

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



SKRIPSI

**PERANCANGAN DAN PEMBUATAN ALAT
PENCATAT PEMAKAIAN AIR
BERBASIS MIKROKONTROLER AT89S8252
MENGUNAKAN SENSOR *HALL EFFECT* (UGN 3503)
DENGAN TAMPILAN LCD**

Disusun oleh :
RUSDIL BADI'
02.17.080

MARET 2007

INSTITUT TEKNOLOGI SEPULUH
AGUSTUS SURABAYA
JURUSAN TEKNIK ELEKTRO
LABORATORIUM SISTEM ELEKTRO

ABSTRAK

PERANAN SISTEM MANAJEMEN
DALAM PENINGKATAN
KINERJA PERUSAHAAN
(STUDI KASUS: PT. BUKITI
KEMANGKARAN BERKAS (PBBK))

Disusun oleh:
NAMA: NAMA
NO. 12345

TAHUN 2003

LEMBAR PERSETUJUAN



**PERANCANGAN DAN PEMBUATAN ALAT PENCATAT
PEMAKAIAN AIR BERBASIS MIKROKONTROLER AT89S8252
MENGUNAKAN SENSOR HALL EFFECT (UGN 3503)
DENGAN TAMPILAN LCD**

SKRIPSI

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

Disusun oleh:

**RUSDIL BADI'
NIM: 02.17.080**

Diperiksa dan Disetujui

Dosen Pembimbing I

Ir. Poerwanto, MT
NIP.131574847

Dosen Pembimbing II

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



Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT
NIP.P: 1039500274

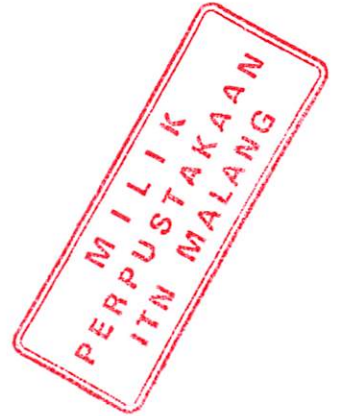
**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL**

2007



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

**Berita Acara Ujian Skripsi
Fakultas Teknologi Industri**



Nama Mahasiswa : Rusdil Badi'
NIM : 02.17.080
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroler AT89S8252 Menggunakan Sensor *Hall Effect (UGN 3503)* dengan Tampilan LCD

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

Hari : Jum'at
Tanggal : 16 Maret 2007
Dengan Nilai : 84,45 (A) *by*

Panitia Penguji Skripsi



Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

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

Anggota Penguji

Penguji Pertama

Ir. Teguh Herbasuki, MT

Penguji Kedua

M. Ashar, ST, MT

**Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis
Mikrokontroler AT89S8252 Menggunakan Sensor *Hall Effect* (UGN 3503)
dengan Tampilan LCD**

ABSTRAK

Perancangan dan pembuatan alat pencatat pemakaian air ini dilengkapi dengan tampilan harga dan alat ini terdiri dari beberapa komponen pendukung diantaranya adalah sensor *Hall effect* (UGN 3503) yang berfungsi mensensor aliran air yang melewati pipa, sensor ini akan bekerja (menghasilkan pulsa) bila didekatkan dengan magnet. Komponen yang kedua adalah mikrokontroler AT89S8252 yang berfungsi sebagai otak dari sistem, dalam mikrokontroler ini terdapat EEPROM internal sebesar 2 Kbyte yang berfungsi untuk menyimpan data dari sensor. Alat ini dilengkapi dengan rangkaian selektor yang terdiri dari 5 buah tombol *push button*, tiga diantaranya digunakan untuk memilih tampilan pada LCD yaitu tampilan volume air yang digunakan, tampilan harga dan tampilan pemakaian komulatif, sedangkan dua diantaranya digunakan untuk *mereset* jumlah pemakaian (pemakaian volume air dan harga), tetapi untuk pemakain komulatif tidak dapat *direset*, dan satu tombol yang digunakan untuk mematikan LCD. Untuk merubah harga yang telah ditetapkan maka alat ini dilengkapi dengan *keypad* 4 x 4, *keypad* ini akan dipasang bila dibutuhkan saja. Untuk tampilan digunakan sebuah LCD M1632 yang berfungsi untuk menampilkan data yang telah diproses oleh mikrokontroler.

KATA PENGANTAR

Alhamdulillah, dengan memanjatkan puji syukur kehadirat Allah SWT, yang telah memberikan rahmat, hidayah serta karunia-Nya, tak lupa sholawat dan salam kepada Nabi Muhammad SAW. dan keluarganya, akhirnya penyusun dapat dapat menyelesaikan skripsi ini yang berjudul “Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroler AT89S8252 Menggunakan Sensor *Hall Effect (UGN 3503)* dengan Tampilan LCD”, laporan Skripsi ini merupakan salah satu persyaratan kelulusan Program Strata 1 Teknik Elektro Konsentrasi Elektronika Institut Teknologi Nasional Malang.

Keberhasilan penyusunan laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terimakasih kepada:

1. Bapak dan Ibu tercinta yang selalu memberikan kasih sayang, do'a restu, dukungan dan bantuan secara moril maupun materiil kepada penyusun.
2. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
3. Bapak Ir. Mochtar Asroni, MSME. selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
4. Bapak Ir. F. Yudi Limpraptono, MT. selaku Ketua Jurusan Teknik Elektro S-1 Institut Teknologi Nasional Malang
5. Bapak Ir. Poerwanto, MT. selaku Dosen Pembimbing I
6. Bapak I Komang Somawirata ST, MT. selaku Dosen Pembimbing II

7. Teman-teman yang telah memberikan motivasi serta bantuan baik berupa tenaga maupun pikiran dalam proses penyelesaian skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam penyelesaian laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang bersifat membangun demi kesempurnaan laporan ini.

Malang, Maret 2007

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN.....	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR GRAFIK.....	xiii

BAB I PENDAHULUAN

1.1 Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah	2
1.4. Tujuan.....	2
1.5. Metodologi	3
1.6.Sistematika Penulisan.....	3

BAB II LANDASAN TEORI

2.1. Sensor <i>Hall-Effect UGN 3503</i>	4
2.2. Komparator	5
2.3. <i>Schmitt Trigger</i>	6
2.4. <i>Keypad</i>	8

2.5. <i>Push Button</i>	9
2.6. Mikrokontroler AT89S8252	9
2.6.1. Teori Umum	9
2.6.2. Konfigurasi Pin Pada Mikrokontroler AT89S8252.....	13
2.6.3. Masukan dan Keluaran.....	17
2.6.4. Data Memori (EEPROM) dan RAM	17
2.6.5. Osilator.....	20
2.6.6. SFR (<i>Special Function Register</i>).....	21
2.6.7. <i>Timer</i> dan <i>Counter</i>	25
2.6.8. Sistem Interupsi	27
2.6.9. <i>Reset</i>	28
2.7. Transistor Bipolar.....	29
2.7.1. Arus bias	29
2.7.2. Arus <i>Emitter</i>	29
2.7.3 Alpha (α)	30
2.7.4 Beta (β)	31
2.7.5. <i>Common Emitter</i> (CE)	32
2.7.6 .Kurva Basis	33
2.7.7 .Kurva Kolektor	34
2.7.8. Daerah Aktif	35
2.7.9. Daerah Saturasi	36
2.7.10. Daerah <i>Cut-off</i>	36

18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

2.7.11. Daerah <i>Breakdown</i>	38
2.7.12. Datasheet transistor	38
2.8. Dioda	39
2.9. LCD (<i>Liquid Crystal Display</i>)	42
2.9.1. Sinyal <i>interface</i> M1632.....	46

BAB III PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan.....	48
3.2. Perencanaan Dan Pembuatan Perangkat Keras	51
3.2.1 Perancangan Mekanik	51
3.2.2. Perancangan Rangkaian Sensor.....	54
3.2.3. Perancangan Rangkaian Selektor	56
3.2.4. Perancangan Rangkaian <i>Keypad</i>	57
3.2.5. Sistem Mikrokontroler AT89S8252	57
3.2.5.1. Rangkaian <i>Reset</i>	59
3.2.5.2. Rangkaian <i>Clock</i>	60
3.2.6. Rangkaian LCD M1632.....	61
3.3. Perencanaan Perangkat Lunak	64
3.3.1. Diagram Alir Prinsip Kerja Secara Umum	65
3.3.1. Diagram Alir Program	66

BAB IV PENGUJIAN ALAT DAN PEMBAHASAN HASIL

4.1. Pengujian Rangkaian Sensor	70
4.1.1. Tujuan	70

4.1.2. Peralatan yang digunakan.....	70
4.1.3. Langkah Pengujian	71
4.1.4. Hasil Pengujian	72
4.2. Pengujian Rangkaian Selektor	72
4.2.1. Tujuan	72
4.2.2. Peralatan yang digunakan.....	73
4.2.3. Langkah Pengujian	73
4.2.4. Hasil Pengujian.....	74
4.3. Rangkaian LCD.....	74
4.3.1. Tujuan	74
4.3.2. Peralatan yang digunakan.....	74
4.3.3. Langkah Pengujian	75
4.3.4. Hasil Pengujian.....	75
4.4. Pengujian keseluruhan Sistem	76
4.5. Cara pengoprasian Alat	78

BAB V PENUTUP

5.1. Kesimpulan	81
5.2. Saran	82

DAFTAR PUSTAKA	83
-----------------------------	-----------

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1 Konfigurasi Pin <i>Hall Effect</i> UGN 3503	4
Gambar 2.2 Rangkaian komparator	5
Gambar 2-3. Rangkaian Dasar <i>Schmitt Trigger</i>	7
Gambar 2.4 <i>Keypad</i> dot matrik 4x4.....	8
Gambar 2.5 Simbol <i>push Button</i>	9
Gambar 2.6 Diagram Blok AT89S8252	10
Gambar 2.7 Konfigurasi Pin AT89S8252.....	13
Gambar 2.8 Organisasi RAM <i>Internal</i>	20
Gambar 2.9 Karakteristik Osilator	21
Gambar 2.10 Letak SFR Pada Mikrokontroler AT89S8252	22
Gambar 2.11 Koneksi SPI Master dan Slave.....	25
Gambar 2.12 Arus Emitor.....	30
Gambar 2.13. Rangkaian CE.....	32
Gambar 2.14. Rangkaian Driver LED.....	37
Gambar 2.15. Resistansi Bulk.....	40
Gambar 2.16 Modul LCD 2 ×16 karakter.....	43
Gambar 2.17 Mengirim/Mengambil Data Ke/Dari M1632	47
Gambar 3.1. Blok Diagram Sistem.....	49
Gambar 3.1. Meteran air	52
Gambar 3.2 Rangkaian sensor.....	54
Gambar 3.3 Rangkaian <i>Trimport</i>	55

Gambar 3.4. Rangkaian Selektor.....	56
Gambar 3.5. Rangkaian Mikrokontroler AT89S8252	58
Gambar 3.6. Rangkaian <i>Reset</i> pada Mikrokontroler AT89S8252	59
Gambar 3.7. Rangkaian <i>Clock</i> pada Mikrokontroler AT89S8252.....	60
Gambar 3.8 Perancangan <i>Liquid Crystal Display (LCD)</i>	63
Gambar 4.1 Rangkaian Sensor	71
Gambar 4.2 Rangkaian Pengujian Selektor	73
Gambar 4.3 Rangkaian LCD.....	75
Gambar 4.4 Tampilan pada LCD	76
Gambar 4.5 Tampilan pada LCD pada saat saklar ON	78
Gambar 4.6 Tampilan pada LCD pada saat saklar volume ditekan	78
Gambar 4.7 Tampilan pada LCD pada saat tombol harga ditekan	79
Gambar 4.8 Tampilan pada LCD pada saat tombol komulatif ditekan	79
Gambar 4.9 Tampilan pada LCD pada saat proses perubahan harga.....	79
Gambar 4.10 Tampilan pada LCD setelah tombol reset ditekan	80

1DAFTAR TABEL

Tabel 2.1. Fungsi khusus pada <i>Port</i> 1	14
Tabel 2.2. Fungsi Alternatif <i>Port</i> 3	14
Tabel 2.3. Pengaturan RS 0 - RS 1 Untuk <i>Select Register Bank</i>	19
Tabel 2.4. 128 <i>Byte Special Function Register</i>	22
Tabel 2.5. <i>Mode</i> Operasi <i>Timer/Counter</i> 0 dan 1	26
Tabel 2.6. <i>Mode</i> Operasi <i>Timer</i> 2	27
Tabel 2.7 Pemilihan <i>Register</i> Pada LCD M1632.....	42
Tabel 2.8 Fungsi Pin – Pin LCD	44
Table 3.1 Hasil percobaan untuk 1 liter	53
Table 3.2 Hasil percobaan untuk 2 liter	53
Tabel 4.1 Hasil Pengujian Rangkaian Sensor	72
Tabel 4.2 Hasil Pengujian Rangkaian Selektor	74
Table 4.3 Hasil pengujian untuk 1 liter	76
Table 4.4 Hasil pengujian untuk 2 liter	77

DAFTAR GRAFIK

Grafik 2.1. Kurva $I_B - V_{BE}$	33
Grafik 2.2. Kurva Kolektor	35
Grafik 2.3. Arus Forward.....	40
Grafik 2.4. Karakteristik Dioda Silikon.....	41

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dunia elektronika adalah dunia teknologi yang sangat luas, yang selalu berkembang dari waktu ke waktu. Seiring dengan perkembangannya banyak hal di dunia ini yang dapat dilakukan oleh dunia elektronika. Dari hal-hal yang bersifat rumit hingga yang sederhana seperti yang ada pada kehidupan sehari-hari.

Bidang elektronika mulai berkembang pesat dengan diperkenalkannya rangkaian terintegrasi yang pertama pada tahun 1960. Perkembangan bidang ini ditandai dengan pencapaian kepadatan element, kecepatan operasi serta fungsi yang dapat dilakukan oleh rangkaian terintegrasi.

Mikrokontroler merupakan kombinasi antara mikroprosesor, I/O ROM dan RAM yang memungkinkan untuk berhubungan dengan dunia luar. dengan segala kelebihan yang dimiliki Mikrokontroler, untuk itu kami mencoba membuat sebuah alat yang bermanfaat bagi kehidupan masyarakat yaitu alat yang dapat memonitor banyaknya air yang kita konsumsi dalam rumah, alat ini dapat mengkonversi volume air yang telah dikonsumsi ke bentuk harga (rupiah).

Dalam pembuatan skripsi ini diharapkan pengguna air dapat memonitor volume air yang dikonsumsi dan jumlah harga (rupiah), serta jumlah pemakaian kumulatif

1.2. Rumusan Masalah

1. Bagaimana merencanakan sensor volume air yang akan dikonsumsi oleh konsumen
2. Bagaimana merancang *hardware* dan *software* yang sesuai dengan sensor tersebut sehingga dapat bekerja sesuai dengan apa yang diharapkan
3. Bagaimana alat ini dapat difungsikan secara baik dan volume air, harga serta pemakaian kumulatif dapat ditampilkan dalam satu *display* yang dilengkapi selektor untuk memilih tampilan yang akan ditampilkan.

1.3. Batasan Masalah

1. Penggunaan alat hanya pada pemakaian rumah tangga saja
2. Volume air maksimum yang ditampilkan 6 digit yaitu 999999 liter atau 999 m³
3. Maksimum harga yang ditampilkan 6 digit yaitu 999999 Rupiah

1.4. Tujuan

Tujuan dari skripsi ini adalah merancang dan membuat alat yang dapat memonitor volume pemakaian air serta dapat mengkonversi kedalam bentuk harga (rupiah) yang dilengkapi dengan LCD sebagai *display* serta selektor yang digunakan untuk memilih tampilan yang akan ditampilkan pada LCD

1.5. Metodologi

1. *Study literature* tentang teori sistem kontrol menggunakan mikrokontroler secara umum, kemudian mempelajari teori mikrokontroler.
2. Setelah melakukan pengumpulan *literature*, maka dilakukan perencanaan perangkat keras dan perangkat lunak
3. Pengujian terhadap peralatan serta pengukuran data hasil pemantauan peralatan.
4. Penyusunan laporan skripsi.

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Pada bab ini berisi tentang latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan, metodologi, dan sistematika penulisan.

BAB II LANDASAN TEORI

Berisi landasan teori yang menunjang dalam perencanaan dan pembuatan alat.

BAB III PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini dibahas tentang perencanaan dan pembuatan alat.

BAB IV PENGUJIAN ALAT DAN PEMBAHASAN HASIL

Meliputi proses pengujian alat serta hasil dari pengujian alat.

BAB V PENUTUP

Meliputi kesimpulan dan saran

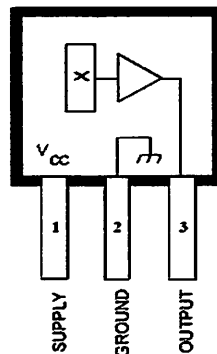
BAB II

LANDASAN TEORI

Bab ini akan menguraikan tentang dasar-dasar teori yang dapat menunjang dalam perencanaan dan pembuatan alat. Uraian teori dalam bab ini meliputi teori tentang Sensor *hall effect* (UGN 3503), IC AT89S8252, *Keypad*, LCD dan komponen-komponen penunjang lainnya.

2.1. Sensor *Hall-Effect* UGN 3503

Sensor *Hall Effect* dirancang untuk memberikan tanggapan terhadap intensitas medan magnet yang ada disekitarnya (www.allegromicro.com). Sensor ini memiliki tiga buah terminal, apabila tidak terdapat medan magnet didekatnya, tegangan yang dihasilkan piranti ini besarnya setengah dari tegangan catu daya.



Gambar 2.1 Konfigurasi Pin *Hall Effect* UGN 3503

Sumber: www.allegromicro.com

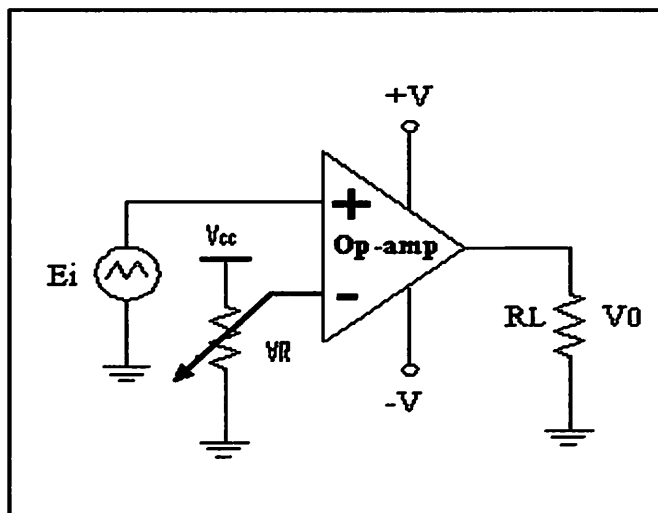
Sensor *Hall Effect* UGN 3503 banyak dijumpai di pasaran, sensor ini membutuhkan tegangan catu daya antara 4,5 V hingga 6 V. sensor ini dapat digunakan sebagai saklar dengan sekedar menggerakkan sebuah magnet mendekati atau menjauhinya.

Agar dapat bekerja sensor diletakkan pada didalam meteran air (PDAM), sedangkan magnet yang digunakan untuk mengaktifkan sensor diletakkan pada baling-baling tembahan yang dipasang didalam meteran tersebut.

2.2. Komparator

Komparator merupakan sebuah pembanding untuk membandingkan tegangan isyarat pada satu masukan dengan suatu tegangan acuan pada masukan lainnya (Couglin,1992: 19). *Op-amp* serba guna telah digunakan sebagai tambahan untuk IC yang telah dirancang untuk pembanding.

Tegangan keluaran *op-amp* serbaguna dapat dirubah sesuai batas-batas yang ditetapkan oleh tegangan-tegangan jenuh, $+V_{sat}$ dan $-V_{sat}$ untuk tegangan V_{cc} -nya menggunakan tegangan DC 6V, sehingga tegangan keluarannya dapat mengeluarkan tegangan sebesar 5V kerana piranti-piranti yang lain memerlukan taraf tegangan antara 0 dan +5V.



Gambar 2.2 Rangkaian komparator

Sumber: Couglin,1992: 19

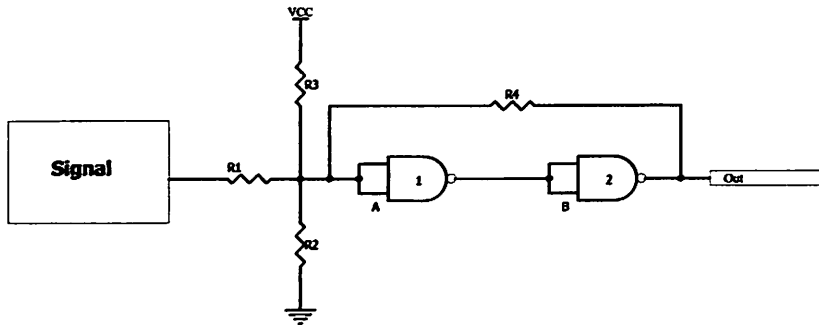
2.3. Schmitt Trigger

Schmitt Trigger, serupa dengan bistabil, astabil, dan monostabil, merupakan rangkaian *regenerative* dua *gate* dengan *feedback* diantara *input* dan *outputnya*. Suatu IC *schmitt trigger* seperti yang diperlihatkan dalam gambar 2-3, merupakan suatu *voltage level detector*. Bila tegangan *inputnya* lebih rendah dari suatu batas tegangan tertentu maka disebut sebagai *upper trip level (UTL)*, tegangan *outputnya* tepat pada batas logik 0 dan bilamana tegangan *input* tersebut diatas UTL-nya maka tegangan *outputnya* tepat pada batas *logic 1*.

Pada mulanya tegangan *input* pada suatu batas *high* maka rangkaian akan berubah menjadi *output* dengan level 0 pada suatu tegangan *input* yang disebut *Lower Trip Level (LTL)*. LTL ini lebih rendah dibandingkan dengan UTL dan perbedaan tegangan merupakan sesuatu yang *hysteresis*, diukur dalam volts :

$$\text{Hysteresis} = \text{UTL} - \text{LTL} \dots \dots \dots (2-1)$$

Rangkaiannya adalah *regenerative*, dan sebagai level trips yang mencapai transisi yang terjadi dengan kecepatan besar. Disamping itu digunakan untuk mengubah suatu *slowly changing signal* menjadi sesuatu dengan transisi yang cepat. Ini mempunyai manfaat yang besar dalam proses *couple circuitry* dimana kopling kapasitornya dapat diperkecil. Dalam gambar 2-3, tahanan R2 dapat digunakan untuk mengatur *trips point*-nya.



Gambar 2-3. Rangkaian Dasar Schmitt Trigger

Sumber : *Buku Panduan Praktikum Teknik Digital, Proyek PUSDIK Migas Cepu, Jurusan Instrumen dan Elektronika, ITS Surabaya, Hal 173.*

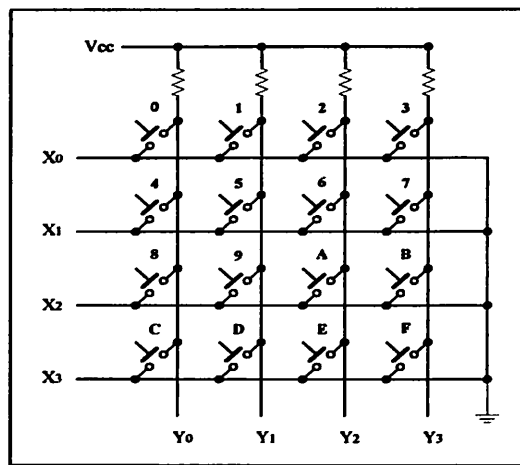
Rangkaian gambar diatas dioperasikan dalam cara berikut ini. Jika tegangan signal dan lebih rendah dari tegangan pada titik A, *input* pada gate satu akan lebih rendah/dibawah tegangan *threshold*-nya.

Hal ini dibantu oleh level 0 dari gate 2 melalui R4. Bila *signal input*-nya pada suatu *high* yang hampir mengenai tegangan (UTL) menjadikan tegangan pada A sama dengan tegangan *threshold* dari *gate*-nya, *gate* 1 mengubah level *output*-nya dan saat itu pula menjadikan *output* gate 2 menjadi level 1. Ini membantu menjaga/memelihara titik A berada pada tegangan transisinya. (Pekerjaan ini merupakan *regenerative* dan mengambil posisi secara cepat).

Sekarang jika kita memulai memperkecil tegangan *input*nya, kita harus memperkecilnya sehingga tegangan pada titik A mencapai pada suatu tegangan *threshold* lagi. Ini harus lebih kecil daripada UTL sebab R4 sekarang dihubungkan pada suatu level yang lebih tinggi. Pada tegangan ini, LTL, circuit *regenerative*-nya segera menuju *output* lagi.

2.4. Keypad

Keypad merupakan salah satu alat untuk memasukkan suatu data kekomputer atau minimum sistem (Mazidi, 1995: 277). Dalam perencanaan ini, tombol masukan (*Keypad*) yang digunakan adalah *Keypad* matrik 4x4. *Keypad* ini berfungsi untuk merubah harga yang telah ada sebelumnya, Prinsip kerja *Keypad* matrik dapat dilihat dari Gambar 2.4.



Gambar 2.4 *Keypad* dot matrik 4x4

Sumber: Mazidi, 1995: 277

Gambar 2.4. terlihat bahwa *Keypad* matrik ini terdiri atas 4 lajur baris (X_0 - X_3) dan 4 lajur kolom (Y_0 - Y_3), dan *Keypad* ini akan bekerja dengan menggunakan prinsip *scanning* pada lajur baris dan lajur kolom tersebut. Jika terdeteksi adanya persambungan antara baris dan kolom *valid*, maka data dari *Keypad* tersebut akan diterjemahkan pada mikrokontroler

2.5. Push Button

Push button merupakan sebuah saklar tekan yang bekerja apabila ditekan, cara kerjanya adalah jika ada penekanan maka kondisi saklar ON, sebaliknya bila tidak ditekan maka kondisi saklar menjadi OFF, berikut dibawah ini adalah simbol *push button*.



Gambar 2.5 Simbol *push Button*

2.6. Mikrokontroler AT89S8252

2.6.1. Teori Umum

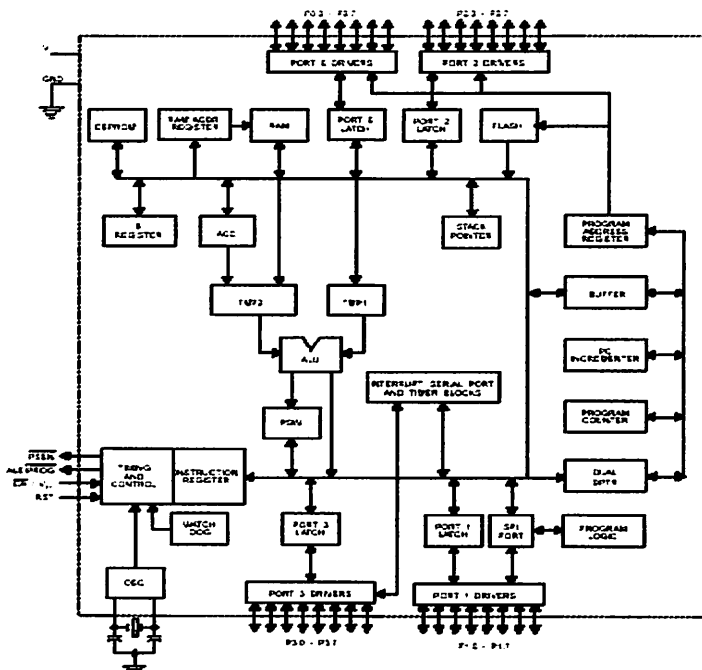
SCM (*single on-chip*) adalah suatu mikrokontroler lengkap yang dibuat dalam sebuah IC yang mempunyai struktur seperti CPU, Osilator, *Timer*, RAM, EPROM dan *Buffer* (alamat, data dan *input-output*). Mikrokontroler AT89S8252 merupakan mikrokontroler 8-bit kompatibel dengan standar industri MCS-51™ baik dari segi pemrograman maupun kaki tiap pin. Mikrokontroler AT89S8252 mempunyai 8 Kbyte PEROM (*Programmable and Erasable Read Only Memory*).

Pada dasarnya mikrokontroler terdiri atas mikroprosesor, *Timer*, *Counter*, perangkat I/O dan *internal* memori. Mikrokontroler termasuk perangkat yang mudah didesain dalam bentuk *chip* tunggal (*single chip*). Pada dasarnya mikrokontroler mempunyai fungsi yang sama dengan mikroprosesor yaitu untuk mengontrol suatu kerja sistem.

Di dalam mikrokontroler juga terdapat CPU, ALU, PC, SP dan *register* seperti dalam mikroprosesor, tetapi juga ditambah dengan perangkat-perangkat

lain seperti RAM, ROM, PIO, SIO, *Counter* dan sebuah rangkaian *Clock*. RAM pada dasarnya merupakan suatu *flip-flop* yang dapat *diset/direset*, sifat ini membuat RAM dapat dibaca atau ditulis. Karena transistor yang menyusun *flip-flop* membutuhkan suatu tegangan DC agar tetap aktif, maka sel RAM akan kehilangan datanya bila *Power* dimatikan. Hal ini dalam dunia komputer disebut bersifat *volatile*. Sedangkan ROM memiliki beberapa tipe diantaranya yaitu *Mask Programmable ROM*, *Fusible Link PROM*, *UV Light Erasable PROM (EPROM)* dan *EEPROM*. Tidak seperti RAM, data yang ada didalam ROM tidak akan hilang bila *Power* dimatikan. Hal ini disebut bersifat *non-volatile*, suatu pemrogram khusus yang diperlukan untuk menulis data ke ROM. Karena *non-volatilitasnya* maka ROM sering dipetakan ke alamat *reset* dari mikrokontroler, dalam hal ini sangat diperlukan pada saat melakukan *booting*.

Diagram blok dari IC AT89S8252 seperti gambar 2.6 dibawah ini.



Gambar 2.6 Diagram Blok AT89S8252

Sumber: ATMEL Datasheet Book, 3

Mikrokontroler didesain dengan instruksi-instruksi lebih luas dan 8-bit instruksi yang digunakan untuk membaca data instruksi dari *internal* memori ke ALU. Sebagai suatu sistem kontrol, mikrokontroler bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. 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.

Di samping itu pula mikrokontroler AT89S8252 membutuhkan daya rendah, memiliki *performance* yang tinggi dan merupakan mikrokontroler 8-bit yang dilengkapi 2 Kbyte EEPROM (*Electrical Erasable and Programmable Read Only Memory*) dan 256 byte RAM *internal*, program memori yang dapat diprogram ulang dalam sistem atau menggunakan *programmer non-volatile memory* konvensional.

AT89S8252 juga mempunyai 2 buah *Power Saving Mode* yang dapat diatur melalui *software*, yaitu *IDE Mode* yang akan menghentikan CPU sebagai RAM, dimana *Timer/Counter*, *Serial Port* dan Sistem Interupsi tetap berfungsi. Sedangkan *Power Down Mode* yang akan menyimpan data di RAM dan akan

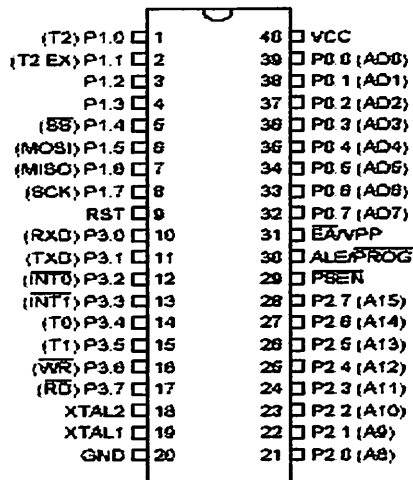
menahan osilator untuk tidak mengaktifkan *chip* yang lain sampai terjadi *reset* secara hardware.

Adapun secara umum, konfigurasi yang dimiliki mikrokontroler AT89S8252 adalah sebagai berikut :

- Sebuah CPU 8-bit dengan menggunakan teknologi dari Atmel.
- 8 Kbyte *Downloadable Flash Memory*.
- 2 Kbyte EEPROM.
- Sebuah *port* serial dengan kontrol *full duplex* UART.
(*Universal Asynchronous Receiver Transmitter*).
- 256 byte RAM *internal*.
- 32 I/O yang dapat dipakai semuanya.
- 3 buah *Timer/Counter* 16-bit.
- 6 Sumber Interupsi.
- SPI *Serial Interface*.
- *Programmable Watchdog Timer*.
- *Dual data pointer*.
- Frekuensi kerja 0 – 24 MHz.
- Tegangan operasi 2,7 V sampai 6 V.
- *Power-of flag*.
- Kemampuan melaksanakan operasi perkalian, pembagian dan operasi *Boolean* (bit).

2.6.2. Konfigurasi Pin Pada Mikrokontroler AT89S8252

Konfigurasi kaki-kaki mikrokontroler AT89S8252 terdiri dari 40 pin seperti yang terlihat pada gambar berikut ini.



Gambar 2.7 Konfigurasi Pin AT89S8252

Sumber: ATMEL Datasheet Book, 2

Adapun fungsi dari tiap pin akan dijelaskan sebagai berikut :

- Pin 1 – 8.

Port 1 yang terdiri atas pin 1 – 8 merupakan saluran masukan/keluaran dua arah dengan *internal pull up* dan mempunyai fungsi khusus seperti yang terlihat pada tabel.

Tabel 2.1. Fungsi khusus pada Port 1

Port Pin	Fungsi
P1.0	T2 (masukan eksternal untuk <i>Timer/Counter 2</i>)
P1.1	T2EX (<i>Timer/Counter 2 capture/reload trigger</i> dan kontrol arah)
P1.2	-
P1.3	-
P1.4	SS (<i>slave port select input</i>)
P1.5	MOSI (<i>master data output, slave data input</i> untuk chanel SPI)
P1.6	MISO (<i>master data input, slave data output</i> untuk chanel SPI)
P1.7	SCK (<i>master clock output, slave clock input</i> untuk chanel SPI)

Sumber: ATMEL Datasheet Book, 4

- Pin 9.

RST merupakan saluran 2 masukan untuk mereset mikrokontroler dengan cara memberi masukan logika tinggi.

- Pin 10 – 17.

Port 3 yang terdiri atas pin 10 – 17 merupakan saluran masukan/keluaran dua arah dengan *internal pull up* dan mempunyai fungsi khusus seperti yang terlihat pada tabel.

Tabel 2.2. Fungsi Alternatif Port 3

Port Pin	Fungsi
P3.0	RXD (<i>port serial input</i>)
P3.1	TXD (<i>port serial output</i>)

P3.2	INT0 (Interupsi eksternal 0)
P3.3	INT1 (Interupsi eksternal 1)
P3.4	T0 (<i>input</i> eksternal <i>Timer</i> 0)
P3.5	T1 (<i>input</i> eksternal <i>Timer</i> 1)
P3.6	WR (menulis data ke memori eksternal)
P3.7	RD (membaca data dari <i>memory</i> eksternal)

Sumber: *ATMEL Datasheet Book, 5*

- Pin 18 – 19.

