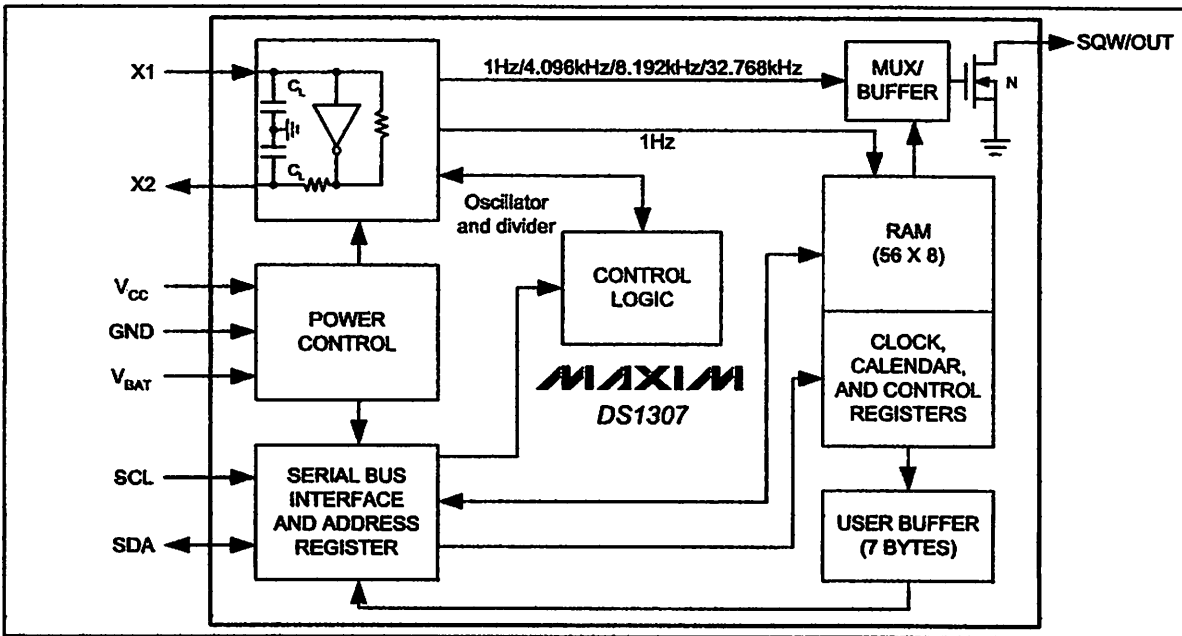
PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C <sub>IO</sub>				10	pF
Capacitance Load for Each Bus Line	C <sub>B</sub>	(Note 7)			400	pF

**Note 1:** All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I<sub>CCS</sub> specified with V<sub>CC</sub> = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V<sub>IH(MIN)</sub> of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t<sub>HD:DAT</sub> only has to be met if the device does not stretch the LOW period (t<sub>LOW</sub>) of the SCL signal.**Note 7:** C<sub>B</sub>—total capacitance of one bus line in pF.

**TIMING DIAGRAM**



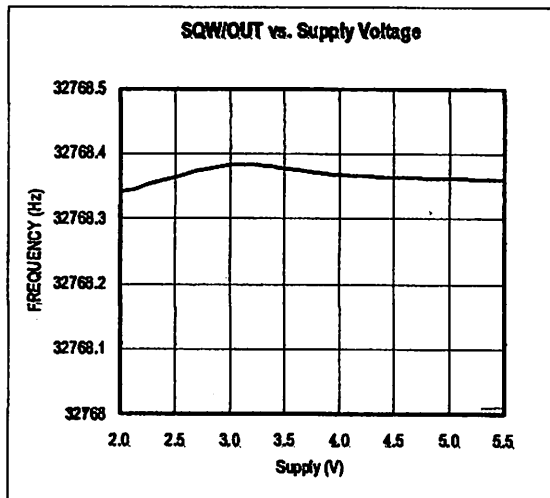
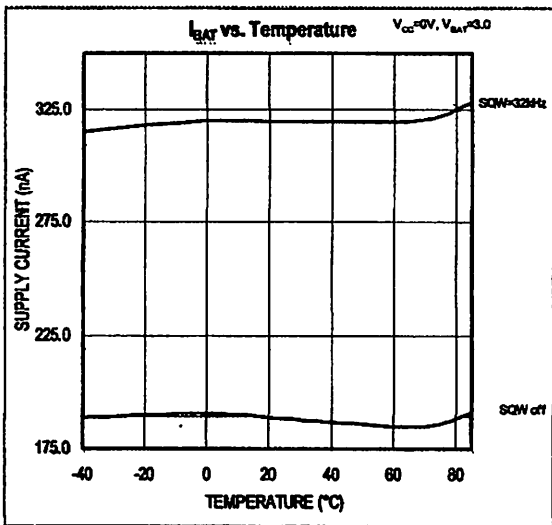
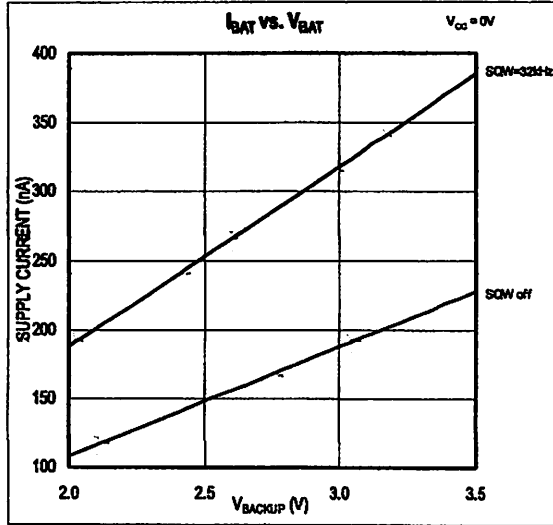
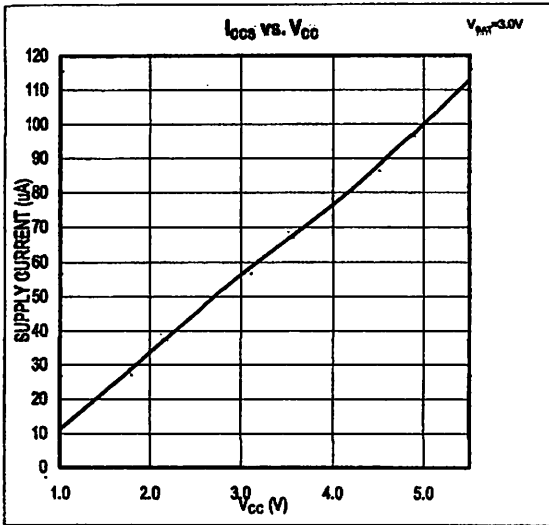
**Figure 1. Block Diagram**





**TYPICAL OPERATING CHARACTERISTICS**

(V<sub>CC</sub> = 5.0V, T<sub>A</sub> = +25°C, unless otherwise noted.)



**PIN DESCRIPTION**

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance ( $C_L$ ) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.  <b>Note:</b> For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
2	X2	
3	V <sub>BAT</sub>	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V <sub>BAT</sub> pin may prevent proper operation. If a backup supply is not required, V <sub>BAT</sub> must be grounded. The nominal power-fail trip point (V <sub>PF</sub> ) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V <sub>BAT</sub> nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C.  UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: <a href="http://www.maxim-ic.com/qa/info/ul/">www.maxim-ic.com/qa/info/ul/</a> .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I <sup>2</sup> C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
6	SCL	Serial Clock Input. SCL is the clock input for the I <sup>2</sup> C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V <sub>CC</sub> or V <sub>BAT</sub> applied. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> . If not used, this pin can be left floating.
8	V <sub>CC</sub>	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V <sub>CC</sub> is below V <sub>TP</sub> , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

**DETAILED DESCRIPTION**

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I<sup>2</sup>C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V<sub>CC</sub> falls below 1.25 x V<sub>BAT</sub>, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V<sub>CC</sub> falls below V<sub>BAT</sub>, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V<sub>CC</sub> when V<sub>CC</sub> is greater than V<sub>BAT</sub> + 0.2V and recognizes inputs when V<sub>CC</sub> is greater than 1.25 x V<sub>BAT</sub>. The block diagram in Figure 1 shows the main elements of the serial RTC.

## OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

## CLOCK ACCURACY

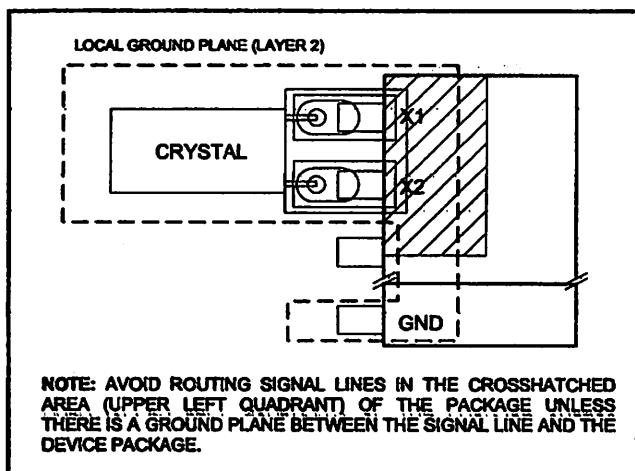
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: *Crystal Considerations with Dallas Real-Time Clocks* for detailed information.

**Table 1. Crystal Specifications\***

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	$f_0$		32.768		kHz
Series Resistance	ESR			45	k $\Omega$
Load Capacitance	$C_L$		12.5		pF

\*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: *Crystal Considerations for Dallas Real-Time Clocks* for additional specifications.

**Figure 2. Recommended Layout for Crystal**



## RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

## CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current ( $I_{BATDR}$ ).

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

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I<sup>2</sup>C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I<sup>2</sup>C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

**Table 2. Timekeeper Registers**

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year				Year				Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

**CONTROL REGISTER**

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

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

**Bit 7: Output Control (OUT).** This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

**Bit 4: Square-Wave Enable (SQWE).** This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

**Bits 1 and 0: Rate Select (RS[1:0]).** These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.096kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

## I<sup>2</sup>C DATA BUS

The DS1307 supports the I<sup>2</sup>C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I<sup>2</sup>C bus.

Figures 3, 4, and 5 detail how data is transferred on the I<sup>2</sup>C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

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

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

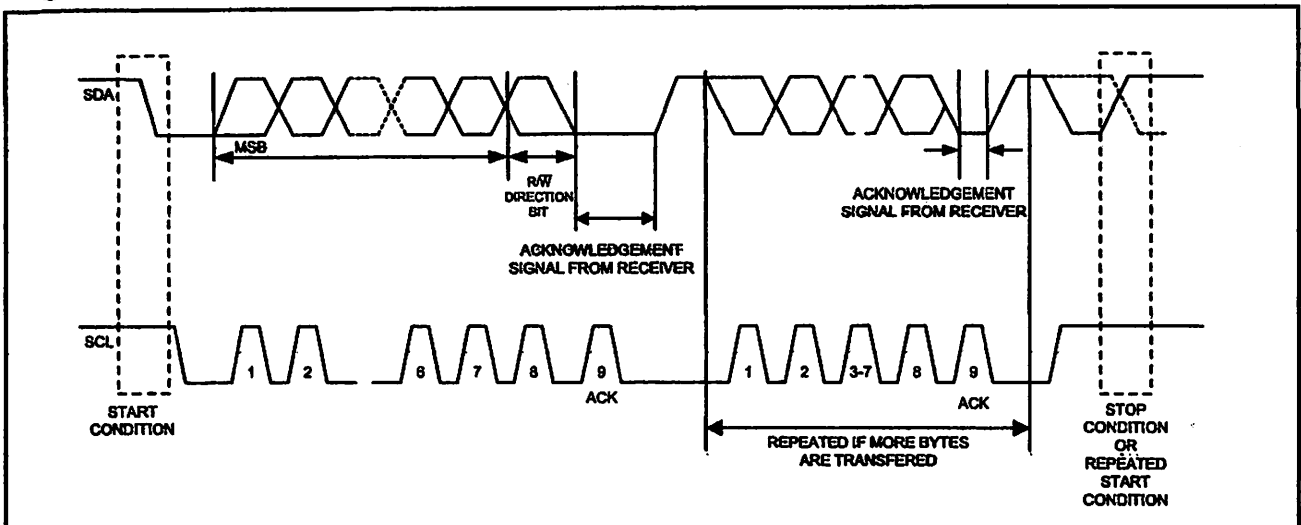
**STOP data transfer:** A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

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

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

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

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

**Figure 3. Data Transfer on I<sup>2</sup>C Serial Bus**

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

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

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

The DS1307 can operate in the following two modes:

- 1. Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ( $R/\bar{W}$ ), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
- 2. Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit ( $R/\bar{W}$ ), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

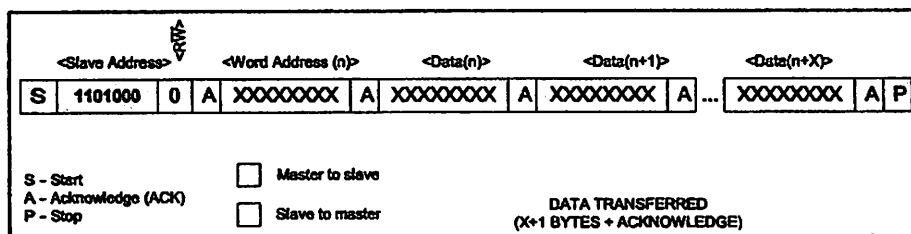
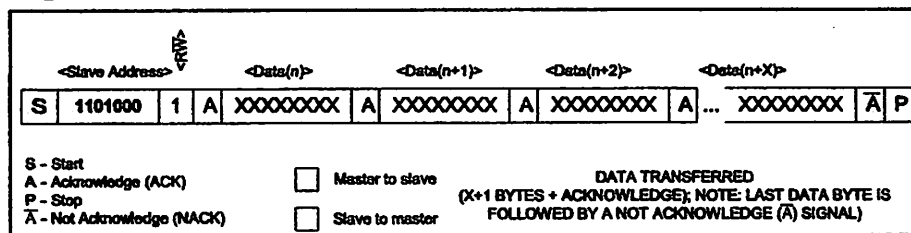
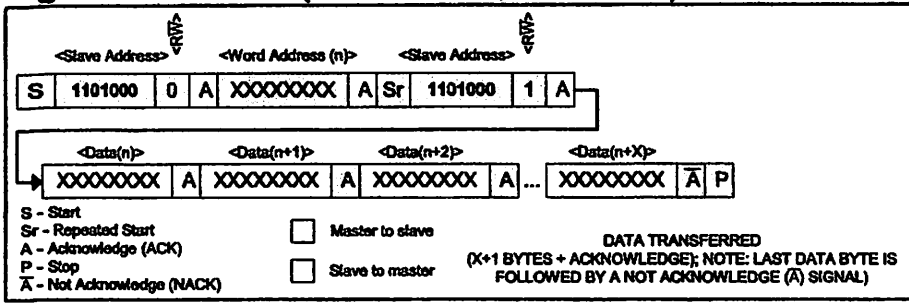


Figure 5. Data Read—Slave Transmitter Mode





**Figure 6. Data Read (Write Pointer, Then Read)—Slave Receive and Transmit**



**PACKAGE INFORMATION**

For the latest package outline information and land patterns, go to [www.maxim-ic.com/packages](http://www.maxim-ic.com/packages).

