

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



**PERANCANGAN DAN PEMBUATAN OVEN TERPROGRAM  
DENGAN SISTEM MENU BARBASIS MIKROKONTROLLER  
AT89S52**

**SKRIPSI**

**Disusun oleh :**

**Mochamad Yusuf  
9917181**

**MARET 2005**



# LEMBAR PERSETUJUAN

## PERANCANGAN DAN PEMBUATAN OVEN TERPROGRAM DENGAN SISTEM MENU BERBASIS MIKROKONTROLLER

AT89S52

### SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi dan Memenuhi Syarat  
Guna Mencapai Gelar Sarjana Teknik*

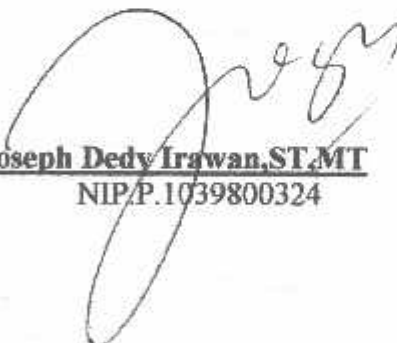
Disusun oleh :

**Mochamad Yusuf**

9917181

  
Mengetahui  
Ketua Jurusan Teknik Elektro-S1  
  
Ir. F. Yusi Lampraptono, MT  
NIP. Y. 1079500274

Diperiksa dan Disetujui  
Dosen Pembimbing

  
Joseph Dedy Irawan, ST, MT  
NIP. P. 1039800324

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



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

## BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Mochamad Yusuf  
NIM : 99.17.181  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan Dan Pembuatan Oven Terprogram  
Dengan Sistem Menu Berbasis Mikrokontroler  
AT89S52

Dipertahankan dihadapan Team Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa  
Tanggal : 29 Maret 2005  
Nilai : 76,6 ( B )

### Panitia Ujian Skripsi



Ketua

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

Sekretaris

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

### Anggota Penguji

Penguji Pertama

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

Penguji Kedua

(Ir. H. Erfan Ahmad Dahlan)  
NIP. 131124663

## ABSTRAKSI

( Mochamad Yusuf , 9917181 , Jurusan Teknik Elektro S-1/Elektronika )

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

Seiring perkembangan jaman, manusia semakin dituntut untuk melakukan segala sesuatu bukan hanya secara cepat, tetapi juga harus tepat. Untuk itu, manusia senantiasa mencari akal guna mempermudah pekerjaannya sehari-hari.

Berkaitan dengan hal tersebut, manusia juga cenderung untuk mencari alternatif pemecahan masalah bagi pekerjaan yang monoton dan sering dilakukan secara manual sehingga pekerjaan tersebut dapat dikerjakan secara otomatis dengan sedikit campur tangannya.

Saat ini terdapat beberapa jenis oven baik yang manual maupun otomatis. Kekurangan pada oven yang terdapat dipasaran adalah tidak adanya fasilitas untuk penambahan menu masakan atau dengan kata lain, seting waktu telah ditentukan oleh pabrik. Pada tugas akhir ini akan didesain suatu oven yang dapat menambah menu masakan sesuai dengan keinginan kita.

Jadi sangatlah relevan bila judul yang saya angkat dalam penyusunan Tugas Akhir ini adalah

**PERANCANGAN DAN PEMBUATAN OVEN TERPROGRAM DENGAN SISTEM MENU BERBASIS MIKROKONTROLLER AT89S52.**



## KATA PENGANTAR

Puji syukur kehadirat Allah Yang Maha Pengasih dan Penyayang , karena dengan petunjuk dan rahmat – Nyalah penyusunan Tugas Akhir ini dapat terselesaikan dengan baik.

Penyusunan Laporan Tugas Akhir yang berjudul Perancangan Dan Pembuatan Oven Terprogram Dengan Sistem Menu Berbasis Mikrokontroler AT89S52 ini , selain guna memenuhi syarat kelulusan pada Jurusan Teknik Elektro Program Studi SI , juga agar dapat mengaplikasikanya secara nyata teori – teori yang telah didapat selama masa kuliah dan perkuliahan sehingga pemahaman yang diperoleh akan lebih mantab.