XTAL₁ dan XTAL₂ merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan *internal* maupun eksternal. Pin ini dihubungkan dengan kristal bila menggunakan osilator *internal*. XTAL₁ merupakan masukan ke rangkaian osilator *internal* sedangkan XTAL₂ merupakan keluaran dari rangkaian osilator *internal*.

- Pin 20.

V_{SS} merupakan hubungan ke *ground* dari rangkaian.

- Pin 21 – 28.

Port 2 yang terdiri atas pin 21 – 28 merupakan saluran masukan/keluaran dua arah dengan *internal pull up*. *Port* ini mengeluarkan 8-bit bagian alamat tinggi (A₈ – A₁₅) selama pengambilan instruksi dari memori program eksternal dan pengambilan data memori eksternal menggunakan *mode* pengalamatan 16-bit.

- Pin 29.
PSEN (*Program Store Enable*) merupakan sinyal baca untuk mengaktifkan memori program eksternal.
- Pin 30.
ALE/PROG (*Address Latch Enable*) merupakan pulsa yang berfungsi untuk mengeluarkan alamat rendah ($A_0 - A_7$) dalam *port* 0, selama proses baca/tulis memori eksternal. Frekuensi ALE adalah 1/6 kali frekuensi osilator dan dapat digunakan sebagai pewaktu. Pin ini juga berfungsi sebagai saluran program selama dilakukan pemrograman jika menggunakan memori program *internal*.
- Pin 31.
EA/VPP (*External Access Enable*) untuk mengatur penggunaan memori program eksternal dan *internal*. Pin ini harus dihubungkan dengan *ground* bila menggunakan memori program eksternal dan dihubungkan dengan VPP sebesar 12 V jika menggunakan memori program *internal*. Dapat diberikan logika rendah (*Ground*) atau logika tinggi (+5V), jika diberikan logika tinggi maka mikrokontroler akan mengakses program dari ROM *internal* (EEPROM/*Flash Memory*), dan jika diberikan logika rendah maka mikrokontroler akan mengakses program dari memori eksternal.

- Pin 32 – 39.

Port 0 yang terdiri atas pin 32 – 39 merupakan saluran masukan/keluaran dua arah tanpa *internal pull up*. *Port 0* merupakan saluran alamat rendah ($A_0 - A_7$) yang dimultipleks dengan saluran bus data ($D_0 - D_7$).

- Pin 40.

V_{CC} merupakan saluran masukan untuk catu daya positif sebesar 5 Volt DC dengan toleransi kurang lebih 1 %.

2.6.3. Masukan dan Keluaran

Untuk saluran masukan dan keluaran terdapat 4 buah *port* yang masing-masing 8-bit. Saluran ini bersifat dua arah (*bidirectional*) yang berarti dapat difungsikan sebagai masukan/keluaran, serta dapat dialamati per bit. *Port 3* selain digunakan sebagai *port* masukan dan keluaran juga dapat digunakan sebagai fungsi pengganti sebagaimana yang terdapat dalam tabel 2.2. AT89S8252 juga memiliki fitur tambahan yang terdapat pada *port 1* seperti dalam tabel 2.1.

2.6.4. Data Memori (EEPROM) dan RAM

Berbeda dengan mikrokontroler standar MCS-51, untuk AT89S8252 terdapat 2 *Kbytes* memori dalam EEPROM untuk penyimpanan data dan 256 *byte* untuk RAM. Dibagian atas 128 *byte* RAM ditempati paralel untuk SFR. Bagian atas 128 *byte* mempunyai alamat sama dengan SFR, tetapi secara fisik terpisah dari SFR.

EEPROM *on-chip* ini diakses dengan mengeset bit EEMEN pada *register* WMCON pada alamat 96_H. Alamat EEPROM ini adalah 000_H sampai 7FF_H. Dan selama EEPROM memprogram, yang dibaca dari EEPROM akan mengambil *byte* yang sedang ditulis dengan melengkapi MSB. Instruksi *movx* digunakan untuk mengakses EEPROM *internal* ini. Bit EEMWE pada *register* WMCON harus diset ke “1” sebelum sembarang lokasi pada EEPROM dapat ditulis. Program pengguna harus *mereset* bit EEMWE ke “0” jika proses penulisan ke EEPROM tidak diperlukan lagi. Proses penulisan ke EEPROM dapat dilihat dengan membaca bit RDY/BSY pada SFR WMCON. Jika bit ini berlogika rendah maka berarti penulisan EEPROM sedang berlangsung, tapi jika bit ini berlogika tinggi berarti penulisan sudah selesai dan penulisan lain dapat dimulai lagi.

Sedangkan RAM yang ada pada mikrokontroler AT89S8252 berkapasitas 256 *byte* dan kompatibel dengan RAM yang ada pada mikrokontroler standar MCS-51. Pada *lower* 128-bit lokasi memori dapat dibagi menjadi 3 bagian, yaitu :

1. Empat *Bank Register* .

Setiap *bank* terdiri dari 8 *register* (R0-R7), sehingga jumlah *register* untuk keempat *bank register* (*bank* 0 – *bank* 3) menjadi 32 buah *register* yang menempati ruang alamat 00_H – 1F_H. Untuk mengaktifkan salah satu *bank register* dapat dilakukan dengan mengatur RS0-RS1 melalui pengaturan pada PSW (*Program Status Word*).

2. *Bit Addressable*.

Terdiri dari 16-bit yang berada pada alamat $20_H - 2F_H$. Masing- masing dari 128-bit lokasi ini dapat dialamat secara langsung yaitu dari $00_H - 7F_H$.

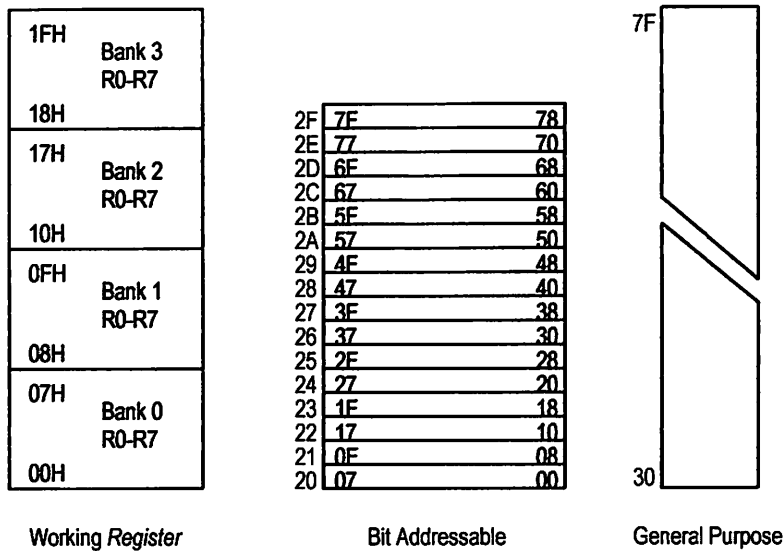
3. *Scratch Pad Area.*

Terdiri dari 80-byte yang menempati alamat $30_H - 7F_H$ yang dapat dialamat secara langsung dan dapat digunakan untuk keperluan umum (*General Purpose RAM*). Misalnya digunakan untuk lokasi *Stack*.

Tabel 2.3. Pengaturan RS 0 - RS 1 Untuk *Select Register Bank*

RS 1	RS 0	<i>Select Register Bank</i>
0	0	<i>Bank 0</i>
0	1	<i>Bank 1</i>
1	0	<i>Bank 2</i>
1	1	<i>Bank 3</i>

Sumber : *Teknik Antarmuka dan Pemrograman AT89C51, Elekmedia Komputindo*



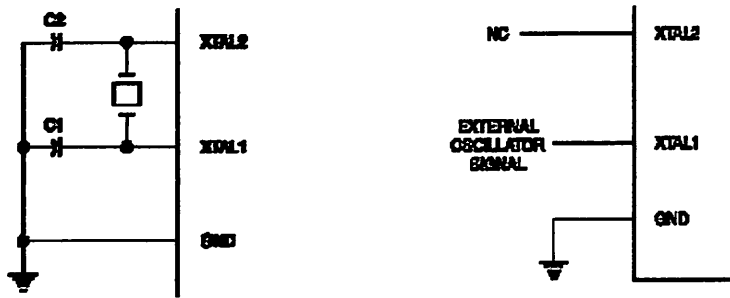
Gambar 2.8 Organisasi RAM Internal

Sumber : *Teknik Antarmuka dan Pemrograman AT89C51, Elekmedia Komputindo*

2.6.5. Osilator

Jantung dari AT89S8252 adalah rangkaian yang membangkitkan pulsa *clock* yang mensinkronkan semua operasi *internal*. Mikrokontroler AT89S8252 memiliki osilator *internal* (*on-chip oscillator*) yang dapat digunakan sebagai sumber pewaktu (*clock*) bagi CPU. Untuk menggunakan osilator *internal* diperlukan sebuah kristal atau resonator keramik antara pin XTAL₁ dan XTAL₂ dan sebuah kapasitor ke *ground*. XTAL₂ dan XTAL₁ secara berurutan merupakan *input* dan *output* dari sebuah *inverting amplifier* yang dapat dikonfigurasi penggunaannya sebagai *on-chip oscillator* seperti yang ditunjukkan pada gambar 2.9a.

Untuk memberikan IC AT89S8252 sumber *Clock* eksternal, maka pin XTAL₂ dibiarkan tidak berhubungan dengan sumber *Clock* eksternal dan XTAL₁ dihubungkan dengan sumber *Clock* eksternal seperti pada gambar 2.9b.



a). *Oscillator Connector*

b). *External Clock Drive Configuration*

Gambar 2.9 Karakteristik Osilator

Sumber: *ATMEL Datasheet Book, 21*

2.6.6. SFR (*Special Function Register*)

Area memori AT89S8252 disebut dengan SFR (*Special Function Register*) yang merupakan *register* dengan tugas khusus. Gambar berikut akan menjelaskan letak masing-masing SFR.

0F8H								0FFH
0F4H	B 00000000							0F7H
0E8H								0EFH
0E4H	ACC 00000000							0E7H
0D8H								0DFH
0D4H	PSW 00000000				SPCR 00000100			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C4H								0C7H
0B8H	IP XX000000							0BFH
0B4H	P3 11111111							0B7H
0A8H	P2 00000000		SPSR 00XXXXXX					0AFH
0A4H	P1 11111111							0A7H
098H	SCON 00000000	SBUF XXXXXXXX						09FH
094H	P0 11111111						VBROCN 00000000	09FH
088H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		08FH
084H	P0 11111111	SP 00001111	DPL 00000000	DPH 00000000	DPIL 00000000	DPH 00000000	SPSR XXXXXXXX	08FH

Gambar 2.10 Letak SFR Pada Mikrokontroler AT89S8252

Sumber: ATMEL Datasheet Book. 6

Tidak semua *address* digunakan sebagai SFR, hanya *address* tertentu seperti yang dijelaskan oleh tabel 2.4 berikut ini.

Tabel 2.4. 128 Byte Special Function Register

SYMBOL	NAME	ADDRESS
ACC	ACCUMULATOR	0E0H
B	B REGISTER	0F0H
PSW	PROGRAM STATUS WORD	0D0H
SP	STACK POINTER	81H
DPTR	DATA POINTER 2 BYTE	
DPL	LOW BYTE	82H
DPH	HIGH BYTE	83H
P0	PORT 0	80H
P1	PORT 1	90H
P2	PORT 2	0A0H
P3	PORT 3	080H

IP	<i>INTERUPSI PERIORITY CONTROL</i>	088H
IE	<i>INTERUPSI ENABLE CONTROL</i>	0ABH
TMOD	<i>TIMER/COUNTER MODE CONTROL</i>	89H
TCON	<i>TIMER/COUNTER CONTROL</i>	88H
+TCON	<i>TIMER/COUNTER 2 CONTROL</i>	0CBH
TH0	<i>TIMER/COUNTER 0 HIGH CONTROL</i>	8CH
TL0	<i>TIMER/COUNTER 0 LOW CONTROL</i>	8DH
TH1	<i>TIMER/COUNTER 1 HIGH CONTROL</i>	8DH
TL1	<i>TIMER/COUNTER 1 LOW CONTROL</i>	8CH
TH2	<i>TIMER/COUNTER 2 HIGH CONTROL</i>	0CDH
TL2	<i>TIMER/COUNTER 2 LOW CONTROL</i>	0CCH
RCAP2H	<i>T/C 2 CAPTURE REG. HIGH BYTE</i>	0CBH
+RCAP2L	<i>T/C 2 CAPTURE REG. LOW BYTE</i>	0CAH
SCON	<i>SERIAL CONTROL</i>	98H
SBUF	<i>SERIAL DATA BUFFER</i>	99H
PCON	<i>POWER CONTROL</i>	87H

Sumber : *Teknik Antarmuka dan Pemrograman AT89C51, Elekmedia Komputindo*

Selain itu mikrokontroler AT89S8252 memiliki tambahan SFR. Hal ini tak lain adalah karena terdapatnya tambahan fitur pada mikrokontroler ini. SFR tambahan ini meliputi T2CON (*Timer 2 Control* dengan alamat 0C8_H), T2MOD (*Timer 2 Mode* dengan alamat 0C9_H), WMCON (*Watchdog and Memory Control Register* dengan alamat 96_H), SPCR (*SPI Control Register* dengan alamat D5_H), SPSR (*SPI Status Register* dengan alamat AA_H), SPDR (*SPI Data Register* dengan alamat 86_H).

- SFR untuk *Timer 2*

Mikrokontroler AT89S8252 terdapat tambahan sebuah *Timer/Counter* yang diberi nama *Timer 2* (sehingga AT89S8252 memiliki 3 *Timer/Counter* yaitu *Timer/Counter 0*, *Timer/Counter 1*, *Timer/Counter 2*). Pada *Timer/Counter 2* ini dikendalikan oleh SFR yang bernama T2CON (*Timer 2 Control*), T2MOD (*Timer 2 Mode*) dan sepasang *register* RCAP2H, RCAP2L yang merupakan *register capture/reload* untuk *Timer 2* dalam 16-bit *capture mode/auto-reload mode*.

- SFR untuk *Watchdog* dan Memori

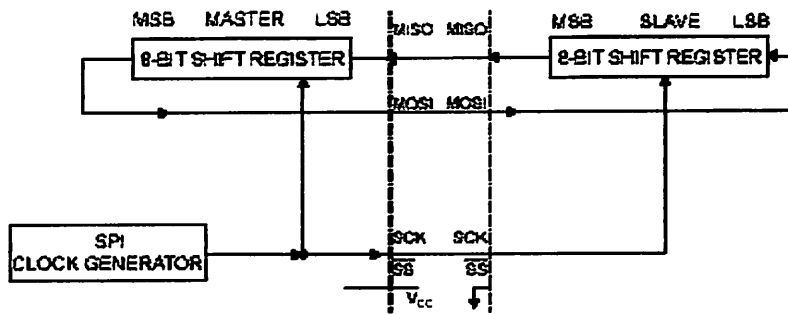
Untuk menggunakan *Watchdog Timer/Memori*, maka dapat dilakukan dengan mengatur SFR yang bernama WMCON dengan alamat 96H

- SFR pengontrol SPI

Berbeda dengan mikrokontroler MCS-51, AT89S8252 memiliki fasilitas SPI (*Serial Peripheral Interface*). Fasilitas ini memungkinkan transfer data kecepatan tinggi secara sinkron antara mikrokontroler dengan peripheral atau antar mikrokontroler AT89S8252. Fitur ini meliputi :

- a. *Full Duplex*, 3 kawat dengan transfer data secara sinkron
- b. Operasi *Master* atau *Slave*
- c. Frekuensi maksimum 6 MHz
- d. 4-bit *rate* terprogram
- e. Sistem data transfer MSB dahulu atau LSB
- f. *Write Collision Flag Protection*

Gambar berikut menunjukkan hubungan antara CPU *master* dan *slave*.



Gambar 2.11 Koneksi SPI Master dan Slave

Sumber: ATMEL Datasheet Book, 21

2.6.7. *Timer dan Counter*

Dalam mikrokontroler AT89S8252 terdapat 3 buah pewaktu/pencacah (*Timer/Counter*) 16-bit yang dapat diatur melalui perangkat lunak, yaitu pewaktu/pencacah 0 dan pewaktu/pencacah 1. *Timer/Counter* ini diatur oleh SFR (*Special Function Register*) yaitu *Timer/Counter Control* (TCON dengan alamat 88H) dan *Timer/Counter Mode Control* (TMOD dengan alamat 89H). Selain itu nilai *byte* bawah dan *byte* atas dari *Timer/Counter* disimpan dalam *register* TL dan TH.

Jika difungsikan sebagai *Timer*, maka akan menggunakan sistem *clock* sebagai sumber masukan pulsanya. Jika sebagai *Counter* (pencacah), maka akan menggunakan pulsa dari luar (eksternal) sebagai masukan pulsanya. Pada *port* 3 terdapat fungsi khusus yaitu T0 (masukan luar untuk *Timer/Counter* 0) dan T1 (masukan luar untuk *Timer/Counter* 1). Pemilihan *mode Timer/Counter* dikontrol oleh *register* TMOD. Dengan memberikan nilai tertentu pada *register* TMOD,

dapat dipilih *mode* operasi untuk *Timer/Counter 0* dan *Timer/Counter 1* seperti terlihat dalam tabel.

Tabel 2.5. Mode Operasi *Timer/Counter 0* dan *1*

<i>Mode</i>	<i>Timer/Counter 0</i>	<i>Timer/Counter 1</i>
0	13-bit <i>Timer</i>	13-bit <i>Timer</i>
1	16-bit <i>Timer</i>	16-bit <i>Timer</i>
2	8-bit auto-reload	8-bit auto-reload
3	Dua 8-bit <i>Timer</i>	Tidak bekerja

Sumber : *MCS-51 Mikrokontroler User's Manual*

Pada mikrokontroler terdapat tambahan *Timer 2*. *Timer* yang lain adalah *Timer 0* dan *Timer 1*. *Timer 2* ini merupakan *Timer/Counter* 16-bit dan memiliki 3 *mode* operasi yaitu capture, auto-reload (up-down counting) dan *baund rate generator*. Untuk memilih *mode* ini dilakukan dengan mengatur bit pada SFR *T2CON* (*Timer 2 Control Register*). *Timer 2* ini terdiri dari 2 buah *Timer* 8-bit register yaitu *TH2* dan *TL2*. Pada fungsi *Timer*, register *TL2* dinaikkan (*increament*) tiap siklus mesin. Karena siklus mesin terdiri dari 12 periode osilasi, maka *count rate* menjadi 1/12 dari frekuensi osilator. Sedangkan pada fungsi *Counter*, register dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pin yang bersesuaian (dalam hal ini pin *T2* atau *P1.0*). Tabel berikut menunjukkan *mode* operasi yang dapat dijalankan pada *Timer 2*.

Tabel 2.6. Mode Operasi Timer 2

RCLK + TCLK	CP/RL2	TR2	MODE
0	0	1	<i>16-bit auto-reload</i>
0	1	1	<i>16-bit capture</i>
1	X	1	<i>Baund Rate Generator</i>
X	X	0	<i>Off</i>

Sumber: *ATMEL Datasheet Book*

Keterangan:

- RCLK = *Receive Clock Enable*.

Jika diset menyebabkan serial *port* menggunakan pulsa *Overflow Timer 2* sebagai detak penerimaan pada serial *port*. Jika RCLK = 0, maka *Timer 1* yang digunakan

- TCLK = *Transmit Clock Enable*.

Jika diset menyebabkan serial *port* menggunakan pulsa *Overflow Timer 2* sebagai detak pengiriman. Jika TCLK = 0, maka pulsa *Overflow Timer 1* yang digunakan

- CP/RL2 = *Pemilihan Capture/Reload*.

Jika diset maka proses *capture* yang terjadi sedangkan jika bit ini diclear maka proses *reload*

2.6.8. Sistem Interupsi

AT89S8252 memiliki 6 buah sumber Interupsi, 2 eksternal Interupsi (INT0 dan INT1), 3 *Timer* Interupsi (*Timer 0,1 dan 2*) dan satu serial *port* Interupsi.

INT0 = Interupsi pada P3.2 (kaki 12)

INT1 = Interupsi pada P3.3 (kaki 13)

Timer 0 = Timer pada P3.4 (kaki 14)

Timer 1 = Timer pada P3.5 (kaki 15)

Port serial = jika pengiriman/penerimaan suatu *frame* telah lengkap

Saat terjadinya Interupsi, mikrokontroler secara otomatis akan menuju *subroutine* pada alamat tersebut. Setelah Interupsi *service* selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Dua sumber eksternal adalah INT0 dan INT1, kedua interupsi eksternal akan aktif, transisi tergantung isi IT0 dan IT1 pada *register* TCON. Interupsi T0 dan T1 aktif pada saat *Timer* yang sesuai mengalami *roll over*. Interupsi serial akan dibangkitkan dengan melakukan operasi OR pada R1 dan T1 tiap-tiap sumber interupsi dapat *Enable* atau *disable* secara *software*. Tingkat prioritas semua sumber interupsi dapat diprogram sendiri-sendiri dengan *set* atau *clear bit* pada SFR IP (Interupsi *priority*). *Registe* yang akan berperan dalam mengatur aktif tidaknya interupsi adalah Interupsi *Enable Register*.

2.6.9. Reset

Rangkaian *Power on reset* diperlukan untuk *mereset* mikrokontroler secara otomatis setiap catu daya *on*. Ketika catu daya diaktifkan, rangkaian *reset* menahan logika tinggi pin RST dengan jangka waktu yang ditentukan oleh besarnya pengisian muatan C.

2.7. Transistor Bipolar

Prinsip kerja transistor adalah arus bias basis-emitor yang kecil mengatur besar arus kolektor-emitor.

Bagian penting berikutnya adalah bagaimana caranya memberi arus bias yang tepat sehingga transistor dapat bekerja optimal.

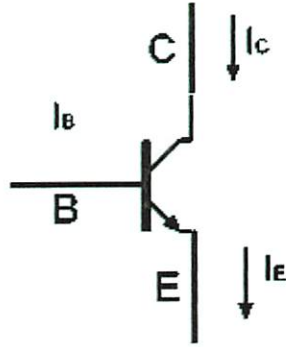
2.7.1. Arus bias

Ada tiga cara yang umum untuk memberi arus bias pada transistor, yaitu rangkaian CE (*Common Emitter*), CC (*Common Collector*) dan CB (*Common Base*). Namun saat ini akan lebih detail dijelaskan bias transistor rangkaian CE. Dengan menganalisa rangkaian CE akan dapat diketahui beberapa parameter penting dan berguna terutama untuk memilih transistor yang tepat untuk aplikasi tertentu. Tentu untuk aplikasi pengolahan sinyal frekuensi *audio* semestinya tidak menggunakan transistor *Power* misalnya.

2.7.2. Arus *Emitter*

Dari hukum Kirchhoff diketahui bahwa jumlah arus yang masuk kesatu titik akan sama jumlahnya dengan arus yang keluar. Jika teorema tersebut diaplikasikan pada transistor, maka hukum itu menjelaskan hubungan :

$$I_E = I_C + I_B \quad (2.2)$$



Gambar 2.12 Arus Emitor

Sumber : www.electroniclab.com, rubrik elka analog, aswan hamonangan

Persamaan (2.1) tersebut mengatakan arus emitor I_E adalah jumlah dari arus kolektor I_C dengan arus basis I_B . Karena arus I_B sangat kecil sekali atau disebutkan $I_B \ll I_C$, maka dapat di nyatakan :

$$I_E = I_C \quad (2.3)$$

2.7.3. Alpha (a)

Pada tabel data transistor (*databook*) sering dijumpai spesifikasi a_{dc} (alpha dc) yang tidak lain adalah :

$$a_{dc} = I_C/I_E \quad (2.4)$$

Defenisinya adalah perbandingan arus kolektor terhadap arus emitor.

Karena besar arus kolektor umumnya hampir sama dengan besar arus emitor maka idealnya besara_{dc} adalah = 1 (satu). Namun umumnya transistor yang ada memiliki a_{dc} kurang lebih antara 0.95 sampai 0.99.

2.7.4. Beta (b)

Beta didefinisikan sebagai besar perbandingan antara arus kolektor dengan arus basis.

$$b = I_C/I_B \quad (2.5)$$

Dengan kata lain, b adalah parameter yang menunjukkan kemampuan penguatan arus (*Current Gain*) dari suatu transistor. Parameter ini ada tertera di *databook* transistor dan sangat membantu para perancang rangkaian elektronika dalam merencanakan rangkaiannya.

Misalnya jika suatu transistor diketahui besar $b=250$ dan diinginkan arus kolektor sebesar 10 mA, maka berapakah arus bias basis yang diperlukan. Tentu jawabannya sangat mudah yaitu :

$$I_B = I_C/b = 10\text{mA}/250 = 40 \mu\text{A}$$

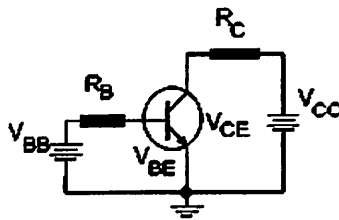
Arus yang terjadi pada kolektor transistor yang memiliki $b = 200$ jika diberi arus bias basis sebesar 0.1mA adalah :

$$I_C = b I_B = 200 \times 0.1\text{mA} = 20 \text{mA}$$

Dari rumusan ini lebih terlihat definisi penguatan arus transistor, yaitu sekali lagi, arus basis yang kecil menjadi arus kolektor yang lebih besar.

2.7.5. Common Emitter (CE)

Rangkaian CE adalah rangkain yang paling sering digunakan untuk berbagai aplikasi yang menggunakan transistor. Dinamakan rangkaian CE, sebab titik *ground* atau titik tegangan 0 Volt dihubungkan pada titik emitor.



Gambar 2.13. Rangkaian CE

Sumber : www.electroniclab.com, rubrik elka analog, aswan hamonangan

Sekilas tentang notasi, ada beberapa notasi yang sering digunakan untuk menunjukkan besar tegangan pada suatu titik maupun antar titik. Notasi dengan 1 *subscript* adalah untuk menunjukkan besar tegangan pada satu titik, misalnya V_C = tegangan kolektor, V_B = tegangan basis dan V_E = tegangan emitor.

Ada juga notasi dengan 2 *subscript* yang dipakai untuk menunjukkan besar tegangan antar 2 titik, yang disebut juga dengan tegangan jepit. Diantaranya adalah:

V_{CE} = tegangan jepit kolektor- emitor

V_{BE} = tegangan jepit basis - emitor

V_{CB} = tegangan jepit kolektor - basis

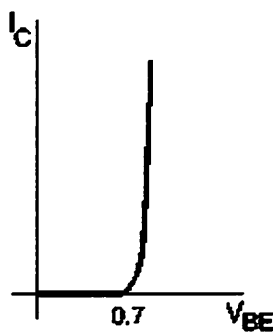
Notasi seperti V_{BB} , V_{CC} , V_{EE} berturut-turut adalah besar sumber tegangan yang masuk ke titik basis, kolektor dan emitor.

2.7.6. Kurva Basis

Hubungan antara I_B dan V_{BE} tentu saja akan berupa kurva dioda. Karena memang telah diketahui bahwa *junction* basis-emitor tidak lain adalah sebuah dioda. Jika hukum Ohm diterapkan pada *loop base* diketahui adalah :

$$I_B = (V_{BB} - V_{BE}) / R_B \quad (2.6)$$

V_{BE} adalah tegangan jepit dioda *junction* basis-emitor. Arus hanya akan mengalir jika tegangan antara basis-emitor lebih besar dari V_{BE} . Sehingga arus I_B mulai aktif mengalir pada saat nilai V_{BE} tertentu.



Grafik 2.1. Kurva $I_B - V_{BE}$

Sumber : www.electroniclab.com, rubrik elka dasar, aswan hamonangan

Besar V_{BE} umumnya tercantum di dalam *databook*. Tetapi untuk penyederhanaan umumnya diketahui $V_{BE} = 0.7$ Volt untuk transistor silikon dan $V_{BE} = 0.3$ Volt untuk transistor germanium. Nilai ideal $V_{BE} = 0$ Volt.

Sampai disini akan sangat mudah mengetahui arus I_B dan arus I_C dari rangkaian berikut ini, jika diketahui besar $b = 200$. Katakanlah yang digunakan adalah transistor yang dibuat dari bahan silikon.

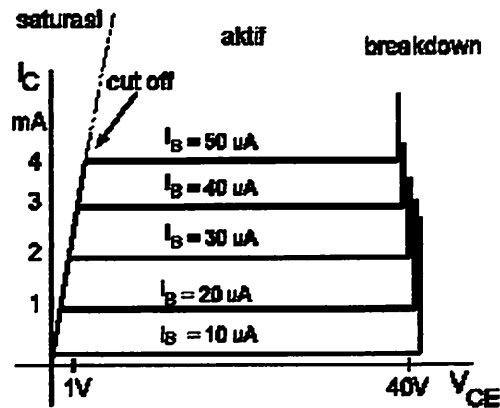
$$\begin{aligned} I_B &= (V_{BB} - V_{BE}) / R_B \\ &= (2V - 0.7V) / 100 K \\ &= 13 \mu A \end{aligned}$$

Dengan $b = 200$, maka arus kolektor adalah :

$$I_C = bI_B = 200 \times 13\mu A = 2.6 \text{ mA}$$

2.7.7. Kurva Kolektor

Sekarang sudah diketahui konsep arus basis dan arus kolektor. Satu hal lain yang menarik adalah bagaimana hubungan antara arus basis I_B , arus kolektor I_C dan tegangan kolektor-emitor V_{CE} . Tegangan V_{BB} dan V_{CC} dapat diatur untuk memperoleh plot garis-garis kurva kolektor. Pada gambar berikut telah diplot beberapa kurva kolektor arus I_C terhadap V_{CE} dimana arus I_B dibuat konstan.



Grafik 2.2. Kurva Kolektor

Sumber : www.electroniclab.com, rubrik elka dasar, aswan hamonangan

Dari kurva ini terlihat ada beberapa region yang menunjukkan daerah kerja transistor. Pertama adalah daerah *saturasi*, lalu daerah *cut-off*, kemudian daerah aktif dan seterusnya daerah *breakdown*.

2.7.8. Daerah Aktif

Daerah kerja transistor yang normal adalah pada daerah aktif, dimana arus I_C konstan terhadap berapapun nilai V_{CE} . Dari kurva ini diperlihatkan bahwa arus I_C hanya tergantung dari besar arus I_B . Daerah kerja ini biasa juga disebut daerah linear (*linear region*).

Jika hukum *Kirchhoff* mengenai tegangan dan arus diterapkan pada *loop* kolektor (rangkaiannya CE), maka dapat diperoleh hubungan :

$$V_{CE} = V_{CC} - I_C R_C \tag{2.7}$$

Dapat dihitung dissipasi daya transistor adalah :

$$P_D = V_{CE} \cdot I_C \quad (2.8)$$

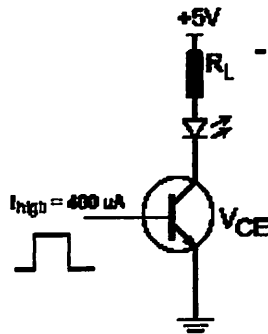
Rumus ini mengatakan jumlah dissipasi daya transistor adalah tegangan kolektor-emitor dikali jumlah arus yang melewatinya. Dissipasi daya ini berupa panas yang menyebabkan naiknya temperatur transistor. Umumnya untuk transistor *Power* sangat perlu untuk mengetahui spesifikasi P_{Dmax} . Spesifikasi ini menunjukkan temperatur kerja maksimum yang diperbolehkan agar transistor masih bekerja normal. Sebab jika transistor bekerja melebihi kapasitas daya P_{Dmax} , maka transistor dapat rusak atau terbakar.

2.7.9. Daerah Saturasi

Daerah saturasi adalah mulai dari $V_{CE} = 0$ Volt sampai kira-kira 0.7 Volt (transistor silikon), yaitu akibat dari efek dioda kolektor-basis yang mana tegangan V_{CE} belum mencukupi untuk dapat menyebabkan aliran elektron.

2.7.10. Daerah *Cut-off*

Jika kemudian tegangan V_{CC} dinaikkan perlahan-lahan, sampai tegangan V_{CE} tertentu tiba-tiba arus I_C mulai konstan. Pada saat perubahan ini, daerah kerja transistor berada pada daerah *cut-off* yaitu dari keadaan saturasi (*OFF*) lalu menjadi aktif (*ON*). Perubahan ini dipakai pada sistem digital yang hanya mengenal angka biner 1 dan 0 yang tidak lain dapat direpresentasikan oleh status transistor *OFF* dan *ON*.



Gambar 2.14. Rangkaian Driver LED

Sumber : www.electroniclab.com, rubrik elka analog, aswan hamonangan

Misalkan pada rangkaian driver LED di atas, transistor yang digunakan adalah transistor dengan $b = 50$. Penyalan LED diatur oleh sebuah gerbang logika (*logic gate*) dengan arus *output high* = $400 \mu\text{A}$ dan diketahui tegangan *forward* LED, $V_{\text{LED}} = 2.4 \text{ Volt}$. Lalu pertanyaannya adalah, berapakah seharusnya resistansi R_L yang dipakai.

$$I_C = bI_B = 50 \times 400 \mu\text{A} = 20 \text{ mA}$$

Arus sebesar ini cukup untuk menyalakan LED pada saat transistor *cut-off*. Tegangan VCE pada saat *cut-off* idealnya = 0, dan aproksimasi ini sudah cukup untuk rangkaian ini.

$$\begin{aligned} R_L &= (V_{CC} - V_{\text{LED}} - V_{\text{CE}}) / I_C \\ &= (5 - 2.4 - 0)\text{V} / 20 \text{ mA} \\ &= 2.6\text{V} / 20 \text{ mA} \\ &= 130 \text{ Ohm} \end{aligned}$$

2.7.11. Daerah *Breakdown*

Dari kurva kolektor, terlihat jika tegangan V_{CE} lebih dari 40V, arus I_C menanjak naik dengan cepat. Transistor pada daerah ini disebut berada pada daerah *breakdown*. Seharusnya transistor tidak boleh bekerja pada daerah ini, karena akan dapat merusak transistor tersebut. Untuk berbagai jenis transistor nilai tegangan $V_{CE\ max}$ yang diperbolehkan sebelum *breakdown* bervariasi. $V_{CE\ max}$ pada *databook* transistor selalu dicantumkan juga.

2.7.12. Datasheet transistor

Sebelumnya telah disinggung beberapa spesifikasi transistor, seperti tegangan $V_{CE\ max}$ dan $P_D\ max$. Sering juga dicantumkan di *datasheet* keterangan lain tentang arus $I_C\ max$, $V_{CB\ max}$ dan $V_{EB\ max}$. Ada juga $P_C\ max$, pada transistor C9012 dicantumkan data-data seperti :

$V_{CB\ max}$ (collector-base voltage)	= -40V
$V_{CE\ max}$ (collector-emitor voltage)	= -20V
$V_{EB\ max}$ (emitor-base voltage)	= -5 V
$I_C\ max$ (collector current)	= -500 mA
$P_C\ max$ (collector dissipation)	= 625 mW
T_{jmax} (junction temperature)	= 150 °C
T_{stg} (storage temperature)	= -55 ~150 °C

2.8. Dioda

Supaya dioda dapat bekerja, maka perlu adanya tegangan yang diberikan pada dioda tersebut. Tegangan itu disebut muka. Tegangan yang diberikan pada dioda ada dua cara, yaitu dengan tegangan muka maju (*forward bias*) dan tegangan muka terbalik (*reverse bias*). Dari pemberian tegangan muka pada dioda tersebut dapat dicari karakteristik dari dioda.