PACKAGE TYPE	PACKAGE CODE	DOCUMENT NO.
8 PDIP	—	<u>21-0043</u>
8 SO	—	<u>21-0041</u>

**REVISION HISTORY**

<b>REVISION DATE</b>	<b>DESCRIPTION</b>	<b>PAGES CHANGED</b>
100208	Moved the <i>Typical Operating Circuit and Pin Configurations</i> to first page.	1
	Removed the leaded part numbers from the <i>Ordering Information</i> table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the <i>Pin Description</i> table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the <i>Clock and Calendar</i> section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the <i>Control Register</i> section.	9
	Updated the <i>Package Information</i> section to reflect new package outline drawing numbers.	13

# **Manual Reference**

## **AT Command Set** (GSM 07.07, GSM 07.05, Siemens specific commands)

for the SIEMENS Mobile Phones

### **S35i, C35i, M35i**

The command descriptions or example sequences in this document imply no liability or warranty in any way. The author therefore will take no responsibility and will accept no liability which results of using the content of this document in any way.

All rights reserved. No part of this work covered by the copyrights hereof may be reproduced or copied in any form or by any means (graphic, electronic, or mechanical, including photocopying, taping, or information storage and retrieval systems) without written permission of the publisher.

**Revisions Overview**

Date	Version	Name	Description of revision
15-03-2000	1.0	Kel	created

**Table of Contents**

Revisions Overview.....2

Table of Contents.....2

1. Software Interface.....3

    1.1. OVERVIEW OF THE SUPPORTED AT COMMAND SET.....3

    1.2. AT COMMAND SET.....5

        1.2.1. *Hayes-Standard Commands*.....6

        1.2.2. *Acknowledgments for Normal Data Communication*.....6

    1.3. AT COMMANDS AND RESPONSES ACCORDING TO GSM 07.07 AND GSM 07.05.....7

        1.3.1. *AT Cellular Commands According to GSM 07.07*.....7

        1.3.2. *AT Commands According to GSM 07.05 for SMS*.....26

        1.3.3. *User-Defined Commands for Controlling the GSM Mobile Phone*.....33

        1.3.4. *Summary of All Unexpected Messages*.....44

APPENDIX A.....45

*Factory settings made by AT&F*.....45

*Features of the Telephone-Book Memory*.....45

*Writing to the FDN Phonebook / FDN Replacement*.....47

*Special hints for using +CPBR/+CPBW command*.....48

APPENDIX B.....50

*Example for creating/interrogation of an organizer entry*.....50

## 1. Software Interface

### 1.1. Overview of the Supported AT Command Set

Page	Commands 07.07	Function
7	AT+CGMI	Issue manufacturer ID code
7	AT+CGMM	Issue model ID code
7	AT+CGMR	Output the GSM telephone version
8	AT+CGSN	Output the serial number (IMEI)
8	AT+GSN	Output the serial number (IMEI)
8	AT+CHUP	Terminate call
8	AT+CEER	Query the reason for disconnection of last call
9	AT+CREG	Network registration
9	AT+COPS	Commands concerning selection of network operator
10	AT+CLK	Switch locks on and off
10	AT+CPWD	Change password to a lock
11	AT+CLIP	Display telephone number of calling party
11	AT+CCFC	Call forwarding
12	AT+CHLD	Call hold and multiparty
12	AT+CPAS	Query the telephone status
13	AT+CPIN	Enter PIN and query lock
13	AT+CBC	Battery charge
14	AT+CSQ	Output signal quality
14	AT+CPBS	Select a telephone book
15	Fehler! Kein gültiges Resultat für Tabelle.	Read a telephone-book entry
15	AT+CPBW	Write a telephone-book entry
16	AT+CMEE	Expanded error messages according to GSM 07.07
17	AT+VTS	Send a DTMF tone
18	AT+VTD	Set duration of a DTMF tone
18	AT+WS46	Select wireless network
18	AT+CSCS	Select TE character set
19	AT+CAOC	Advice of charge
19	AT+CSSN	Supplementary service notifications
20	AT+CRSM	Restricted SIM access
20	AT+CIM	Output of IMSI
21	AT+CACM	Accumulated call meter
21	AT+CAMM	Accumulated call meter maximum
22	AT+CLCC	List Current Calls
23	AT+CCLK	Clock
23	AT+COPN	Read operator names
23	AT+CPUC	Price per unit and currency table
24	AT+CALM	Alert sound mode
24	AT+CRSL	Ringer sound level
24	AT+CLVL	Loudspeaker volume level
24	AT+CMUT	Mute control
25	AT+CVIB	Vibrator mode

Page	Commands 07.05	Function
26	AT+CSMS	Selection of message service
27	AT+CPMS	Selection of SMS memory
27	AT+CMGF	SMS format
28	AT+CSCA	Address of the SMS service center
28	AT+CNMI	Display new incoming SMS
29	AT+CNMA	Acknowledgment of a short message directly output
30	AT+CMGL	List SMS
31	AT+CMGR	Read in an SMS
31	AT+CMGS	Send an SMS
31	AT+CMSS	Send an SMS from the SMS memory
32	AT+CMGW	Write an SMS to the SMS memory
32	AT+CMGD	Delete an SMS in the SMS memory
32	AT+CSCB	Select cell broadcast messages
32	AT+CMGC	Send an SMS command

Page	Siemens- specific commands	Function
33	AT^SPBS	Select a telephone book (including Siemens-specific books)
33	AT^SDLD	Delete the "last number redial" memory
34	AT^SPBC	Seek the first entry in the sorted telephone book which begins with the selected (or next available) letter
34	AT^SPBG	Read entry from the sorted telephone book via the sorted index
35	AT^SLCK	Switch locks (including user-defined locks) on and off
35	AT^SPWD	Change password to a lock (including user-defined locks)
36	AT^SACM	Output ACM (accumulated call meter) and ACMmax
36	AT^SPLM	Read the PLMN
36	AT^SPLR	Read an entry from the preferred-operator list
36	AT^SPLW	Write an entry to the preferred-operator list
37	AT^SCNI	Output call number information
37	AT^SNFV	Set the volume
37	AT^SNFS	Select NF hardware
38	AT^SRTC	Set the ringing tone
38	AT^SCID	Output card ID
38	AT^SCKS	Output SIM card status
39	AT^SPIC	Output PIN counter
39	AT^SMGO	SMS overflow indicator
40	AT^SMGL	List SMS (without status change from unread to read)
40	AT^SMGR	Read SMS record without Changing unread->read
40	AT^SMSO	Switch device off
41	AT^SLNG	Language settings
41	AT^SSTK	SIM Toolkit
41	AT^SBNW	Binary Write
43	AT^SBNR	Binary Read

## **1.2. AT Command Set**

Remote control operation of the GSM mobile telephone runs via a serial interface (data cable or infrared connection), where AT+C commands according to ETSI GSM 07.07 and GSM 07.05 specification as well as several manufacturer specific AT commands are available. These commands are described in more detail later on.

The modem guideline V.25ter applies to the sequence of the interface commands. According to this guideline, commands should begin with the character string "AT" and end with "<CR>" (= 0x0D). The input of a command is acknowledged by the display of "OK" or "ERROR". **A command currently in process is interrupted by each additional character entered.** This means that you should not enter the next command until you have received the acknowledgment; otherwise the current command is interrupted.

The commands supported are listed in the following tables:

### 1.3.3. User-Defined Commands for Controlling the GSM Mobile Phone

Since user-defined commands cannot be implemented according to official syntax, the character string "+C" is replaced by "^S" ("^" = 0x5E). For the future: if a user-defined command is accepted in the same syntax in GSM recommendations, the command can be addressed using both command strings.

<b>AT^SPBS</b>	<b>Select a telephone book (including Siemens-specific books)</b>
<small>Test command</small> <b>AT^SPBS=?</b>	<small>Response</small> <b>^SPBS: (list of supported &lt;sto&gt;s)</b> <b>OK/ERROR/+CME ERROR</b>



**1.3.2. AT Commands According to GSM 07.05 for SMS**

The GSM 07.05 commands are used for operating the SMS functions of the GSM mobile phone. The GSM module MOBILE supports the SMS PDU mode.

<p><b>AT+CSMS</b></p>	<p><b>Selection of message service</b> <b>Revision according to GSM 07.05 Version 5.0.0</b></p>
<p>Test command <b>AT+CSMS=?</b></p>	<p>Response <b>+CSMS: (list of supported &lt;service&gt;S)</b></p> <p>Parameter <b>&lt;service&gt; 0</b> GSM 3.40 and 3.41 <b>1</b> GSM 3.40 and 3.41 and compatibility of the AT command syntax for phase 2+</p> <p><b>NOTE:</b> Deactivating the phase 2+ compatibility is only possible if the direct output of short messages +CNMI=1,2 or +CNMI=1,3 is not activated. If necessary, the latter should be deactivated first.</p>
<p>Read command <b>AT+CSMS?</b></p>	<p>Response <b>+CSMS: &lt;service&gt;,&lt;mt&gt;,&lt;mo&gt;,&lt;bm&gt;</b></p> <p>Parameter <b>&lt;service&gt; 0</b> GSM 3.40 and 3.41 <b>&lt;mt&gt;</b> 1 Mobile terminated messages Type supported <b>&lt;mo&gt;</b> 1 Mobile originated messages Type supported <b>&lt;bm&gt;</b> 1 Broadcast type messages Type supported 0 Type not supported</p>
<p>Write command <b>AT+CSMS= &lt;service&gt;</b></p>	<p>Parameter <b>&lt;service&gt; 0</b> GSM 3.40 and 3.41</p> <p>Response <b>+CSMS: &lt;mt&gt;,&lt;mo&gt;,&lt;bm&gt;</b> <b>OK/ERROR/+CMS ERROR</b></p>

AT+CPMS	Selection of SMS memory Revision according to GSM 07.05 Version 4.7.0
Test command <b>AT+CPMS=?</b>	Response +CPMS: (list of supported <mem1>s),( list of supported <mem2>s) ,(list of supported <mem3>s) Parameter <mem1> Memory from which messages are read and deleted "SM" SIM-messages memory <mem2> Memory to which messages are written and sent "SM" SIM-messages memory <mem3> Memory in which received messages are stored, if forwarding to the PC is not set ("CNMI") "SM" SIM-messages memory
Read command <b>AT+CPMS?</b>	Response +CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2>,<total2> ,<mem3>,<used3>,<total3> Parameter <memx> Memory from which messages are read and deleted <usedx> Number of messages currently in <memx> <totalx> Number of storable messages in <memx>
Write command <b>AT+CPMS=                      &lt;mem1&gt;                      [,&lt;mem2&gt;                      [,&lt;mem3&gt;]]</b>	Parameter <mem1> See Test command <mem2> See Test command <mem3> See Test command  Response +CPMS: <used1>,<total1>,<used2>,<total3>,<used3>,<total3> OK/ERROR/+CMS ERROR

AT+CMGF	SMS format
Test command <b>AT+CMGF=?</b>	Response +CMGF: (list of supported <mode>s) Parameter <mode>: 0 PDU mode
Read command <b>AT+CMGF?</b>	Response +CMGF: <mode> Parameter <mode>: 0 PDU mode
Write command <b>AT+CMGF=[&lt;                      mode&gt;]</b>	Parameter <mode>: 0 PDU mode  Response OK/ERROR

<b>AT+CSCA</b>	<b>Address of the SMS service center</b>
Test command <b>AT+CSCA=?</b>	Response OK
Read command <b>AT+CSCA?</b>	Response <b>+CSCA: &lt;sca&gt;,&lt;tosca&gt;</b>  Parameter <b>&lt;sca&gt;</b> Service-center address in string format <b>&lt;tosca&gt;</b> Service-center address format
Write command <b>AT+CSCA= &lt;sca&gt;[,&lt;tosca&gt;]</b>	Parameter <b>&lt;sca&gt;</b> Service-center address in string format <b>&lt;tosca&gt;</b> Service-center address format  Response OK/ERROR

<b>AT+CNMI</b>	<b>Display new incoming SMS Revision according to GSM 07.05 Version 4.7.0</b>
Test command <b>AT+CNMI=?</b>	Response <b>+CNMI: (list of supported &lt;mode&gt;s),(list of supported &lt;mt&gt;s),(list of supported &lt;bm&gt;s),(list of supported &lt;ds&gt;s),(list of supported &lt;bfr&gt;s)</b>  Parameter <b>&lt;mode&gt;</b> 0 Buffers unexpected messages (but is equivalent to rejecting; see <bfr>) 1 Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved. Otherwise forward them directly to the TE. (only with S25ff) 2 Buffers unexpected messages if serial interface is occupied, otherwise they are output (only models before S25)  <b>&lt;mt&gt;</b> 0 Suppresses unexpected messages for incoming short messages 1 Unexpected messages of a received short message (SMS-DELIVER) that is stored on a chip card are output in the form <b>+CMTI: &lt;mem&gt;,&lt;index&gt;</b> 2 Unexpected messages of a received short message (SMS-DELIVER) (except class 2 and the message "Waiting Indication Group: store message") are output in the form <b>+CMT: [&lt;alpha&gt;],&lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b> (<alpha> is not supported) Class 2 and the message "Waiting Indication Group: store message" are output as <b>&lt;mt&gt;=1</b> 3 Unexpected messages of a received short message (SMS-DELIVER) class 3 are output as <b>&lt;mt&gt;=2</b> . Messages with other data coding schemes are output as <b>&lt;mt&gt;=1</b> .  <b>NOTE: &lt;mt&gt;=2 and &lt;mt&gt;=3 are not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1</b>  <b>&lt;bm&gt;</b> 0 Suppresses unexpected messages for incoming cell broadcast messages 2 Outputs unexpected messages for cell broadcast messages in the form <b>+CBM: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b>  <b>&lt;ds&gt;</b> 0 Suppresses unexpected messages for incoming SMS status reports 2 Outputs unexpected messages for SMS status reports in the form <b>+CDS: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b>

	<p>&lt;bfr&gt; 1 Buffered unexpected messages are rejected when switching from &lt;mode&gt; 0 to &lt;mode&gt; 2.</p> <p>&lt;mem&gt; See +CPMS                  &lt;index&gt; Index of the record on the chip card                  &lt;alpha&gt; alphanumeric representation of the sender address                  &lt;length&gt; Length of &lt;pdu&gt;                  &lt;pdu&gt; See +CMGL</p>
<p>Read command  <b>AT+CNMI?</b></p>	<p>Response                  +CNMI: &lt;mode&gt;,&lt;mt&gt;,&lt;bm&gt;,&lt;ds&gt;,&lt;bfr&gt;</p> <p>Parameter                  &lt;mode&gt; See Test command                  &lt;mt&gt; See Test command                  &lt;bm&gt; See Test command                  &lt;ds&gt; See Test command                  &lt;bfr&gt; See Test command</p>
<p>Write command  <b>AT+CNMI=</b>                  [&lt;mode&gt;                  [,&lt;mt&gt;[,&lt;bm&gt;                  [,&lt;ds&gt;[,&lt;bfr&gt;]]]]</p>	<p>Parameter                  &lt;mode&gt; See Test command                  &lt;mt&gt; See Test command                  &lt;bm&gt; See Test command                  &lt;ds&gt; See Test command                  &lt;bfr&gt; See Test command</p> <p>Response                  OK/ERROR/+CMS ERROR</p>
	<p>Unexpected message                  +CMTI: &lt;mem&gt;,&lt;index&gt; Indication that new message has arrived</p> <p>+CMT: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the short message</p> <p>+CDS: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the status report</p> <p>+CBM: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the cell broadcast message</p>

<p><b>AT+CNMA</b></p>	<p>Acknowledgment of a short message directly output (without storing on the chip card)  <b>Revision according to GSM 07.05 Version 5.0.0</b>  <i>(NOTE: This command is not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1)</i></p>
<p>Test command  <b>AT+CNMA=?</b></p>	<p>Response                  +CNMA: (list of supported &lt;n&gt;s)</p> <p>Parameter                  &lt;n&gt; 0 Mode of functioning analogous to GSM 07.05 text mode</p>
<p>Write command  <b>AT+CNMA[=&lt;n&gt;]</b></p>	<p>Parameter                  &lt;n&gt; See Test command</p> <p>Response                  OK/ERROR/+CMS ERROR: &lt;err&gt;</p>

<p><b>AT+CMGL</b></p>	<p><b>List SMS</b> <b>Revision according to GSM 07.05 Version 4.7.0</b></p>										
<p>Test command <b>AT+CMGL=?</b></p>	<p>Response <b>+CMGL:</b> (list of supported &lt;stat&gt;s)</p> <p>Parameter <b>&lt;stat&gt;</b></p> <table border="0"> <tr> <td style="padding-right: 20px;">0</td> <td>"REC UNREAD": received unread messages (default)</td> </tr> <tr> <td>1</td> <td>"REC READ": received read messages</td> </tr> <tr> <td>2</td> <td>"STO UNSENT": stored unsent messages</td> </tr> <tr> <td>3</td> <td>"STO SENT": stored sent messages</td> </tr> <tr> <td>4</td> <td>"ALL": all messages</td> </tr> </table>	0	"REC UNREAD": received unread messages (default)	1	"REC READ": received read messages	2	"STO UNSENT": stored unsent messages	3	"STO SENT": stored sent messages	4	"ALL": all messages
0	"REC UNREAD": received unread messages (default)										
1	"REC READ": received read messages										
2	"STO UNSENT": stored unsent messages										
3	"STO SENT": stored sent messages										
4	"ALL": all messages										
<p>Write command <b>AT+CMGL</b> <b>[=&lt;stat&gt;]</b></p>	<p>Parameter <b>&lt;stat&gt;</b>                    See Test command</p> <p>Response If PDU mode (+CMGF=0) and command are successful: <b>+CMGL:</b>&lt;index&gt;, &lt;stat&gt;, [<b>&lt;alpha&gt;</b>], &lt;length&gt; &lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;&lt;CR&gt;&lt;LF&gt; <b>+CMGL:</b> &lt;index&gt;, &lt;stat&gt;, [<b>&lt;alpha&gt;</b>], &lt;length&gt; &lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;&lt;CR&gt;&lt;LF&gt; [...]</p>										
	<p>Parameter <b>&lt;pdu&gt;</b>                    The PDU begins with the service-center address (according to GSM04.11), followed by the TPDU according to GSM03.40 in hexadecimal format otherwise: <b>+CMS ERROR:</b> &lt;err&gt;</p>										

<b>AT+CMGR</b>		<b>Read in an SMS</b>	
		<b>Revision according to GSM 07.05 Version 4.7.0</b>	
Test command	<b>AT+CMGR=?</b>	Response	OK
Write command	<b>AT+CMGR= &lt;index&gt;</b>	Parameter	<b>&lt;index&gt;</b> Index of message in selected memory <mem1>
		Response	<b>If PDU mode (+CMGF=0) and command are successful:</b> +CMGR: <stat>,[<alpha>],<length><CR><LF><pdu>
		Parameter	<b>&lt;pdu&gt;</b> Siehe "AT+CMGL" otherwise: +CMS ERROR: <err>

<b>AT+CMGS</b>		<b>Send an SMS</b>	
Test command	<b>AT+CMGS=?</b>	Response	OK
Write command	<b>If PDU mode (+CMGF=0) +CMGS=&lt;length&gt;&lt;CR&gt;PDU is given &lt;ctrl-Z/ESC&gt;</b>	Parameter	<b>&lt;length&gt;</b> Length of PDU <b>&lt;pdu&gt;</b> See "AT+CMGL" <b>&lt;mr&gt;</b> Message reference
		Response	<b>If sending is successful:</b> +CMGS: <mr> <b>If sending is not successful:</b> +CMS ERROR: <err>

<b>AT+CMSS</b>		<b>Send an SMS from the SMS memory</b>	
Test command	<b>AT+CMSS=?</b>	Response	OK
Write command	<b>+CMSS=&lt;index&gt;[,&lt;da&gt;[,&lt;toda&gt;]]</b>	Parameter	<b>&lt;index&gt;</b> Index of message in selected memory <mem1> <b>&lt;da&gt;</b> Destination address in string format <b>&lt;toda&gt;</b> Format of destination address <b>&lt;mr&gt;</b> Message reference
		Response	<b>If sending is successful:</b> +CMSS: <mr> <b>If sending is not successful:</b> +CMS ERROR: <err>

<b>AT+CMGW</b>	<b>Write an SMS to the SMS memory</b>
Test command <b>AT+CMGW=?</b>	Response OK
Write command <b>If PDU mode (+CMGF=0) AT+CMGW=&lt;length&gt;[,&lt;stat&gt;]&lt;CR&gt;PDU is given &lt;ctrl-Z/ESC&gt;</b>	Parameter <length>           Length of PDU <stat>                See command +CMGL <pdu>                 See "AT+CMGL" <index>              Index of message in selected memory <mem1>  Response <b>+CMGW: &lt;index&gt;</b> <b>+CMS ERROR: &lt;err&gt;</b>

<b>AT+CMGD</b>	<b>Delete an SMS in the SMS memory</b>
Test command <b>At+CMGD=?</b>	Response OK
Write command <b>AT+CMGD= &lt;index&gt;</b>	Parameter <index>                Index of message in the selected memory <mem1>  Response OK/ERROR/+CMS ERROR

<b>AT+CSCB</b>	<b>Select cell broadcast messages</b>
Test command <b>AT+CSCB=?</b>	Response <b>+CSCB: (list of supported &lt;mode&gt;s)</b>  Parameter <mode>            0    Accepts messages that are defined in <mids> and <dcss> 1    Does not accept messages that are defined in <mids> and <dcss>
Read command <b>AT+CSCB?</b>	Response <b>+CSCB: &lt;mode&gt;,&lt;mids&gt;,&lt;dcss&gt;</b>  Parameter <mode>            See Test command <mids>            String type; combinations of CBM message IDs <dcss>            String type; combinations of CBM data coding schemes
Write command <b>AT+CSCB=[&lt; mode&gt;[,&lt;mids &gt;[,&lt;dcss&gt;]]]</b>	

<b>AT+CMGC</b>	<b>Send an SMS command</b>
Test command <b>AT+CMGC=?</b>	Response OK
Write command <b>If PDU mode (+CMGF=0) +CMGC=&lt;length&gt;&lt;CR&gt;PDU is given &lt;ctrl-Z/ESC&gt;</b>	Parameter <length>           Length of PDU <pdu>                See "AT+CMGL" <mr>                 Message reference  Response <b>If sending is successful:</b> <b>+CMGC: &lt;mr&gt;</b> <b>If sending is not successful:</b> <b>+CMS ERROR: &lt;err&gt;</b>

**1.3.3. User-Defined Commands for Controlling the GSM Mobile Phone**

Since user-defined commands cannot be implemented according to official syntax, the character string "+C" is replaced by "^S" ("^" = 0x5E). For the future: if a user-defined command is accepted in the same syntax in GSM recommendations, the command can be addressed using both command strings.

<b>AT^SPBS</b>	<b>Select a telephone book (including Siemens-specific books)</b>
Test command <b>AT^SPBS=?</b>	Response ^SPBS: (list of supported <sto>s) OK/ERROR/+CME ERROR Parameter <sto>      "FD" SIM fix-dialing telephone book "SM" SIM telephone book "ME" Telephone book in device "DC" ME Dialed Calls List "ON" Own telephone numbers "LD" SIM last dialing number "MC" ME Missed Calls List "RC" ME Received Calls List  "MD" Last number redial memory in telephone device "OW" Own numbers "BD" Barred dialing numbers "SD" Service dialing numbers "MS" Missed dialing numbers (unanswered calls) "CD" Callback dialing numbers (answered calls) "BL" Blacklist dialing numbers (banned numbers from remote) "MB" Mailbox dialing numbers (network-operator mailbox)  "CS" Common sortable telephone book (sorted combination of "SM", "ME", "FD"; access only via ^SPBC, ^SPBG) "RD" Red book (all entries in "CS" whose name portions have an exclamation point (!) as the final character)
	*For a description of the telephone-book features, see Appendix A
Read command <b>AT^SPBS?</b>	Response ^SPBS: <sto> OK/ERROR/+CME ERROR Parameter <sto>      See Test command
Write command <b>AT^SPBS=                  &lt;sto&gt;</b>	Parameter <sto>      See Test command  Response OK/ERROR/+CME ERROR

<b>AT^SDLD</b>	<b>Delete the "last number redial" memory</b>
Test command <b>AT^SDLD=?</b>	Response OK
Execute command <b>AT^SDLD</b>	Response OK/ERROR/+CME ERROR



<b>AT^SPBC</b>	Seek the first entry in the sorted telephone book which begins with the selected (or next available) letter
Test command <b>AT^SPBC=?</b>	Response ^SPBC: (list of sorted telephone books supported <mem>s) See AT+CPBS/AT^SPBS OK/ERROR/+CME ERROR
Write command <b>AT^SPBC=&lt;char&gt;</b>	Parameter <char> First letter of sought entry "A" to "Z" (with any other character, the index of the first entry that begins with a special character is sent back) <index> Index in the sorted telephone book (access via AT^SPBG) Response ^SPBC: <index> OK/ERROR/+CME ERROR

<b>AT^SPBG</b>	Read entry from the sorted telephone book via the sorted index
Test command <b>AT^SPBG=?</b>	Response ^SPBG: (list of supported <index>s), <nlength>, <tlength> OK/ERROR/+CME ERROR Parameter <index> Location number <nlength> Max. length of telephone number <tlength> Max. length of the text corresponding to the number
Write command <b>AT^SPBG= &lt;index1&gt; [, &lt;index2&gt;]</b>	Response ^SPBG: <index1>, <nummer>, <typ>, <text>[<CR><CL> ^SPBG: ..... ^SPBG: <index2>, <nummer>, <typ>, <text> ] OK/ERROR/+CME ERROR Parameter <index1> Location number where the read of the entry starts <index2> Location number where the read of the entry ends <nummer> Telephone number <typ> Type of number <text> Text corresponding to the telephone number

AT^SLCK	Switch locks (including user-defined locks) on and off
<p><b>Test command</b> AT^SLCK=?</p>	<p><b>Response</b> ^SLCK: (list of supported &lt;fac&gt;s) OK/ERROR/+CME ERROR</p> <p><b>Parameter</b> &lt;fac&gt;</p> <ul style="list-style-type: none"> <li>"PS" Phone locked to SIM (device code)</li> <li>"SC" SIM card (PIN)</li> <li>"FD" FDN lock</li> <li>"AO" BAOC (bar all outgoing calls)</li> <li>"OI" BOIC (bar outgoing international calls)</li> <li>"OX" BOIC-exHC (bar outgoing international calls except to home country)</li> <li>"AI" BAIC (bar all incoming calls)</li> <li>"IR" BIC-Roam (bar incoming calls when roaming outside the home country)</li> <li>"AB" All barring services</li> <li>"AG" All outgoing barring services</li> <li>"AC" All incoming barring services</li> </ul> <ul style="list-style-type: none"> <li>"PN" Network personalization (GSM 02.22)</li> <li>"PC" Corporate personalization (GSM 02.22)</li> <li>"PU" Network subset personalization (GSM 02.22)</li> <li>"PP" Service provider personalization (GSM 02.22)</li> <li>"PF" Phone locked to very first inserted SIM</li> </ul>
<p><b>Write command</b> AT^SLCK = &lt;fac&gt;, &lt;mode&gt; [,&lt;passwd&gt; [,&lt;class&gt;]]</p>	<p><b>Parameter</b> &lt;fac&gt; See Test command &lt;mode&gt; 0 Cancels lock 1 Activates lock 2 Queries lock status &lt;passwd&gt; Password &lt;class&gt; 1 Voice 2 Data 4 Fax 7 All classes (default value)</p> <p><b>Response</b> If &lt;mode&gt;=2 and command is successful ^SLCK: &lt;status&gt;[,&lt;class1&gt;[&lt;CR&gt;&lt;LF&gt; ^SLCK: &lt;status&gt;, class2....]]</p> <p><b>Parameter</b> &lt;status&gt; 0 Off 1 On</p> <p>OK/ERROR/+CME ERROR</p>