Mengikuti cara pemberian tegangan muka dioda, maka karakteristik diodapun ada dua macam, yaitu karakteristik dioda maju pada saat diberi tegangan muka maju dan karakteristik terbalik pada saat diberi tegangan muka terbalik. Pada karakteristik ini dimaksudkan adalah karakteristik tegangan dan arusnya.

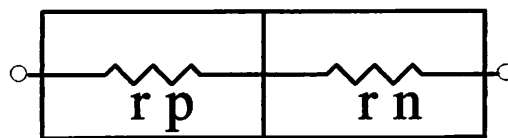
Karakteristik pemberian tegangan muka pada dioda dapat dilihat pada grafik 2.4. Pada saat tegangan maju V_F nol, arus maju I_F masih dalam posisi nol juga. Sedikit demi sedikit tegangan maju ditambah, arus maju I_F masih dalam posisi nol (sangat kecil). Tegangan ditambah terus sampai mencapai 0,7 V dan dioda akan mengalirkan arus I_F secara besar-besaran. Tegangan dimana arus bertambah dengan cepat disebut *tegangan knee* dari dioda. Untuk dioda silikon, tegangan knee sama dengan *potensial barrier* kira-kira 0,7 V.

Diatas *tegangan knee*, pertambahan arus dioda amat cepat, pertambahan tegangan sedikit saja pada dioda menyebabkan pertambahan yang besar pada arus dioda. Alasannya adalah: sesudah melewati *potensial barrier*, semua penghambat arus adalah resistansi daerah p dan n yang dilambangkan sebagai r_p dan r_n dalam gambar 2.20. Karena setiap konduktor mempunyai resistansi, maka kedua daerah

p dan n juga mempunyai resistansi. Jumlah resistansi-resistansi ini disebut resistansi *bulk* dioda.

$$r_B = r_p + r_n \quad (2.9)$$

Nilai resistansi *bulk* r_B tergantung pada doping dan besarnya daerah p dan n, r_B umumnya dari 1 sampai 25 Ohm.

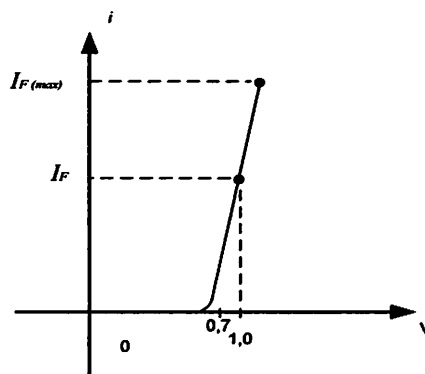


Gambar 2.15. Resistansi Bulk

Sumber : *Prinsip-prinsip Elektronik, Malvino*

Lembar data dari pabrik kerap kali memberikan arus forward I_F pada 1 V (grafik 2.3). Untuk dioda silikon, 0,7 V pertama jatuh pada lapisan pengosongan ; 0,3 V terakhir jatuh pada r_B dari dioda, maka :

$$r_B \cong \frac{0,3}{I_F} \quad (2.10)$$



Grafik 2.3. Arus Forward

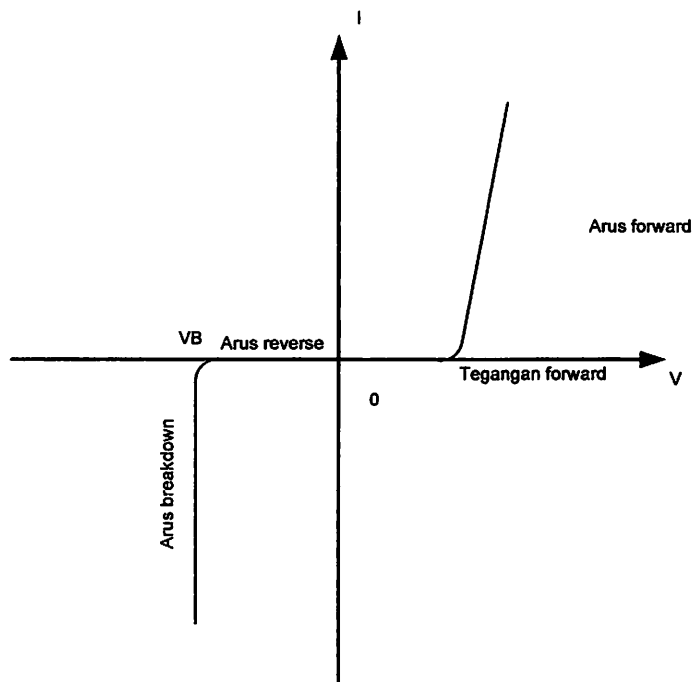
Sumber : *Prinsip-prinsip Elektronik, Malvino*

Dimana I_F adalah arus *forward* DC pada 1 V. (simbol \cong artinya kira-kira sama dengan). Sebagai contoh, dioda silikon 1N456 dengan $I_F = 40 \text{ mA}$ pada 1V. dengan persamaan diatas,

$$r_B \cong \frac{0,3}{40(10^{-3})} = 7,5\Omega$$

Di dalam rangkaian yang menggunakan 1N456, 0,7 pertama muncul pada lapisan pengosongan. Setiap penambahan tegangan dioda, jatuh pada resistansi *bulk* 7,5 Ω .

Bila tegangan muka terbalik sudah sampai V_B , maka akan terjadi tegangan *breakdown*.



Grafik 2.4. Karakteristik Dioda Silikon

Sumber : *Prinsip-prinsip Elektronik, Malvino*

2.9. LCD (*Liquid Crystal Display*)

LCD (*liquid Crystal Display*) adalah suatu jenis piranti *output* yang menggunakan daya rendah dengan pengontrol kontras dan kecerahan. Pengontrol utamanya dan karakter ada pada ROM (*Read Only Memory*) generator dan display data RAM (*Random Access Memory*) yang akan menghasilkan *extended key codes* (kode tombol/*keyboard* standart internasional dalam *Hexsa*) jika padanya diberikan *inputan*. Untuk mendapatkan fungsi dengan baik maka perlu diperhatikan proses inisialisasi yang telah ditentukan oleh pabrik pembuatnya. *Timing* penginisialisasian sangat perlu dipertimbangkan, karena jika meleset sampai orde *millisecond*, maka dapat dipastikan LCD itu tidak dapat berfungsi.

Ada dua jenis *register* yang terdapat dalam LCD M1632 ini, yaitu data *register* dan *instruction register*. Dengan menggunakan pin RS (*Register Select*) pada LCD, pemakaian kedua *register* dapat dipilih. Pemilihan *register* dapat dipilih. Pemilihan *register* pada LCD ditunjukkan dalam table berikut ini :

Tabel 2.7
Pemilihan *Register* Pada LCD M1632

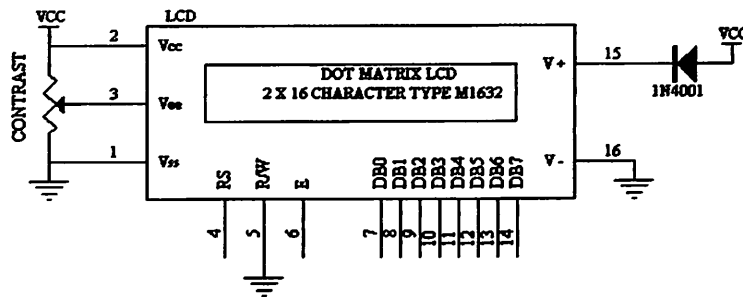
Nama Sinyal	No. Terminal	I/O	Tujuan	Keterangan Sinyal
RS	4	<i>Input</i>	MPU	0 : <i>Instruction Register</i> 0 : <i>Data Register</i>

Sumber : Seiko Instrument Inc, 1987

Jika bagian yang dipilih adalah *instruction register* maka *output* yang dihasilkan adalah meliputi operasional dari LCD, misalnya fungsi *display clear*, *cursor home*, *entry mode set*, *display on/off*, *cursor shift*, dan sejenisnya.

Sebaliknya, jika bagian yang dipilih adalah data *register*, *output* yang dihasilkan adalah meliputi karakter yang tabelnya terdapat pada lampiran data sheet LCD.

Berikut adalah gambar dari LCD dengan pin- pin yang terhubung dengan mikrokontroler AT89S8252 :



Gambar 2.16 Modul LCD 2 ×16 karakter

Sumber : Seiko Instrument Inc, 1987

LCD M1632 mempunyai spesifikasi sebagai berikut :

- ❖ 16 karakter 2 baris dalam bentuk dot matrik 5 × 7 dan kursor.
- ❖ *Duty ratio* 1/16.
- ❖ Memiliki ROM pembangkit karakter untuk 192 jenis karakter.
- ❖ RAM untuk data *display* sebanyak 80 × 8 bit (80 karakter maksimum).
- ❖ Dapat dirangkai dengan MPU (*Mikroprocessor Unit*) 8 bit atau 4 bit.
- ❖ RAM data *display* dan RAM pembangkit karakter dibaca oleh MPU.
- ❖ Memiliki fungsi intruksi : *display ON/OFF*, *cursor ON/OFF*, *display character blink*, *cursor shift* dan *display shift*.
- ❖ Memiliki rangkaian oscillator sendiri.
- ❖ Sumber tegangan tunggal +5 volt.
- ❖ Memiliki rangkaian *reset* otomatis pada catu daya dihidupkan.
- ❖ Temperature operasi 0°-50° C.

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

Tabel 2.8
Fungsi Pin – Pin LCD

No. PIN	Nama PIN	Fungsi
1	Vss	Terminal Ground
2	Vcc	Tegangan Catu + 5 volt
3	Vee	Mengendalikan kecerahan LCD
4	RS	Sinyal pemilihan <i>register</i> 0 = Tulis 1 = Baca
5	R/W	Sinyal seleksi tulis atau baca 0 = Tulis 1 = Baca
6	E	Sinyal operasi awal yang mengaktifkan data tulis atau baca
7 - 14	DB0 – DB7	Merupakan saluran data berisi perintah data yang akan ditampilkan
15	V + BL	Back Light Supply 5 Volt (Volt)
16	V - BL	Back Ligth Supply 0 (Ground)

Sumber : LCD Modul User Manual, Seiko Instrument Ing, 1987

Pada LCD juga terdapat instruksi – instruksi sebagai berikut :

- ❖ *Display clear* : membersihkan tampilan yang ada pada LCD serta menyimpan, sedangkan kursor kembali ke posisi semula.
- ❖ *Cursor home* : hanya membersihkan tampilan dan kursor kembali ke semula.

❖ *Empty mode Set* : layar beraksi sebagai tampilan tulis.

S : 1/0 = menggeser layar.

1/0 : 1 = kursor bergerak ke kanan dan layar bergerak ke kiri.

1/0 : 0 = kursor bergerak ke kiri dan layar bergerak ke kanan

❖ *Display On/Off* kontrol.

D : 1 = layar on

D : 0 = layar off

C : 1 = kursor on

C : 0 = kursor off

B : 1 = kursor berkedip-kedip

B : 0 = kursor tidak berkedip – kedip

Cursor Display Shift

S/C : 1 = LCD diidentifikasi sebagai layar

S/C : 0 = LCD diidentifikasi sebagai kursor

R/L : 1 = menggeser satu spasi ke kanan

R/L : 0 = menggeser satu spasi ke kiri

❖ *Fuction Set*

DL : 1 = panjang data LCD pada 8 bit

DL : 0 = panjang data LCD pada 4 bit

Bit upper ditransfer terlebih dahulu kemudian diikuti dengan 4 bit lower.

N : 1/0 = LCD menggunakan 2 atau 1 baris karakter

P : 1/0 = LCD menggunakan 5 x 10 dot matrik

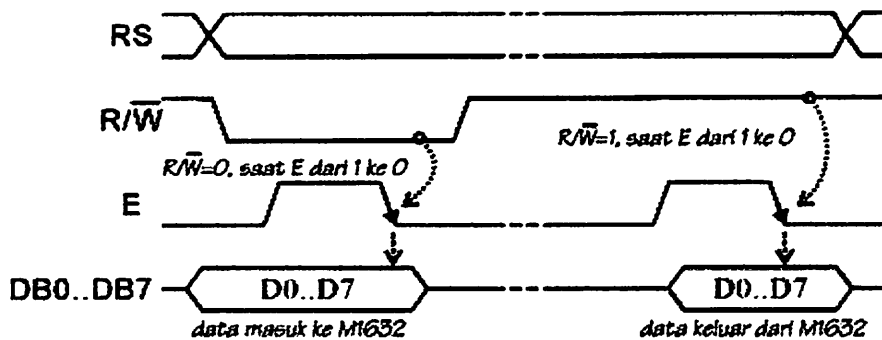
- ❖ *CG RAM address set* : menulis alamat RAM ke karakter
- ❖ *DD RAM address set* : menulis alamat RAM ke tampilan
- ❖ *BF/address set* : BF = 1/0, LCD dalam keadaan sibuk atau tidak sibuk.
- ❖ *Data write to CG RAM or DD RAM* : membaca byte dari alamat terakhir RAM yang dipilih.

2.9.1. Sinyal *interface* M1632

Untuk berhubungan dengan mikrokontroler pemakai 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. Kombinasi lainya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas Intel dengan kombinasi sinyal **WR** dan **RD**. **RS**, singkatan dari *Register Select*, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

Demikian pula saat pengambilan data, saat $RS=0$ data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat $RS=1$ maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

Proses mengirim/mengambil data ke/dari M1632 digambarkan dalam gambar 2-17 bisa dijabarkan sebagai berikut :



Gambar 2.17 Mengirim/Mengambil Data Ke/Dari M1632

Sumber : alds.stts.edu, rubrik analog, STTS Surabaya

1. **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
2. **R/W** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB0..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB0 .. DB7** diterima oleh M1632.
- 3) Untuk mengambil data dari M1632 sinyal **R/W** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada suatu **E** menjadi 1, M1632 akan meletakkan datanya di **DB0 .. DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

BAB III

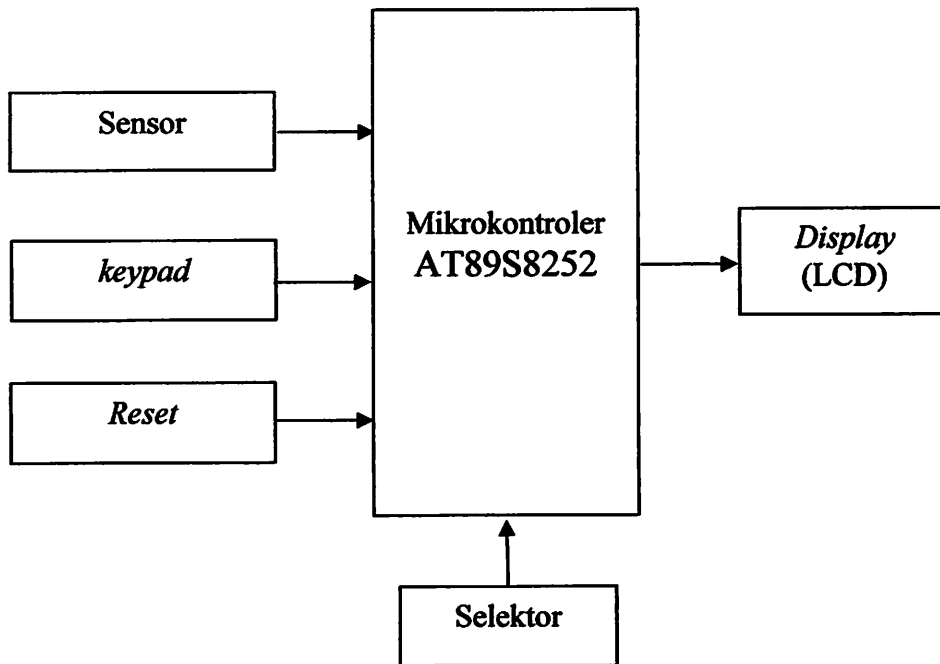
PERENCANAAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Pada Bab ini akan dibahas mengenai Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroler AT89S8252 Menggunakan Sensor *Hall effect (UGN 3503)* dengan Tampilan LCD

Perancangan dan pembuatan alat tersebut meliputi perancangan perangkat keras dan perangkat lunak sebagai sistem pengontrol yang digunakan. Pada perancangan perangkat keras diperlukan alat atau rangkaian penunjang antara lain rangkaian mikrokontroler AT89S8252, sensor *Hall effect (UGN 3503)*, LCD. Sedangkan perangkat lunak, berupa *software* sebagai pengendali perangkat keras yang telah dirancang

Adapun blok diagram sistem dari alat yang akan dirancang adalah sebagai berikut :



Gambar 3.1. Blok Diagram Sistem

Fungsi dari tiap-tiap blok adalah

- **Sensor**
Digunakan untuk mensensor banyaknya air yang mengalir dalam pipa.
- **Reset**
Digunakan untuk *mereset* sistem
- **Mikrokontroler**
Merupakan otak dari sistem ini, dimana mikrokontroler akan menerima data dari sensor kemudian akan mengolah data tersebut sebelum ditampilkan pada *display*.

- *Display*

Display menggunakan LCD yang *type* karakternya 2 x 16 karakter yang berfungsi sebagai *display* jumlah atau volume air, *display* harga, serta *display* komulatif.

- *Keypad*

Keypad menggunakan *Keypad* matrik 4x4, *Keypad* ini berfungsi untuk merubah harga yang telah ada sebelumnya

- Selektor

Terdiri dari beberapa tombol yaitu:

1. Tombol untuk menampilkan volume air
2. Tombol untuk menampilkan Harga dalam Rupiah
3. Tombol untuk menampilkan Jumlah Komulatif
4. Tombol *reset* untuk *mereset* tampilan volume dan tampilan harga
5. Tombol untuk mematikan LCD

Prinsip kerja diagram blok :

Sensor yang digunakan adalah Sensor *Hall effect type UGN 3503*, sensor ini bila di beri catu daya sebesar 5 V dan didekatkan pada sebuah magnet (kutub Selatan) maka akan bekerja (mengeluarkan tegangan) pada kaki outputnya, pada pembuatan alat ini magnet akan dipasang pada ujung baling-baling, jika ada aliran air yang melewati pipa maka baling-baling akan berputar sehingga sensor akan menghasilkan pulsa.

Pulsa yang dihasilkan sensor akan dikirimkan ke mikrokontroler untuk diproses agar pulsa tersebut dapat ditampilkan pada LCD dalam bentuk volume,

jumlah harga, serta jumlah pemakaian kumulatif. Alat ini memiliki tiga tombol selektor yang berfungsi untuk memilih tampilan apa yang ingin kita tampilkan pada LCD serta dilengkapi satu tombol *reset* dan satu tombol untuk mematikan LCD.

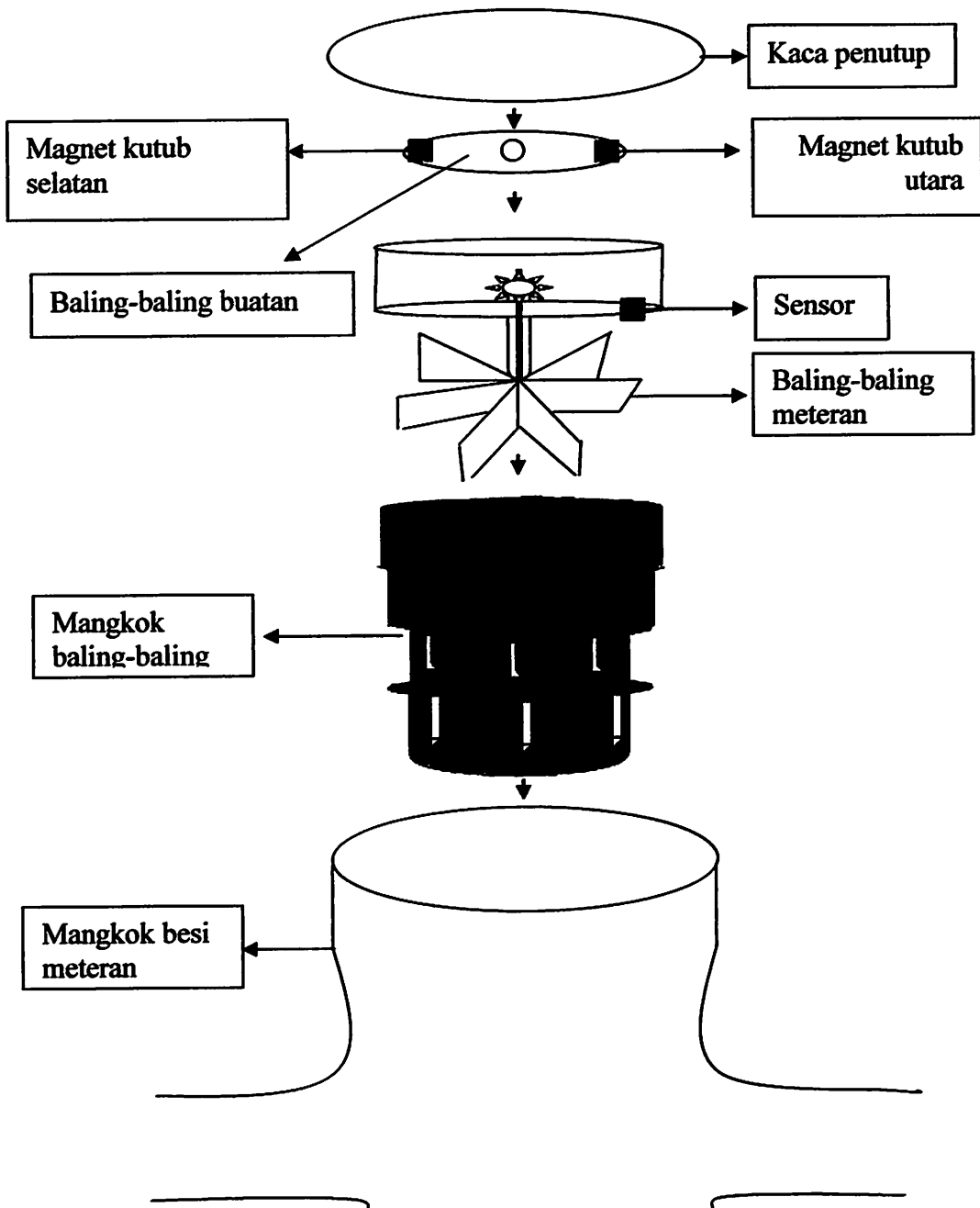
Alat ini juga dilengkapi dengan tombol *reset* yang berfungsi untuk *mereset* pemakaian dalam bentuk volume dan pemakaian dalam bentuk harga ke kondisi awal. Untuk pemakaian kumulatif tidak dapat *direset*, jadi untuk pemakaian kumulatif ini akan bertambah terus dari awal sampai akhir.

Tegangan yang digunakan untuk mensuplay sistem diambil dari jala-jala PLN yang kemudian dimasukkan ke UPS, hal ini bertujuan untuk mengantisipasi agar sistem terus dapat bekerja meskipun jala-jala PLN padam.

3.2. Perencanaan Dan Pembuatan Perangkat Keras

3.2.1 Perancangan Mekanik

Dalam pembuatan alat ini mekaniknya menggunakan meteran air (meteran PDAM) yang dimodifikasi sedemikian rupa agar dapat digunakan sebagai tempat untuk menempatkan sensor serta magnet yang akan digunakan untuk mengaktifkan sensor. Untuk gambar perancangan mekanik adalah sebagai berikut:



Gambar 3.1. Meteran air

Adapun untuk menentukan jumlah putaran baling-baling dalam satu liter adalah dengan cara menghitung jumlah putaran baling-baling dan menampung volume air yang dikeluarkan dalam gelas ukur. Dari percobaan dapat dihasilkan tabel seperti dibawah ini:

Table 4.3 Hasil percobaan untuk 1 liter

No	Jumlah air (L)	Jumlah putaran baling-baling
1	1	19
2	1	20
3	1	21
4	1	20
5	1	21
6	1	20
7	1	20

Rata-rata putaran tiap liter

$$\frac{19+20+21+20+21+20+20}{7} = 20.143 \text{ putaran}$$

Table 4.3 Hasil percobaan untuk 2 liter

No	Jumlah air (L)	Jumlah putaran baling-baling
1	2	38
2	2	41
3	2	40
4	2	40
5	2	42
6	2	40
7	2	41

Rata-rata putaran tiap 2 liter

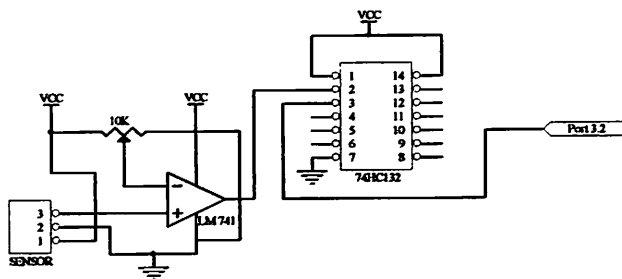
$$\frac{38+41+40+40+42+40+41}{7} = 40.285 \text{ putaran}$$

Dari hasil percobaan diatas maka dapat ditentukan jumlah putaran dalam satu liter adalah 20 putaran, jadi tiap satu liter jumlah pulsa yang dikirimkan sensor adalah 20 pulsa.

Pipa yang digunakan untuk mengalirkan air ke meteran air mempunyai ukuran 1/2", dan dilengkapi dengan *stop kran* yang digunakan untuk memutus aliran air jika sewaktu-waktu diperlukan.

3.2.2. Perancangan Rangkaian Sensor

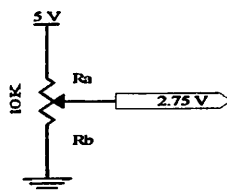
Rangkaian sensor digunakan untuk mengolah data yang dihasilkan oleh sensor agar data tersebut bisa dibaca oleh mikrokontroler, gambar rangkaian sensor adalah sebagai berikut:



Gambar 3.2 Rangkaian sensor

Rangkaian sensor terdiri dari sensor *Hall effect UGN 3503*, komparator 741, *Schmitt Trigger 74HC132*, serta V_r 10 K Ω . Dari beberapa komponen tersebut masing-masing mempunyai fungsi sebagai berikut:

- Sensor *Hall effect UGN 3503*. Sensor ini bila diberi tegangan input sebesar 5 V, maka akan menghasilkan tegangan output sebesar setengah dari tegangan input yaitu sebesar 2.5 V, tetapi bila diberi medan magnet (kutub Selatan) maka tensiangannya akan naik sebesar 3 V.
- Komparator 741 berfungsi membandingkan tegangan yang dihasilkan oleh sensor, kerana perbedaan tegangan yang dikeluarkan oleh sensor sangat kecil yaitu sebesar 0.5 V
- Vr 10 K Ω berfungsi untuk mengatur tegangan pembanding yang akan dimasukkan ke komparator, besar tegangan yang dikeluarkan oleh Vr sebesar 2.75V.



Gambar 3.3 Rangkaian *Trimport*

Dari gambar 3.3 diatas dapat diketahui besar Ra dan Rb dengan menggunakan persamaan dibawah ini:

$$V_o = \frac{R_b}{R_a + R_b} V_{cc} \quad (3.1)$$

$$2.75 = \frac{R_b}{R_a + R_b} 5$$

Maka didapatkan $R_a = 0.82 \Omega$

$$R = R_a + R_b \quad (3.2)$$

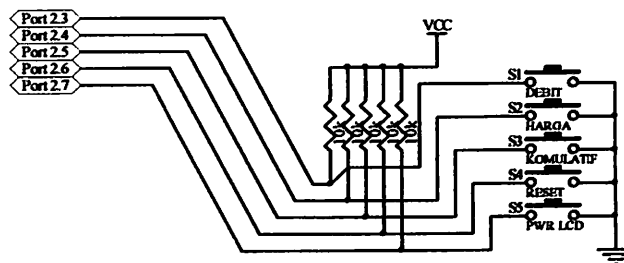
$$R_b = 10 \text{ k}\Omega - 0.82 \Omega$$

$$= 999.18 \Omega$$

- *Schmitt Trigger* 74LS132 berfungsi menerima data yang telah dikeluarkan oleh komparator 741 yang kemudian data tersebut di *NAND*-kan sebelum dimasukkan ke *port* 3.2 pada Mikrokontroller.

3.2.3. Perancangan Rangkaian Selektor

Rangkaian selektor terdiri dari lima buah tombol *push button* yang akan digunakan untuk memilih tampilan yang akan ditampilkan pada LCD, kelima tombol selektor tersebut dilengkapi dengan lima buah resistor *pull-up* sebesar 10K kemudian dihubungkan kemikrokontroller *port* 2.3 – 2.7, berikut gambar rangkaian selektor.



Gambar 3.4. Rangkaian Selektor

Dalam gambar 3-4 rangkaian selektor dipasang tahanan *pull up* yang dihubungkan ke masukan *port* 2.3 - 2.7. Dengan melihat lembar data pada IC mikrokontroler AT89S8252, maka tahanan *pull up* minimum dapat dihitung :

$$R_{\text{pull up}} = \frac{V_{CC} - V_{OL}}{I_{OL}} \quad (3-3)$$

$$R_{\text{pull up}} = \frac{5 - 0,45}{1,6 \cdot 10^{-3}} = 2843,75 \Omega$$

Dalam perancangan $R_{\text{pull up}} = 10 \text{ k}\Omega$ sehingga arus yang mengalir ketika logika rendah dapat diketahui, yaitu :

$$I_{\text{OL}} = \frac{5 - 0,45}{10000} = 0,455 \text{ mA}$$

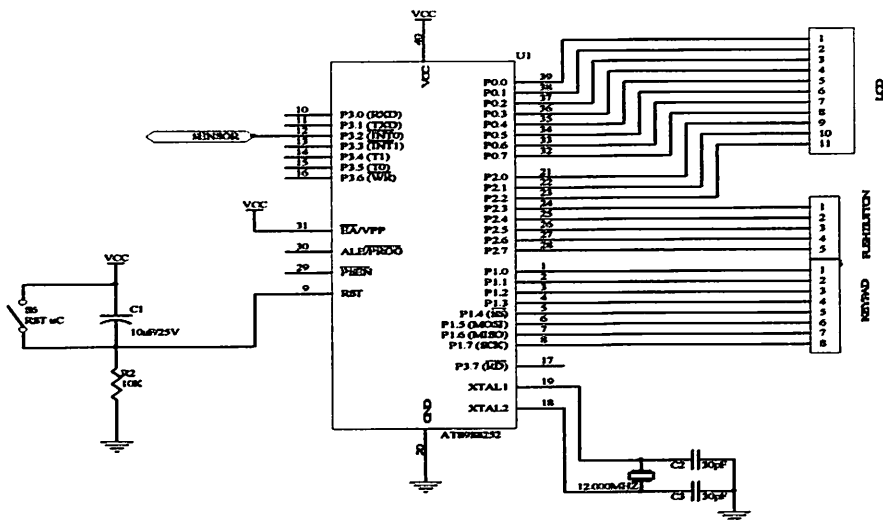
Jadi dengan nilai $R_{\text{pull up}} = 10 \text{ k}\Omega$, maka arus yang terserap ketika keluaran rendah tidak lebih besar dari batas arus I_{OL} nya (umumnya sebesar 1,6 mA).

3.2.4. Perancangan Rangkaian Keypad

Tombol masukan (*Keypad*) yang digunakan adalah *Keypad* matrik 4x4. *Keypad* ini berfungsi untuk merubah harga yang telah ada sebelumnya, *Keypad* dihubungkan dengan mikrokontroller *port* 1.0 – 1.7.

3.2.5. Sistem Mikrokontroler AT89S8252

Komponen utama dari rangkaian ini adalah IC mikrokontroller AT89S8252 dan selanjutnya adalah merencanakan pemakaian tiap-tiap *port* pada IC tersebut. Perancangan pemakaian *port* dapat dilihat dalam gambar berikut :



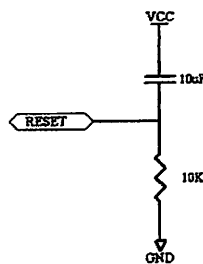
Gambar 3.5. Rangkaian Mikrokontroler AT89S8252

Port-port yang digunakan dalam sistem adalah:

- Pin 12 (*port* 3.2) digunakan sebagai inputan data dari sensor
- Pin 32 - 39 (*Port* 0.0 – 0.7) digunakan untuk mengeluarkan data ke LCD, dan pin 21 – 22 (*port* 2.0 – 2.1) digunakan sebagai RS dan Enable LCD.
- Pin 32 (*port* 2.2) digunakan untuk menghidupkan dan mematikan *backlight* LCD.
- Pin 24 – 28 (*Port* 2.3 – 2.7) digunakan sebagai inputan dari *push button*
- Pin 1 – 8 (*Port* 1.0 – 1.7) digunakan sebagai inputan dari *Keypad*
- Pin 18, 19 (XTAL1 dan XTAL2) digunakan untuk *clock*.
- Pin 9 (*RESET*) , *Reset* aktif tinggi yang terhubung dengan rangkaian *power on reset* dan jika diaktifkan akan *mereset* mikrokontroler.
- Pin 20 (GND) digunakan sebagai *ground*.
- Pin 40 (V_{CC}) digunakan sebagai tegangan sumber.

3.2.5.1.. Rangkaian *Reset*

Untuk *mereset* mikrokontroler AT89S8252, maka pin *reset* diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal *reset*, kapasitor dihubungkan dengan V_{CC} dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam gambar dibawah ini sebagai berikut :



Gambar 3.6. Rangkaian *Reset* pada Mikrokontroler AT89S8252

Sumber: ATMEL Datasheet Book

Rangkaian ini terbentuk oleh komponen R dan C yang sudah ditetapkan oleh ATMEL. Nilai R yang dipakai adalah 10 KΩ dan untuk C sebesar 1µF.

$$F_o = \frac{1}{1,1.RC} \quad (3.4)$$

$$F_o = \frac{1}{1,1.10 \times 10^3 . 10 \times 10^{-6}}$$

$$F_o = 9,09 \text{ Hz.}$$

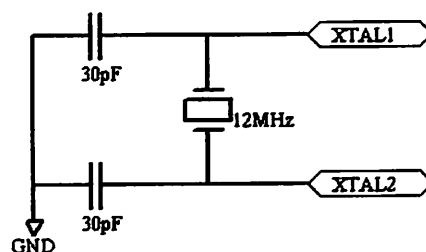
$$\text{Maka periode clock} = \frac{1}{F} \quad (3.5)$$

$$T = \frac{1}{9,09}$$

$$T = 0,11 \text{ detik}$$

3.2.5.2. Rangkaian *Clock*

Kecepatan proses yang diperlukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Mikrokontroler AT89S8252 memiliki internal *clock* generator yang berfungsi sebagai sumber *clock* yang diperlukan. Untuk sistem *clock*nya dipasang kristal dan resonator keramik yang berfungsi sebagai pembangkit *clock* osilator yang ada pada mikrokontroler. Rangkaian ini terdiri dari dua buah kapasitor dan sebuah kristal yang memiliki ketentuan range kristal yang digunakan dari 2 MHz - 24 MHz dengan kapasitor resonator dari 27 pF - 33 pF. Untuk mengendalikan frekuensi osilatornya cukup dengan menghubungkan kristal pada pin 19 (XTAL1) dan pin 18 (XTAL2) serta dua buah kapasitor ke *ground*.