AT^SPWD	Change password to a lock (including user-defined locks)
<p><b>Test command</b> AT^SPWD=?</p>	<p><b>Response</b> ^SPWD: list of supported (&lt;fac&gt;, &lt;pwdlength&gt;)s OK/ERROR/+CME ERROR</p> <p><b>Parameter</b> &lt;fac&gt; "P2" PIN2 otherwise See Test command for the command AT^SLCK, without "FD" &lt;pwdlength&gt; Length of password</p>
<p><b>Write command</b> AT^SPWD = &lt;fac&gt;,&lt;oldpwd&gt;, &lt;newpwd&gt;</p>	<p><b>Parameter</b> &lt;fac&gt; See Test command for the command AT^SLCK &lt;oldpwd&gt;, &lt;newpwd&gt; Old and new password</p> <p><b>Response</b> OK/ERROR/+CME ERROR</p>

<b>AT^SACM</b>	<b>Output ACM (accumulated call meter) and ACMmax</b>
Test command <b>AT^SACM=?</b>	Response ^SACM: (list of supported <n>s)
Execute command <b>AT^SACM</b>	Response ^SACM: <n>,<acm>,<acm_max> OK/ERROR/+CME ERROR Parameter <n> See Test command <acm> Accumulated call meter <acm_max> Maximum accumulated call meter
Write command <b>AT^SACM=&lt;n&gt;</b>	Parameter <n> 0 Suppresses the unexpected message 1 Outputs the unexpected message

<b>AT^SPLM</b>	<b>Read the PLMN list</b>
Test command <b>AT^SPLM=?</b>	Response OK
Execute command <b>AT^SPLM</b>	Response ^SPLM:numeric <oper>,long alphanumeric <oper><CR><LF> ^SPLM:..... OK/ERROR/+CME ERROR Parameter <oper> Network operator in numeric and alphanumeric notation

<b>AT^SPLR</b>	<b>Read an entry from the preferred-operator list</b>
Test command <b>AT^SPLR=?</b>	Response ^SPLR: (list of supported <index>s) OK/ERROR/+CME ERROR Parameter <index> Location numbers
Write command <b>AT^SPLR=&lt;index1&gt; [, &lt;index2&gt;]</b>	Response ^SPLR: <index1>, numeric <oper> ^SPLR: ..... ^SPLR: <index2>, numeric <oper> OK/ERROR/+CME ERROR Parameter <index1> Location number where the read of the entry starts <index2> Location number where the read of the entry ends <oper> Network operator in numeric form

<b>AT^SPLW</b>	<b>Write an entry to the preferred-operator list</b>
Test command <b>AT^SPLW=?</b>	Response ^SPLW: (list of supported <index>s) OK/ERROR/+CME ERROR Parameter <index> Location number
Write command <b>AT^SPLW=&lt;index&gt;[, &lt;oper&gt;]</b>	Parameter <index> Location number at which the entry is written <oper> Network operator in numeric form Response OK/ERROR/+CME ERROR

<http://www.globalkomputer.com/Bahasan/Komunikasi-Data/Topik/PDU/Subtopik/SMS-Deliver-PDU.html>

## SMS Deliver PDU

### SMS Deliver PDU (Mobile Terminated)

SMS Deliver PDU ialah terminal menerima pesan yang datang/masuk dari SMSC (lihat topik Teknologi SMS) dalam format PDU.

Skema format SMS Deliver PDU adalah :

| SCA | PDU Type | OA | PID | DCS | SCTS | UDL | UD |

#### 1. SCA (Service Center Address)

SCA memiliki 3 komponen utama, yaitu:

a. Len (Length) yaitu panjang informasi SMSC dalam oktet

b. Type of Number sebagai type address dari SMSC

nilai 81h untuk format lokal

nilai 91h untuk format internasional

c. BSC Digits berupa nomor SMSC. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal.

Contoh:

SCA bernilai 059126181642. maka,

Len = 05,

Type of Number = 91 menyatakan format internasional

BSC Digits = 26 18 16 42.

Berikut contoh penulisan nomor SMSC untuk masing-masing operator

===Operator===	Nomor SMSC	==== Format PDU =====
Satelindo---->	---62816124	05 91 26 18 16 42 -- --
Excelcomindo->	62818445009	07 91 26 18 48 54 00 F9
Telkomse1---->	-6281100000	06 91 26 18 01 00 00 --
IM3----->	62855000000	05 91 26 58 05 00 00 F0

Pada IM3 Nomor SMSC 62855000000 bernilai ganjil, maka mendapat tambahan F menjadi

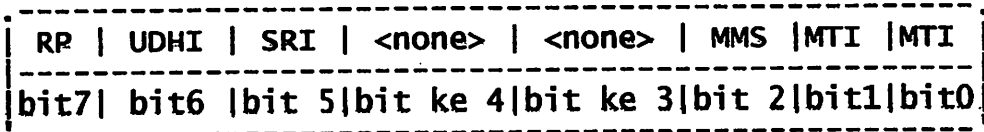
6285500000F

Setelah itu dilakukan perpindahan tempat setiap 2 digit untuk mengubah Nomor SMSC ke Format PDU (swapped nibble presentation) dari 62 85 50 00 00 0F menjadi 26 58 05 00 00 F0

Begitu juga halnya pada Nomor SMSC Excelcomindo.

## 2. PDU Type (Protocol Data Unit Type)

Format PDU Type adalah



Nilai default dari PDU untuk SMS-Deliver adalah 04 heksadesimal, akan tetapi pada PDU Type diubah ke biner menjadi 00000100, dimana nilai "1" berada pada biner ke 2 yaitu MMS.

Contoh:

PDU Type bernilai 24 heksadesimal = 00100100 biner.

Nilai "1" berada pada biner 2 (MMS) dan 5(SRI).

## 3. (OA) Originator Address

OA memiliki 3 komponen utama, yaitu:

a. Len (length) yaitu panjang nomor Originator Address

Format OA pada PDU dengan default = 14 dan bertype lokal

Format OA pada PDU = 0C bila nomor OA berawalan "+" dan bertype internasional

b. Type of Number sebagai type address dari Originator Address

nilai 81h untuk format lokal

nilai 91h untuk format internasional

c. BSC Digits berupa nomor OA. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal.

Contoh:

Bila Nomor OA = ABCDEFGHIJKLMNOPQRST, maka

Len = 14

Type of Number = 81 (lokal)

BSC Digits = BA DC FE HG JI LK NM PO RQ TS

Bila Nomor OA = +ABCDEFGHIJKL, maka

Len = 0C

Type of Number = 91 (internasional)

BSC Digits = BA DC FE HG JI LK

Bila Nomor OA dalam format PDU = 0C91261885980482, Maka

Len = 0C

Type of Number = 91 (internasional)

BSC Digits = 26 18 85 98 04 82

Nomor OA sebenar adalah +628158894028, yaitu nomor yang mengirim SMS ke penerima.

Tanda +62 merupakan asal negara dimana 62 adalah negara Indonesia, sedangkan 8158894028 merupakan nomor Handphone yang ditambah prefiks "0" (no1) menjadi 08158894028.

#### 4. PID (Protocol Identifier)

Nilai default dari PID ialah 00 = "Standard-Text-SMS"

#### 5. DCS (Data Coding Scheme)

DCS dapat menentukan message class

N11 Hexa	- Character- -- Coding --	Message Class
00	default(7 bit)	no class
F0	default(7 bit)	class 0(immediate display)
F1	default(7 bit)	class 1(Mobile Equipment-specific)
F2	default(7 bit)	class 2(SIM specific message)
F3	default(7 bit)	class 3(Terminate Equipment-specific)
F4	8 bit	class 0(immediate display)
F5	8 bit	class 1(Mobile Equipment-specific)
F6	8 bit	class 2(SIM specific message)
F7	8 bit	class 3(Terminate Equipment-specific)

Hal yang perlu diperhatikan, pada beberapa handphone message dengan class 0 dengan encoding 7 bit berupa flash SMS

Sedangkan dengan encoding 16 bit Unicode (ucs2), message yang didahului "0001" dengan class 0 berupa blinking flash SMS.

#### 6. SCTS (Service Center Time Stamp)

Format SCTS menggunakan format 2 digit per satuan waktu dimulai dari Tahun, Bulan, Tanggal, Jam, Menit, Detik dan Zona waktu.

Dalam 2 digit tersebut dilakukan pemindahan posisi (swapped nibble presentation) pada format PDU.

Contoh :

Pada format PDU SCTS bernilai 30503161403482, maka

pemisahan tiap 2 digit untuk pembagian satuan waktu adalah 30 50 31 61 40 34 82

Dimana dilakukan swapped nibble presentation :

Tahun = 30 menjadi 03 yaitu tahun 2003

Bulan = 50 menjadi 05 yaitu bulan Mei

Tanggal = 31 menjadi 13

Jam = 61 menjadi 16

Menit = 40 menjadi 04

Detik = 34 menjadi 43

Zona Waktu = 82 menjadi 28 unit

=>  $15 \times 28 = 420$  menit (1 unit = 15 menit)

=>  $420 \text{ menit} / 60 = 7$  jam

=> GMT+07.00 merupakan zona jam di negara Indonesia, Thailand, Vietnam.

Penerimaan SMS terjadi pada tanggal 13 Mei 2003 jam 16:04:43 waktu Indonesia (GMT+07:00)

### **7. UDL (User Data Length)**

UDL menunjukkan panjang User Data.

Contoh :

bila data yang diterima adalah "terima sms", maka panjang data adalah 10 desimal pada format PDU menjadi 0A heksadesimal.

### **8. User Data**

Data berupa isi sms yang diterima. Pada PDU bernilai heksadesimal diubah ke biner (berbasis 8 digit / oktet) dimana penulisan tiap biner oktet berikutnya di sebelah kiri biner oktet sebelumnya lalu tiap karakter diambil 7 digit (septet) dari kanan ke kiri pada konversi biner.

Berikut tabel karakter dari 7 binari digit (Alphabet Standard GSM 03.38).

				b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1		0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	i	P	z	p
0	0	0	1	1	£	_	!	1	A	Q	a	q
0	0	1	0	2	\$	Φ	"	2	B	R	b	r
0	0	1	1	3	¥	Γ	#	3	C	S	c	s
0	1	0	0	4	è	Λ	π	4	D	T	d	t
0	1	0	1	5	é	Ω	%	5	E	U	e	u
0	1	1	0	6	ù	π	&	6	F	V	f	v
0	1	1	1	7	ì	ψ	'	7	G	W	g	w
1	0	0	0	8	ò	Σ	(	8	H	X	h	x
1	0	0	1	9	ç	θ	)	9	I	Y	i	y
1	0	1	0	10	LF	≡	*	:	J	Z	j	z
1	0	1	1	11	ø	η	+	;	K	Ä	k	ä
1	1	0	0	12	ø	€	,	<	L	ö	l	ö
1	1	0	1	13	CR	æ	-	=	M	ñ	m	ñ
1	1	1	0	14	A	β	.	>	N	ü	n	ü
1	1	1	1	15	á	É	/	?	O	š	o	š

Tanda b1,b2,b3,b4 pada tabel merupakan perwakilan biner septet(7 digit) dari bit ke 1,2,3,4 dimana nilai tiap bit tertera pada baris-baris di bawahnya.

Tanda b5,b6,b7 pada tabel merupakan perwakilan biner septet (7 digit) dari bit ke 5,6,7 dimana nilai tiap bit tertera pada kolom-kolom di sebelah kanannya.

Karakter yang dihasilkan dari oktet adalah gabungan dari baris bit 1,2,3,4 dengan kolom bit 5,6,7 yang membentuk koordinat mengacu pada 1 kolom.

Contoh :

UD pada format PDU adalah "F4B23CDD0E83E6ED39" berupa heksadesimal.

Nilai - nilai tiap 2 karakter heksadesimal diubah ke biner 8 digit dimulai dari yang paling kiri menjadi biner yang paling kanan.



2 karakter heksadesimal itu adalah F4h B2h 3Ch DDh 0Eh 83h E6h EDh 39h

Peletakkannya ke biner menjadi 39h EDh E6h 83h 0Eh DDh 3Ch B2h F4h

Untuk lebih mudah diubah ke format desimal menjadi 57 237 230 131 14 221 60 178 224

Lalu diubah ke biner 8 digit :

57 = 00111001

237 = 11101101

230 = 11100110

131 = 10000011

14 = 00001110

221 = 11011101

60 = 00111100

178 = 10110010

244 = 11110100

Penulisannya menjadi :

00111001 11101101 11100110 10000011 00001110 11011101 00111100 10110010 11110100

Dilakukan penggabungan biner lalu dipisah tiap - tiap 7 digit dimulai dari yang paling kanan ke kiri menjadi :

00 1110011 1101101 1110011 0100000 1100001 1101101 1101001 1110010 1100101 1110100

Dimana "00" tidak akan di konversi karena biner tidak memiliki 7 digit.

Lalu mengubah tiap 7 digit menjadi 1 karakter berdasarkan tabel, menjadi:

"s" "m" "s" " " "a" "m" "i" "r" "e" "t" = "sms amiret"

Kemudian data dibaca dari kanan ke kiri menjadi "terima sms"

UD dapat ditulis sebanyak 140 oktet yaitu  $140 \times 8 / 7 = 160$  Karakter sms.

## Review

Berdasarkan dari penjelasan format SMS deliver PDU diatas, maka bila PDU bernilai

"059126181642240C912618859804820000305031614034820A

F4B23CDD0E83E6ED39" adalah:

05 91 26181642 - 24 - 0C 91 261885980482 - 00 - 00 - 30503161403482 - 0A -

F4B23CDD0E83E6ED39

yaitu :

Melalui service center Satelindo berformat Internasional dimana nomor handphone pengirim

+628158894028 dengan Id protokol = "Standad-Text-SMS" tanpa Message Class melakukan pengiriman sms pada tanggal 13 Mei 2003 jam 16:04:43 waktu Indonesia sebanyak 10 karakter dengan isi sms adalah "terima sms".

## SMS Submit PDU

### SMS SUBmit PDU (Mobile Originated)

SMS Deliver PDU ialah terminal mengirim pesan ke SMSC (lihat topik Teknologi SMS) dalam format PDU.

Skema format SMS Deliver PDU adalah :

| SCA | PDU Type | MR | DA | PID | DCS | VP | UDL | UD |

#### 1. SCA (Service Center Address)

SCA memiliki 3 komponen utama, yaitu:

a. Len (Length) yaitu panjang informasi SMSC dalam oktet

b. Type of Number sebagai type address dari SMSC

nilai 81h untuk format lokal

nilai 91h untuk format internasional

c. BSC Digits berupa nomor SMSC. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal.

Contoh:

SCA bernilai 00. maka,

Len = 00,

Type of Number = 0 bila Len = 0

BSC Digits = 0 bila Len = 0

Hal ini terjadi dikarena SMSC yang digunakan adalah SMSC yang terdapat pada SIM Card berdasarkan perintah "AT+CSCA"

Berikut contoh penulisan nomor SMSC untuk masing-masing operator

===Operator===	Nomor SMSC	==== Format PDU =====
Satelindo---->	---62816124	05 91 26 18 16 42 -- --
Excelcomindo->	62818445009	07 91 26 18 48 54 00 F9
Telkomsel---->	-6281100000	06 91 26 18 01 00 00 --
IM3----->	62855000000	05 91 26 58 05 00 00 F0

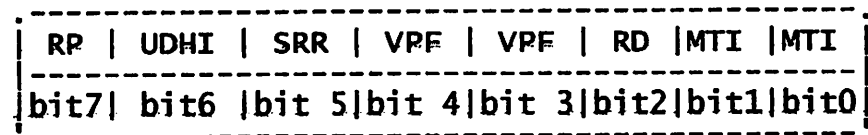
Pada IM3 Nomor SMSC 6285500000 bernilai ganjil, maka mendapat tambahan F menjadi 6285500000F

Setelah itu dilakukan perpindahan tempat setiap 2 digit untuk mengubah Nomor SMSC ke Format PDU (swapped nibble presentation) dari 62 85 50 00 00 0F menjadi 26 58 05 00 00 F0

Begitu juga halnya pada Nomor SMSC Excelcomindo.

## 2. PDU Type (Protocol Data Unit Type)

Format PDU Type adalah



Nilai default dari PDU untuk SMS-Deliver adalah 04 heksadesimal, akan tetapi pada PDU Type diubah ke biner menjadi 00000100, dimana nilai "1" berada pada biner ke 2 yaitu MMS.

Contoh:

PDU Type bernilai 11 heksadesimal = 00010001 biner.

Nilai "1" berada pada biner 1 (MTI) dan 4(VPF).

## 3. (MR) Message Reference

Adalah referensi pesan

## 4. (DA) Destination Address

DA memiliki 3 komponen utama, yaitu:

a. Len (length) yaitu panjang nomor Destination Address

Format DA pada PDU dengan default = 14 dan bertype lokal

Format DA pada PDU = 0C bila nomor OA berawalan "+" dan bertype internasional

b. Type of Number sebagai type address dari Destination Address

nilai 81h untuk format lokal

nilai 91h untuk format internasional

c. BSC Digits berupa nomor DA. Jika panjangnya ganjil, pada akhir karakter ditambahkan "F" heksadesimal.

Contoh:

Bila Nomor DA = ABCDEFGHIJKLMNOPQRST, maka

Len = 14  
 Type of Number = 81 (lokal)  
 BSC Digits = BA DC FE HG JI LK NM PO RQ TS

Bila Nomor DA = +ABCDEFGHJKLM, maka  
 Len = 0C  
 Type of Number = 91 (internasional)  
 BSC Digits = BA DC FE HG JI LK

Bila Nomor DA dalam format PDU = 0C91261885980482, Maka  
 Len = 0C  
 Type of Number = 91 (internasional)  
 BSC Digits = 26 18 85 98 04 82

Nomor DA sebenar adalah +628158894028, yaitu nomor yang mengirim SMS ke penerima. Tanda +62 merupakan asal negara dimana 62 adalah negara Indonesia, sedangkan 8158894028 merupakan nomor Handphone yang ditambah prefiks "0" (nol) menjadi 08158894028.

### 5. PID (Protocol Identifier)

Nilai default dari PID ialah 00 = "Standard-Text-SMS"

### 6. DCS (Data Coding Scheme)

DCS dapat menentukan message class

N11 Hexa	- character- -- Coding --	Message Class
00	default(7 bit)	no class
F0	default(7 bit)	class 0(immediate display)
F1	default(7 bit)	class 1(Mobile Equipment-specific)
F2	default(7 bit)	class 2(SIM specific message)
F3	default(7 bit)	class 3(Terminate Equipment-specific)
F4	— 8 bit —	class 0(immediate display)
F5	— 8 bit —	class 1(Mobile Equipment-specific)
F6	— 8 bit —	class 2(SIM specific message)
F7	— 8 bit —	class 3(Terminate Equipment-specific)

Hal yang perlu diperhatikan, pada beberapa handphone message dengan class 0 dengan enkoding 7 bit berupa flash SMS

Sedangkan dengan enkoding 16 bit Unicode (ucs2), message yang didahului "0001" dengan class 0 berupa blinking flash SMS.

## 7. VP (Validity Period)

Berupa validasi batas pengiriman pesan. Bila pada batas waktu yang telah ditentukan pesan tidak juga sampai ke tujuan, maka pesan akan di usangkan oleh SMSC

Nilai VP	Nilai validitas Periode
0 - 143	$(VP+1)*5$ menit (interval 5 menit hingga 12 jam)
144 - 167	12 jam + $((TP-VP)*30)$ menit
168 - 196	$(VP - 166) * 1$ hari
197 - 255	$(VP - 192) * 1$ minggu

Contoh :

Pada format PDU VP bernilai AA heksadesimal, maka AA heksadesimal = 170.  $170 - 166 = 4$  hari

## 8. UDL (User Data Length)

UDL menunjukkan panjang User Data.

Contoh :

bila data yang diterima adalah "kirim sms", maka panjang data adalah 9 desimal pada format PDU menjadi 09 heksadesimal.

## 9. User Data

Data berupa isi sms yang diterima. Pada PDU bernilai heksadesimal diubah ke biner (berbasis 8 digit / oktet) dimana penulisan tiap biner oktet berikutnya di sebelah kiri biner oktet sebelumnya lalu tiap karakter diambil 7 digit (septet) dari kanan ke kiri pada konversi biner.

Berikut tabel karakter dari 7 binari digit (Alphabet Standard GSM 03.38).

					b7	0	0	0	0	1	1	1	1
					b6	0	0	1	1	0	0	1	1
					b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1			0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	i	P	z	p	
0	0	0	1	1	£	_	!	1	A	Q	a	q	
0	0	1	0	2	\$	Φ	"	2	B	R	b	r	
0	0	1	1	3	¥	Γ	#	3	C	S	c	s	
0	1	0	0	4	è	Λ	π	4	D	T	d	t	
0	1	0	1	5	é	Ω	%	5	E	U	e	u	
0	1	1	0	6	ù	π	&	6	F	V	f	v	
0	1	1	1	7	ì	ψ	'	7	G	W	g	w	
1	0	0	0	8	ò	Σ	(	8	H	X	h	x	
1	0	0	1	9	ç	θ	)	9	I	Y	i	y	
1	0	1	0	10	LF	≡	*	:	J	Z	j	z	
1	0	1	1	11	ø	η	+	;	K	Ä	k	ä	
1	1	0	0	12	ø	Æ	,	<	L	Ö	l	ö	
1	1	0	1	13	CR	æ	-	=	M	Ñ	m	ñ	
1	1	1	0	14	À	β	.	>	N	Ü	n	ü	
1	1	1	1	15	á	É	/	?	O	Š	o	š	

Tanda b1,b2,b3,b4 pada tabel merupakan perwakilan biner septet(7 digit) dari bit ke 1,2,3,4 dimana nilai tiap bit tertera pada baris-baris di bawahnya.

Tanda b5,b6,b7 pada tabel merupakan perwakilan biner septet (7 digit) dari bit ke 5,6,7 dimana nilai tiap bit tertera pada kolom-kolom di sebelah kanannya.

Karakter yang dihasilkan dari oktet adalah gabungan dari baris bit 1,2,3,4 dengan kolom bit 5,6,7 yang membentuk koordinat mengacu pada 1 kolom.

Contoh :

UD pada format PDU adalah "EBB43CDD06CDDDB73" berupa heksadesimal.

Nilai - nilai tiap 2 karakter heksadesimal diubah ke biner 8 digit dimulai dari yang paling kiri menjadi biner yang paling kanan.

2 karakter heksadesimal itu adalah EBh B4h 3Ch DDh 06h CDh DBh 73h

Peletakkannya ke biner menjadi 73h DBh CDh 06h DDh 3Ch B4h EBh

Untuk lebih mudah diubah ke format desimal menjadi 115 219 205 006 221 060 180 235

Lalu diubah ke biner 8 digit :

115= 01110011

219= 11011011

205= 11001101

006= 00000110

221= 11011101

060= 00111100

180= 10110100

235= 11101011

Penulisannya menjadi :

01110011 11011011 11001101 00000110 11011101 00111100 10110100 11101011

Dilakukan penggabungan biner lalu dipisah tiap - tiap 7 digit dimulai dari yang paling kanan ke kiri menjadi :

1110011 1101101 1110011 0100000 1101101 1101001 1110010 1101001 1101011

Lalu mengubah tiap 7 digit menjadi 1 karakter berdasarkan tabel, menjadi:

"s" "m" "s" " " "m" "i" "r" "i" "k" = "sms mirik"

Kemudian data dibaca dari kanan ke kiri menjadi "kirim sms"

UD dapat ditulis sebanyak 140 oktet yaitu  $140 \times 8/7 = 160$  Karakter sms.

## Review

Berdasarkan dari penjelasan format SMS deliver PDU diatas, maka bila PDU bernilai

"0011000C912618859804820000AA09EBB43CDD06CDDDB73"

adalah:

00 - 11 - 00 - 0C 91 261885980482 - 00 - 00 - AA - 09 - EBB43CDD06CDDDB73

yaitu :

Pengiriman sms dengan menggunakan SMSC yang terdapat pada SIM Card ke nomor handphone +628158894028 dengan Id protokol = "Standad-Text-SMS" tanpa Message Class bervalidasi waktu 4 hari sebanyak 9 karakter dengan isi sms adalah "kirim sms".

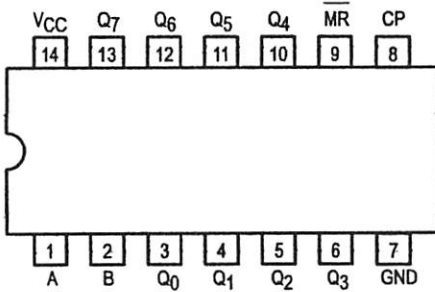


# SERIAL-IN PARALLEL-OUT SHIFT REGISTER

The SN54/74LS164 is a high speed 8-Bit Serial-In Parallel-Out Shift Register. Serial data is entered through a 2-Input AND gate synchronous with the LOW to HIGH transition of the clock. The device features an asynchronous Master Reset which clears the register setting all outputs LOW independent of the clock. It utilizes the Schottky diode clamped process to achieve high speeds and is fully compatible with all Motorola TTL products.

- Typical Shift Frequency of 35 MHz
- Asynchronous Master Reset
- Gated Serial Data Input
- Fully Synchronous Data Transfers
- Input Clamp Diodes Limit High Speed Termination Effects
- ESD > 3500 Volts

CONNECTION DIAGRAM DIP (TOP VIEW)



NOTE:  
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

**PIN NAMES**

A, B	Data Inputs
CP	Clock (Active HIGH Going Edge) Input
MR	Master Reset (Active LOW) Input
Q <sub>0</sub> -Q <sub>7</sub>	Outputs (Note b)

**LOADING (Note a)**

	HIGH	LOW
A, B	0.5 U.L.	0.25 U.L.
CP	0.5 U.L.	0.25 U.L.
MR	0.5 U.L.	0.25 U.L.
Q <sub>0</sub> -Q <sub>7</sub>	10 U.L.	5 (2.5) U.L.

NOTES:  
a) 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.  
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

## SN54/74LS164

### SERIAL-IN PARALLEL-OUT SHIFT REGISTER LOW POWER SCHOTTKY



J SUFFIX  
CERAMIC  
CASE 632-08



N SUFFIX  
PLASTIC  
CASE 646-06

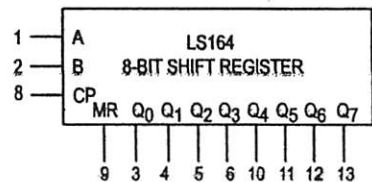


D SUFFIX  
SOIC  
CASE 751A-02

**ORDERING INFORMATION**

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

**LOGIC SYMBOL**

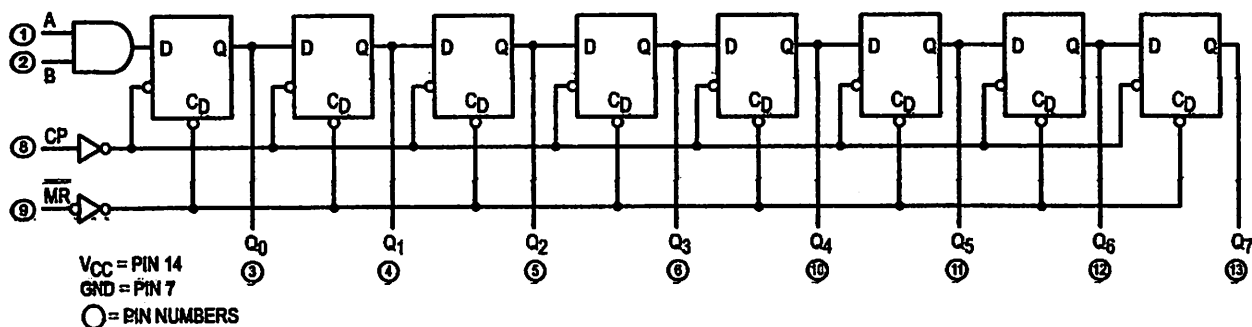


VCC = PIN 14  
GND = PIN 7



# SN54/74LS164

## LOGIC DIAGRAM



## FUNCTIONAL DESCRIPTION

The LS164 is an edge-triggered 8-bit shift register with serial data entry and an output from each of the eight stages. Data is entered serially through one of two inputs (A or B); either of these inputs can be used as an active HIGH Enable for data entry through the other input. An unused input must be tied HIGH, or both inputs connected together.

Each LOW-to-HIGH transition on the Clock (CP) input shifts data one place to the right and enters into Q<sub>0</sub> the logical AND of the two data inputs (A•B) that existed before the rising clock edge. A LOW level on the Master Reset (MR) input overrides all other inputs and clears the register asynchronously, forcing all Q outputs LOW.

## MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	MR	A	B	Q <sub>0</sub>	Q <sub>1</sub> -Q <sub>7</sub>
Reset (Clear)	L	X	X	L	L-L
Shift	H	l	l	L	q <sub>0</sub> -q <sub>6</sub>
	H	l	h	L	q <sub>0</sub> -q <sub>6</sub>
	H	h	l	L	q <sub>0</sub> -q <sub>6</sub>
	H	h	h	H	q <sub>0</sub> -q <sub>6</sub>

L (l) = LOW Voltage Levels

H (h) = HIGH Voltage Levels

X = Don't Care

q<sub>n</sub> = Lower case letters indicate the state of the referenced input or output one set-up time prior to the LOW to HIGH clock transition.

## GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V <sub>CC</sub>	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T <sub>A</sub>	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I <sub>OH</sub>	Output Current — High	54, 74			-0.4	mA
I <sub>OL</sub>	Output Current — Low	54			4.0	mA
		74			8.0	

# SN54/74LS164

## DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
$V_{IL}$	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
$V_{IK}$	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{IN} = -18 \text{ mA}$	
$V_{OH}$	Output HIGH Voltage	54	2.5	3.5	V	$V_{CC} = \text{MIN}$ ; $I_{O\overline{H}} = \text{MAX}$ ; $V_{IN} = V_{IH}$ or $V_{IL}$ per Truth Table	
		74	2.7	3.5			
$V_{OL}$	Output LOW Voltage	54, 74		0.25	0.4	V	$I_{OL} = 4.0 \text{ mA}$
		74		0.35	0.5	V	$I_{OL} = 8.0 \text{ mA}$
$I_{IH}$	Input HIGH Current			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{IN} = 2.7 \text{ V}$	
				0.1	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 7.0 \text{ V}$	
$I_{IL}$	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 0.4 \text{ V}$	
$I_{OS}$	Short Circuit Current (Note 1)	-20		-100	mA	$V_{CC} = \text{MAX}$	
$I_{CC}$	Power Supply Current			27	mA	$V_{CC} = \text{MAX}$	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

## AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$f_{MAX}$	Maximum Clock Frequency	25	36		MHz	$V_{CC} = 5.0 \text{ V}$ $C_L = 15 \text{ pF}$
$t_{PHL}$	Propagation Delay MR to Output Q		24	36	ns	
$t_{PLH}$ $t_{PHL}$	Propagation Delay Clock to Output Q		17 21	27 32	ns	

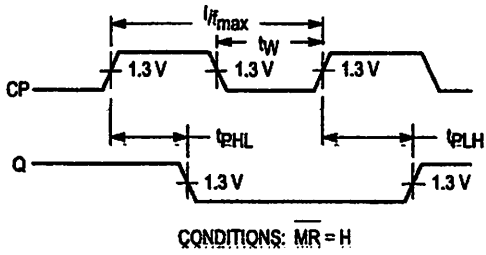
## AC SETUP REQUIREMENTS ( $T_A = 25^\circ\text{C}$ )

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
$t_W$	CP, MR Pulse Width	20			ns	$V_{CC} = 5.0 \text{ V}$
$t_s$	Data Setup Time	15			ns	
$t_h$	Data Hold Time	5.0			ns	
$t_{rec}$	MR to Clock Recovery Time	20			ns	

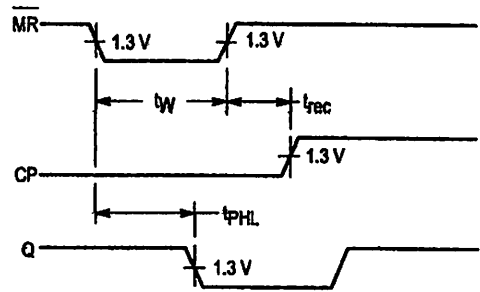
# SN54/74LS164

## AC WAVEFORMS

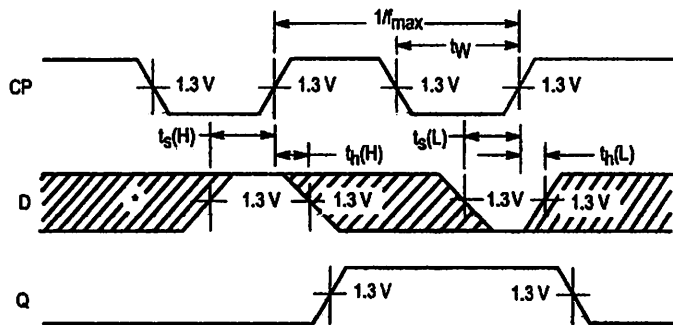
\*The shaded areas indicate when the input is permitted to change for predictable output performance.



**Figure 1. Clock to Output Delays and Clock Pulse Width**



**Figure 2. Master Reset Pulse Width, Master Reset to Output Delay and Master Reset to Clock Recovery Time**



**Figure 3. Data Setup and Hold Times**

# MOSPEC

## COMPLEMENTARY SILICON PLASTIC POWER TRANSISTORS

... designed for use in general purpose power amplifier and switching applications.

### FEATURES:

\* Collector-Emitter Sustaining Voltage -

$V_{CE(sus)}$  = 40V(Min)- TIP41, TIP42  
 60V(Min)- TIP41A, TIP42A  
 80V(Min)- TIP41B, TIP42B  
 100V(Min)- TIP41C, TIP42C

\* Collector-Emitter Saturation Voltage-  $V_{CE(sat)}$  = 1.5V(Max) @  $I_C = 6.0A$

\* Current Gain-Bandwidth Product  $f_T = 3.0$  MHz (Min) @  $I_C = 500mA$

NPN	PNP
TIP41	TIP42
TIP41A	TIP42A
TIP41B	TIP42B
TIP41C	TIP42C

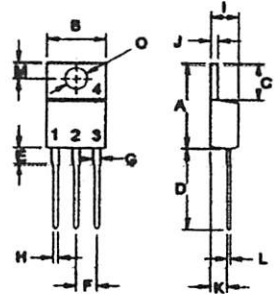
6 AMPERE  
COMPLEMENTARY SILICON  
POWER TRANSISTORS  
40 -100 VOLTS  
65 WATTS

### MAXIMUM RATINGS

Characteristic	Symbol	TIP41 TIP42	TIP41A TIP42A	TIP41B TIP42B	TIP41C TIP42C	Unit
Collector-Emitter Voltage	$V_{CEO}$	40	60	80	100	V
Collector-Base Voltage	$V_{CBO}$	40	60	80	100	V
Emitter-Base Voltage	$V_{EBO}$	5				V
Collector Current - Continuous - Peak	$I_C$	6 10				A
Base Current	$I_B$	2				A
Total Power Dissipation @ $T_C = 25^\circ C$ Derate above $25^\circ C$	$P_D$	65 0.52				W W/°C
Operating and Storage Junction Temperature Range	$T_J, T_{STG}$	-65 to +150				°C



TO-220

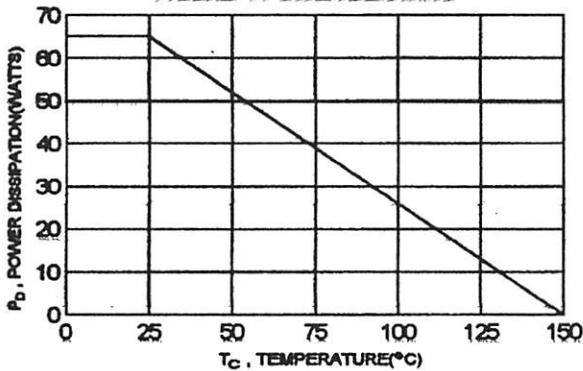


PIN 1.BASE  
2.COLLECTOR  
3.EMITTER  
4.COLLECTOR(CASE)

### THERMAL CHARACTERISTICS

Characteristic	Symbol	Max	Unit
Thermal Resistance Junction to Case	$R_{\theta jc}$	1.92	°C/W

FIGURE -1 POWER DERATING



DIM	MILLIMETERS	
	MIN	MAX
A	14.68	15.31
B	9.78	10.42
C	5.01	6.52
D	13.08	14.62
E	3.57	4.07
F	2.42	3.68
G	1.12	1.36
H	0.72	0.98
I	4.22	4.98
J	1.14	1.38
K	2.20	2.97
L	0.33	0.55
M	2.48	2.88
O	3.70	3.90

**TIP41, TIP41A, TIP41B, TIP41C NPN / TIP42, TIP42A, TIP42B, TIP42C PNP**

**ELECTRICAL CHARACTERISTICS (  $T_c = 25^\circ\text{C}$  unless otherwise noted )**

Characteristic	Symbol	Min	Max	Unit
----------------	--------	-----	-----	------

**OFF CHARACTERISTICS**

Collector-Emitter Sustaining Voltage(1) ( $I_c = 30\text{ mA}$ , $I_B = 0$ )	TIP41, TIP42 TIP41A, TIP42A TIP41B, TIP42B TIP41C, TIP42C	$V_{CE0(\text{max})}$	40 60 80 100	V
Collector Cutoff Current ( $V_{CE} = 30\text{ V}$ , $I_B = 0$ ) ( $V_{CE} = 60\text{ V}$ , $I_B = 0$ )	TIP41, TIP42, TIP41A, TIP42A TIP41B, TIP42B, TIP41C, TIP42C	$I_{CBO}$	0.7 0.7	mA
Collector Cutoff Current ( $V_{CE} = 40\text{ V}$ , $V_{EB} = 0$ ) ( $V_{CE} = 60\text{ V}$ , $V_{EB} = 0$ ) ( $V_{CE} = 80\text{ V}$ , $V_{EB} = 0$ ) ( $V_{CE} = 100\text{ V}$ , $V_{EB} = 0$ )	TIP41, TIP42 TIP41A, TIP42A TIP41B, TIP42B TIP41C, TIP42C	$I_{CES}$	0.4 0.4 0.4 0.4	mA
Emitter Cutoff Current ( $V_{EB} = 5.0\text{ V}$ , $I_C = 0$ )		$I_{EBO}$	1.0	mA

**ON CHARACTERISTICS (1)**

DC Current Gain ( $I_C = 0.3\text{ A}$ , $V_{CE} = 4.0\text{ V}$ ) ( $I_C = 3.0\text{ A}$ , $V_{CE} = 4.0\text{ V}$ )	$h_{FE}$	30 15	75	
Collector-Emitter Saturation Voltage ( $I_C = 6.0\text{ A}$ , $I_B = 600\text{ mA}$ )	$V_{CE(\text{sat})}$		1.5	V
Base-Emitter On Voltage ( $I_C = 6.0\text{ A}$ , $V_{CE} = 4.0\text{ V}$ )	$V_{BE(\text{on})}$		2.0	V

**DYNAMIC CHARACTERISTICS**

Current Gain - Bandwidth Product (2) ( $I_C = 500\text{ mA}$ , $V_{CE} = 10\text{ V}$ , $f_{\text{TEST}} = 1\text{ MHz}$ )	$f_T$	3.0		MHz
Small Signal Current Gain ( $I_C = 500\text{ mA}$ , $V_{CE} = 10\text{ V}$ , $f = 1\text{ kHz}$ )	$h_{fe}$	20		

(1) Pulse Test: Pulse width  $\leq 300\ \mu\text{s}$ , Duty Cycle  $\leq 2.0\%$

(2)  $f_T = |h_{fe}| \cdot f_{\text{TEST}}$

TIP41, TIP41A, TIP41B, TIP41C NPN / TIP42, TIP42A, TIP42B, TIP42C PNP

FIGURE 2 - SWITCHING TIME TEST CIRCUIT

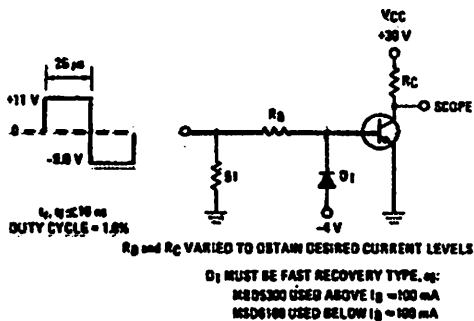


FIG-4 DC CURRENT GAIN

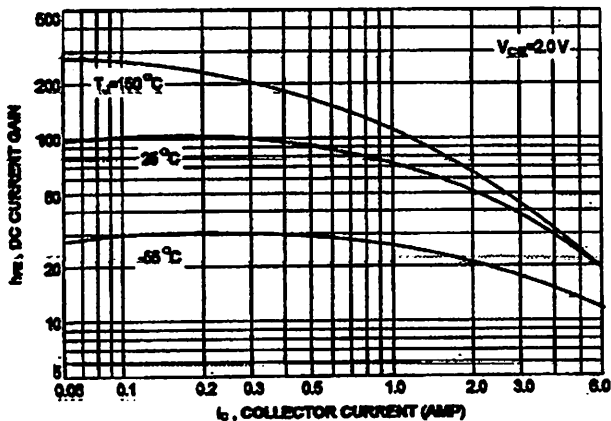


FIG-3 TURN-ON TIME

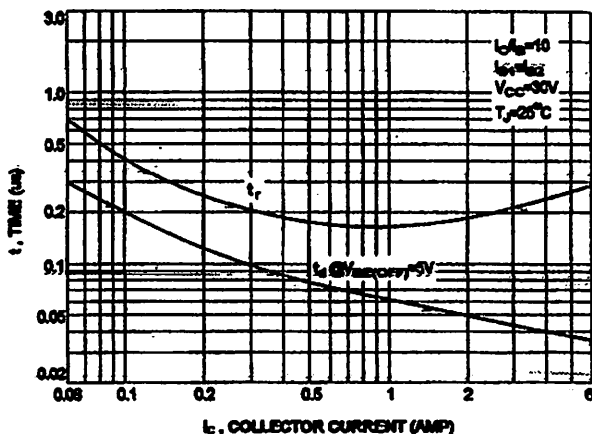


FIG-5 TURN-OFF TIME

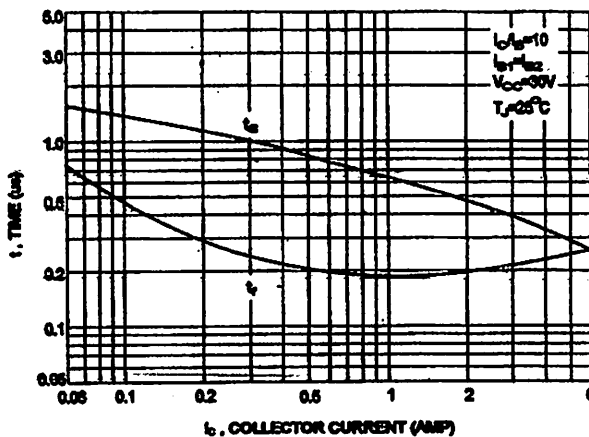
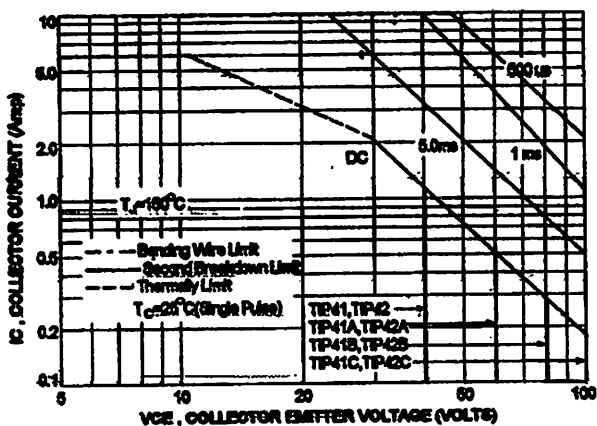


FIG-6 ACTIVE REGION SAFE OPERATING AREA



There are two limitation on the power handling ability of a transistor: average junction temperature and second breakdown safe operating area curves indicate  $I_C - V_{CE}$  limits of the transistor that must be observed for reliable operation i.e., the transistor must not be subjected to greater dissipation than curves indicate.