Gambar 3.7. Rangkaian *Clock* pada Mikrokontroler AT89S8252

Sumber: ATMEL Datasheet Book

Untuk nilai arus yang melalui mikrokontroler telah memiliki ketentuan, yakni untuk arus *input* pada *port* 1, 2, 3 saat $V_{in} = 0,45\text{ V}$ adalah sebesar $50\ \mu\text{A}$. Sedangkan untuk arus transisi pada *port* 1, 2, 3 adalah sebesar $-650\ \mu\text{A}$ saat $V_{in} = 2\text{ V}$, $V_{cc} = 5\text{ V}$ dengan toleransi sebesar 10%.

Nilai arus maksimum yang ada per *port* pin adalah sebesar 10 mA.

Sedangkan arus *output* maksimum per 8-bit *port* adalah sebesar :

Port 0 = 26 mA

Port 1, 2, 3 = 15 mA

Total arus maksimum untuk semua keluaran pin adalah sebesar 71 mA dan V_{cc} minimum untuk *power down* adalah 3 V.

3.2.6. Rangkaian LCD M1632

LCD *Display Module M1632* buatan *Seiko Instrument Inc.* adalah komponen *display* yang paling umum digunakan saat ini. LCD M1632 merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing-masing baris bisa menampung 16 huruf/angka.

Untuk berhubungan dengan mikrokontroler, 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.

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

Demikian pula saat pengambilan data, saat RS=0 data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat RS=1 maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

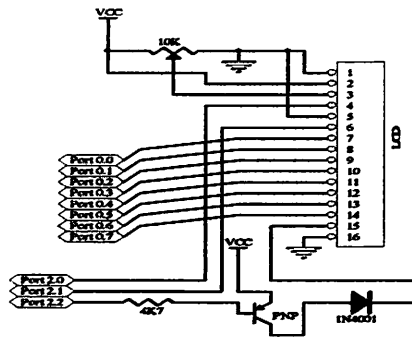
Proses mengirim/mengambil data ke/dari M1632 bisa dijabarkan sebagai berikut :

- RS harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
- R/W* di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di DB0..DB7, sesaat kemudian sinyal E di-satu-kan dan di-nol-kan kembali. Sinyal E merupakan sinyal sinkronisasi, saat E berubah dari 1 menjadi 0 data di DB0 .. DB7 diterima oleh M1632.
- Untuk mengambil data dari M1632 sinyal R/W* di-satu-kan, menyusul sinyal E di-satu-kan. Pada saat E menjadi 1, M1632 akan meletakkan datanya di DB0 .. DB7, data ini harus diambil sebelum sinyal E di-nol-kan kembali.

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali cursor pada baris huruf pertama baris pertama, menghidup/matikan tampilan dan lain sebagainya, semua itu dibahas secara terperinci dalam Lembar Data M1632.

Untuk tampilan dipergunakan LCD *Dot Matrik* 2 x 16 karakter. Sinyal-sinyal yang diperlukan oleh LCD adalah RS dan *Enable*, sinyal RS dan Enable dipergunakan sebagai input yang outputnya dipakai untuk mengaktifkan LCD. LCD akan aktif apabila mikrokontroller memberikan instruksi tulis pada LCD. Saat kondisi RS don't care dan Enable 0 maka LCD tetap pada kondisi semula,

pengiriman data ke LCD dilakukan saat RS berlogika 1 dan enable berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 0 dan Enable 1. Pin LCD ini untuk data terkoneksi pada *Port 0* mikrokontoller. Kemudian untuk RS dihubungkan pada *Port 2.0*, tulis/baca (*Read/Write*) diberikan logika *low* karena disini LCD bersifat menulis data, dan yang terakhir *Enable* (E) dikendalikan dengan *Port 2.1* adapun *back light* dihubungkan dengan *port 2.2*. Gambar rangkaian LCD ditunjukkan pada gambar 3.8 sebagai berikut :



Gambar 3.8 Perancangan *Liquid Crystal Display* (LCD)

Untuk mencari harga Resistansi minimal pada basis transistor 9012, maka dengan data-data pengukuran Hfe transistor didapat :

$$\begin{aligned}
 I_b &= \frac{I_c}{h_{fe}} \\
 &= \frac{500mA}{120} \\
 &= 4.16 \text{ mA}
 \end{aligned}$$

$$R_b = \frac{V_{cc} - V_{Be}}{I_b}$$

$$R_b = \frac{5 - 0.7}{4.16 \text{ mA}}$$

$$= 1.03 \text{ K}\Omega$$

Karena resistansi minimal untuk mengaktifkan transistor sebesar 1.03 K Ω , dan pada perancangan digunakan resistor sebesar 4.7K Ω

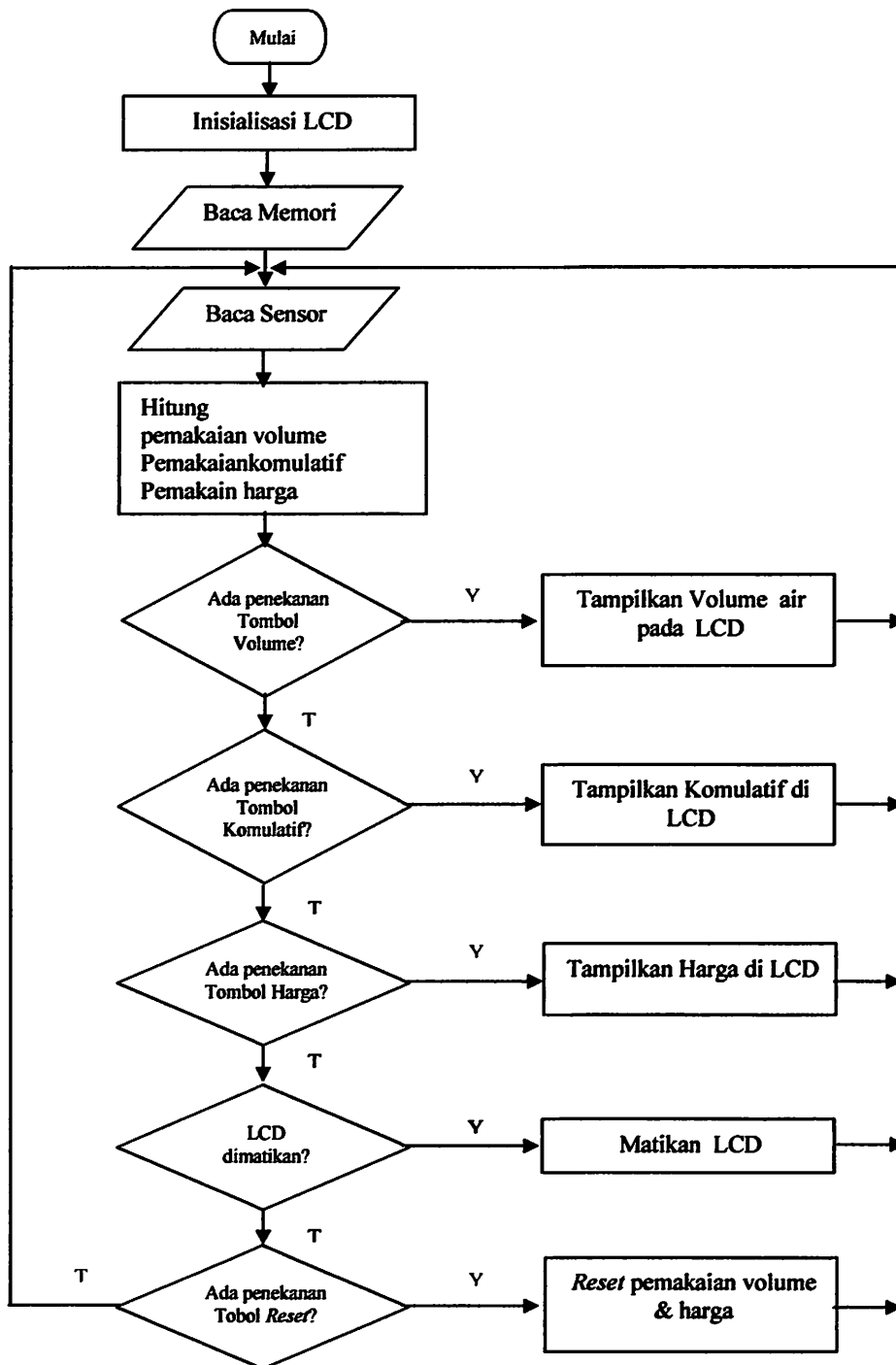
3.2.4. Perencanaan Perangkat Lunak

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

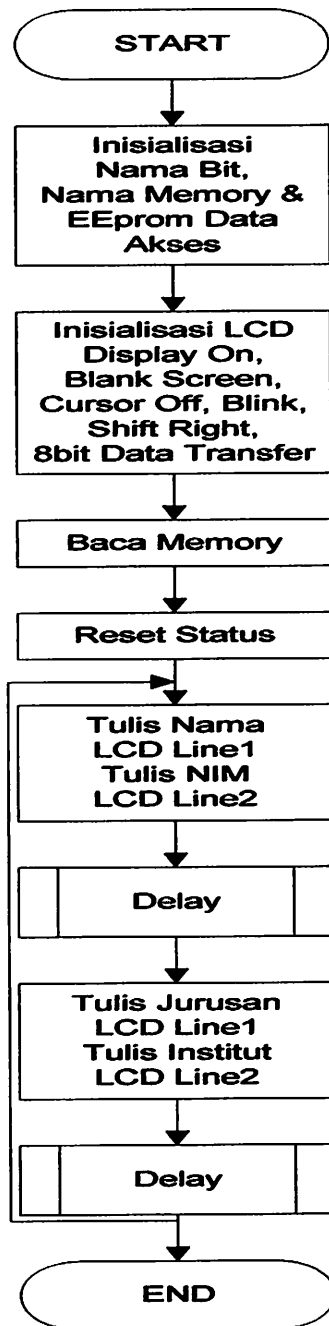
Dalam mikrokontroler memori merupakan fasilitas utama karena disini disimpan perintah-perintah yang harus dikerjakan. Memori disini dapat dibedakan menurut fungsinya menjadi memori program dan memori data.

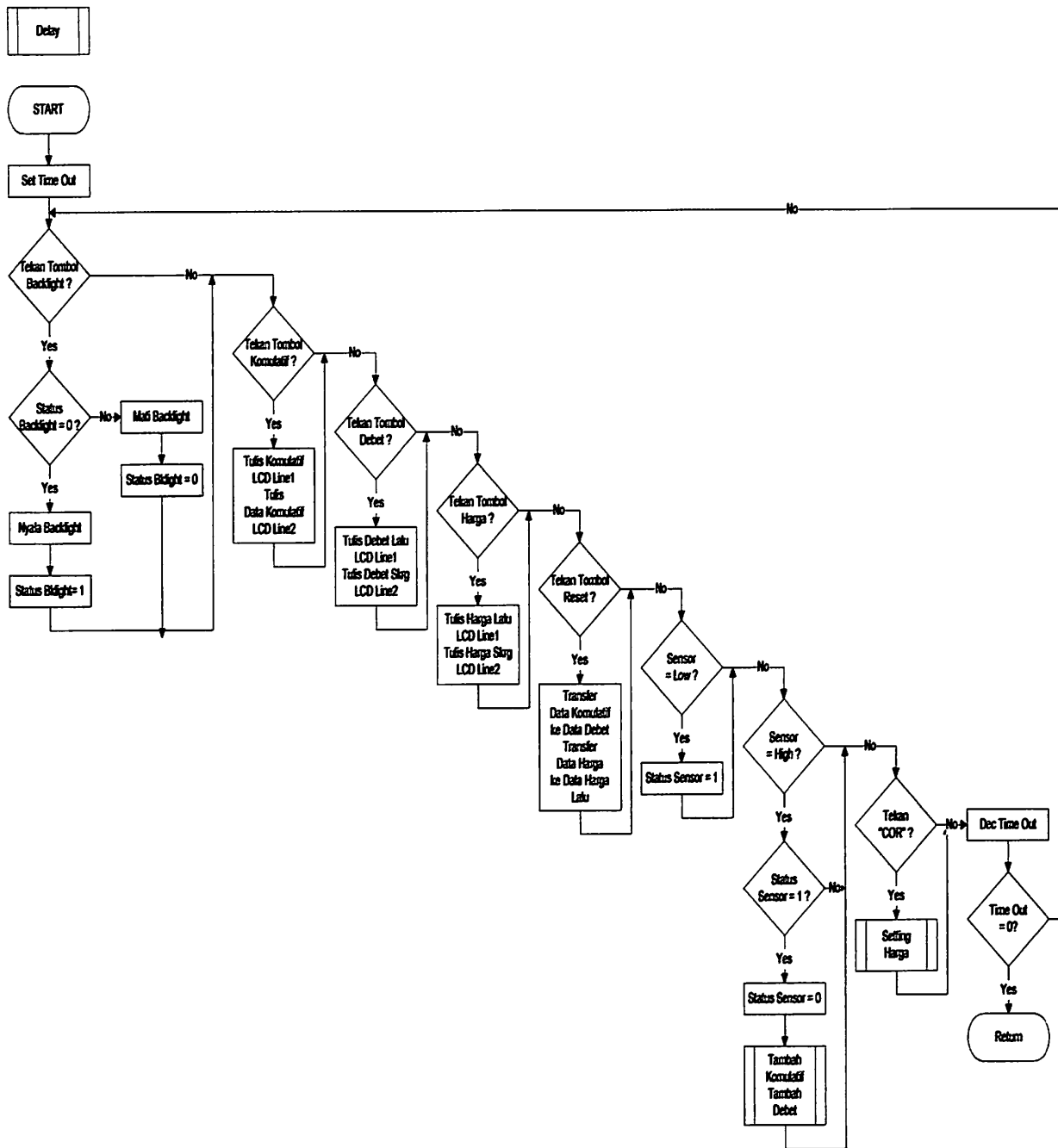
Menurut letaknya memori dapat dibedakan menjadi memori dalam dan memori luar. Memori dalam adalah memori yang disediakan didalam mikrokontroler itu sendiri. Memori luar adalah memori tambahan dari luar mikrokontroler. Memori ini diberikan bila memori didalam mikrokontroler tidak mencukupi untuk menampung semua program dan data. Perencanaan perangkat lunak (*software*) didasarkan perencanaan perangkat keras yang telah dibuat sebelumnya, untuk mendapatkan sistem kerja yang diharapkan *software* dari alat tersebut terdapat di bagian lampiran dan diagram alir (*flowchart*) dari alat tersebut adalah sebagai berikut:

3.2.4.1. Diagram Alir Prinsip Kerja Secara Umum

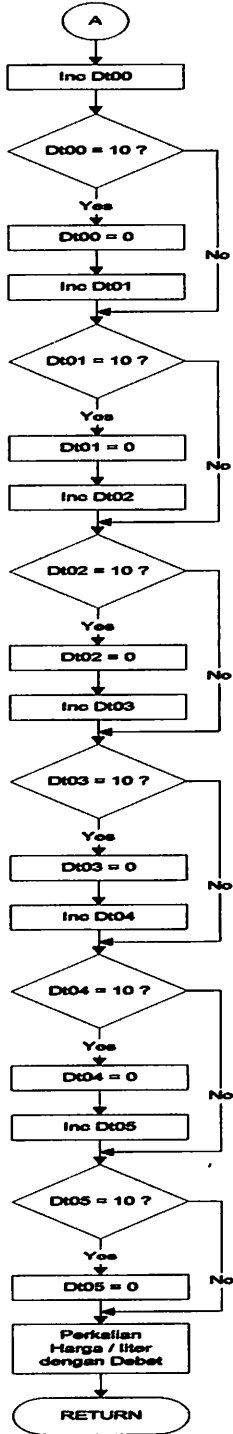
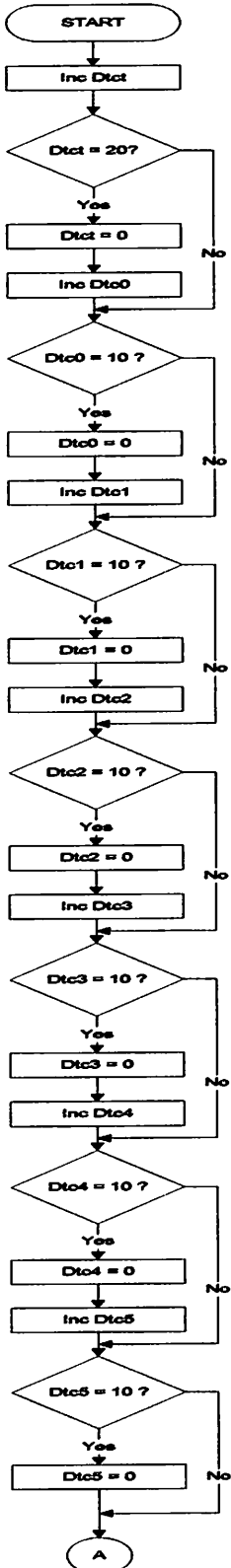


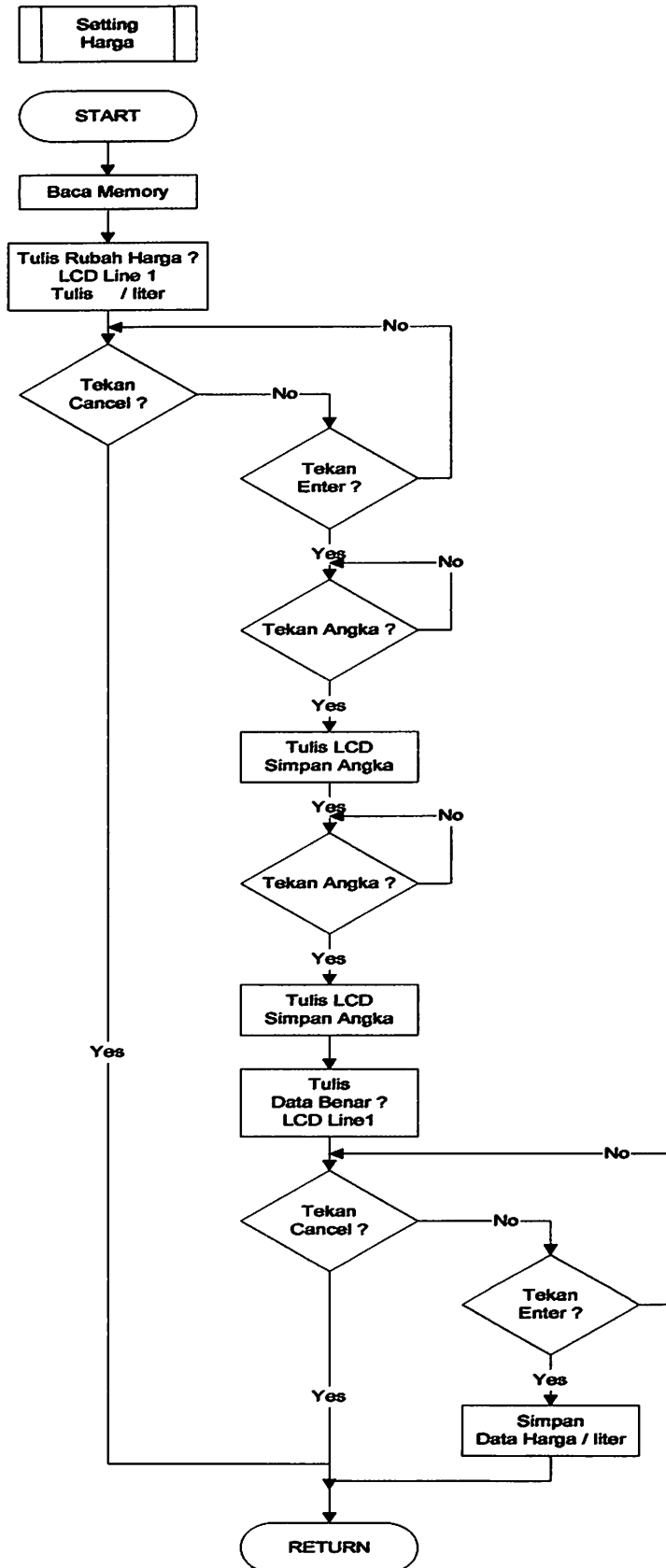
3.2.4.2. Diagram Alir Program





Tambah Komudatif
 Tambah Debet
 Perkalian Harga Debet





BAB IV

PENGUJIAN ALAT DAN PEMBAHASAN HASIL

Pada bab ini dilakukan pengujian dan pengukuran dari sistem yang telah dibuat. Dari pengujian dan pengukuran ini dapat diketahui apakah sistem yang dibuat telah bekerja dengan baik atau masih ada kekurangannya. Pengujian dan pengukuran dilakukan pada tiap-tiap blok untuk mengetahui kerja sistem secara keseluruhan.

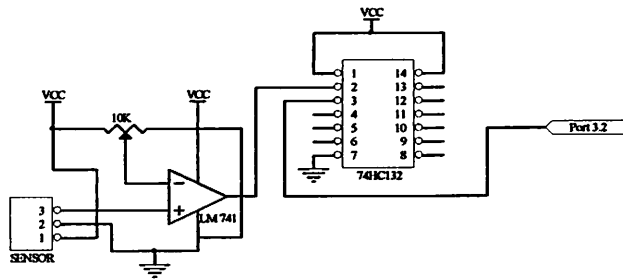
4.1. Pengujian Rangkaian Sensor

4.1.1. Tujuan

Bertujuan untuk mengetahui berapa besar tegangan yang dikeluarkan sensor, serta untuk mengetahui apakah tegangan yang dihasilkan sensor dalam keadaan *high* atau *low*.

4.1.2. Peralatan yang digunakan

- Multimeter Digital
- Rangkaian sensor
- Catu Daya 5 Volt DC
- Catu Daya 6 Volt DC
- Magnet



Gambar 4.1 Rangkaian Sensor

4.1.3. Langkah Pengujian

- Memberikan catu daya sebesar 5 Volt pada sensor
- Mengukur tegangan yang dikeluarkan oleh sensor
- Mendekatkan sensor pada magnet
- Mengukur kembali besar tegangan yang dikeluarkan oleh sensor
- Mencatat tegangan yang dihasilkan sensor dan ditabelkan
- Memberikan catu daya sebesar 6 Volt kekomparoator 741 (Pin 7), serta menghubungkan pin nomor empat ke ground
- Memasukkan tegangan yang dihasilkan sensor kekomparator (Pin 3)
- Memasukkan tegangan pembanding yang dihasilkan oleh Vr sebesar 2.75 Volt kekomparator (pin 2).
- Mengukur tegangan yang dikeluarkan oleh komparator pada pin nomor 6
- Mendekatkan sensor pada megnet, kemudian mengukur kembali tegangan yang dikeluarkan oleh komparator.
- Mencatat tegangan yang dihasilkan sensor dan ditabelkan

- Memberikan catu daya sebesar 5 volt ke schmitt trigger pin nomor 14, dan menghubungkan pin nomor 7 ke ground
- Memasukkan tegangan yang dihasilkan komparator *schmitt trigger* 74HC132 pin nomor dua, serta menghubungkan pin nomor satu ke Vcc
- Mengukur tegangan yang dikeluarkan pada pin nomor tiga
- Mendekatkan sensor pada magnet, kemudian mengukur kembali tegangan yang dikeluarkan oleh *schmitt trigger*
- Mencatat tegangan yang dihasilkan sensor dan ditabelkan

4.1.4. Hasil Pengujian

Tabel 4.1 Hasil Pengujian Rangkaian Sensor

Komponen	Tegangan yang dikeluarkan (V)	
	Tanpa Magnet	Dengan Magnet
Sensor Hall Effect	2.50	3.00
Komparator 741	1.54	4.95
Schmitt Trigger 74HC132	5.00	0.00

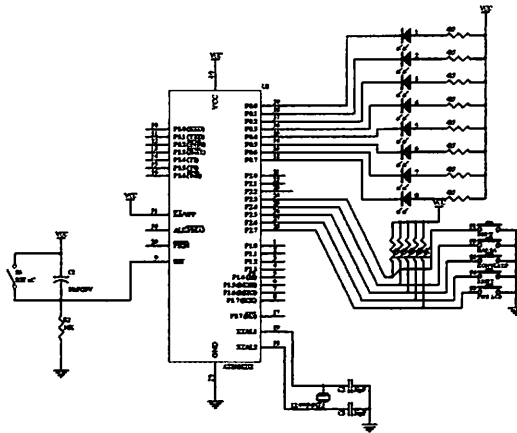
4.2. Pengujian Rangkaian Selektor

4.2.1. Tujuan

Tujuannya adalah untuk mengetahui apakah rangkaian selektor sudah terhubung dengan benar dan untuk mengetahui apakah rangkaian sudah dapat bekerja dengan baik.

4.2.2. Peralatan yang digunakan

- Komputer
- Rangkaian minimum sistem AT89S8252
- Rangkaian selektor
- Delapan buah LED dan delapan buah Resistor $4k7\Omega$
- Catu daya 5 Volt DC



Gambar 4.2 Rangkaian Pengujian Selektor

4.2.3. Langkah Pengujian

- Mempersiapkan *software* mikrokontroler yang berfungsi untuk menyalakan LED bila ada penekanan pada tombol, *software* pengujian rangkaian selektor seperti pada lampiran 1.
- Mengisi mikrokontroler AT89S8252 dengan *software* yang telah dipersiapkan diatas dengan menggunakan *downloader* AT89S8252.
- Mempersiapkan rangkaian pengujian seperti pada gambar 4.2

4.2.4. Hasil Pengujian

Tabel 4.2 Hasil Pengujian Rangkaian Selektor

Tombol yang ditekan	LED yang hidup							
	1	2	3	4	5	6	7	8
S 1	○	○	○	○	○	○	●	●
S 2	○	○	○	○	●	●	○	○
S 3	○	○	●	●	○	○	○	○
S 4	●	●	○	○	○	○	○	○
S 5	●	●	●	●	●	●	●	●

Keterangan: ● (hidup)
○ (mati)

Dari hasil pengujian diatas maka dapat disimpulkan bahwa rangkaian selektor dapat berfungsi dengan baik.

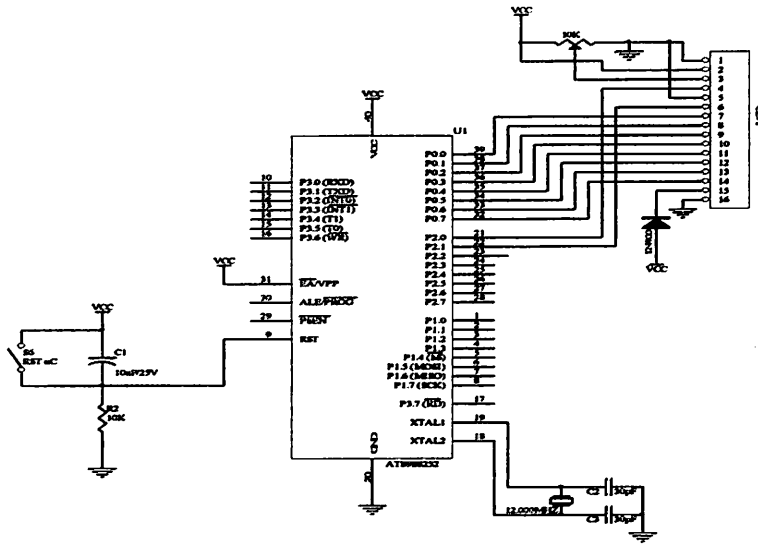
4.3. Rangkaian LCD

4.3.1. Tujuan

Tujuan dari pengujian rangkaian LCD ini adalah untuk mengetahui apakah rangkaian display (LCD) yang dirancang dapat bekerja sesuai dengan apa yang direncanakan.

4.3.2. Peralatan yang digunakan

- komputer
- Minimum sistem AT89S8252
- Catu daya sebesar 5 Volt DC
- Rangkaian LCD



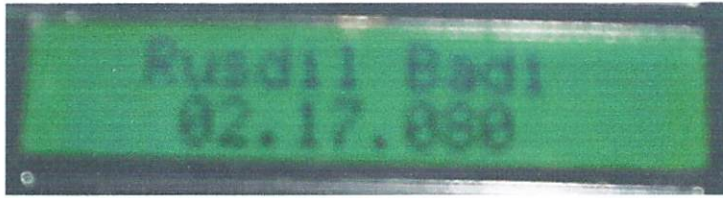
Gambar 4.3 Rangkaian LCD

4.3.3. Langkah Pengujian

- Mempersiapkan *software* mikrokontroler yang berfungsi untuk inialisasi LCD. Sekaligus untuk menuliskan suatu kalimat “RUSDIL BADI” dan 02.17.080, *software* pengujian LCD dapat dilihat pada lampiran 2.
- Mengisi mikrolontroler AT89S8252 dengan *software* yang telah dipersiapkan diatas dengan menggunakan *downloader* AT89S8252
- Mempersiapkan rangkaian penguji seperti pada gambar 4.3

4.3.4. Hasil Pengujian

Setelah melakukan pengujian LCD diatas maka didapatkan tulisan ”coba” pada baris pertama LCD, dengan demikian setelah mengetahui hasil pengujian rangkaian LCD maka dapat disimpulkan bahwa rangkaian LCD tersebut berfungsi dengan baik.



Gambar 4.4 Tampilan pada LCD

4.4. Pengujian keseluruhan Sistem

Pengujian mekanik yang telah direncanakan dan dibuat, bertujuan untuk mengetahui apakah sesuai dengan apa yang direncanakan, caranya dengan melakukan beberapa kali percobaan. Untuk hasil pengujian dapat dilihat pada table dibawah ini.

Table 4.3 Hasil pengujian untuk 1 liter

No	Tampilan volume air pada LCD (L)	Volume air yang dikeluarkan	Kesalahan (Error)
1	1	1.01	0.01
2	1	0.98	-0.02
3	1	1.01	0.01
4	1	1.03	0.03
5	1	1.01	0.01
6	1	0.97	-0.03
7	1	1.01	0.01

Dari data diatas maka dapat ditentukan nilai rata-rata dan persentase kesalahan putaran setiap liternya:

Rata-rata kesalahan tiap liter

$$\frac{0.01+(-0.02)+0.01+0.03+0.01+(-0.03)+0.01}{7} = 0.020 \text{ Liter}$$

Persentase kesalahan tiap liter

$$\frac{0.020}{1} \times 100\% = 2 \%$$

Persentase ketelitian : $100\% - 2 = 98 \%$

Table 4.3 Hasil pengujian untuk 2 liter

No	Tampilan volume air pada LCD (L)	Volume air yang dikeluarkan	Kesalahan (Error)
1	2	1.99	-0.01
2	2	2.02	0.02
3	2	1.98	-0.02
4	2	2.03	0.01
5	2	2.03	0.03
6	2	2.02	0.02
7	2	2.01	0.01

Dari data diatas maka dapat ditentukan nilai rata-rata dan persentase kesalahan putaran untuk 2 liter:

Rata-rata kesalahan untuk 2 liter:

$$\frac{-0.01+0.02+(-0.02)+0.01+0.03+0.02+0.01}{7} = 0.060 \text{ Liter}$$

Persentase kesalahan untuk 2 liter:

$$\frac{0.060}{2} \times 100\% = 3\%$$

Persentase ketelitian : $100\% - 3 = 97\%$

4.5. Cara Pengoperasian Alat

1. Hidupkan saklar ke posisi ON, maka akan muncul tulisan seperti gambar di bawah ini secara bergantian. Kemudian tekan power LCD untuk menyalakan *backlight*.



Gambar 4.5 Tampilan pada LCD pada saat saklar ON

2. Untuk mengetahui jumlah volume air yang dipakai tekan tombol volume, dan akan muncul tampilan seperti dibawah ini



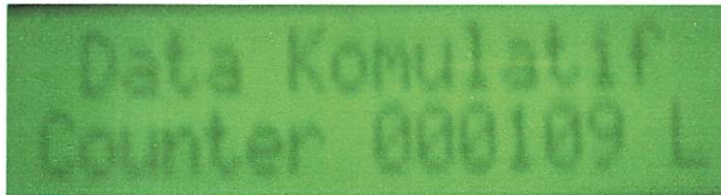
Gambar 4.6 Tampilan pada LCD pada saat saklar volume ditekan

3. Untuk mengetahui jumlah harga maka tekan tombol harga, dan akan muncul tampilan seperti dibawah ini



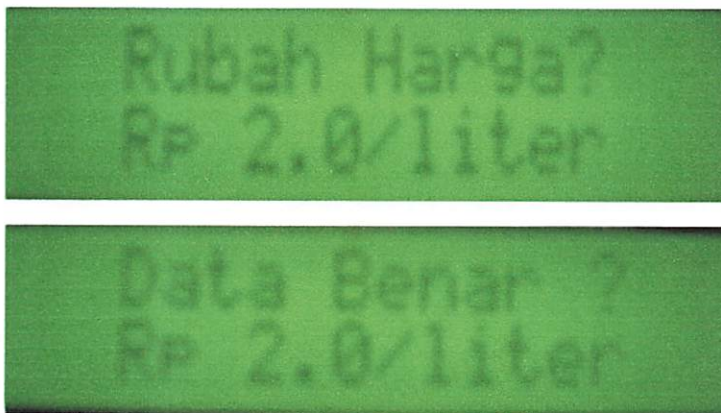
Gambar 4.7 Tampilan pada LCD pada saat tombol harga ditekan

4. Untuk mengetahui jumlah komulatif maka tekan tombol komulatif, dan akan muncul tampilan seperti dibawah ini



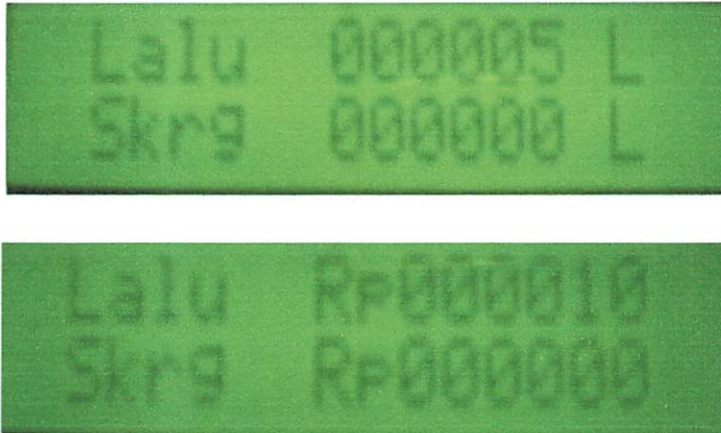
Gambar 4.8 Tampilan pada LCD pada saat tombol komulatif ditekan

5. Untuk merubah harga, pasang *keypad* kemudian tekan tombol COR, kemudian tekan tombol ENT, lalu masukkan harga yang baru dan diakhiri dengan menekan tombol ENT, dan untuk membatalkan perubahan harga maka tekan tombol CAN.



Gambar 4.9 Tampilan pada LCD pada saat proses perubahan harga

6. untuk mereset pemakaian volume air dan harga, tekan tombol reset dan akan muncul tampilan seperti gambar dibawah ini.



Gambar 4.10 Tampilan pada LCD setelah tombol reset ditekan

BAB V

PENUTUP

5.1. Kesimpulan

Setelah dilakukan pengukuran dan percobaan peralatan, maka dapat disimpulkan beberapa hal yang berhubungan dengan kinerja peralatan:

- ❖ Sensor Hall Effect (UGN 3503) dapat mengeluarkan tegangan sebesar setengah dari tegangan input yaitu 2,5 V, dan bila didekatkan pada magnet maka akan mengeluarkan tegangan sebesar 3 V
- ❖ Jarak pemasangan sensor dengan magnet tidak boleh lebih dari 2 milimeter
- ❖ Tegangan keluaran dari trimpot tidak boleh kurang dari 2.5 V dan tidak boleh lebih dari 3 V agar rangkaian komparator dapat membedakan apakah sensor dalam keadaan *low* atau *high*
- ❖ Tegangan yang digunakan untuk mensuplay komparator sebesar 6 V, agar komparator menghasilkan V_o sebesar 5V.
- ❖ Volume air yang dikeluarkan mempunyai rata-rata kesalahan tiap liternya sebesar 0.02 liter atau sebesar 2 %

5.2. Saran

Dengan memandang dari segi kepraktisan suatu peralatan, maka penulis mempunyai beberapa saran untuk pengembangan alat yang dibuat apabila ada pihak yang berminat mengembangkan sekaligus memasarkannya secara luas.

Antara lain ;

- ❖ Alat ini alangkah baiknya dilengkapi dengan alat yang dapat mengirimkan data langsung ke kantor pelayanan
- ❖ Untuk meningkatkan ketelitian alat sebaiknya digunakan lebih dari satu sensor

Daftar Pustaka

- 1) Wasito S, 1984, *Vademekum Elektronika Edisi Kedua*, PT Gramedia Pustaka Utama, Jakarta.
- 2) Ibnu Malik, Moh, 2003, *Belajar Mikrokontroler ATMEL AT89S8252*, Gava Media, Yogyakarta.
- 3) Atmel, 2003, AT89S8252, *8-bit Microcontroller with 8K Bytes Flash*, <http://www.atmel.com>.
- 4) www.atmel.com, *download data sheet AT89S8252*.
- 5) www.allegromicro.com, *download data sheet hall effect UGN 3503*
- 6) www.electroniclab.com, *rubrik elka analog, aswan hamonangan*

LAMPYRAN



FORMULIR PERBAIKAN SKRIPSI

Dari hasil ujian komprehensif jenjang Strata Satu (S-1) Jurusan Teknik Elektro, Konsentrasi Elektronika yang diselenggarakan pada:

Hari : Jum'at
Tanggal : 16 Maret 2007

Telah dilakukan perbaikan skripsi oleh :

Nama : Rusdil Badi'
N.I.M : 02.17.080
Judul : Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroler AT89S8252 Menggunakan Sensor *Hall Effect* (UGN 3503) dengan Tampilan LCD

Perbaikan Meliputi :

No	Materi Perbaikan	Paraf Dosen Penguji
1.	<i>Flowchart</i>	
2.	Kesimpulan	
3	Pengoperasian Alat	
4	Gambar 2.11, 2.7, 2.6	

Disetujui Oleh:

Penguji Kedua

M. Ashar, ST, MT

Dosen Pembimbing I

Ir. Poerwanto, MT
NIP.131574847

Dosen Pembimbing II

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



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

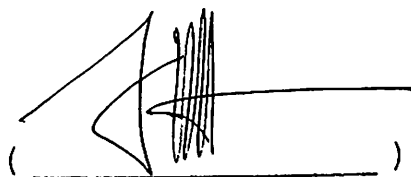
Formulir Perbaikan Ujian Skripsi

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

NAMA : RUSDI BADI'
NIM : 0217086
Perbaikan meliputi :

1. Flowchart
2. Kesimpulan
3. Pengoperasian Alat.
4. Gbr 2.11, 2.7, 2.6.

Malang,

()
M. ASHAR



FORMULIR BIMBINGAN SKRIPSI

Nama : Rusdil Badi'
Nim : 02 17 080
Masa Bimbingan : 12 Desember 2006 s/d 12 Juni 2007
Judul Skripsi : Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroller AT89S8252 Menggunakan Sensor *Hall Effect* (UGN 3503) dengan Tampilan LCD

NO	Tanggal	Uraian	Paraf Pembimbing
1.	21-02-07	Bab I	
		- Latar belakang	
		- Rumusan masalah	
		- Tujuan diringkas	
		Bab II	
		Ref. dicantumkan	
		- Rencana daftar isi	
		ditandatangani	
		Bab III	
		ditandatangani oleh dosen pembimbing	

Malang,.....
Dosen Pembimbing I

Ir. Poerwanto, MT
NIP.P.131.574.847



FORMULIR BIMBINGAN SKRIPSI

Nama : Rusdil Badi'
Nim : 02 17 080
Masa Bimbingan : 12 Desember 2006 s/d 12 Juni 2007
Judul Skripsi : Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroller AT89S8252 Menggunakan Sensor *Hall Effect* (UGN 3503) dengan Tampilan LCD

NO	Tanggal	Uraian	Paraf Pembimbing
2.	28/02-07	Dokumen diperbaiki.	
		- Rancangan Sensor & Transistor.	
		- Rancangan Hasil.	

Malang.....
Dosen Pembimbing I

Ir. Poerwanto, MT
NIP.P.131.574.847



FORMULIR BIMBINGAN SKRIPSI

Nama : Rusdil Badi'
Nim : 02 17 080
Masa Bimbingan : 12 Desember 2006 s/d 12 Juni 2007
Judul Skripsi : Perancangan dan Pembuatan Alat Pencatat Pemakaian Air Berbasis Mikrokontroler AT89S8252 Menggunakan Sensor Hall Effect (UGN 3503) dengan Tampilan LCD

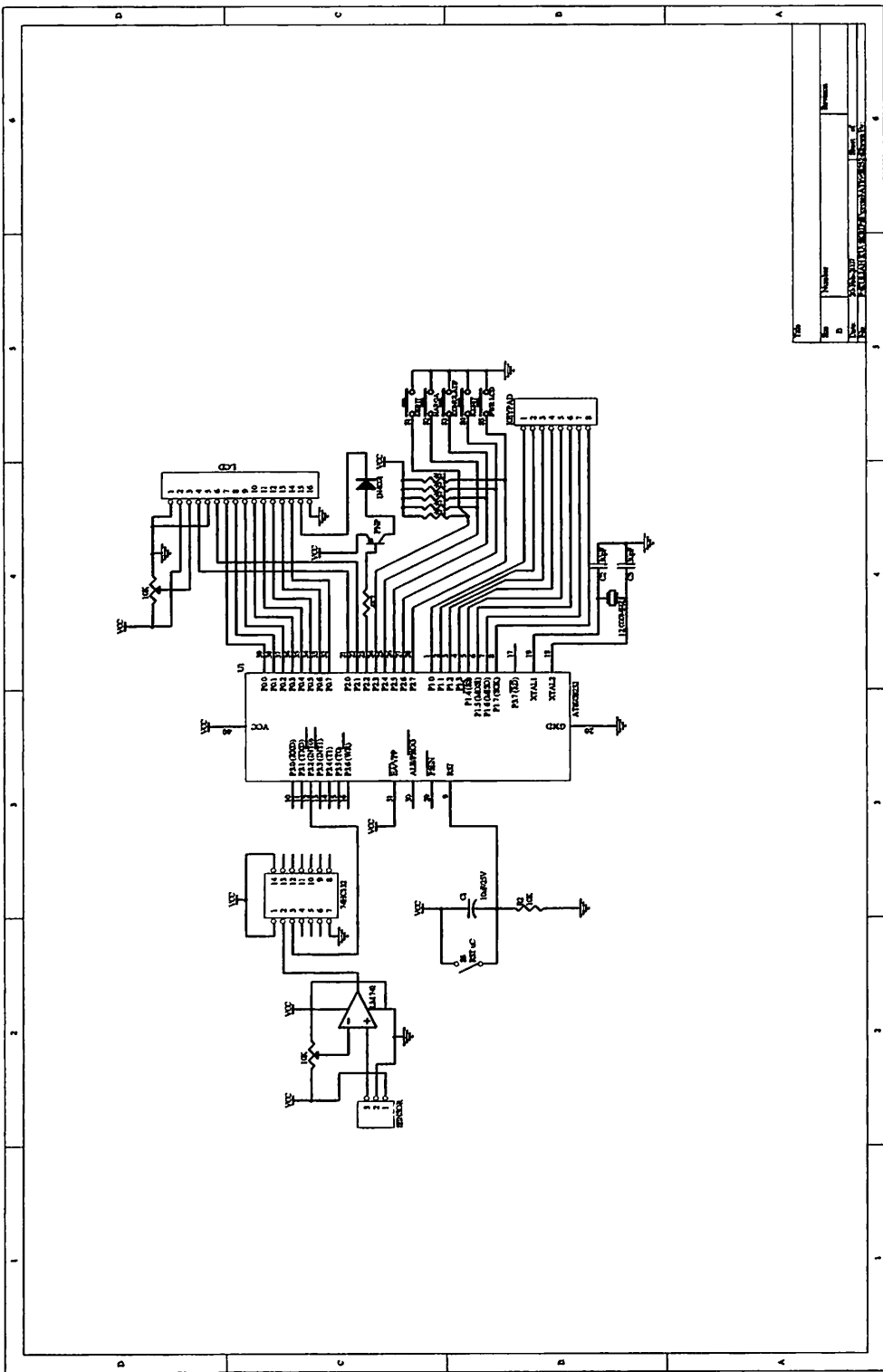
NO	Tanggal	Uraian	Paraf Pembimbing
1.	15/12 ⁰⁶	Materi skripsi	
2.	10/01 ⁰⁷	Konsultasi Range Sensor.	
3.	31/01 ⁰⁷	Konsultasi Bab I, Bab II & Bab III	
4.	19/02 ⁰⁷	Revisi Bab I, Bab II & Bab III	
5.	27/02 ⁰⁷	Komentar Bab IV & Bab V	
6.	01/03 ⁰⁷	Revisi Skripsi	

Malang, 10-03-2007

Dosen Pembimbing II

I Komang Somawirata, ST, MT
NIP 1030100361

Form. S-4a



Rev	Number	Revision
1	10000000	10000000
2	10000000	10000000
3	10000000	10000000
4	10000000	10000000
5	10000000	10000000
6	10000000	10000000
7	10000000	10000000
8	10000000	10000000
9	10000000	10000000
10	10000000	10000000

Lampiran 1

```
org 0h
mulai: mov a,p2
      cjne a,#01111111b,terus
      mov p0,#0
      sjmp mulai

      terus: cjne a,#10111111b,terus1
            mov p0,#0fch
            sjmp mulai

            terus1: cjne a,#11011111b,terus2
                    mov p0,#0f3h
                    sjmp mulai

                    terus2: cjne a,#11101111b,terus3
                              mov p0,#0cfh
                              sjmp mulai

                              terus3: cjne a,#11110111b,mulai
                                        mov p0,#3fh
                                        sjmp mulai
                                        end
```

lampiran 2

```
org 0h
;
Enbl  Bit P2.0
Rest  Bit P2.1
Hurf  Equ 31h
Dly0  Equ 32h
Dly1  Equ 33h
Dly2  Equ 34h
;
init: acall lcd_op
;
mulai: mov  DPTR,#nama
      acall line1
      mov  hurf,#16
      acall tulis
      acall line2
      mov  hurf,#16
      acall tulis
      acall delay1
      sjmp mulai
;
line1:  mov  R0,#80h
```

```

    acall w_ins
    ret
;
line2:  mov  R0,#0C0h
    acall w_ins
    ret
;
tulis: clr  A
    movc  A,@A+DPTR
    mov  R0,A
    inc  DPTR
    acall w_chr
    djnz huruf,tulis
    ret
;
wr_chr: movc  A,@A+DPTR
    mov  R0,A
    acall w_chr
    ret
;
W_ins:  clr  Enbl
    clr  Rest
    mov  P0,R0
    setb Enbl
    clr  Enbl
    acall jeda
    ret
;
W_chr:  clr  Enbl
    setb Rest
    mov  P0,R0
    setb Enbl
    clr  Enbl
    acall jeda
    ret
;
Lcd_op: acall delays
    mov  R0,#03Fh
    acall w_ins
    acall w_ins
    mov  R0,#0Dh
    acall w_ins
    mov  R0,#06h
    acall w_ins
    mov  R0,#01h
    acall w_ins
    mov  R0,#0Ch
    acall w_ins
    acall delays
    ret

```

```
;
jeda: djnz Dly0,$
      ret
;
delays: mov Dly1,#255
dlys:  acall jeda
      djnz Dly1,dlys
      ret
;
delayl: mov Dly2,#20
dlyl:  acall delays
      djnz Dly2,dlyl
      ret
;
nama:  DB 'RUSDIL BADI'
      DB ' 02.17.080 '
;
      end
```

```

1: lampiran 3
2:
3: ;counter 6 digit
4: ;penambahan per 20 put = 1 liter
5: ;harga / liter = Rp1.0 - 1.9
6: ;
7: ;routin tulis nama, nim, ITN malang
8: ;ditekan PWRLCD -> nyala/mati backlight
9: ;ditekan DEBIT -> langsung nyala backlight -> matinya ditekan PWRLCD -> tampil
10: debit 000.000 L -> reset 0 -> yang lama naik
11: ;ditekan HARGA -> langsung nyala backlight -> matinya ditekan PWRLCD -> tampil
12: harga Rp 1.523 -> reset 0 -> yang lama naik
13: ;ditekan KOMULATIF -> langsung nyala backlight -> matinya ditekan PWRLCD ->
14: tampil counter 6 digit -> reset ga ngefek
15: ;keypad hanya merubah data harga
16: ;
17:         org         0h
18: ;
19:         Wmcn         Data 96h           ; data write memory control
20:         Eemn         Equ 00001000b      ; data eeprom read
21:         Eemw         Equ 00010000b      ; data eeprom write
22:         Wtdg         Equ 00000010b      ; data watchdog
23:         Rest         Bit P2.0           ; reset LCD
24:         Enbl         Bit P2.1           ; enable LCD
25:         Bklg         Bit P2.2           ; driver backlight LCD
26:         Tbdb         Bit P2.3           ; tombol debit
27:         Tbhg         Bit P2.4           ; tombol harga
28:         TbkM         Bit P2.5           ; tombol komulatif
29:         Tbrs         Bit P2.6           ; tombol reset
30:         Tbbk         Bit P2.7           ; tombol backlight
31:         Snsf         Bit P3.2           ; sensor flow
32:         Dtct         Equ 30h             ; data counter meter
33:         Dtc0         Equ 31h             ; data counter 0
34:         Dtc1         Equ 32h             ; data counter 1
35:         Dtc2         Equ 33h             ; data counter 2
36:         Dtc3         Equ 34h             ; data counter 3
37:         Dtc4         Equ 35h             ; data counter 4
38:         Dtc5         Equ 36h             ; data counter 5
39:         Dt00         Equ 37h             ; data meter jalan 0
40:         Dt01         Equ 38h             ; data meter jalan 1
41:         Dt02         Equ 39h             ; data meter jalan 2
42:         Dt03         Equ 3Ah             ; data meter jalan 3
43:         Dt04         Equ 3Bh             ; data meter jalan 4
44:         Dt05         Equ 3Ch             ; data meter jalan 5
45:         Dt10         Equ 3Dh             ; data meter lalu 0
46:         Dt11         Equ 3Eh             ; data meter lalu 1
47:         Dt12         Equ 3Fh             ; data meter lalu 2
48:         Dt13         Equ 40h             ; data meter lalu 3
49:         Dt14         Equ 41h             ; data meter lalu 4
50:         Dt15         Equ 42h             ; data meter lalu 5

```

```

51:      Dt20      Equ 43h      ; data harga jalan 0
52:      Dt21      Equ 44h      ; data harga jalan 1
53:      Dt22      Equ 45h      ; data harga jalan 2
54:      Dt23      Equ 46h      ; data harga jalan 3
55:      Dt24      Equ 47h      ; data harga jalan 4
56:      Dt25      Equ 48h      ; data harga jalan 5
57:      Dt30      Equ 49h      ; data harga lalu 0
58:      Dt31      Equ 4Ah      ; data harga lalu 1
59:      Dt32      Equ 4Bh      ; data harga lalu 2
60:      Dt33      Equ 4Ch      ; data harga lalu 3
61:      Dt34      Equ 4Dh      ; data harga lalu 4
62:      Dt35      Equ 4Eh      ; data harga lalu 5
63:      Dbf0      Equ 50h      ; data buffer 0
64:      Dbf1      Equ 51h      ; data buffer 1
65:      Dbf2      Equ 52h      ; data buffer 2
66:      Dbf3      Equ 53h      ; data buffer 3
67:      Dbf4      Equ 54h      ; data buffer 4
68:      Dbf5      Equ 55h      ; data buffer 5
69:      Dbf6      Equ 56h      ; data buffer 6
70:      Dbf7      Equ 57h      ; data buffer 7
71:      Dbf8      Equ 58h      ; data buffer 8
72:      Dbf9      Equ 59h      ; data buffer 9
73:      DbfA      Equ 5Ah      ; data buffer A
74:      DbfB      Equ 5Bh      ; data buffer B
75:      DbfC      Equ 5Ch      ; data buffer C
76:      DbfD      Equ 5Dh      ; data buffer D
77:      DbfE      Equ 5Eh      ; data buffer E
78:      DbfF      Equ 5Fh      ; data buffer F
79:      Dhg0      Equ 60h      ; data harga 0 (per liter)
80:      Dhg1      Equ 61h      ; data harga 1 (per liter)
81:      Hurf      Equ 62h
82:      Dly0      Equ 63h
83:      Dly1      Equ 64h
84:      Dly2      Equ 65h
85: ;
86: init:  acall   lcd_in
87:         acall   rd_mem      ; baca memory
88:         mov     R1,#0        ; reset status sensor flow
89:         mov     R2,#0        ; reset status tombol backlight
90:         mov     R3,#0        ; reset status layer
91: ;
92: mulai:  mov     DPTR,#nama
93:         acall   line1
94:         mov     Hurf,#16
95:         acall   tulis
96:         acall   line2
97:         mov     Hurf,#16
98:         acall   tulis
99:         acall   delay1
100:        acall   line1

```

```

101:      mov     Hurf,#16
102:      acall   tulis
103:      acall   line2
104:      mov     Hurf,#16
105:      acall   tulis
106:      acall   delay1
107:      sjmp    mulai
108: ;
109: bc_sns: jb     Snsf,bcsns0
110:      mov     R1,#1
111: bcsns0: jnb    Snsf,bcsns1
112:      cjne   R1,#1,bcsns1
113:      mov     R1,#0
114:      acall   conter
115: bcsns1: ret
116: ;
117: conter: inc    Dtct
118:      mov     A,Dtct
119:      cjne   A,#20,cntr0
120:      mov     Dtct,#0
121:      acall   dbmetr
122:      inc     Dtc0
123: cntr0: mov     A,Dtc0
124:      cjne   A,#10,cntr1
125:      mov     Dtc0,#0
126:      inc     Dtc1
127: cntr1: mov     A,Dtc1
128:      cjne   A,#10,cntr2
129:      mov     Dtc1,#0
130:      inc     Dtc2
131: cntr2: mov     A,Dtc2
132:      cjne   A,#10,cntr3
133:      mov     Dtc2,#0
134:      inc     Dtc3
135: cntr3: mov     A,Dtc3
136:      cjne   A,#10,cntr4
137:      mov     Dtc3,#0
138:      inc     Dtc4
139: cntr4: mov     A,Dtc4
140:      cjne   A,#10,cntr5
141:      mov     Dtc4,#0
142:      inc     Dtc5
143: cntr5: mov     A,Dtc5
144:      cjne   A,#10,cntr6
145:      mov     Dtc5,#0
146: cntr6: acall   wr_mm0          ; tulis memory
147:      ret
148: ;
149: dbmetr: inc    Dt00
150:      mov     A,Dt00

```



```

151:      cjne    A,#10,metr0
152:      mov     Dt00,#0
153:      inc     Dt01
154: metr0:  mov     A,Dt01
155:      cjne    A,#10,metr1
156:      mov     Dt01,#0
157:      inc     Dt02
158: metr1:  mov     A,Dt02
159:      cjne    A,#10,metr2
160:      mov     Dt02,#0
161:      inc     Dt03
162: metr2:  mov     A,Dt03
163:      cjne    A,#10,metr3
164:      mov     Dt03,#0
165:      inc     Dt04
166: metr3:  mov     A,Dt04
167:      cjne    A,#10,metr4
168:      mov     Dt04,#0
169:      inc     Dt05
170: metr4:  mov     A,Dt05
171:      cjne    A,#10,metr5
172:      mov     Dt05,#0
173: metr5:  ret
174: ;
175: komltf: mov     SP,#07h           ; reset RAM
176:      mov     R3,#1             ; status layer 1
177:      clr     Bklg
178:      mov     R2,#1
179:      mov     DPTR,#tdtkml
180:      acall   line1
181:      mov     Hurf,#16
182:      acall   tulis
183: kmllop: mov     DPTR,#tcontr
184:      acall   line2
185:      mov     Hurf,#8
186:      acall   tulis
187:      mov     DPTR,#tangka
188:      mov     A,Dtc5
189:      acall   wr_chr
190:      mov     A,Dtc4
191:      acall   wr_chr
192:      mov     A,Dtc3
193:      acall   wr_chr
194:      mov     A,Dtc2
195:      acall   wr_chr
196:      mov     A,Dtc1
197:      acall   wr_chr
198:      mov     A,Dtc0
199:      acall   wr_chr
200:      mov     A,#10

```

```

201:      acall   wr_chr
202:      mov     P0,#'L'           ; character L
203:      acall   w_chr
204:      jnb    TbkM,$
205:      acall   delays
206:      sjmp   kmllop
207: ;
208: debit: mov     SP,#07h         ; reset RAM
209:      mov     R3,#1           ; status layer 1
210:      clr     Bklg
211:      mov     R2,#1
212:      mov     DPTR,#tdtdll
213:      acall   line1
214:      mov     Hurf,#7
215:      acall   tulis
216:      mov     DPTR,#tangka
217:      mov     A,Dt15
218:      acall   wr_chr
219:      mov     A,Dt14
220:      acall   wr_chr
221:      mov     A,Dt13
222:      acall   wr_chr
223:      mov     A,Dt12
224:      acall   wr_chr
225:      mov     A,Dt11
226:      acall   wr_chr
227:      mov     A,Dt10
228:      acall   wr_chr
229:      mov     A,#10
230:      acall   wr_chr
231:      mov     P0,#'L'         ; character L
232:      acall   w_chr
233:      mov     A,#10
234:      acall   wr_chr
235: dbtlop: mov     DPTR,#tdtdsk
236:      acall   line2
237:      mov     Hurf,#7
238:      acall   tulis
239:      mov     DPTR,#tangka
240:      mov     A,Dt05
241:      acall   wr_chr
242:      mov     A,Dt04
243:      acall   wr_chr
244:      mov     A,Dt03
245:      acall   wr_chr
246:      mov     A,Dt02
247:      acall   wr_chr
248:      mov     A,Dt01
249:      acall   wr_chr
250:      mov     A,Dt00

```

```

251:      acall   wr_chr
252:      mov     A,#10
253:      acall   wr_chr
254:      mov     P0,#'L'          ; character L
255:      acall   w_chr
256:      mov     A,#10
257:      acall   wr_chr
258:      jnb    Tbdb,$
259:      acall   delays
260:      sjmp   dbtlop
261:      ret
262: ;
263: harga: mov     SP,#07h          ; reset RAM
264:      mov     R3,#1            ; status layer 1
265:      clr    Bklg
266:      mov     R2,#1
267:      mov     DPTR,#tdtdll
268:      acall   line1
269:      mov     Hurf,#7
270:      acall   tulis
271:      mov     DPTR,#tangka
272:      mov     P0,#'R'          ; character R
273:      acall   w_chr
274:      mov     P0,#'p'          ; character p
275:      acall   w_chr
276:      mov     A,Dt35
277:      acall   wr_chr
278:      mov     A,Dt34
279:      acall   wr_chr
280:      mov     A,Dt33
281:      acall   wr_chr
282:      mov     A,Dt32
283:      acall   wr_chr
284:      mov     A,Dt31
285:      acall   wr_chr
286:      mov     A,Dt30
287:      acall   wr_chr
288:      mov     A,#10
289:      acall   wr_chr
290: hrglop: acall   ht_hrg
291:      mov     DPTR,#tdtdsk
292:      acall   line2
293:      mov     Hurf,#7
294:      acall   tulis
295:      mov     DPTR,#tangka
296:      mov     P0,#'R'          ; character R
297:      acall   w_chr
298:      mov     P0,#'p'          ; character p
299:      acall   w_chr
300:      mov     A,Dt25

```

```

301:      acall   wr_chr
302:      mov     A,Dt24
303:      acall   wr_chr
304:      mov     A,Dt23
305:      acall   wr_chr
306:      mov     A,Dt22
307:      acall   wr_chr
308:      mov     A,Dt21
309:      acall   wr_chr
310:      mov     A,Dt20
311:      acall   wr_chr
312:      mov     A,#10
313:      acall   wr_chr
314:      jnb    Tbdb,$
315:      acall   delays
316:      sjmp   hrglop
317:      ret
318: ;
319: rstdta: mov     SP,#07h           ; reset RAM
320:      clr    Bklg
321:      mov    R2,#1
322:      mov    DPTR,#trstdt
323:      acall  line1
324:      mov    Hurf,#16
325:      acall  tulis
326:      acall  line2
327:      mov    Hurf,#16
328:      acall  tulis
329:      mov    Dt10,Dt00
330:      mov    Dt11,Dt01
331:      mov    Dt12,Dt02
332:      mov    Dt13,Dt03
333:      mov    Dt14,Dt04
334:      mov    Dt15,Dt05
335:      mov    Dt30,Dt20
336:      mov    Dt31,Dt21
337:      mov    Dt32,Dt22
338:      mov    Dt33,Dt23
339:      mov    Dt34,Dt24
340:      mov    Dt35,Dt25
341:      acall  wr_mm1
342:      mov    Dt00,#0
343:      mov    Dt01,#0
344:      mov    Dt02,#0
345:      mov    Dt03,#0
346:      mov    Dt04,#0
347:      mov    Dt05,#0
348:      acall  wr_mm0
349:      acall  delays
350:      acall  delays

```

```

351:      setb    Bklg
352:      mov     R2,#0
353:      ljmp    mulai
354: ;
355: seting: mov     SP,#07h          ; reset RAM
356:      clr     Bklg
357:      mov     R2,#1
358:      mov     DPTR,#trbhrq
359:      acall   line1
360:      mov     Hurf,#16
361:      acall   tulis
362:      mov     DPTR,#trpiah
363:      acall   line2
364:      mov     Hurf,#5
365:      acall   tulis
366:      mov     DPTR,#tangka
367:      mov     A,Dhg1
368:      acall   wr_chr
369:      mov     P0,#'.'
370:      acall   w_chr
371:      mov     A,Dhg0
372:      acall   wr_chr
373:      mov     DPTR,#tprltr
374:      mov     Hurf,#8
375:      acall   tulis
376:      acall   tg_lps
377: ;
378: bole0: acall   scnkp
379:      cjne    R0,#11,bole1
380:      setb    Bklg
381:      mov     R2,#0
382:      ljmp    mulai
383: bole1: cjne    R0,#12,bole0
384:      mov     DPTR,#trpiah
385:      acall   line2
386:      mov     Hurf,#5
387:      acall   tulis
388:      mov     DPTR,#tangka
389:      acall   tg_tkn
390:      mov     Dhg1,R0
391:      mov     A,R0
392:      acall   wr_chr
393:      acall   tg_lps
394:      mov     P0,#'.'
395:      acall   w_chr
396:      acall   tg_tkn
397:      mov     Dhg0,R0
398:      mov     A,R0
399:      acall   wr_chr
400:      acall   tg_lps

```

```

401: ;
402:     mov     DPTR,#tdtbnr
403:     acall  line1
404:     mov     Hurf,#16
405:     acall  tulis
406: ;
407: bole2: acall  scnkpd
408:     cjne   R0,#11,bole3
409:     ljmp   mulai
410: bole3: cjne   R0,#12,bole2
411:     acall  wr_mm0
412:     setb  Bklg
413:     mov   R2,#0
414:     ljmp  mulai
415: ;
416: ht_hrg: mov   DbfF,#0           ; reset buffer kali
417:     mov   A,Dt00              ;\
418:     mov   B,Dhg0              ; | Dt00 kali Dhg0
419:     acall rmskli              ; | sisa simpan -> Dbf0
420:     mov   Dbf0,B              ;/
421:     mov   A,Dt01              ;\
422:     mov   B,Dhg0              ; | Dt01 kali Dhg0
423:     acall rmskli              ; | sisa simpan -> Dbf1
424:     mov   Dbf1,B              ;/
425:     mov   A,Dt02              ;\
426:     mov   B,Dhg0              ; | Dt02 kali Dhg0
427:     acall rmskli              ; | sisa simpan -> Dbf2
428:     mov   Dbf2,B              ;/
429:     mov   A,Dt03              ;\
430:     mov   B,Dhg0              ; | Dt03 kali Dhg0
431:     acall rmskli              ; | sisa simpan -> Dbf3
432:     mov   Dbf3,B              ;/
433:     mov   A,Dt04              ;\
434:     mov   B,Dhg0              ; | Dt04 kali Dhg0
435:     acall rmskli              ; | sisa simpan -> Dbf4
436:     mov   Dbf4,B              ;/
437:     mov   A,Dt05              ;\
438:     mov   B,Dhg0              ; | Dt05 kali Dhg0
439:     acall rmskli              ; | sisa simpan -> Dbf5
440:     mov   Dbf5,B              ;/
441:     mov   A,Dt00              ;\
442:     mov   B,Dhg1              ; | Dt00 kali Dhg1
443:     acall rmskli              ; | sisa simpan -> Dbf6
444:     mov   Dbf6,B              ;/
445:     mov   A,Dt01              ;\
446:     mov   B,Dhg1              ; | Dt01 kali Dhg1
447:     acall rmskli              ; | sisa simpan -> Dbf7
448:     mov   Dbf7,B              ;/
449:     mov   A,Dt02              ;\
450:     mov   B,Dhg1              ; | Dt02 kali Dhg1

```

```

451:      acall   rmskli           ; | sisa simpan -> Dbf8
452:      mov     Dbf8,B           ;/
453:      mov     A,Dt03          ;\
454:      mov     B,Dhg1          ; | Dt03 kali Dhg1
455:      acall   rmskli           ; | sisa simpan -> Dbf9
456:      mov     Dbf9,B           ;/
457:      mov     A,Dt04          ;\
458:      mov     B,Dhg1          ; | Dt04 kali Dhg1
459:      acall   rmskli           ; | sisa simpan -> DbfA
460:      mov     DbfA,B           ;/
461:      mov     A,Dt05          ;\
462:      mov     B,Dhg1          ; | Dt05 kali Dhg1
463:      acall   rmskli           ; | sisa simpan -> DbfB
464:      mov     DbfB,B           ;/
465: ;
466:      mov     DbfE,#0          ; reset buffer tambah
467:      mov     A,Dbf1           ;\
468:      mov     B,Dbf6           ; | Dbf1 tambah Dbf6
469:      acall   rmstbh           ; | hasil simpan -> Dt20
470:      mov     Dt20,B           ;/
471:      mov     A,Dbf2           ;\
472:      mov     B,Dbf7           ; | Dbf2 tambah Dbf7
473:      acall   rmstbh           ; | hasil simpan -> Dt21
474:      mov     Dt21,B           ;/
475:      mov     A,Dbf3           ;\
476:      mov     B,Dbf8           ; | Dbf3 tambah Dbf8
477:      acall   rmstbh           ; | hasil simpan -> Dt22
478:      mov     Dt22,B           ;/
479:      mov     A,Dbf4           ;\
480:      mov     B,Dbf9           ; | Dbf4 tambah Dbf9
481:      acall   rmstbh           ; | hasil simpan -> Dt23
482:      mov     Dt23,B           ;/
483:      mov     A,Dbf5           ;\
484:      mov     B,DbfA           ; | Dbf5 tambah DbfA
485:      acall   rmstbh           ; | hasil simpan -> Dt24
486:      mov     Dt24,B           ;/
487:      mov     A,DbfF           ;\
488:      mov     B,DbfB           ; | DbfF tambah DbfB
489:      add     A,B              ; | hasil simpan -> Dt25
490:      mov     Dt25,B           ;/
491:      ret
492: ;
493: rmskli: mul     AB            ;\
494:      mov     B,DbfF           ; | Dt0x kali Dhgx
495:      add     A,B              ; | tambah hasil kali sebelumnya
496:      mov     B,#10            ; | hasil bagi 10
497:      div     AB               ; | hasil simpan -> DbfF
498:      mov     DbfF,A           ;/
499:      ret
500: ;

```

```

501: rmstbh: add    A,B
502:          mov    B,DbfE
503:          add    A,B
504:          mov    B,#10
505:          div   AB
506:          mov    DbfE,A
507:          ret
508: ;
509: line1:  mov    P0,#80h
510:          acall  w_ins
511:          ret
512: ;
513: line2:  mov    P0,#0C0h
514:          acall  w_ins
515:          ret
516: ;
517: tulis:  clr    A
518:          movc   A,@A+DPTR
519:          mov    P0,A
520:          acall  w_chr
521:          inc   DPTR
522:          djnz  Hurf,tulis
523:          ret
524: ;
525: wr_chr: movc   A,@A+DPTR
526:          mov    P0,A
527:          acall  w_chr
528:          ret
529: ;
530: w_ins:  clr    Enbl
531:          clr    Rest
532:          setb  Enbl
533:          clr    Enbl
534:          acall  jeda
535:          ret
536: ;
537: w_chr:  clr    Enbl
538:          setb  Rest
539:          setb  Enbl
540:          clr    Enbl
541:          acall  jeda
542:          ret
543: ;
544: lcd_in: acall  delays
545:          mov    P0,#3Fh                ; function set
546:          acall  w_ins
547:          acall  w_ins
548:          mov    P0,#0Eh                ; display, cursor, blink
549:          acall  w_ins
550:          mov    P0,#06h                ; entry mode

```



```

551:      acall   w_ins
552:      mov     P0,#01h           ; display clear
553:      acall   w_ins
554:      acall   delays
555:      ret
556: ;
557: scnkpd: mov     R0,#10
558:      acall   jeda
559: col1:  mov     P1,#11111110b
560:      mov     A,P1
561: c1b1:  cjne   A,#11101110b,c1b2
562:      mov     R0,#1
563: c1b2:  cjne   A,#11011110b,c1b3
564:      mov     R0,#2
565: c1b3:  cjne   A,#10111110b,c1b4
566:      mov     R0,#3
567: c1b4:  cjne   A,#01111110b,col2
568:      mov     R0,#13
569: ;
570: col2:  mov     P1,#11111101b
571:      mov     A,P1
572: c2b1:  cjne   A,#11101101b,c2b2
573:      mov     R0,#4
574: c2b2:  cjne   A,#11011101b,c2b3
575:      mov     R0,#5
576: c2b3:  cjne   A,#10111101b,c2b4
577:      mov     R0,#6
578: c2b4:  cjne   A,#01111101b,col3
579:      mov     R0,#14
580: ;
581: col3:  mov     P1,#11111011b
582:      mov     A,P1
583: c3b1:  cjne   A,#11101011b,c3b2
584:      mov     R0,#7
585: c3b2:  cjne   A,#11011011b,c3b3
586:      mov     R0,#8
587: c3b3:  cjne   A,#10111011b,c3b4
588:      mov     R0,#9
589: c3b4:  cjne   A,#01111011b,col4
590:      mov     R0,#15
591: ;
592: col4:  mov     P1,#11110111b
593:      mov     A,P1
594: c4b1:  cjne   A,#11100111b,c4b2
595:      mov     R0,#11
596: c4b2:  cjne   A,#11010111b,c4b3
597:      mov     R0,#0
598: c4b3:  cjne   A,#10110111b,c4b4
599:      mov     R0,#12
600: c4b4:  cjne   A,#01110111b,back