The data of FIG-6 curve is base on  $T_{J,avg} = 150^\circ C$ ;  $T_C$  is variable depending on power level. second breakdown pulse limits are valid for duty cycles to 10% provided  $T_{J,avg} \leq 150^\circ C$ . At high case temperatures, thermal limitation will reduce the power that can be handled to values less than the limitations imposed by second breakdown.

TIP41, TIP41A, TIP41B, TIP41C NPN / TIP42, TIP42A, TIP42B, TIP42C PNP

FIG-7 COLLECTOR SATURATION REGION

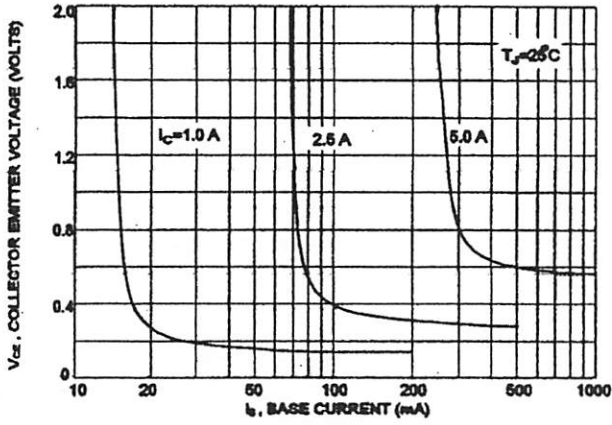


FIG-8 CAPACITANCES

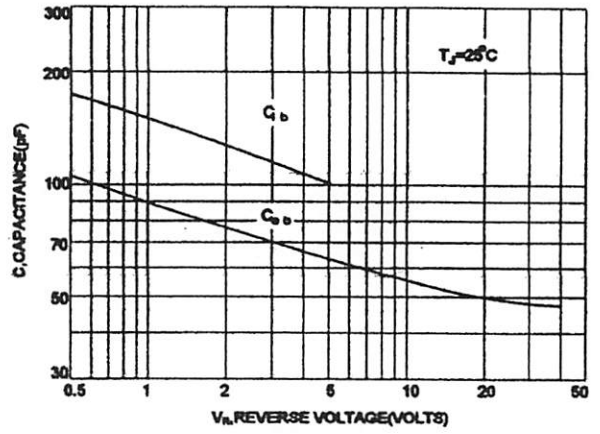


FIG-9 "ON" VOLTAGE

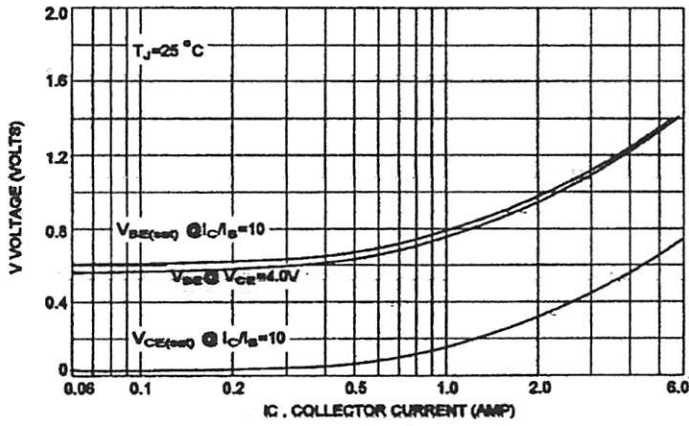
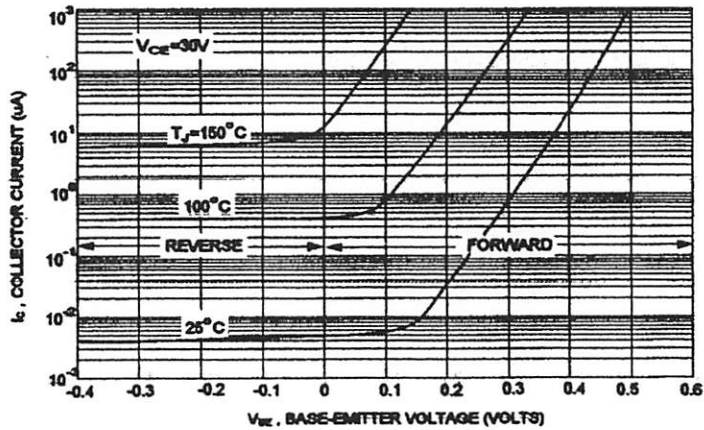


FIG-10 COLLECTOR CUT-OFF REGION



**PNP TRANSISTOR**

**A733**

**-100mA**



- AF OUTPUT AMPLIFIER

**MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS ( Ta=25°C )**

PARAMETERS	SYMBOL	MIN	TYP	MAX	UNIT	CONDITION
Collector-Emitter Breakdown Voltage	BV <sub>ceo</sub>	-50			V	I <sub>c</sub> =1mA
Collector-Base Breakdown Voltage	BV <sub>cb0</sub>	-60			V	I <sub>c</sub> =5μA
Emitter-Base Breakdown Voltage	BV <sub>eb0</sub>	-5			V	I <sub>e</sub> =50μ A
Collector-Base Leakage	I <sub>cbo</sub>			-0.1	μA	V <sub>cb</sub> =-60V
Emitter-Base Leakage	I <sub>ebo</sub>			-0.1	μA	V <sub>eb</sub> =-5V
Collector-Emitter Saturation Voltage	V <sub>ce ( sat )</sub>		-0.18	-0.3	V	I <sub>c</sub> =-100mA, I <sub>b</sub> =-10mA
DC Current Gain	H <sub>fe</sub>	90	200	600		V <sub>ce</sub> =-6.0V, I <sub>c</sub> =-1.0mA
Collector Current	I <sub>c</sub>		-100		mA	
Current Gain Bandwidth	f <sub>r</sub>	100	180		MHz	V <sub>ce</sub> =-6V, I <sub>e</sub> =10mA
Output Capacitance	C <sub>ob</sub>		4.5	6.0	pF	V <sub>cb</sub> =-10V, I <sub>e</sub> =0, f=1MHz
Power Dissipation	P <sub>c</sub>			0.25	W	
Junction Temperature	T <sub>j</sub>			125	°C	
Storage Temperature	T <sub>stg</sub>	-55		125	°C	

**Classification of Hfe**

Rank	R	Q	P	K
Range	90-180	135-270	200-400	300-600



**STANSON TECHNOLOGY**

120 Bentley Square, Mountain View, Ca 94040 USA

TEL: (650) 9389294 FAX: (650) 9389295





**M1632**

**LIQUID CRYSTAL DISPLAY MODULE**

**USER'S MANUAL**

## **1. GENERAL**

### **1.1 General**

The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view 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 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

## 2. OPERATING INSTRUCTIONS

### 2.1 Terminal Functions

**Table 1 Terminal functions**

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

**Table 3 Correspondence between character codes and character patterns**

Upper 4 bits Lower 4 bits	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	1	2	3	4	5	6	7	8	9	0
xxxx0001	(2)	1	2	3	4	5	6	7	8	9	0	1	2
xxxx0010	(3)	3	4	5	6	7	8	9	0	1	2	3	4
xxxx0011	(4)	4	5	6	7	8	9	0	1	2	3	4	5
xxxx0100	(5)	5	6	7	8	9	0	1	2	3	4	5	6
xxxx0101	(6)	6	7	8	9	0	1	2	3	4	5	6	7
xxxx0110	(7)	7	8	9	0	1	2	3	4	5	6	7	8
xxxx0111	(8)	8	9	0	1	2	3	4	5	6	7	8	9
xxxx1000	(1)	9	0	1	2	3	4	5	6	7	8	9	0
xxxx1001	(2)	0	1	2	3	4	5	6	7	8	9	0	1
xxxx1010	(3)	1	2	3	4	5	6	7	8	9	0	1	2
xxxx1011	(4)	2	3	4	5	6	7	8	9	0	1	2	3
xxxx1100	(5)	3	4	5	6	7	8	9	0	1	2	3	4
xxxx1101	(6)	4	5	6	7	8	9	0	1	2	3	4	5
xxxx1110	(7)	5	6	7	8	9	0	1	2	3	4	5	6
xxxx1111	(8)	6	7	8	9	0	1	2	3	4	5	6	7

**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 ← Upper bit    Lower bit →		5 4 3 ← Upper bit	2 1 0 Lower bit →		7 6 5 4 3 2 1 0 ← Upper bit    Lower bit →			
0 0 0 0 * 0 0 0		0 0 0	0 0 0	↑ ↓	* * *	0 0	Example of character pattern (R)	
			0 0 1					
			0 1 0					
			0 1 1					
			1 0 0					
			1 0 1					
			1 1 0					
			1 1 1					
0 0 0 0 * 0 0 1		0 0 1	0 0 0	↑ ↓	* * *	0 0	Example of character pattern (ヰ)	
			0 0 1					
			0 1 0					
			0 1 1					
			1 0 0					
			1 0 1					
			1 1 0					
			1 1 1					
0 0 0 0 * 1 1 1		1 1 1	0 0 0	↑ ↓	* * *	0 0 0 0 0 0 0 0		
			0 0 1					
0 0 0 0 * 1 1 1		1 1 1	1 0 0	↑ ↓	* * *	0 0 0 0 0 0 0 0		
			1 0 1					
			1 1 0					
			1 1 1					

**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.

## 2.3 Instruction Outline

Table 5 List of instructions

Instruction	Code											Function	Execution time*
	RS	RAW	DB <sub>7</sub>	DB <sub>6</sub>	DB <sub>5</sub>	DB <sub>4</sub>	DB <sub>3</sub>	DB <sub>2</sub>	DB <sub>1</sub>	DB <sub>0</sub>			
(1) 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
(2) Cursor Home	0	0	0	0	0	0	0	0	0	1	*	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
(3) Entry Mode Set	0	0	0	0	0	0	0	0	1	UD	S	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 $\mu$ s
(4) Display ON/OFF control	0	0	0	0	0	0	1	D	C	B	*	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 $\mu$ s
(5) Cursor/Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	*	Moves cursor and shifts display without changing DD RAM contents	40 $\mu$ s
(6) Function Set	0	0	0	0	1	DL	N	F	*	*	*	Sets interface data length (DL), the number of display lines (N), and character fonts (F)	40 $\mu$ s
(7) CG RAM Address Set	0	0	0	1	A <sub>CG</sub>						Sets CG RAM address to start transmitting or receiving CG RAM data	40 $\mu$ s	
(8) DD RAM Address Set	0	0	1	A <sub>DD</sub>						Sets DD RAM address to start transmitting or receiving DD RAM data	40 $\mu$ s		
(9) BF/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 $\mu$ s		
(10) Data Write to CG RAM or DD RAM	1	0	Write Data						Writes data into DD RAM or CG RAM	40 $\mu$ s			
(11) Data Read from CG RAM or DD RAM	1	1	Read Data						Reads data from DD RAM or CG RAM	40 $\mu$ s			

\* : Invalid bit

A<sub>CG</sub>: CG RAM addressA<sub>DD</sub>: DD RAM address

AC: Address counter

UD = 1 : Increment

UD = 0 : Decrement

S = 1 : Display shift

S = 0 : No display shift

D = 1 : Display ON

D = 0 : Display OFF

C = 1 : Cursor ON

C = 0 : Cursor OFF

B = 1 : Blink ON

B = 0 : Blink OFF

S/C = 1 : Display shift

S/C = 0 : Cursor movement

R/L = 1 : Right shift

R/L = 0 : Left shift

DL = 1 : 8 bits

DL = 0 : 4 bits

N = 1 : 2 lines

N = 0 : 1 line

F = 1 : 5x10 dot matrix

F = 0 : 5x7 dot matrix

BF = 1 : Internal operation in progress

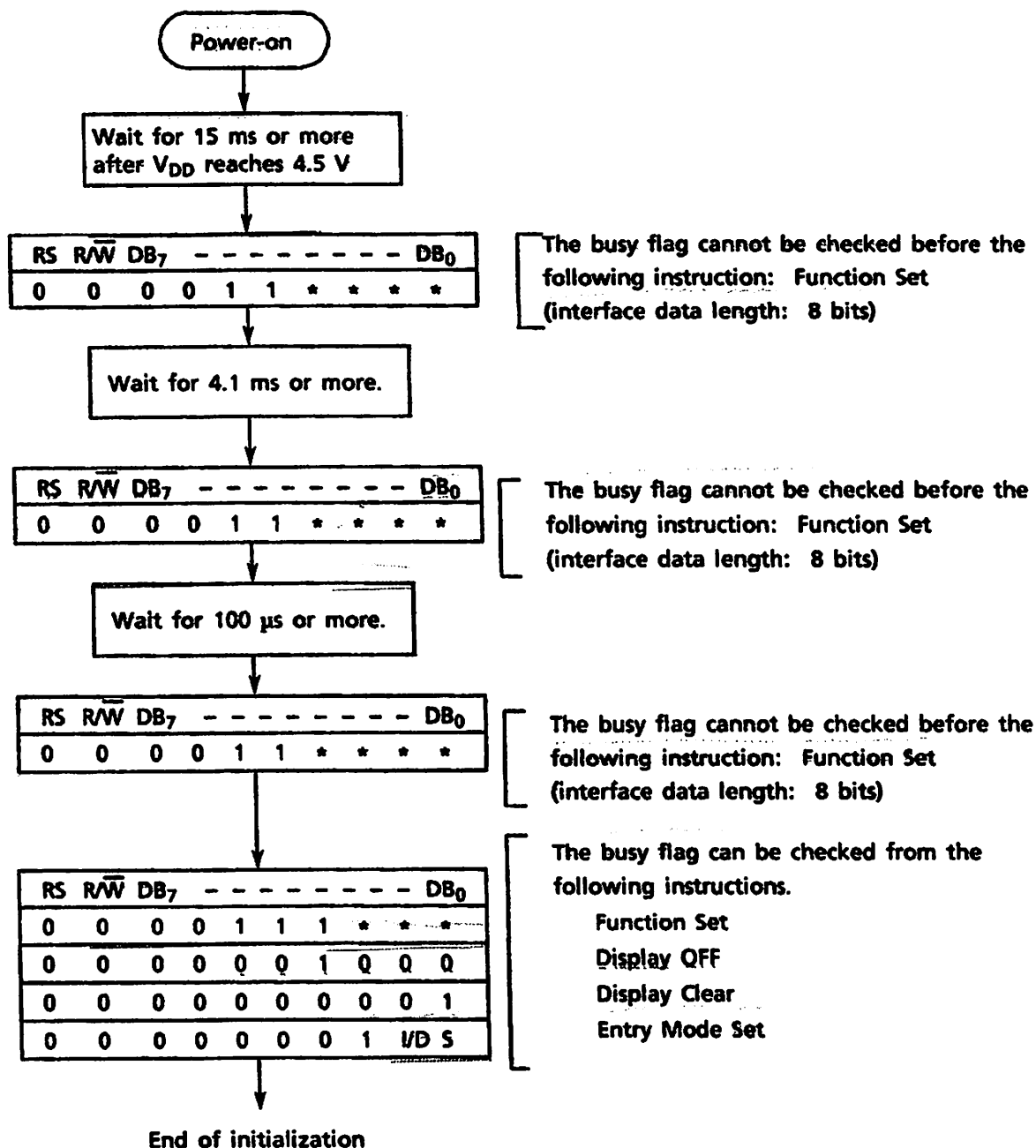
BF = 0 : Instruction can be accepted

\* An execution time indicates maximum value when  $f_{osc}$  is 250 kHz. It changes at the inverse proportion of  $f_{osc}$ .