```

```

601:      mov      R0,#16
602: back:  ret
603: ;
604: tg_tkn: acall  scnkp
605: tg_tk0: cjne  R0,#16,tg_tk1
606:      sjmp    tg_tkn
607: tg_tk1: cjne  R0,#15,tg_tk2
608:      sjmp    tg_tkn
609: tg_tk2: cjne  R0,#14,tg_tk3
610:      sjmp    tg_tkn
611: tg_tk3: cjne  R0,#13,tg_tk4
612:      sjmp    tg_tkn
613: tg_tk4: cjne  R0,#12,tg_tk5
614:      sjmp    tg_tkn
615: tg_tk5: cjne  R0,#11,tg_tk6
616:      ljmp    mulai
617: tg_tk6: cjne  R0,#10,tg_tk7
618:      sjmp    tg_tkn
619: tg_tk7: ret
620: ;
621: tg_lps: acall  scnkp
622:      cjne  R0,#10,tg_lps
623:      ret
624: ;
625: snctb1: jb    Tbdb,snctb0          ; tombol debet
626:      ljmp    debit
627: snctb0: jb    Tbhg,snctb1          ; tombol harga
628:      ljmp    harga
629: snctb1: jb    Tbkm,snctb2          ; tombol komulatif
630:      ljmp    komltf
631: snctb2: jb    Tbrs,snctb3          ; tombol debet
632:      ljmp    rstda
633: snctb3: jb    Tbbk,snctb6          ; tombol backlight
634:      cjne  R2,#0,snctb4
635:      mov    R2,#1
636:      clr    Bklg
637:      sjmp    snctb5
638: snctb4: mov    R2,#0
639:      setb  Bklg
640:      cjne  R3,#1,snctb5
641:      mov    R3,#0
642:      ljmp    mulai
643: snctb5: jnb   Tbbk,$
644: snctb6: ret
645: ;
646: stwrmm: orl   Wmcn,#Eemn
647:      orl   Wmcn,#Eemw
648:      ret
649: ;
650: enwrmm: xrl   Wmcn,#Eemw

```

```

651:      xrl      Wmcn, #Eemn
652:      ret
653: ;
654: strdmm: orl      Wmcn, #Eemn
655:      ret
656: ;
657: enrddmm: xrl     Wmcn, #Eemn
658:      ret
659: ;
660: wt_wr:  mov     A, Wmcn
661:      anl     A, #Wtdg
662:      jz     wt_wr
663:      ret
664: ;
665: wr_mm0: mov     DPTR, #00          ; start address memory 00
666:      acall  stwrmm              ; start tulis memory
667:      mov     A, Dtc0            ; \
668:      movx   @DPTR, A           ; |
669:      acall  wt_wr              ; |
670:      inc    DPTR                ; |
671:      mov     A, Dtc0            ; |
672:      movx   @DPTR, A           ; |
673:      acall  wt_wr              ; |
674:      inc    DPTR                ; |
675:      mov     A, Dtc1            ; |
676:      movx   @DPTR, A           ; |
677:      acall  wt_wr              ; |
678:      inc    DPTR                ; |
679:      mov     A, Dtc2            ; |
680:      movx   @DPTR, A           ; |
681:      acall  wt_wr              ; |
682:      inc    DPTR                ; |
683:      mov     A, Dtc3            ; |
684:      movx   @DPTR, A           ; |
685:      acall  wt_wr              ; |
686:      inc    DPTR                ; |
687:      mov     A, Dtc4            ; |
688:      movx   @DPTR, A           ; |
689:      acall  wt_wr              ; |
690:      inc    DPTR                ; |
691:      mov     A, Dtc5            ; |
692:      movx   @DPTR, A           ; |
693:      acall  wt_wr              ; |
694: ;
695:      mov     DPTR, #07          ; start address memory 07
696:      mov     A, Dt00            ; \
697:      movx   @DPTR, A           ; |
698:      acall  wt_wr              ; |
699:      inc    DPTR                ; |
700:      mov     A, Dt01            ; |

```

```

701:      movx   @DPTR,A           ; |
702:      acall  wt_wr            ; |
703:      inc    DPTR             ; |
704:      mov    A,Dt02           ; |
705:      movx   @DPTR,A           ; |
706:      acall  wt_wr            ; |
707:      inc    DPTR             ; | memory debit sekarang
708:      mov    A,Dt03           ; |
709:      movx   @DPTR,A           ; |
710:      acall  wt_wr            ; |
711:      inc    DPTR             ; |
712:      mov    A,Dt04           ; |
713:      movx   @DPTR,A           ; |
714:      acall  wt_wr            ; |
715:      inc    DPTR             ; |
716:      mov    A,Dt05           ; |
717:      movx   @DPTR,A           ; |
718:      acall  wt_wr            ;/
719: ;
720:      mov    DPTR,#25         ; start address memory 25
721:      mov    A,Dhg0           ;\
722:      movx   @DPTR,A           ; |
723:      acall  wt_wr            ; |
724:      inc    DPTR             ; | memory harga
725:      mov    A,Dhg1           ; |
726:      movx   @DPTR,A           ; |
727:      acall  wt_wr            ;/
728:      acall  enwrmm
729:      ret
730: ;
731: wr_mml: mov    DPTR,#13     ; start address memory 13
732:      acall  stwrmm           ; start tulis memory
733:      mov    A,Dt10           ;\
734:      movx   @DPTR,A           ; |
735:      acall  wt_wr            ; |
736:      inc    DPTR             ; |
737:      mov    A,Dt11           ; |
738:      movx   @DPTR,A           ; |
739:      acall  wt_wr            ; |
740:      inc    DPTR             ; |
741:      mov    A,Dt12           ; |
742:      movx   @DPTR,A           ; | memory debit lalu
743:      acall  wt_wr            ; |
744:      inc    DPTR             ; |
745:      mov    A,Dt13           ; |
746:      movx   @DPTR,A           ; |
747:      acall  wt_wr            ; |
748:      inc    DPTR             ; |
749:      mov    A,Dt14           ; |
750:      movx   @DPTR,A           ; |

```

```

751:      acall   wt_wr      ; |
752:      inc     DPTR       ; |
753:      mov     A,Dt15     ; |
754:      movx    @DPTR,A    ; |
755:      acall   wt_wr      ;/
756: ;
757:      mov     DPTR,#19   ;\
758:      mov     A,Dt30     ; |
759:      movx    @DPTR,A    ; |
760:      acall   wt_wr      ; |
761:      inc     DPTR       ; |
762:      mov     A,Dt31     ; |
763:      movx    @DPTR,A    ; |
764:      acall   wt_wr      ; |
765:      inc     DPTR       ; |
766:      mov     A,Dt32     ; |
767:      movx    @DPTR,A    ; |
768:      acall   wt_wr      ; |
769:      inc     DPTR       ; | memory harga lalu
770:      mov     A,Dt33     ; |
771:      movx    @DPTR,A    ; |
772:      acall   wt_wr      ; |
773:      inc     DPTR       ; |
774:      mov     A,Dt34     ; |
775:      movx    @DPTR,A    ; |
776:      acall   wt_wr      ; |
777:      inc     DPTR       ; |
778:      mov     A,Dt35     ; |
779:      movx    @DPTR,A    ; |
780:      acall   wt_wr      ;/
781:      acall   enwrmm     ;   end tulis memory
782:      ret
783: ;
784: rd_mem: mov     DPTR,#00 ;   start address memory 00
785:      acall   strdmm     ;   start baca memory
786:      movx    A,@DPTR    ;\
787:      mov     Dtc0,A     ; |
788:      inc     DPTR       ; |
789:      movx    A,@DPTR    ; |
790:      mov     Dtc1,A     ; |
791:      inc     DPTR       ; |
792:      movx    A,@DPTR    ; |
793:      mov     Dtc2,A     ; |
794:      inc     DPTR       ; |
795:      movx    A,@DPTR    ; | memory counter meter
796:      mov     Dtc3,A     ; |
797:      inc     DPTR       ; |
798:      movx    A,@DPTR    ; |
799:      mov     Dtc4,A     ; |
800:      inc     DPTR       ; |

```

```

801:      movx   A,@DPTR      ; |
802:      mov    Dtc4,A        ; |
803:      inc    DPTR          ; |
804:      movx   A,@DPTR      ; |
805:      mov    Dtc5,A        ;/
806: ;
807:      mov    DPTR,#07      ;   start address memory 25
808:      inc    DPTR          ;\
809:      movx   A,@DPTR      ; |
810:      mov    Dt00,A        ; |
811:      inc    DPTR          ; |
812:      movx   A,@DPTR      ; |
813:      mov    Dt01,A        ; |
814:      inc    DPTR          ; |
815:      movx   A,@DPTR      ; |
816:      mov    Dt02,A        ; | memory debet sekarang
817:      inc    DPTR          ; |
818:      movx   A,@DPTR      ; |
819:      mov    Dt03,A        ; |
820:      inc    DPTR          ; |
821:      movx   A,@DPTR      ; |
822:      mov    Dt04,A        ; |
823:      inc    DPTR          ; |
824:      movx   A,@DPTR      ; |
825:      mov    Dt05,A        ;/
826: ;
827:      mov    DPTR,#13      ;   start address memory 13
828:      movx   A,@DPTR      ;\
829:      mov    Dt10,A        ; |
830:      inc    DPTR          ; |
831:      movx   A,@DPTR      ; |
832:      mov    Dt11,A        ; |
833:      inc    DPTR          ; |
834:      movx   A,@DPTR      ; |
835:      mov    Dt12,A        ; | memory debet lalu
836:      inc    DPTR          ; |
837:      movx   A,@DPTR      ; |
838:      mov    Dt13,A        ; |
839:      inc    DPTR          ; |
840:      movx   A,@DPTR      ; |
841:      mov    Dt14,A        ; |
842:      inc    DPTR          ; |
843:      movx   A,@DPTR      ; |
844:      mov    Dt15,A        ;/
845: ;
846:      mov    DPTR,#19      ;   start address memory 19
847:      movx   A,@DPTR      ;\
848:      mov    Dt30,A        ; |
849:      inc    DPTR          ; |
850:      movx   A,@DPTR      ; |

```

```

851:      mov     Dt31,A           ; |
852:      inc     DPTR             ; |
853:      movx    A,@DPTR          ; |
854:      mov     Dt32,A           ; | memory harga lalu
855:      inc     DPTR             ; |
856:      movx    A,@DPTR          ; |
857:      mov     Dt33,A           ; |
858:      inc     DPTR             ; |
859:      movx    A,@DPTR          ; |
860:      mov     Dt34,A           ; |
861:      inc     DPTR             ; |
862:      movx    A,@DPTR          ; |
863:      mov     Dt35,A           ;/
864: ;
865:      mov     DPTR,#25         ; start address memory 25
866:      movx    A,@DPTR          ;\
867:      mov     Dhg0,A           ; | memory harga per liter
868:      inc     DPTR             ; |
869:      movx    A,@DPTR          ; |
870:      mov     Dhg1,A           ;/
871:      acall   enrmmm           ; end baca memory
872: ;
873: cekmem: mov     A,Dtct
874:      cjne    A,#0FFh,cekmm0
875:      mov     Dtct,#0
876: cekmm0: mov     A,Dtc0
877:      cjne    A,#0FFh,cekmm1
878:      mov     Dtc0,#0
879: cekmm1: mov     A,Dtc1
880:      cjne    A,#0FFh,cekmm2
881:      mov     Dtc1,#0
882: cekmm2: mov     A,Dtc2
883:      cjne    A,#0FFh,cekmm3
884:      mov     Dtc2,#0
885: cekmm3: mov     A,Dtc3
886:      cjne    A,#0FFh,cekmm4
887:      mov     Dtc3,#0
888: cekmm4: mov     A,Dtc4
889:      cjne    A,#0FFh,cekmm5
890:      mov     Dtc4,#0
891: cekmm5: mov     A,Dtc5
892:      cjne    A,#0FFh,cekmm6
893:      mov     Dtc5,#0
894: cekmm6: mov     A,Dt00
895:      cjne    A,#0FFh,cekmm7
896:      mov     Dt00,#0
897: cekmm7: mov     A,Dt01
898:      cjne    A,#0FFh,cekmm8
899:      mov     Dt01,#0
900: cekmm8: mov     A,Dt02

```

```
901:      cjne    A, #0FFh, cekmm9
902:      mov     Dt02, #0
903: cekmm9: mov     A, Dt03
904:      cjne    A, #0FFh, cekmmA
905:      mov     Dt03, #0
906: cekmmA: mov     A, Dt04
907:      cjne    A, #0FFh, cekmmB
908:      mov     Dt04, #0
909: cekmmB: mov     A, Dt05
910:      cjne    A, #0FFh, cekmmC
911:      mov     Dt05, #0
912: cekmmC: mov     A, Dt10
913:      cjne    A, #0FFh, cekmmD
914:      mov     Dt10, #0
915: cekmmD: mov     A, Dt11
916:      cjne    A, #0FFh, cekmmE
917:      mov     Dt11, #0
918: cekmmE: mov     A, Dt12
919:      cjne    A, #0FFh, cekmmF
920:      mov     Dt12, #0
921: cekmmF: mov     A, Dt13
922:      cjne    A, #0FFh, cekmmG
923:      mov     Dt13, #0
924: cekmmG: mov     A, Dt14
925:      cjne    A, #0FFh, cekmmH
926:      mov     Dt14, #0
927: cekmmH: mov     A, Dt15
928:      cjne    A, #0FFh, cekmmI
929:      mov     Dt15, #0
930: cekmmI: mov     A, Dt30
931:      cjne    A, #0FFh, cekmmJ
932:      mov     Dt30, #0
933: cekmmJ: mov     A, Dt31
934:      cjne    A, #0FFh, cekmmK
935:      mov     Dt31, #0
936: cekmmK: mov     A, Dt32
937:      cjne    A, #0FFh, cekmmL
938:      mov     Dt32, #0
939: cekmmL: mov     A, Dt33
940:      cjne    A, #0FFh, cekmmM
941:      mov     Dt33, #0
942: cekmmM: mov     A, Dt34
943:      cjne    A, #0FFh, cekmmN
944:      mov     Dt34, #0
945: cekmmN: mov     A, Dt35
946:      cjne    A, #0FFh, cekmmO
947:      mov     Dt35, #0
948: cekmmO: ret
949: ;
950: jeda:  djnz   Dly0, $
```



```

951:         ret
952: ;
953: delays: mov     Dly1,#255
954: dlys0:  acall   bc_sns
955:         acall   scntbl
956:         acall   scnkpdl
957:         cjne   R0,#13,dlys1
958:         ljmp   seting
959: dlys1:  djnz   Dly1,dlys0
960:         ret
961: ;
962: delay1: mov     Dly2,#20
963: dly1:   acall   delays
964:         djnz   Dly2,dly1
965:         ret
966: ;
967: nama:   DB      ' Rusdil Badi '
968:         DB      ' 02.17.080 '
969:         DB      ' Teknik Elektro '
970:         DB      ' ITN Malang '
971: trbhrgr: DB      ' Rubah Harga? '
972: trpiah: DB      ' Rp '
973: tprltr: DB      '/liter '
974: tdtkml: DB      ' Data Komulatif '
975: tcontr: DB      'Counter '
976: tdtddl: DB      ' Lalu '
977: tdtask: DB      ' Skrg '
978: trstdd: DB      ' Reset Data '
979:         DB      ' Please Wait... '
980: tdtbnr: DB      ' Data Benar ? '
981: tangka: DB      '0123456789 '
982: ;
983:         end
984:

```

3503

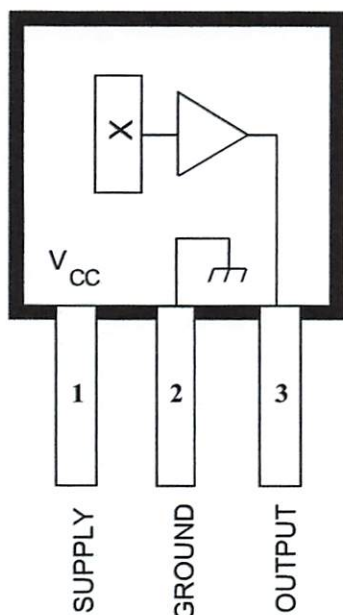
RATIOMETRIC, LINEAR HALL-EFFECT SENSORS

The UGN3503LT, UGN3503U, and UGN3503UA Hall-effect sensors accurately track extremely small changes in magnetic flux density—changes generally too small to operate Hall-effect switches.

As motion detectors, gear tooth sensors, and proximity detectors, they are magnetically driven mirrors of mechanical events. As sensitive monitors of electromagnets, they can effectively measure a system's performance with negligible system loading while providing isolation from contaminated and electrically noisy environments.

Each Hall-effect integrated circuit includes a Hall sensing element, linear amplifier, and emitter-follower output stage. Problems associated with handling tiny analog signals are minimized by having the Hall cell and amplifier on a single chip.

Three package styles provide a magnetically optimized package for most applications. Package suffix 'LT' is a miniature SOT-89/TO-243AA transistor package for surface-mount applications; suffix 'U' is a miniature three-lead plastic SIP, while 'UA' is a three-lead ultra-mini-SIP. All devices are rated for continuous operation over the temperature range of -20°C to +85°C.



Dwg. PH-006

Pinning is shown viewed from branded side.

ABSOLUTE MAXIMUM RATINGS

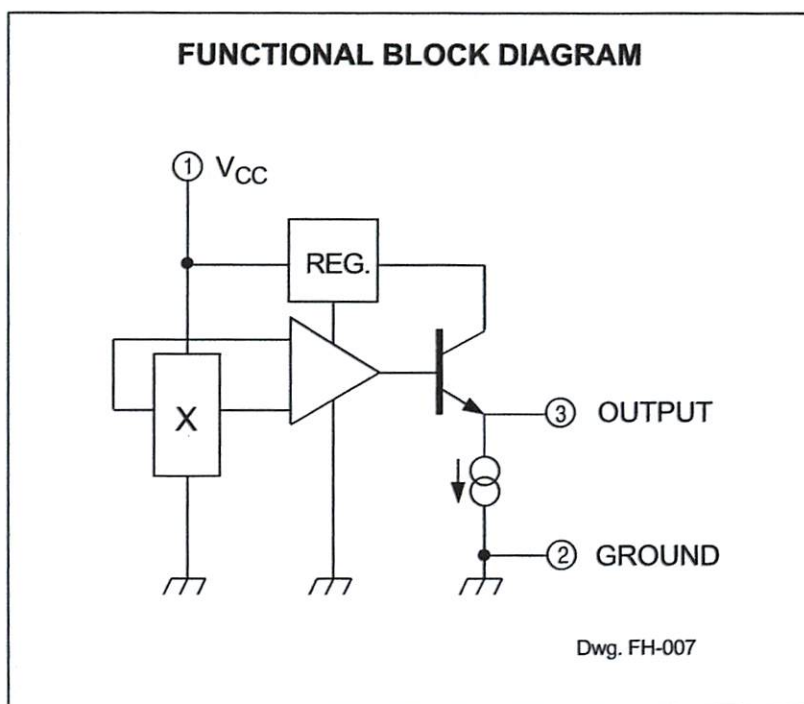
Supply Voltage, V_{CC}	8 V
Magnetic Flux Density, B	Unlimited
Operating Temperature Range,	
T_A	-20°C to +85°C
Storage Temperature Range,	
T_S	-65°C to +150°C

FEATURES

- Extremely Sensitive
- Flat Response to 23 kHz
- Low-Noise Output
- 4.5 V to 6 V Operation
- Magnetically Optimized Package

Always order by complete part number, e.g., **UGN3503UA**.

3503
RATIOMETRIC,
LINEAR
HALL-EFFECT SENSORS



ELECTRICAL CHARACTERISTICS at $T_A = +25^\circ\text{C}$, $V_{CC} = 5\text{ V}$

Characteristic	Symbol	Test Conditions	Limits			Units
			Min.	Typ.	Max.	
Operating Voltage	V_{CC}		4.5	—	6.0	V
Supply Current	I_{CC}		—	9.0	13	mA
Quiescent Output Voltage	V_{OUT}	$B = 0\text{ G}$	2.25	2.50	2.75	V
Sensitivity	ΔV_{OUT}	$B = 0\text{ G to } \pm 900\text{ G}$	0.75	1.30	1.75	mV/G
Bandwidth (-3 dB)	BW		—	23	—	kHz
Broadband Output Noise	V_{out}	$BW = 10\text{ Hz to } 10\text{ kHz}$	—	90	—	μV
Output Resistance	R_{OUT}		—	50	220	Ω

All output-voltage measurements are made with a voltmeter having an input impedance of at least 10 k Ω .

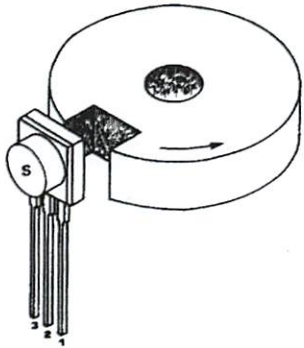
Magnetic flux density is measured at most sensitive area of device located 0.0165" (0.42 mm) below the branded face of the 'U' package; 0.0195" (0.50 mm) below the branded face of the 'UA' package; and 0.0305" (0.775 mm) below the branded face of the 'T' package.



115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000
 Copyright © 1985, 2002 Allegro Microsystems, Inc.

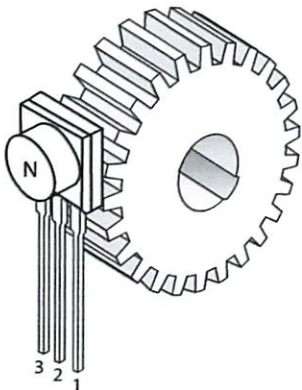
3503 RATIOMETRIC, LINEAR HALL-EFFECT SENSORS

NOTCH SENSOR



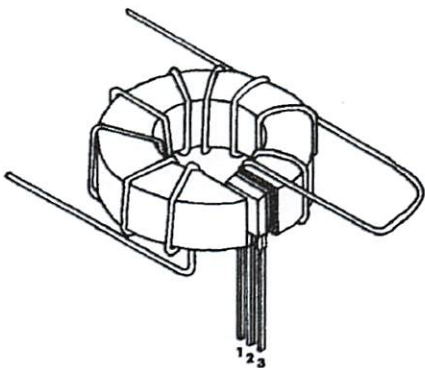
Dwg. A-12,574

GEAR TOOTH SENSOR



Dwg. A-12,512

CURRENT MONITOR



Dwg. A-12,513

OPERATION

The output null voltage ($B = 0$ G) is nominally one-half the supply voltage. A south magnetic pole, presented to the branded face of the Hall-effect sensor will drive the output higher than the null voltage level. A north magnetic pole will drive the output below the null level.

In operation, instantaneous and proportional output-voltage levels are dependent on magnetic flux density at the most sensitive area of the device. Greatest sensitivity is obtained with a supply voltage of 6 V, but at the cost of increased supply current and a slight loss of output symmetry. The sensor's output is usually capacitively coupled to an amplifier that boosts the output above the millivolt level.

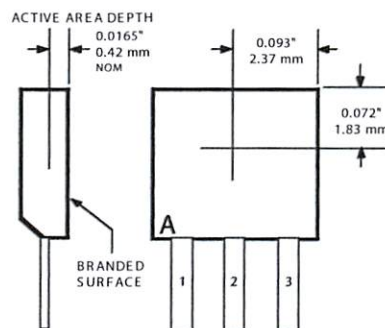
In two applications shown, a permanent bias magnet is attached with epoxy glue to the back of the epoxy package. The presence of ferrous material at the face of the package acts as a flux concentrator.

The south pole of a magnet is attached to the back of the package if the Hall-effect IC is to sense the presence of ferrous material. The north pole of a magnet is attached to the back surface if the integrated circuit is to sense the absence of ferrous material.

Calibrated linear Hall devices, which can be used to determine the actual flux density presented to the sensor in a particular application, are available.

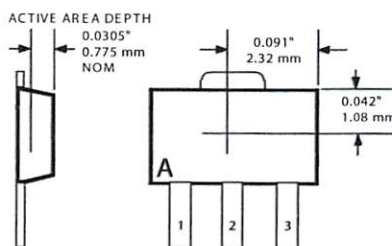
SENSOR LOCATIONS

SUFFIX "U"



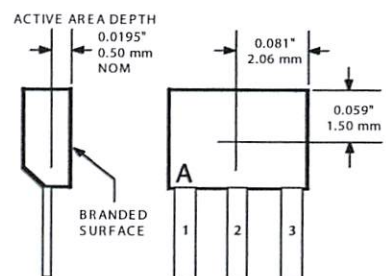
Dwg. MH-002-5D

SUFFIX "LT"



Dwg. MH-008-9A

SUFFIX "UA"



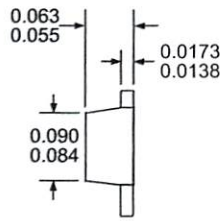
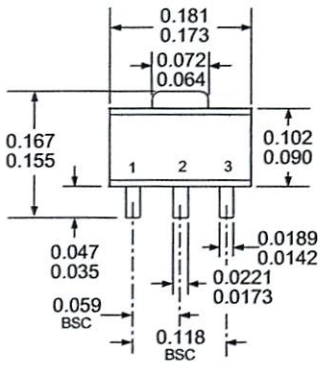
Dwg. MH-011-3D

3503 RATIOMETRIC, LINEAR HALL-EFFECT SENSORS

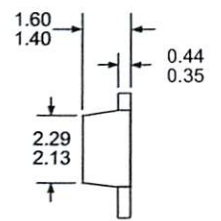
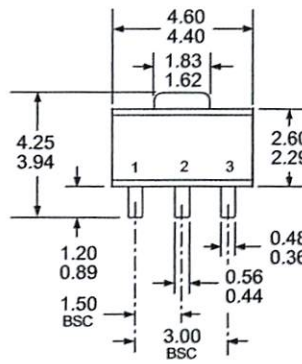
UGN3503LT (SOT89/TO-243AA)

Dimensions in Inches
(for reference only)

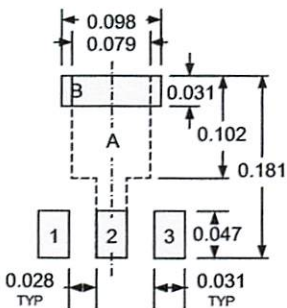
Dimensions in Millimeters
(controlling dimensions)



Dwg. MA-009-3A in

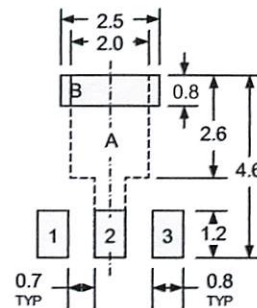


Dwg. MA-009-3A mm



Pads 1, 2, 3, and A — Standard SOT89 Layout
 Pads 1, 2, 3, and B — Low-Stress Version
 Pads 1, 2, and 3 only — Lowest Stress, But Not Self Aligning

Dwg. MA-012-3 in



Pads 1, 2, 3, and A — Standard SOT89 Layout
 Pads 1, 2, 3, and B — Low-Stress Version
 Pads 1, 2, and 3 only — Lowest Stress, But Not Self Aligning

Dwg. MA-012-3 mm

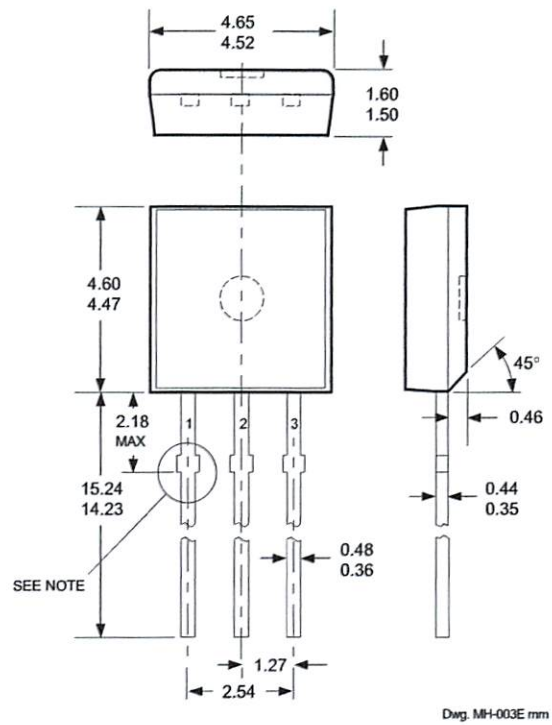
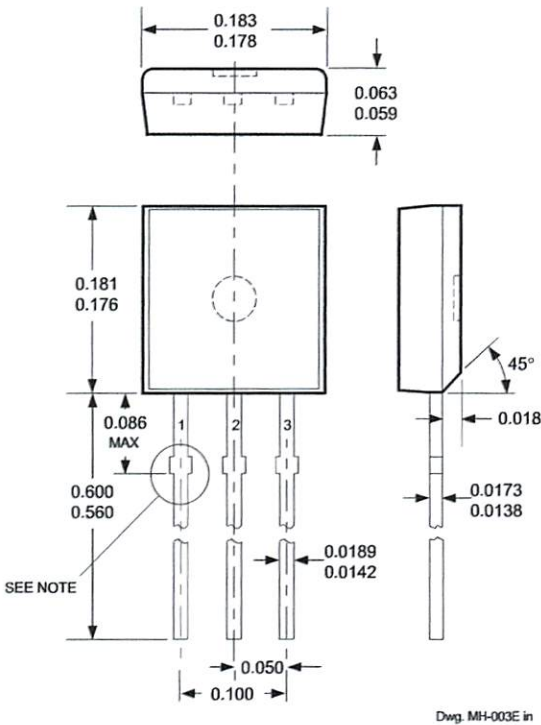
- NOTES:
1. Exact body and lead configuration at vendor's option within limits shown.
 2. Supplied in bulk pack (500 pieces per bag) or add "TR" to part number for tape and reel.
 3. Only low-temperature ($\leq 240^{\circ}\text{C}$) reflow-soldering techniques are recommended for SOT89 devices.

3503
RATIOMETRIC,
LINEAR
HALL-EFFECT SENSORS

UGN3503U

Dimensions in Inches
 (controlling dimensions)

Dimensions in Millimeters
 (for reference only)



Devices in the 'U' package are
NOT RECOMMENDED FOR NEW DESIGN

- NOTES:
1. Tolerances on package height and width represent allowable mold offsets. Dimensions given are measured at the widest point (parting line).
 2. Exact body and lead configuration at vendor's option within limits shown.
 3. Height does not include mold gate flash.
 4. Recommended minimum PWB hole diameter to clear transition area is 0.035" (0.89 mm).
 5. Minimum lead length was 0.500" (12.70 mm). If existing product to the original specifications is not acceptable, contact sales office before ordering.

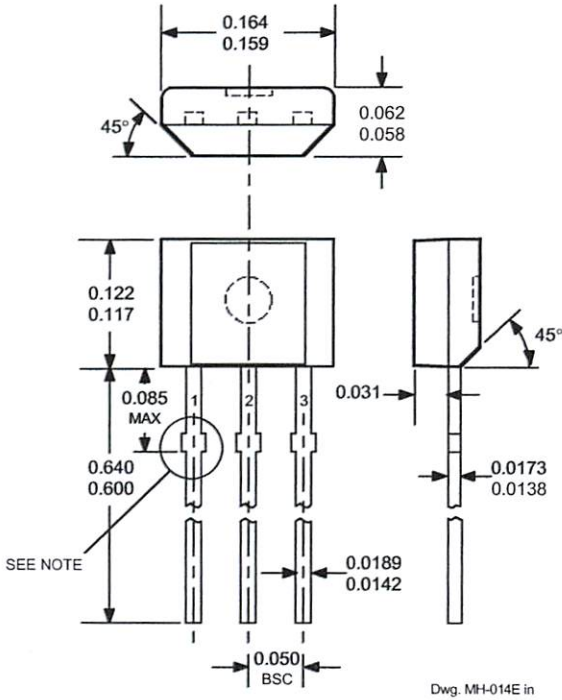


115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000

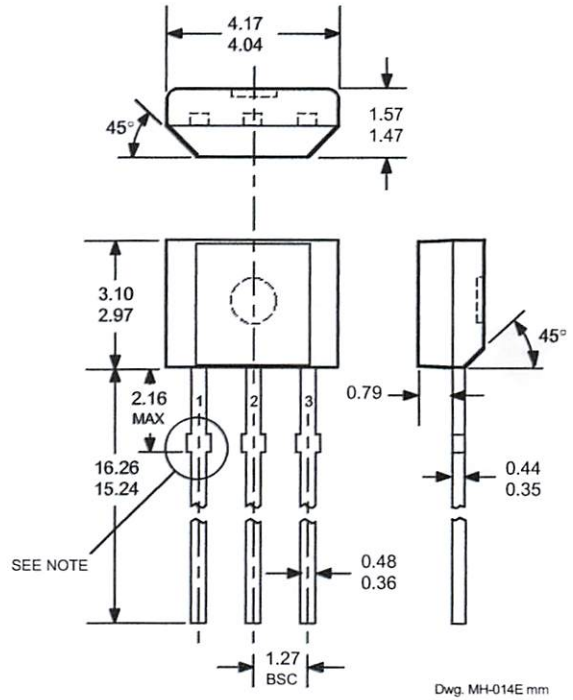
3503 RATIOMETRIC, LINEAR HALL-EFFECT SENSORS

UGN3503UA

Dimensions in Inches
(controlling dimensions)

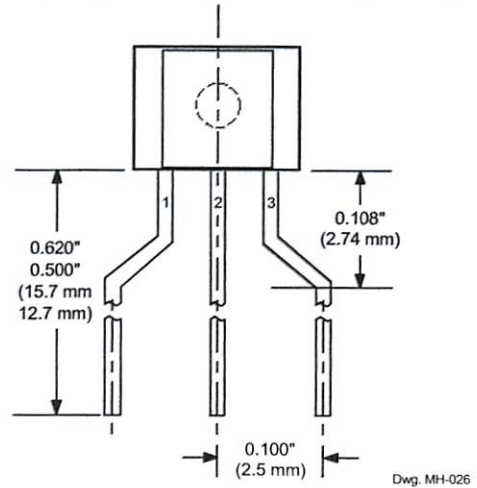


Dimensions in Millimeters
(for reference only)



- NOTES: 1. Tolerances on package height and width represent allowable mold offsets. Dimensions given are measured at the widest point (parting line).
 2. Exact body and lead configuration at vendor's option within limits shown.
 3. Height does not include mold gate flash.
 4. Recommended minimum PWB hole diameter to clear transition area is 0.035" (0.89 mm).
 5. Where no tolerance is specified, dimension is nominal.
 6. Supplied in bulk pack (500 pieces per bag).