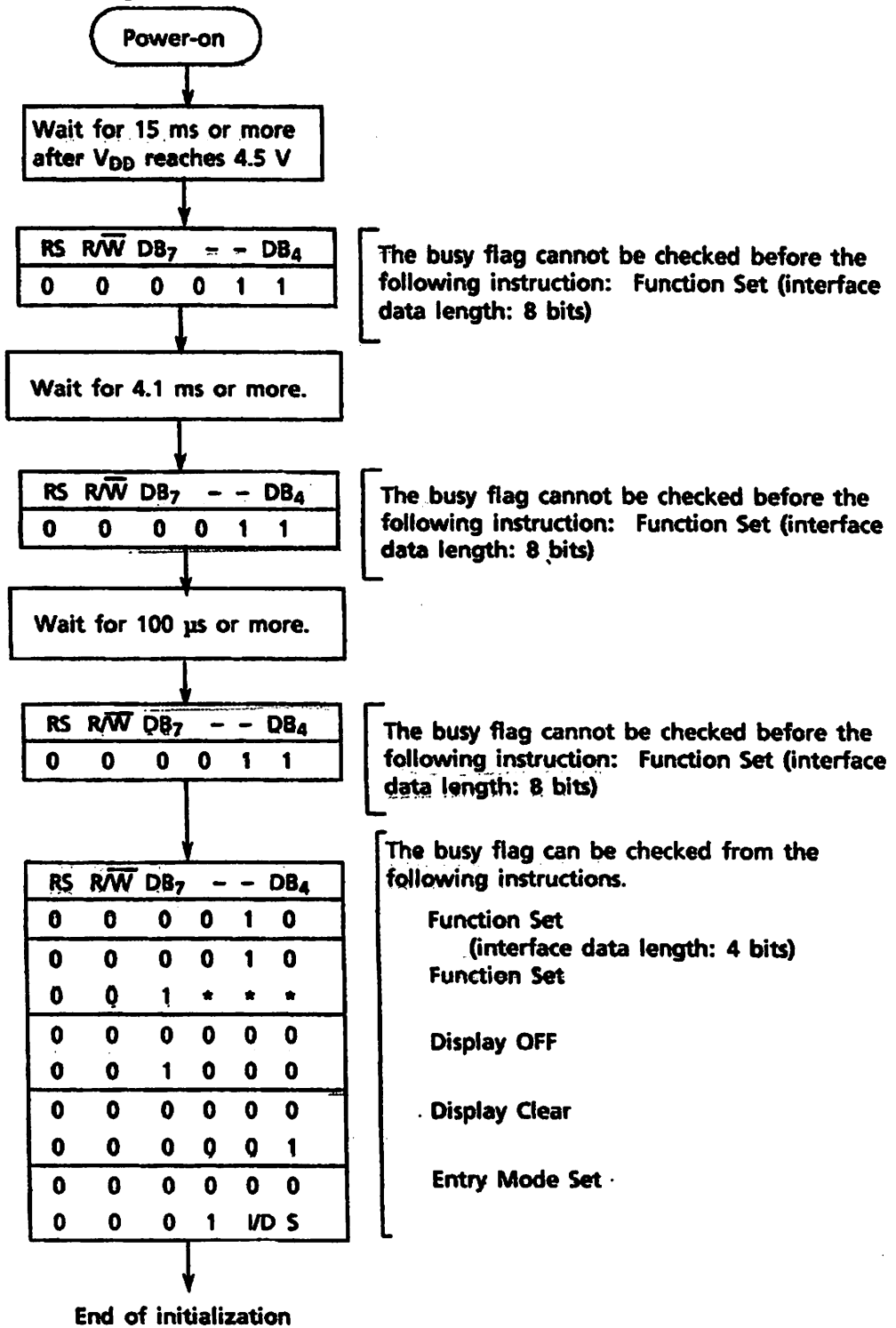
## 2. Initialization by instructions

If automatic initialization is not executed because the above power supply conditions are not satisfied, use the following instructions. Since it is unknown whether the interface data length is set to eight bits or four bits at power on, execute **Function Set** twice to set the interface data length to eight bits. And then for setting the required interface data length execute further **Function Set** instruction.

### (a) Interface data length : Eight bits

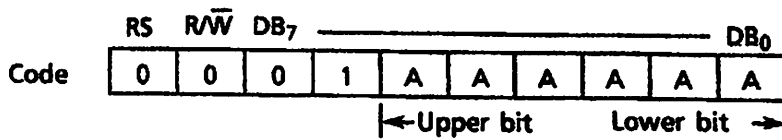


(b) Interface data length: Four bits



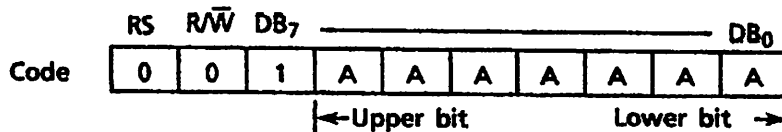


## (7) CG RAM Address Set



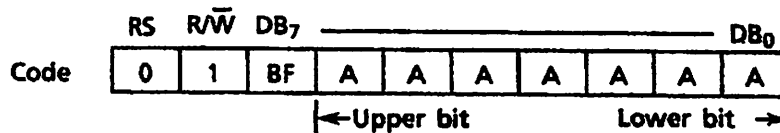
CG RAM addresses expressed as binary AAAAAA are set to the AC. Then data in CG RAM is written from or read to the MPU.

## (8) DD RAM Address Set



DD RAM addresses expressed as binary AAAAAA 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 (AAAAAA) are 00H to 27H and those for line 2 (AAAAAA) are 40H to 67H.

## (9) Busy Flag/Address Read

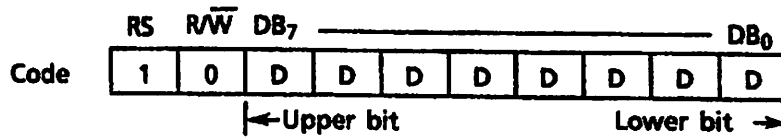


The BF signal is read out, indicating that the module is working internally because of the previous instruction.

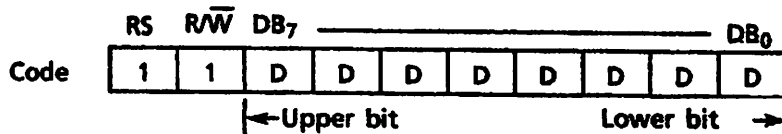
When BF = 1, the module is working internally and the next instruction cannot be accepted until the BF value becomes 0.

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 AAAAAA 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.

**(10) Data Write to CG RAM or DD RAM**

Binary eight-bit data DDDDDDDD 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.

**(11) Data Read from CG RAM or DD RAM**

Binary eight-bit data DDDDDDDD 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.




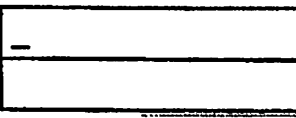
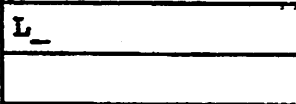
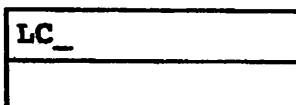
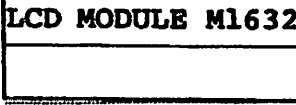
**Note :** The AC is automatically incremented or decremented by one according to the entry mode after a write instruction is executed to write data in CG RAM or DD RAM. However, the data of the RAM selected by the AC are not read out even if a read instruction is executed immediately afterwards.

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.

## 2.5 Examples of Instruction Use

### 2.5.1 Interface data length: Eight bits

No.	Instruction	Display	Operation										
1	Power-on <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	/	/	/	/	/		The built-in reset circuit initializes the module.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>									
/	/	/	/	/									
2	Function Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></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 <sub>7</sub>	—	DB <sub>0</sub>	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 <sub>7</sub>	—	DB <sub>0</sub>									
0	0	0 0 1 1 1	*	*									
3	Display ON/OFF Control <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 1 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	0	0	0 0 0 0 1 1 1 0				The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>									
0	0	0 0 0 0 1 1 1 0											
4	Entry Mode Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 0 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	0	0	0 0 0 0 0 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 <sub>7</sub>	—	DB <sub>0</sub>									
0	0	0 0 0 0 0 1 1 0											
5	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 1 1 0 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	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 <sub>7</sub>	—	DB <sub>0</sub>									
1	0	0 1 0 0 1 1 0 0											
6	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 0 0 1 1</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	1	0	0 1 0 0 0 0 1 1				C is written.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>									
1	0	0 1 0 0 0 0 1 1											
7	..... ..... .....	..... ..... .....											
8	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>1</td> <td>0</td> <td>0 0 1 1 0 0 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	1	0	0 0 1 1 0 0 1 0				2 is written in digit 16. Cursor disappears.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>									
1	0	0 0 1 1 0 0 1 0											

No.	Instruction	Display	Operation												
9	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0 0 0 0 0 0 0</td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	0	0	1	1	0 0 0 0 0 0 0	<table border="1"> <tr> <td>LCD MODULE M1632</td> </tr> <tr> <td>_</td> </tr> </table>	LCD MODULE M1632	_	The DD RAM address is set so that the cursor appears at digit 1 of line 2.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
0	0	1	1	0 0 0 0 0 0 0											
LCD MODULE M1632															
_															
10	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></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 <sub>7</sub>	—	DB <sub>0</sub>	1	0	0	0	1 1 0 0 0 1	<table border="1"> <tr> <td>LCD MODULE M1632</td> </tr> <tr> <td>1_</td> </tr> </table>	LCD MODULE M1632	1_	1 is written.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
1	0	0	0	1 1 0 0 0 1											
LCD MODULE M1632															
1_															
11	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></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 <sub>7</sub>	—	DB <sub>0</sub>	1	0	0	0	1 1 0 1 1 0	<table border="1"> <tr> <td>LCD MODULE M1632</td> </tr> <tr> <td>16_</td> </tr> </table>	LCD MODULE M1632	16_	6 is written.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
1	0	0	0	1 1 0 1 1 0											
LCD MODULE M1632															
16_															
12	⋮	⋮													
13	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></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 <sub>7</sub>	—	DB <sub>0</sub>	1	0	0	1	0 1 0 0 1 1	<table border="1"> <tr> <td>LCD MODULE M1632</td> </tr> <tr> <td>16DIGITS, 2LINES</td> </tr> </table>	LCD MODULE M1632	16DIGITS, 2LINES	S is written.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
1	0	0	1	0 1 0 0 1 1											
LCD MODULE M1632															
16DIGITS, 2LINES															
14	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0 0 0 0 0 0 0</td> </tr> </table>	RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>	0	0	1	0	0 0 0 0 0 0 0	<table border="1"> <tr> <td>LCD MODULE M1632</td> </tr> <tr> <td>16DIGITS, 2LINES</td> </tr> </table>	LCD MODULE M1632	16DIGITS, 2LINES	The cursor returns to the home position.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
0	0	1	0	0 0 0 0 0 0 0											
LCD MODULE M1632															
16DIGITS, 2LINES															
15	Display clear <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub></td> <td>—</td> <td>DB<sub>0</sub></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 <sub>7</sub>	—	DB <sub>0</sub>	0	0	0	0	0 0 0 0 0 0 0 1	<table border="1"> <tr> <td>_</td> </tr> <tr> <td> </td> </tr> </table>	_		All the display disappears and the cursor remains at the home position.
RS	R/W	DB <sub>7</sub>	—	DB <sub>0</sub>											
0	0	0	0	0 0 0 0 0 0 0 1											
_															
16	⋮	⋮													

## 2.5.2 Interface data length: Four bits

No.	Instruction	Display	Operation									
1	<p>Power-on</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB <sub>7</sub> — DB <sub>4</sub>	/	/	/		The built-in reset circuit initializes the module.			
RS	R/W	DB <sub>7</sub> — DB <sub>4</sub>										
/	/	/										
2	<p>Function Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></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 <sub>7</sub> — DB <sub>4</sub>	0	0	0 0 1 0	/	/	/		Four-bit operation mode is set. *Eight-bit operation mode is set by initialization, and the instruction is executed only once.
RS	R/W	DB <sub>7</sub> — DB <sub>4</sub>										
0	0	0 0 1 0										
/	/	/										
3	<p>Function Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></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 <sub>7</sub> — DB <sub>4</sub>	0	0	0 0 1 0	0	0	1 * * *		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.
RS	R/W	DB <sub>7</sub> — DB <sub>4</sub>										
0	0	0 0 1 0										
0	0	1 * * *										
4	<p>Display ON/OFF Control</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></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 <sub>7</sub> — DB <sub>4</sub>	0	0	0 0 0 0	0	0	1 1 1 0		The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB <sub>7</sub> — DB <sub>4</sub>										
0	0	0 0 0 0										
0	0	1 1 1 0										
5	<p>Entry Mode Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></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 <sub>7</sub> — DB <sub>4</sub>	0	0	0 0 0 0	0	0	0 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 <sub>7</sub> — DB <sub>4</sub>										
0	0	0 0 0 0										
0	0	0 1 1 0										
6	<p>Write to CG RAM or DD RAM</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB<sub>7</sub> — DB<sub>4</sub></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 <sub>7</sub> — DB <sub>4</sub>	1	0	0 1 0 0	1	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 <sub>7</sub> — DB <sub>4</sub>										
1	0	0 1 0 0										
1	0	1 1 0 0										
7												