Radial Lead Form (order UGN3503UA-LC)



NOTE: Lead-form dimensions are the nominals produced on the forming equipment. No dimensional tolerance is implied or guaranteed for bulk packaging (500 pieces per bag).

KA741

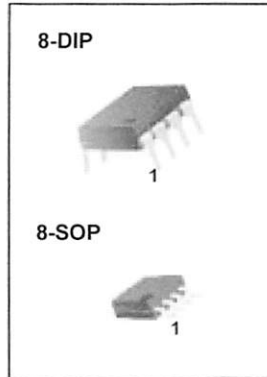
Single Operational Amplifier

Features

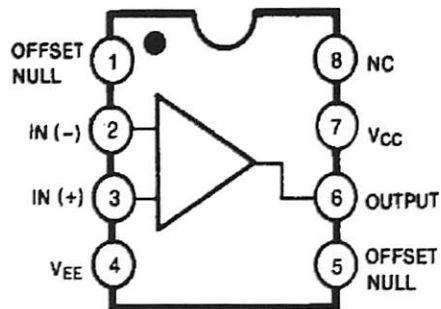
- Short circuit protection
- Excellent temperature stability
- Internal frequency compensation
- High Input voltage range
- Null of offset

Description

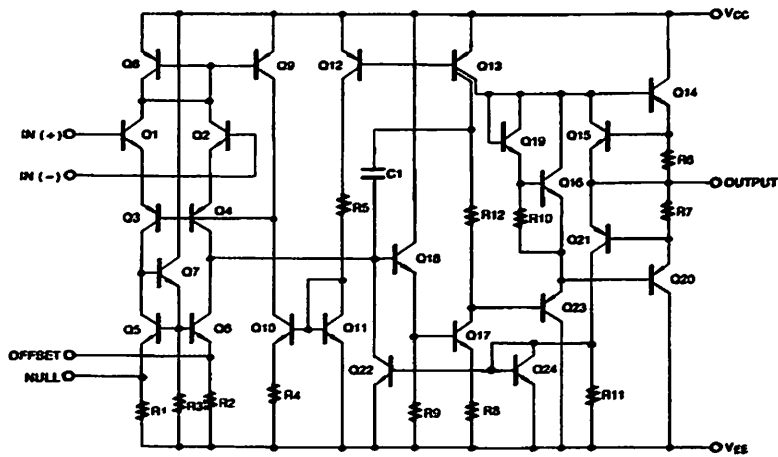
The KA741 series are general purpose operational amplifiers. It is intended for a wide range of analog applications. The high gain and wide range of operating voltage provide superior performance in integrator, summing amplifier, and general feedback applications.



Internal Block Diagram



Schematic Diagram



Absolute Maximum Ratings (TA = 25°C)

Parameter	Symbol	Value	Unit
Supply Voltage	VCC	±18	V
Differential Input Voltage	VI(DIFF)	30	V
Input Voltage	VI	±15	V
Output Short Circuit Duration	-	Indefinite	
Power Dissipation	PD	500	mW
Operating Temperature Range KA741 KA741I	TOPR	0 ~ + 70 -40 ~ +85	°C
Storage Temperature Range	TSTG	-65 ~ + 150	°C

Electrical Characteristics

($V_{CC} = 15V$, $V_{EE} = -15V$, $T_A = 25^\circ C$, unless otherwise specified)

Parameter		Symbol	Conditions	KA741/KA741I			Unit	
				Min.	Typ.	Max.		
Input Offset Voltage		V_{IO}	$R_S \leq 10K\Omega$	-	2.0	6.0	mV	
			$R_S \leq 50\Omega$	-	-	-		
Input Offset Voltage Adjustment Range		$V_{IO(R)}$	$V_{CC} = \pm 20V$	-	± 15	-	mV	
Input Offset Current		I_{IO}	-	-	20	200	nA	
Input Bias Current		I_{BIAS}	-	-	80	500	nA	
Input Resistance (Note1)		R_I	$V_{CC} = \pm 20V$	0.3	2.0	-	M Ω	
Input Voltage Range		$V_{I(R)}$	-	± 12	± 13	-	V	
Large Signal Voltage Gain		G_V	$R_L \geq 2K\Omega$	$V_{CC} = \pm 20V$, $V_{O(P-P)} = \pm 15V$		-	V/mV	
				$V_{CC} = \pm 15V$, $V_{O(P-P)} = \pm 10V$		20		200
Output Short Circuit Current		I_{SC}	-	-	25	-	mA	
Output Voltage Swing		$V_{O(P-P)}$	$V_{CC} = \pm 20V$	$R_L \geq 10K\Omega$	-	-	V	
				$R_L \geq 2K\Omega$	-	-		-
			$V_{CC} = \pm 15V$	$R_L \geq 10K\Omega$	± 12	± 14		-
				$R_L \geq 2K\Omega$	± 10	± 13		-
Common Mode Rejection Ratio		CMRR	$R_S \leq 10K\Omega$, $V_{CM} = \pm 12V$	70	90	-	dB	
			$R_S \leq 50\Omega$, $V_{CM} = \pm 12V$	-	-	-		
Power Supply Rejection Ratio		PSRR	$V_{CC} = \pm 15V$ to $V_{CC} = \pm 15V$ $R_S \leq 50\Omega$	-	-	-	dB	
			$V_{CC} = \pm 15V$ to $V_{CC} = \pm 15V$ $R_S \leq 10K\Omega$	77	96	-		
Transient Response	Rise Time	T_R	Unity Gain	-	0.3	-	μs	
	Overshoot	OS		-	10	-	%	
Bandwidth		BW	-	-	-	-	MHz	
Slew Rate		SR	Unity Gain	-	0.5	-	V/ μs	
Supply Current		I_{CC}	$R_L = \infty\Omega$	-	1.5	2.8	mA	
Power Consumption		P_C	$V_{CC} = \pm 20V$	-	-	-	mW	
			$V_{CC} = \pm 15V$	-	50	85		

Note:

1. Guaranteed by design.

Electrical Characteristics

($V_{CC} = \pm 15V$, unless otherwise specified)

The following specification apply over the range of $0^{\circ}C \leq T_A \leq +70^{\circ}C$ for the KA741; and the $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ for the KA741I

Parameter	Symbol	Conditions	KA741/KA741I			Unit	
			Min.	Typ.	Max.		
Input Offset Voltage	V_{IO}	$R_S \leq 50\Omega$	-	-	-	mV	
		$R_S \leq 10K\Omega$	-	-	7.5		
Input Offset Voltage Drift	$\Delta V_{IO}/\Delta T$	-	-	-	-	$\mu V/^{\circ}C$	
Input Offset Current	I_{IO}	-	-	-	300	nA	
Input Offset Current Drift	$\Delta I_{IO}/\Delta T$	-	-	-	-	$nA/^{\circ}C$	
Input Bias Current	I_{BIAS}	-	-	-	0.8	μA	
Input Resistance (Note1)	R_I	$V_{CC} = \pm 20V$	-	-	-	$M\Omega$	
Input Voltage Range	$V_{I(R)}$	-	± 12	± 13	-	V	
Output Voltage Swing	$V_{O(P-P)}$	$V_{CC} = \pm 20V$	$R_S \geq 10K\Omega$	-	-	-	V
			$R_S \geq 2K\Omega$	-	-	-	
		$V_{CC} = \pm 15V$	$R_S \geq 10K\Omega$	± 12	± 14	-	
			$R_S \geq 2K\Omega$	± 10	± 13	-	
Output Short Circuit Current	I_{SC}	-	10	-	40	mA	
Common Mode Rejection Ratio	CMRR	$R_S \leq 10K\Omega, V_{CM} = \pm 12V$	70	90	-	dB	
		$R_S \leq 50\Omega, V_{CM} = \pm 12V$	-	-	-		
Power Supply Rejection Ratio	PSRR	$V_{CC} = \pm 20V$ to $\pm 5V$	$R_S \leq 50\Omega$	-	-	-	dB
			$R_S \leq 10K\Omega$	77	96	-	
Large Signal Voltage Gain	GV	$R_S \geq 2K\Omega$	$V_{CC} = \pm 20V,$ $V_{O(P-P)} = \pm 15V$	-	-	-	V/mV
			$V_{CC} = \pm 15V,$ $V_{O(P-P)} = \pm 10V$	15	-	-	
			$V_{CC} = \pm 15V,$ $V_{O(P-P)} = \pm 2V$	-	-	-	

Note :

1. Guaranteed by design.

Quad 2-input NAND Schmitt trigger

74HC/HCT132

FEATURES

- Output capability: standard
- I_{CC} category: SSI

GENERAL DESCRIPTION

The 74HC/HCT132 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT132 contain four 2-input NAND gates which accept standard input signals. They are capable of transforming slowly changing input signals into sharply defined, jitter-free output signals.

The gate switches at different points for positive and negative-going signals. The difference between the positive voltage V_{T+} and the negative voltage V_{T-} is defined as the hysteresis voltage V_H .

QUICK REFERENCE DATA

GND = 0 V; $T_{amb} = 25\text{ }^{\circ}\text{C}$; $t_r = t_f = 6\text{ ns}$

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/t_{PLH}	propagation delay nA, nB to nY	$C_L = 15\text{ pF}$; $V_{CC} = 5\text{ V}$	11	17	ns
C_i	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per gate	notes 1 and 2	24	20	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz

f_o = output frequency in MHz

$\sum (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

C_L = output load capacitance in pF

V_{CC} = supply voltage in V

2. For HC the condition is $V_i = \text{GND to } V_{CC}$
For HCT the condition is $V_i = \text{GND to } V_{CC} - 1.5\text{ V}$

ORDERING INFORMATION

See "74HC/HCT/HCU/HCMOS Logic Package Information".

Quad 2-input NAND Schmitt trigger

74HC/HCT132

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 4, 9, 12	1A to 4A	data inputs
2, 5, 10, 13	1B to 4B	data inputs
3, 6, 8, 11	1Y to 4Y	data outputs
7	GND	ground (0 V)
14	V _{CC}	positive supply voltage

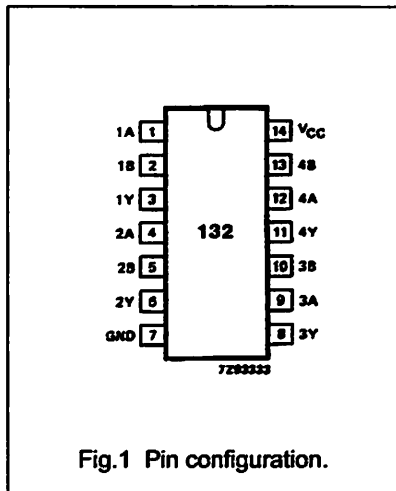


Fig.1 Pin configuration.

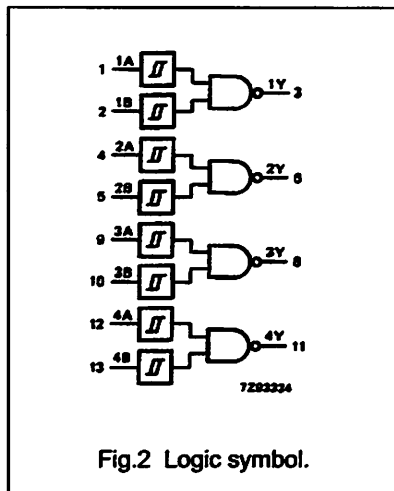


Fig.2 Logic symbol.

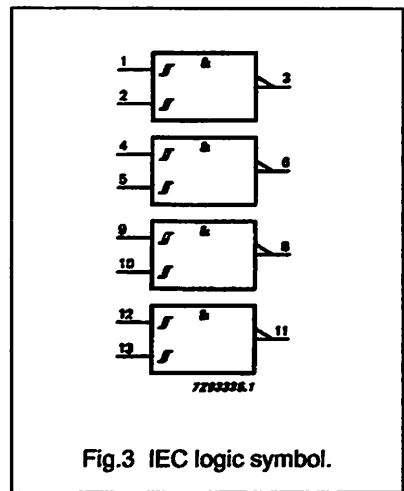


Fig.3 IEC logic symbol.

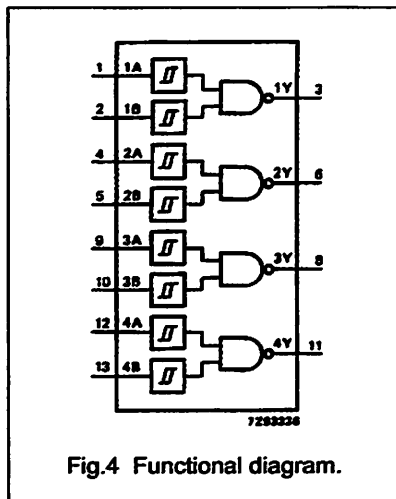


Fig.4 Functional diagram.

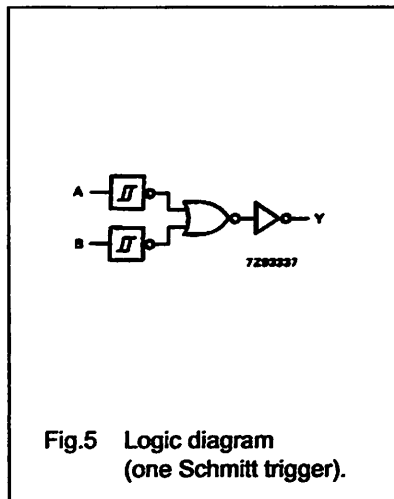


Fig.5 Logic diagram (one Schmitt trigger).

FUNCTION TABLE

INPUTS		OUTPUT
nA	nB	nY
L	L	H
L	H	H
H	L	H
H	H	L

Notes

- H = HIGH voltage level
L = LOW voltage level

APPLICATIONS

- Wave and pulse shapers
- Astable multivibrators
- Monostable multivibrators

Quad 2-input NAND Schmitt trigger

74HC/HCT132

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications". Transfer characteristics are given below.

Output capability: standard

I_{CC} category: SSI

Transfer characteristics for 74HC

Voltages are referenced to GND (ground = 0 V)

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} (V)	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
V _{T+}	positive-going threshold	0.7	1.18	1.5	0.7	1.5	0.7	1.5	V	2.0	Figs 6 and 7	
		1.7	2.38	3.15	1.7	3.15	1.7	3.15				
		2.1	3.14	4.2	2.1	4.2	2.1	4.2				
V _{T-}	negative-going threshold	0.3	0.63	1.0	0.3	1.0	0.3	1.0	V	2.0	Figs 6 and 7	
		0.9	1.67	2.2	0.9	2.2	0.9	2.2				
		1.2	2.26	3.0	1.2	3.0	1.2	3.0				
V _H	hysteresis (V _{T+} - V _{T-})	0.2	0.55	1.0	0.2	1.0	0.2	1.0	V	2.0	Figs 6 and 7	
		0.4	0.71	1.4	0.4	1.4	0.4	1.4				
		0.6	0.88	1.6	0.6	1.6	0.6	1.6				

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} (V)	WAVEFORMS
		+25			-40 TO +85		-40 TO +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay nA, nB to nY		36	125		155		190	ns	2.0	Fig.13	
			13	25		31		38				
			10	21		26		32				
t _{THL} / t _{TLH}	output transition time		19	75		95		110	ns	2.0	Fig.13	
			7	15		19		22				
			6	13		16		19				

Quad 2-input NAND Schmitt trigger

74HC/HCT132

DC CHARACTERISTICS FOR 74HCT

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications". Transfer characteristics are given below.

Output capability: standard

I_{CC} category: SSI

Notes to HCT types

The value of additional quiescent supply current (ΔI_{CC}) for a unit load of 1 is given in the family specifications.

To determine ΔI_{CC} per input, multiply this value by the unit load coefficient shown in the table below.

INPUT	UNIT LOAD COEFFICIENT
nA, nB	0.3

Transfer characteristics for 74HCT

Voltages are referenced to GND (ground = 0 V)

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HCT									V _{CC} (V)	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
V _{T+}	positive-going threshold	1.2	1.41	1.9	1.2	1.9	1.2	1.9	V	4.5	Figs 6 and 7	
		1.4	1.59	2.1	1.4	2.1	1.4	2.1				
V _{T-}	negative-going threshold	0.5	0.85	1.2	0.5	1.2	0.5	1.2	V	4.5	Figs 6 and 7	
		0.6	0.99	1.4	0.6	1.4	0.6	1.4				
V _H	hysteresis (V _{T+} - V _{T-})	0.4	0.56	–	0.4	–	0.4	–	V	4.5	Figs 6 and 7	
		0.4	0.60	–	0.4	–	0.4	–				

AC CHARACTERISTICS FOR 74HCT

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HCT									V _{CC} (V)	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} /t _{PLH}	propagation delay nA, nB to nY		20	33		41		50	ns	4.5	Fig.13	
t _{THL} /t _{TLH}	output transition time		7	15		19		22	ns	4.5	Fig.13	

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 microcontroller 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 programmed 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 microcontroller, 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 external 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 bits have been activated.



**8-bit
Microcontroller
with 8K Bytes
Flash**

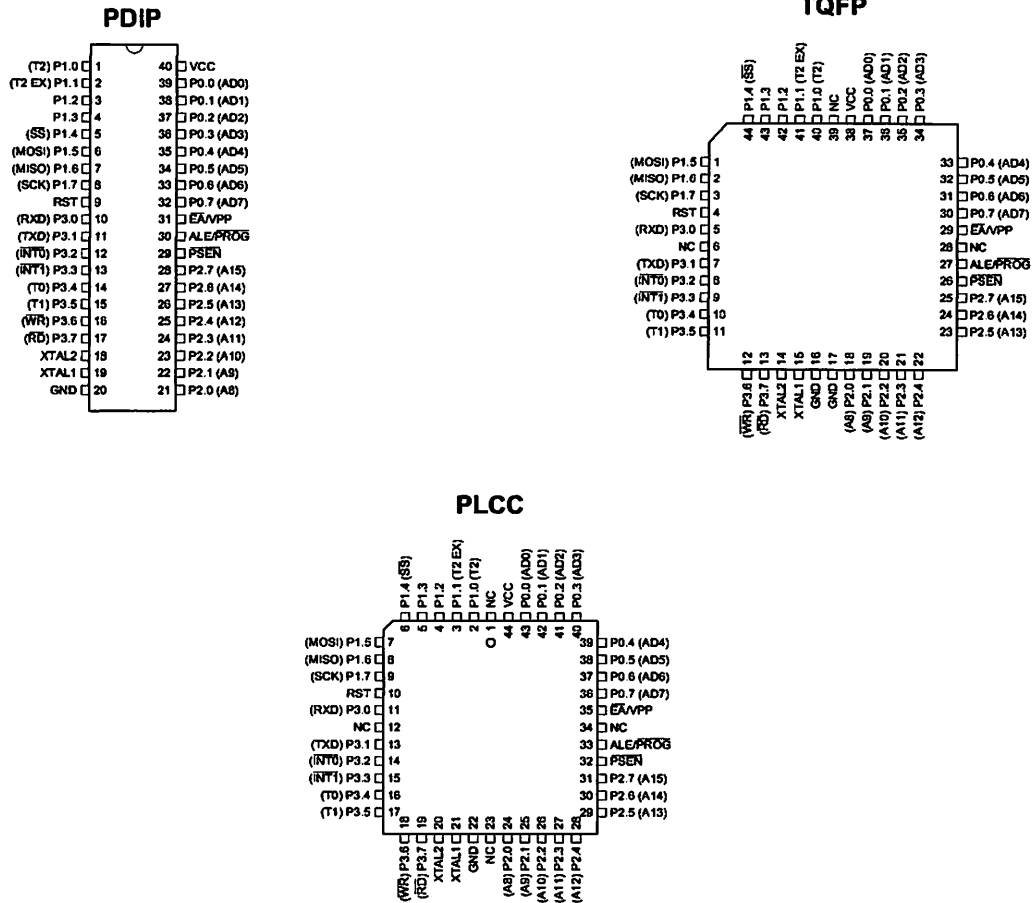
AT89S8252

0401F-MICRO-11/03





Pin Configurations



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 pull-ups.

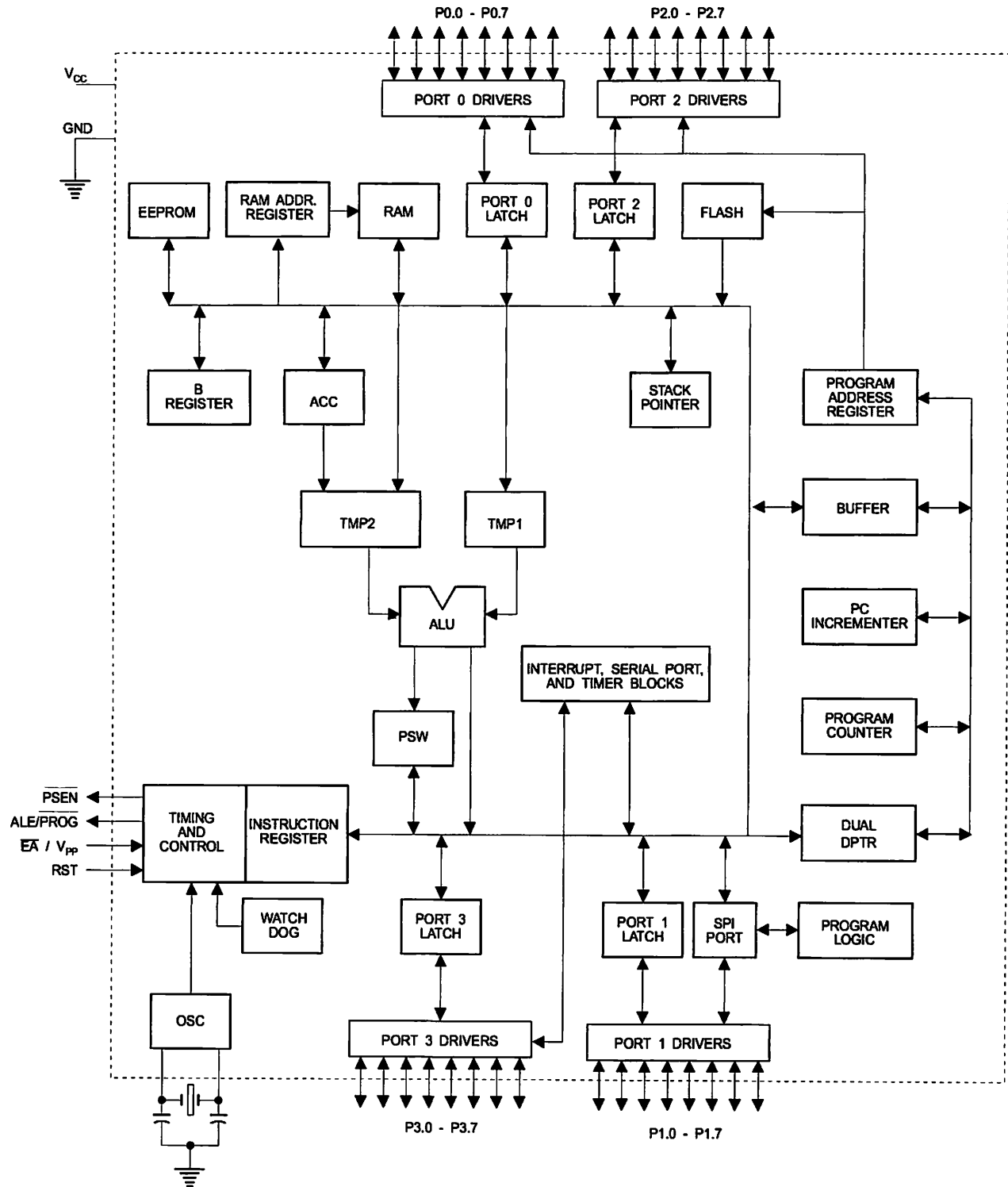
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1

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

AT89S8252

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.

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

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

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

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

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

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

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

ST

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

$\overline{\text{LE/PROG}}$

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 (PROG) during Flash programming.

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

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

$\overline{\text{SEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

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

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

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

TAL1

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

TAL2

Output from the inverting oscillator amplifier.



Special Function Registers

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

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

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 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

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H				Reset Value = 0000 0000B			
3-bit Addressable							
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
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 PS 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
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: <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">SPR1</td> <td style="padding-right: 10px;">SPR0</td> <td>SCK = F_{osc} divided by</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">4</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">16</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">64</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">128</td> </tr> </table>	SPR1	SPR0	SCK = F_{osc} divided by	0	0	4	0	1	16	1	0	64	1	1	128
SPR1	SPR0	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			
	SPIF	WCOL	–	–	–	–	–
Bit	7	6	5	4	3	2	1

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 reading/writing 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				
	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Bit	7	6	5	4	3	2	1	0

Data Memory – EPROM 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 $\overline{RDY/BSY} = 1$ 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 internal 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 WDTRST 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 and AT89C52. For further information on the timers' operation, refer to the Atmel web site (<http://www.atmel.com>). From the home page, select "Products", then "Microcontrollers, then "8051-Architecture". Click on "Documentation", then on "Other Documents". Open the document "AT89 Series Hardware Description".

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 $C/\overline{T2}$ 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.

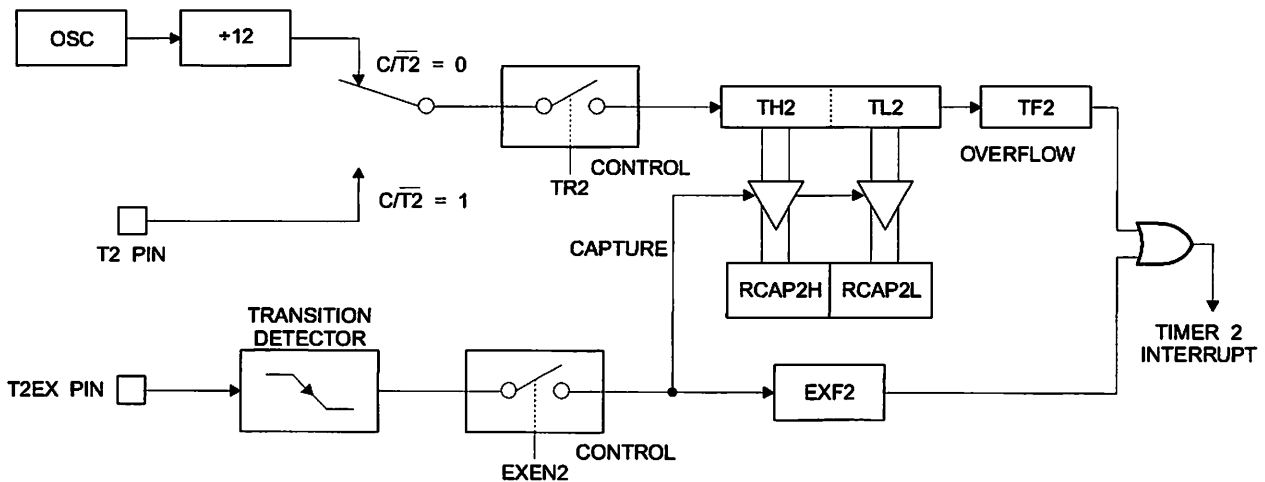
Table 8. Timer 2 Operating Modes

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)

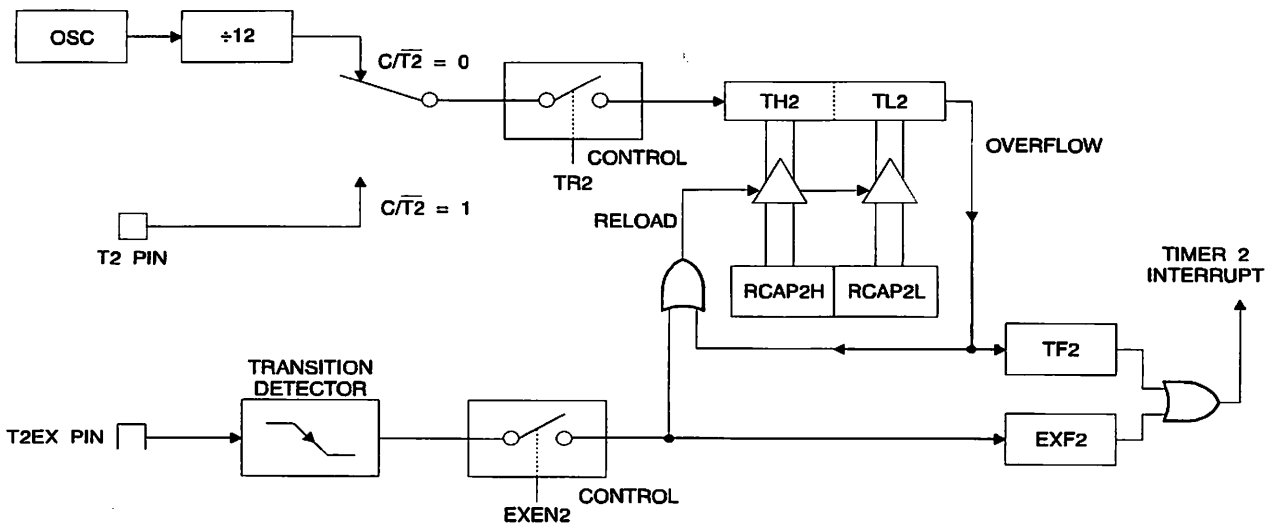


Table 9. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	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)

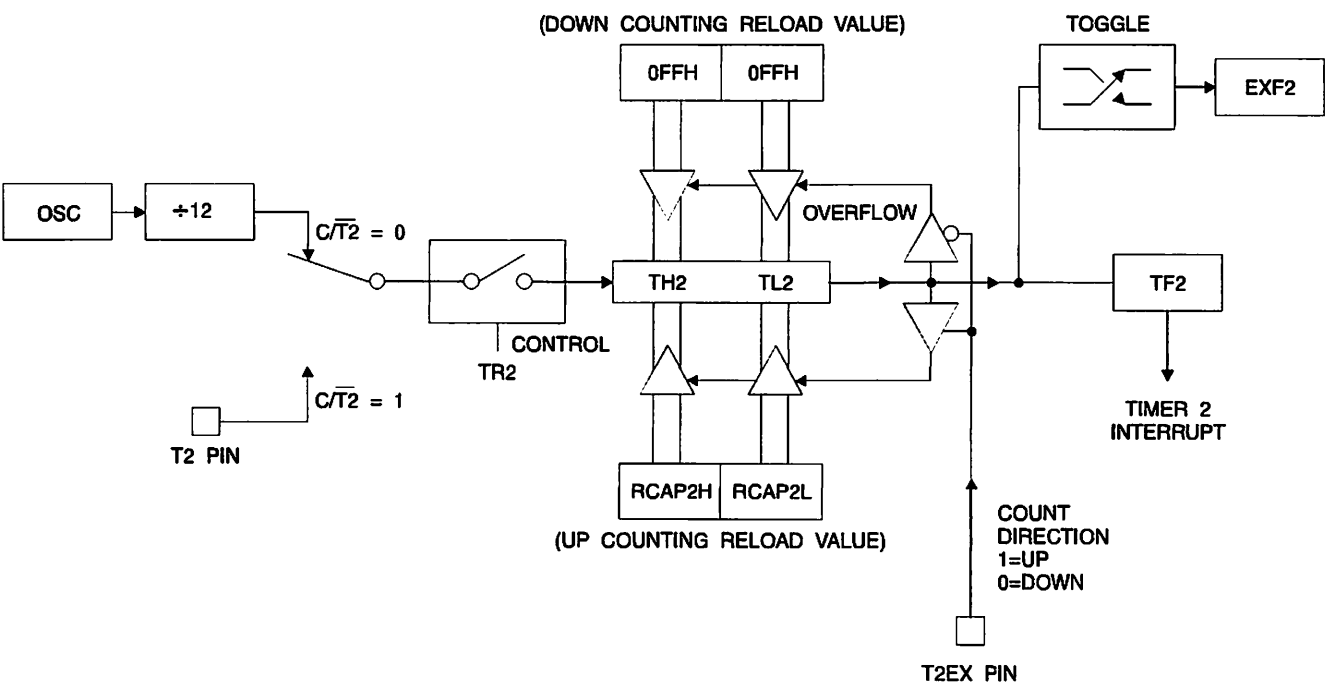
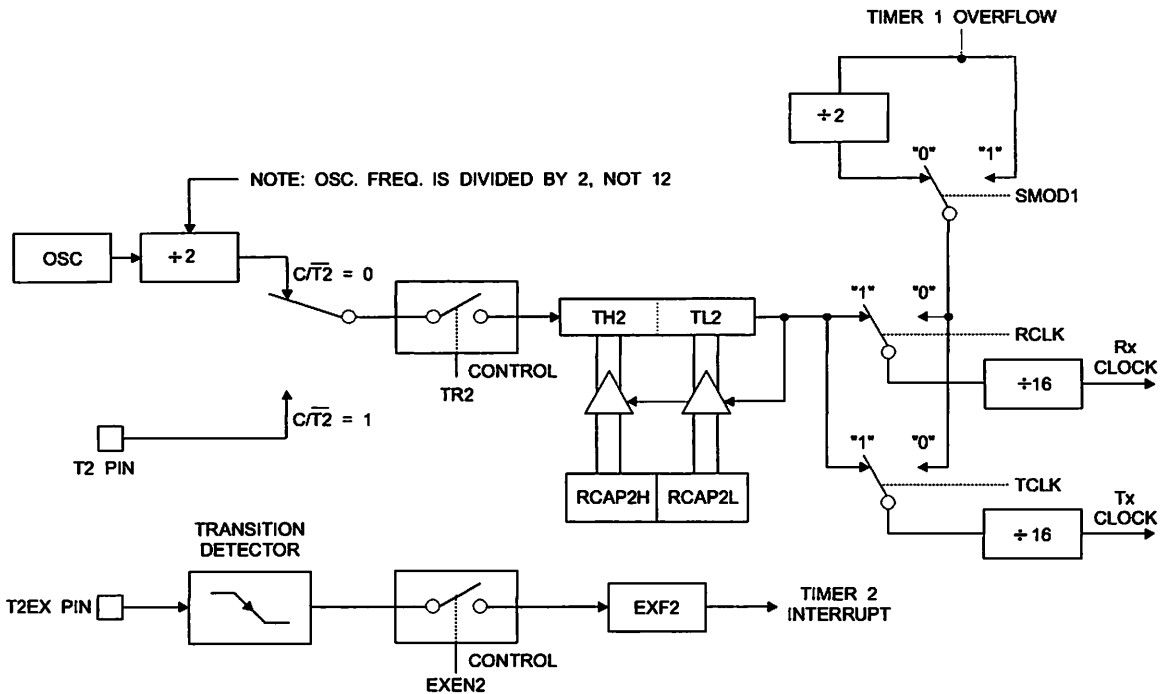


Figure 4. Timer 2 in Baud Rate Generator Mode



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.

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

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 (for a 16-MHz operating frequency).

To configure the Timer/Counter 2 as a clock generator, bit C/T $\bar{2}$ (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

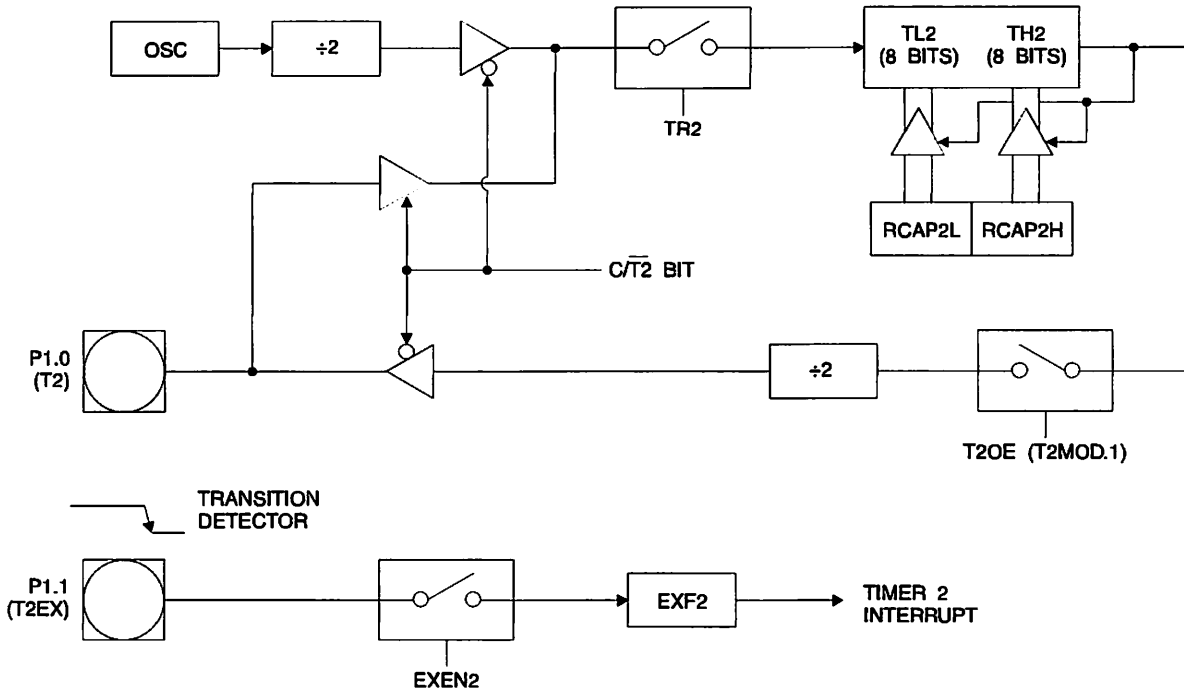
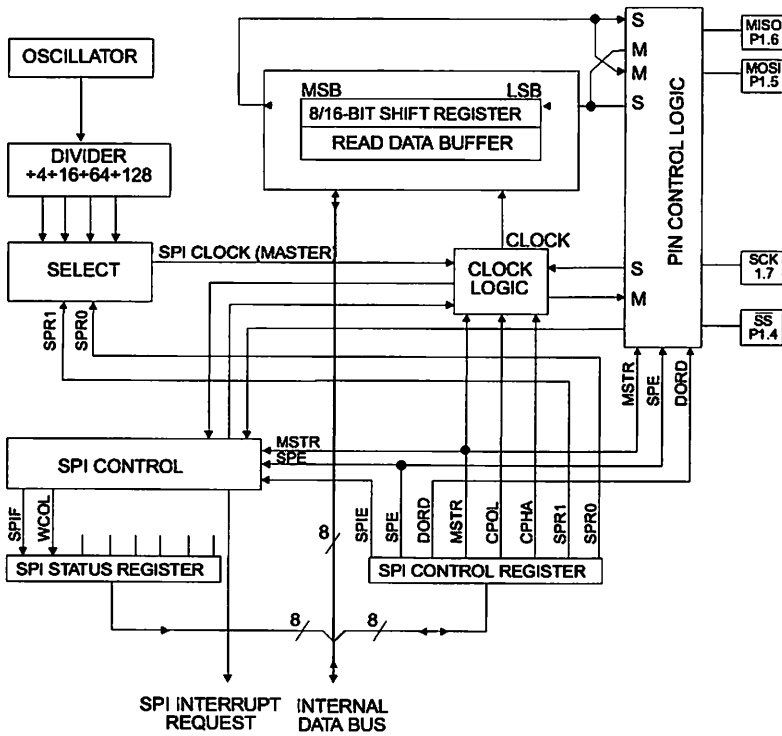


Figure 6. SPI Block Diagram



UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the Atmel web site (<http://www.atmel.com>). From the home page, select "Products", then "Microcontrollers", then "8051-Architecture". Click on "Documentation", then on "Other Documents". Open the document "AT89 Series Hardware Description".

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

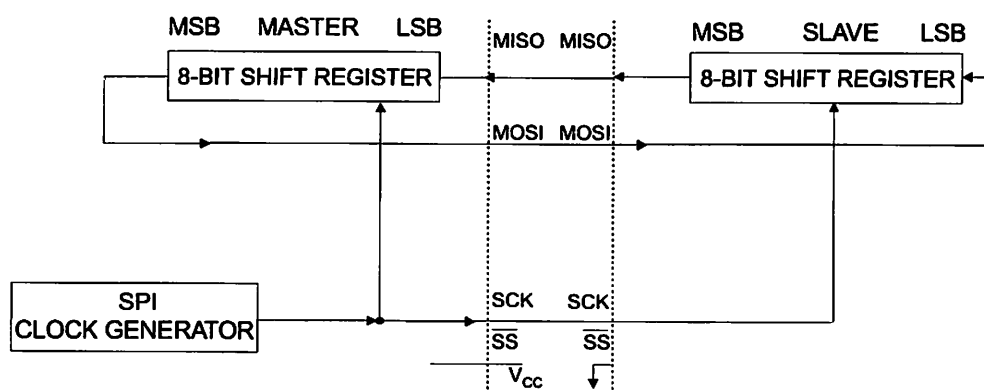
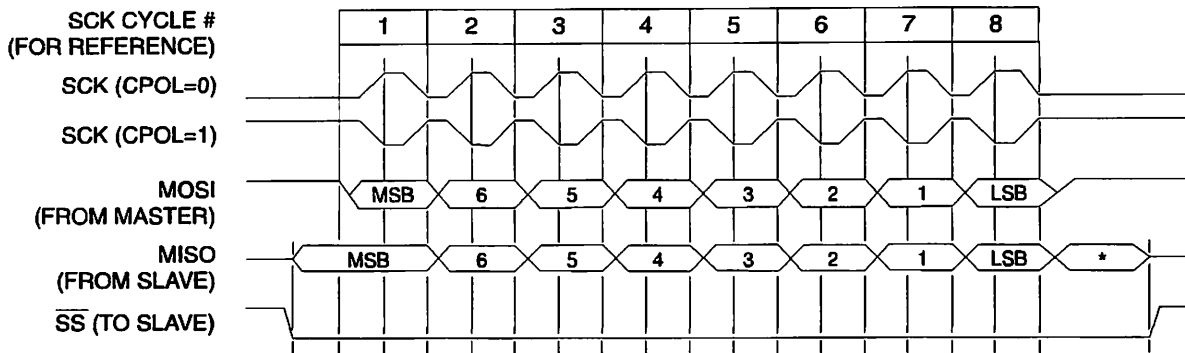
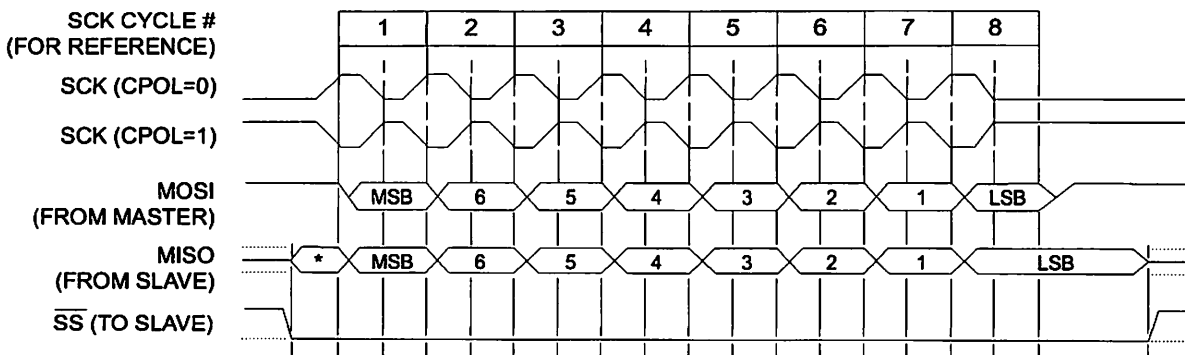


Figure 8. SPI transfer Format with CPHA = 0



Note: *Not defined but normally MSB of character just received

Figure 9. SPI Transfer Format with CPHA = 1



Note: *Not defined but normally LSB of previously transmitted character.

Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), 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.

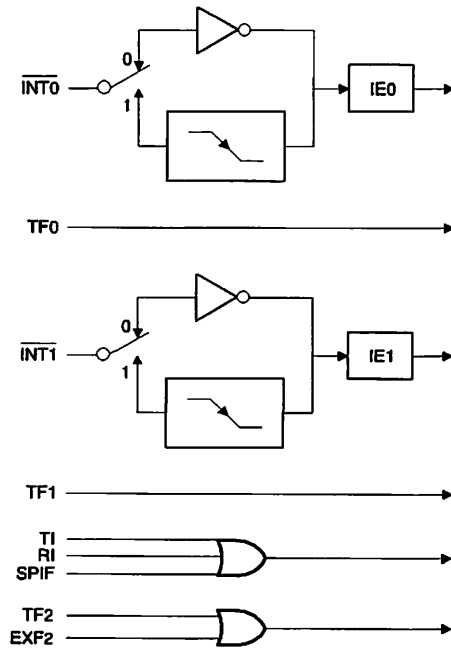


Table 10. Interrupt Enable (IE) Register

(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.

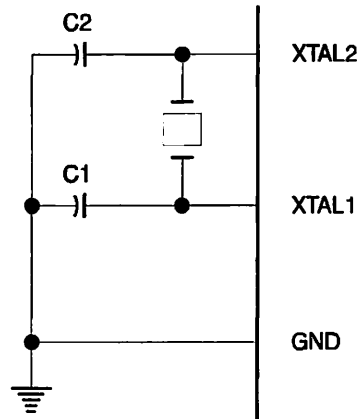
Figure 10. Interrupt Sources



Oscillator
Characteristics

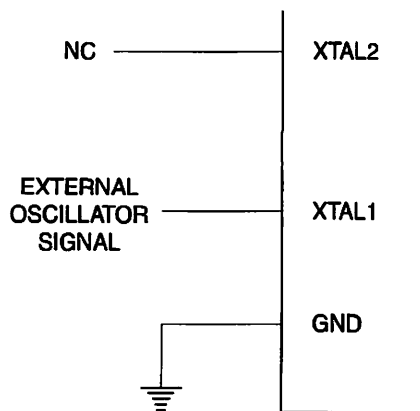
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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.

Figure 11. Oscillator Connections



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

Figure 12. External Clock Drive Configuration





Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special 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	$\overline{\text{PSEN}}$	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 $\overline{\text{EA}}$ 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 $\overline{\text{EA}}$ 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⁽¹⁾⁽²⁾

	Program Lock Bits			Protection Type
	LB1	LB2	LB3	
1	U	U	U	No internal memory lock feature.
2	P	U	U	MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{\text{EA}}$ 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

Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-voltage (12-V V_{PP}) Parallel programming mode and a Low-voltage (5-V V_{CC}) Serial programming mode. The serial programming mode provides a convenient way to reprogram the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming Algorithm: To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set \overline{PSEN} pin to "L"
 - ALE pin to "H"
 - \overline{EA} pin to "H" and all other pins to "H".
3. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
 - Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise \overline{EA}/V_{PP} to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/ \overline{PROG} once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:
 - Set XTAL1 to "L".
 - Set RST and \overline{EA} pins to "L".
 - Turn V_{CC} power off.





In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Data Polling: The AT89S8252 features $\overline{\text{DATA}}$ Polling to indicate the end of a byte write cycle. During a byte write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{DATA}}$ Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate $\overline{\text{BUSY}}$. P3.4 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

Chip Erase: Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

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

Serial Programming Fuse: A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT89S8252 is shipped with the Serial Programming Mode enabled.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 72H indicates 89S8252

Programming Interface

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

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

Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between VCC and GND pins.
 - Set RST pin to "H".
 - If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.
6. Power-off sequence (if needed):
 - Set XTAL1 to "L" (if a crystal is not used).
 - Set RST to "L".
 - Turn V_{CC} power off.





Serial Programming Instruction

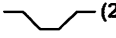

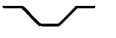
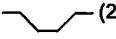

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

Instruction Set

Instruction	Input Format			Operation
	Byte 1	Byte 2	Byte 3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa a001	low addr	xxxx xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa a010	low addr	data in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa a101	low addr	xxxx xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa a110	low addr	data in	Write data to Data memory location at selected address.
Write Lock Bits	1010 1100	LB2 LB1 x111	xxxx xxxx	Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits.

- Notes:
1. DATA polling is used to indicate the end of a byte write cycle which typically takes less than 2.5 ms at 5V.
 2. "aaaaa" = high order address.
 3. "x" = don't care.

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V _{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h ⁽¹⁾	h ⁽¹⁾	x						
Chip Erase	H	L	 (2)	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	 (2)	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	 (2)	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

- Notes:
1. "h" = weakly pulled "High" internally.
 2. Chip Erase and Serial Programming Fuse require a 10 ms $\overline{\text{PROG}}$ pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
 3. P3.4 is pulled Low during programming to indicate RDY/BSY.
 4. "X" = don't care



Figure 13. Programming the Flash/EEPROM Memory

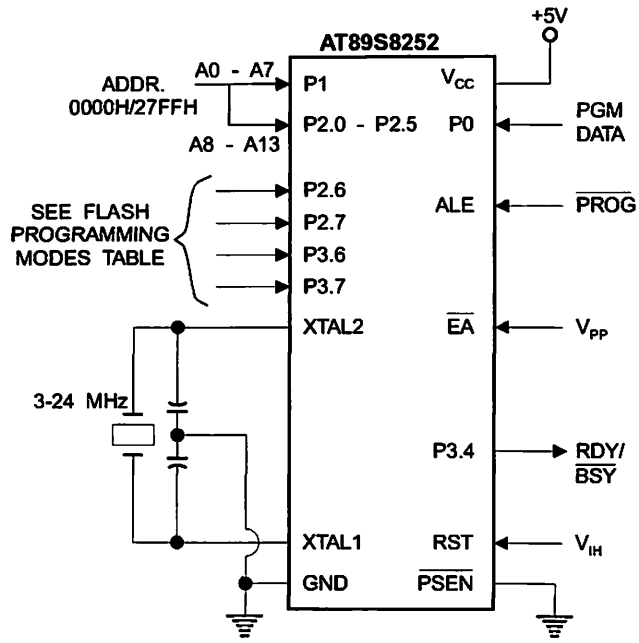


Figure 15. Flash/EEPROM Serial Downloading

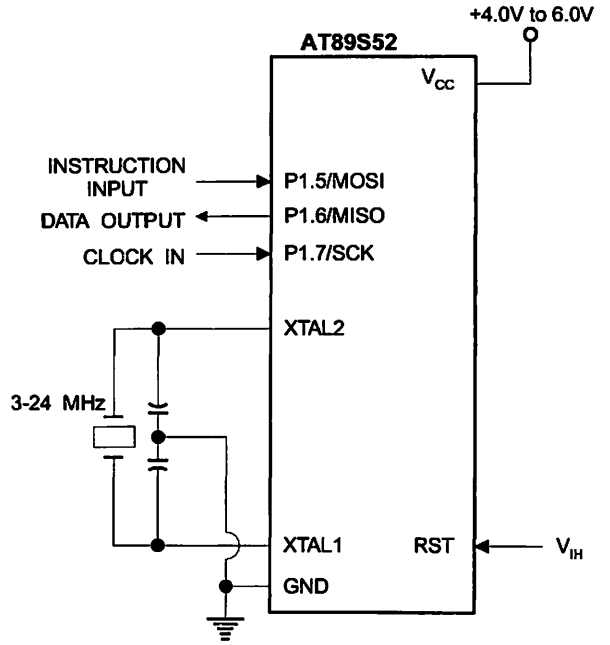
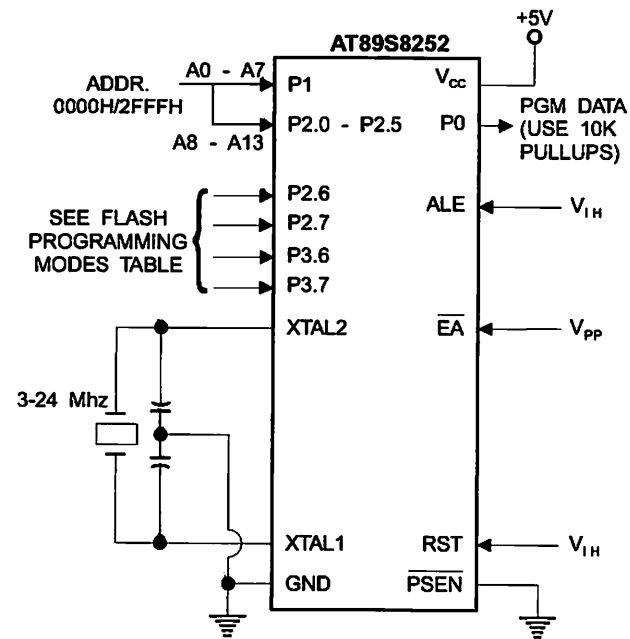


Figure 14. Verifying the Flash/EEPROM Memory

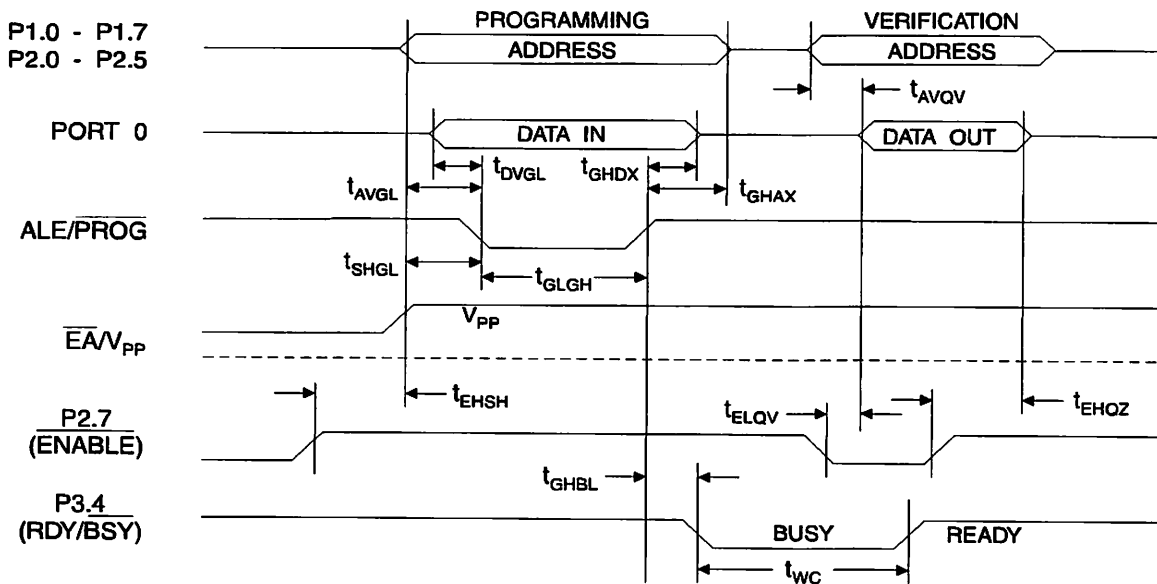


Flash Programming and Verification Characteristics – Parallel Mode

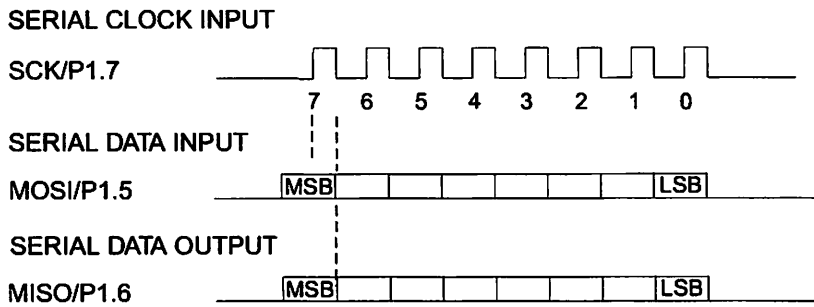
$T_A = 0^\circ\text{C to } 70^\circ\text{C}, V_{CC} = 5.0\text{V} \pm 10\%$

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold after $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHS}	P2.7 (ENABLE) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	1	110	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		2.0	ms

Flash/EEPROM Programming and Verification Waveforms – Parallel Mode



Serial Downloading Waveforms



Serial Programming Characteristics

Figure 16. Serial Programming Timing

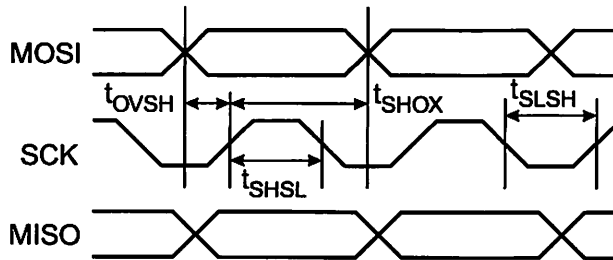


Table 11. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 6.0\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		24	MHz
t_{CLCL}	Oscillator Period	41.6			ns
t_{SHSL}	SCK Pulse Width High	$24 t_{CLCL}$			ns
t_{SLSH}	SCK Pulse Width Low	$24 t_{CLCL}$			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	$2 t_{CLCL}$			ns

Absolute Maximum Ratings*

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

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

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 5.0\text{V} \pm 20\%$, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low-voltage (\overline{EA})		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low-voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.5	V
V_{OL1}	Output Low-voltage ⁽¹⁾ (Port 0, ALE, \overline{PSEN})	$I_{OL} = 3.2 \text{ mA}$		0.5	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, \overline{PSEN})	$I_{OH} = -60 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}$, $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$, $V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{LI}	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pull-down Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 6\text{V}$		100	μA
		$V_{CC} = 3\text{V}$		40	μA

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



FAIRCHILD
SEMICONDUCTOR®

SS9012

1W Output Amplifier of Potable Radios in Class B Push-pull Operation.

- High total power dissipation. ($P_T=625\text{mW}$)
- High Collector Current. ($I_C=-500\text{mA}$)
- Complementary to SS9013
- Excellent h_{FE} linearity.



1 TO-92
1. Emitter 2. Base 3. Collector

PNP Epitaxial Silicon Transistor

Absolute Maximum Ratings $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Ratings	Units
V_{CBO}	Collector-Base Voltage	-40	V
V_{CEO}	Collector-Emitter Voltage	-20	V
V_{EBO}	Emitter-Base Voltage	-5	V
I_C	Collector Current	-500	A
P_C	Collector Power Dissipation	625	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	-55 ~ 150	$^\circ\text{C}$

Electrical Characteristics $T_a=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Typ.	Max.	Units
BV_{CBO}	Collector-Base Breakdown Voltage	$I_C = -100\mu\text{A}, I_E = 0$	-40			V
BV_{CEO}	Collector-Emitter Breakdown Voltage	$I_C = -1\text{mA}, I_B = 0$	-20			V
BV_{EBO}	Emitter-Base Breakdown Voltage	$I_E = -100\mu\text{A}, I_C = 0$	-5			V
I_{CBO}	Collector Cut-off Current	$V_{CB} = -25\text{V}, I_E = 0$			-100	nA
I_{EBO}	Emitter Cut-off Current	$V_{EB} = -3\text{V}, I_C = 0$			-100	nA
h_{FE1} h_{FE2}	DC Current Gain	$V_{CE} = -1\text{V}, I_C = -50\text{mA}$ $V_{CE} = -1\text{V}, I_C = -500\text{mA}$	64 40	120 90	202	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	$I_C = -500\text{mA}, I_B = -50\text{mA}$		-0.18	-0.6	V
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	$I_C = -500\text{mA}, I_B = -50\text{mA}$		-0.95	-1.2	V
$V_{BE(on)}$	Base-Emitter On Voltage	$V_{CE} = -1\text{V}, I_C = -10\text{mA}$	-0.6	-0.87	-0.7	V

h_{FE} Classification

Classification	D	E	F	G	H
h_{FE1}	64 ~ 91	78 ~ 112	96 ~ 135	112 ~ 166	144 ~ 202

1. GENERAL

1.1 General

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

1.2 Features

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

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

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

2. OPERATING INSTRUCTIONS

2.1 Terminal Functions

Table 1 Terminal functions

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

1.7.2 Recommended operating voltage

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

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

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

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

1.7 Optical Characteristics

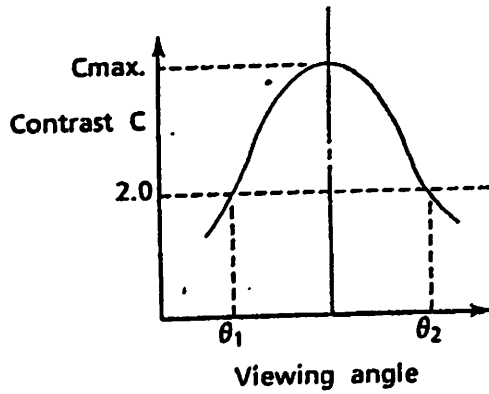
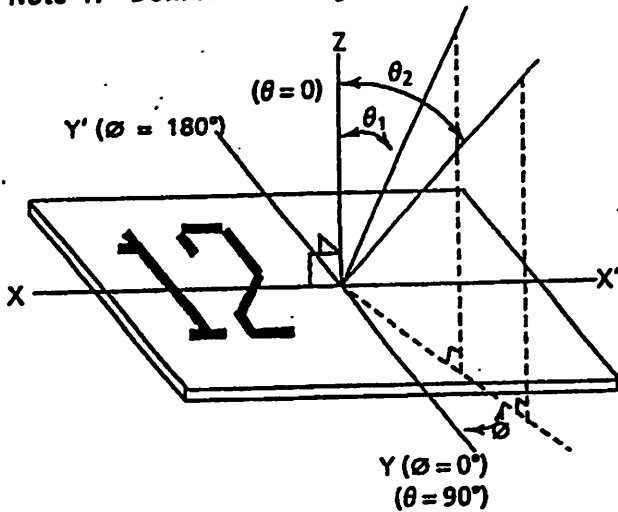
1.7.1 Optical characteristics

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

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

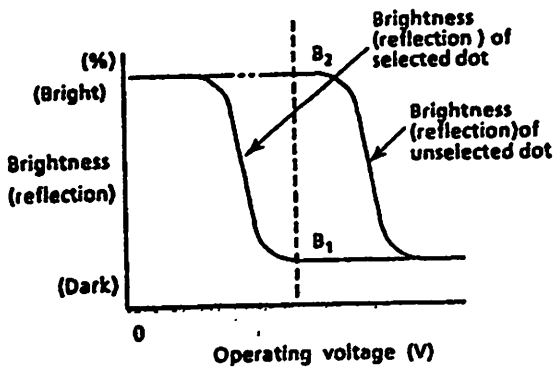
Note 1: Definition of angles \varnothing and θ

Note 2: Definition of viewing angles θ_1 and θ_2

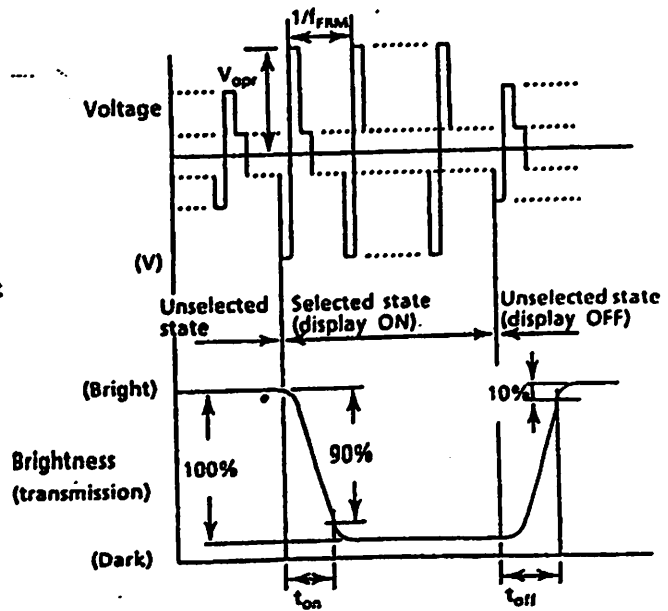


Note 3: Definition of contrast C

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



Note 4: Definition of response time



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

1.5 Absolute Maximum Ratings

$V_{SS} = 0V$

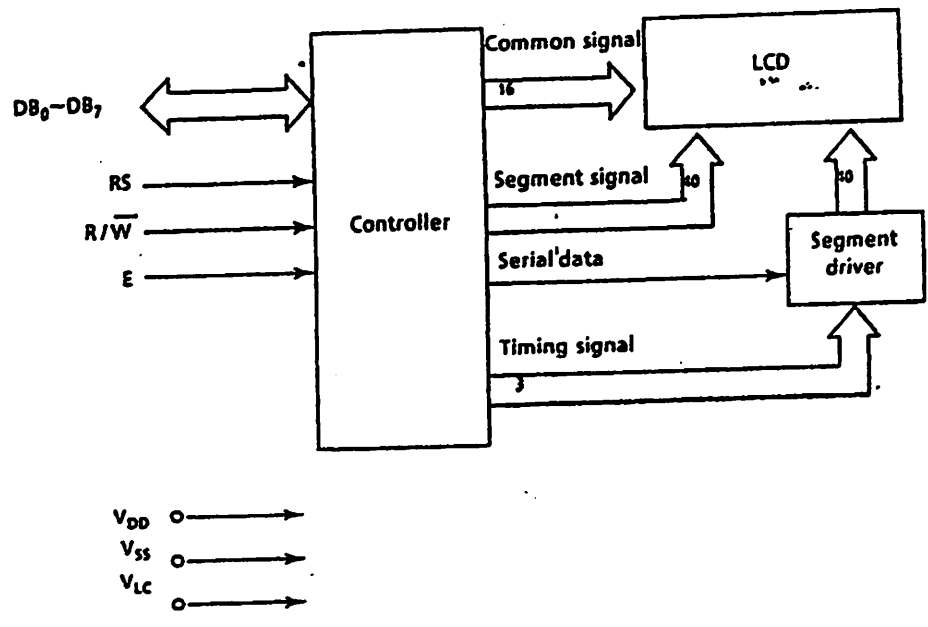
Item	Symbol	Standard	Unit	Remarks
Power supply voltage	V_{DD}	-0.3 to +7.0	V	
	V_{LC}	$V_{DD} - 13.5$ to $V_{DD} + 0.3$	V	
Input voltage	V_{in}	-0.3 to $V_{DD} + 0.3$	V	
Operating temperature	T_{opr}	0 to +50	°C	
Storage temperature	T_{stg}	-20 to +60	°C	At 50% RH

1.6 Electrical Characteristics

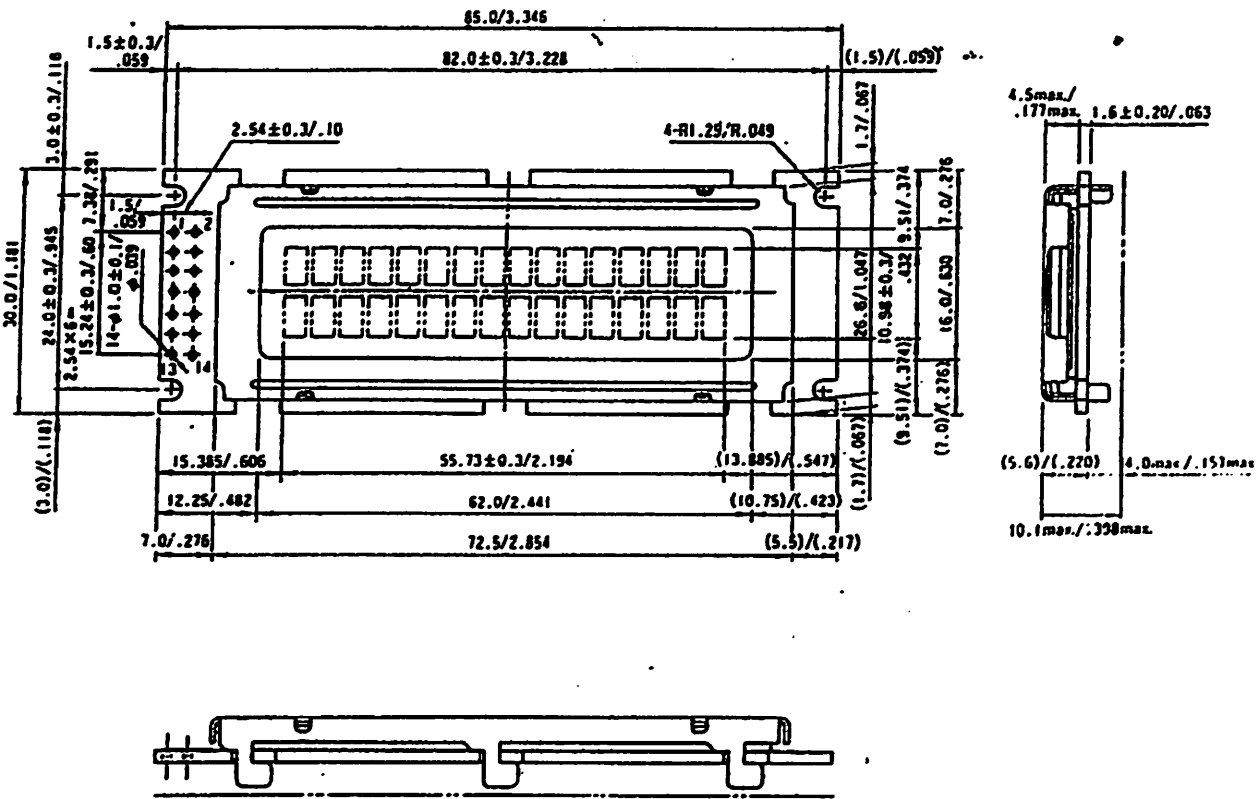
$V_{DD} = 5V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ C$ to $50^\circ C$

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

1.4 Block Diagram



1.3 Dimensions Diagram



Unit : mm/inch
General tolerance : ± 0.5 mm

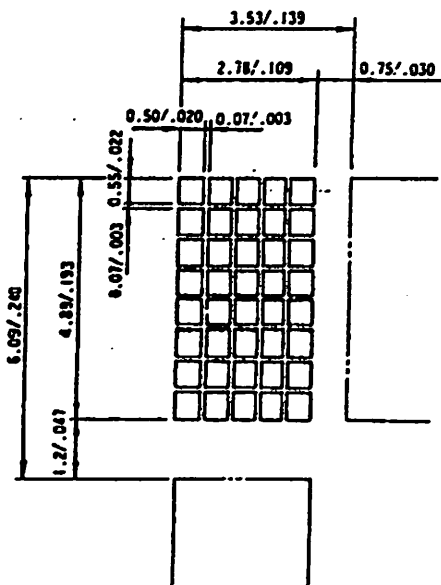


Figure 1 Dimensions diagram

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