Dalam penyusunan Laporan Tugas Akhir ini , penyusun banyak mengalami kesulitan baik dari segi Teknis maupun Non Teknis. Oleh karena itu penyusun menyampaikan banyak terimakasih kepada :

1. Bapak Dr.Ir.Abraham Lomi,MSEE, selaku Rektor ITN Malang
2. Bapak Ir Mochtar Asroni,MSME selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. Bapak Ir.F.Yudi Limpraptono,MT, selaku Ketua Jurusan Teknik Elektro S1 ITN Malang.
4. Ibu Ir Mimien Mustikawati, selaku Sekretaris Jurusan Teknik Elektro S1 ITN Malang.

5. Bapak Joseph Dedy Irawan,ST,MT , selaku Dosen Pembimbing yang telah memberikan Bimbingan dengan baik hingga akhir penyusunan Laporan Tugas Akhir saya ini.
7. Bapak dan Ibu dosen ITN Malang bersama Staf.
8. Rekan – rekan dan semua pihak yang telah memberikan bantuan dalam bermacam bentuk.

Penyusunan Laporan Tugas Akhir ini masih jauh dari kesempurnaan , mengingat keterbatasan Ilmu Pengetahuan dan kemampuan yang Penyusun miliki. Sehingga Penyusun mengharapkan adanya kritik dan saran dari para pembaca yang bersifat membangun.

Akhirnya semoga penyusunan Laporan Tugas Akhir ini dapat bermanfaat bagi Penyusun sendiri dan Pembaca semua yang tertarik dengan Tugas Akhir ini.

Malang , Maret 2005

Penyusun

## DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN .....	ii
BERITA ACARA UJIAN SKRIPSI.....	iii
ABSTRAKSI.....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xii
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	1
1.3. Batasan Masalah.....	2
1.4. Tujuan Pembahasan.....	2
1.5. Metode Penelitian.....	3
1.6. Sistematika Pembahasan .....	5
<b>BAB II DASAR TEORI.....</b>	<b>6</b>
2.1. Minimum Sistem AT89S52 .....	6

2.1.1. Mikrokontroller AT89S52 .....	6
2.1.2. Memori.....	15
2.1.3. PPI 8255.....	17
2.2. Real Time Clock DS12887A .....	22
2.3. Penampil Kristal Cair.....	25
2.4. EPROM.....	27
2.5. EEPROM.....	27
<b>BAB III PERENCANAAN DAN REALISASI ALAT .....</b>	<b>28</b>
3.1. Perangkat Keras .....	29
3.1.1. Sistem Mikrokontroller .....	29
3.1.1.1. Mikrokontroller AT89S52.....	29
3.1.1.2. Dekoder Dan Peta Alamat.....	34
3.1.1.3. Latch .....	37
3.1.1.4. EEPROM.....	38
3.2. PPI 8255 .....	39
3.3. LCD.....	40
3.4. Papan Tombol .....	41
3.5. Rangkaian Sensor.....	43
3.6. Rangkaian Penguat Sinyal.....	45
3.7. Rangkaian Pengubah Data Analog ke Digital .....	47
3.8. Driver Buzzer .....	50

3.9. Perancangan Perangkat Lunak .....	51
<b>BAB IV PENGUJIAN DAN ANALISIS .....</b>	<b>54</b>
4.1. Umum .....	54
4.2. Pengujian Papan Tombol .....	55
4.3. Pengujian Penguat Sinyal .....	58
4.4. Pengujian Rangkaian ADC .....	61
4.5. Pengujian Sistem Mikrokontroller .....	63
4.6. Pengujian Rangkaian Tampilan .....	66
4.7. Pengujian Pencatatan Data Pada EEPROM .....	66
4.8. Pengujian RTC .....	68
4.9. Pengujian Transduser Suhu .....	69
<b>BAB V PENUTUP .....</b>	<b>73</b>
5.1. Kesimpulan .....	73
5.2. Saran .....	73

## DAFTAR PUSTAKA

## LAMPIRAN

## DAFTAR GAMBAR

GAMBAR	HALAMAN
2 – 1 DIAGRAM BLOK MCS 52.....	8
2 – 2 BLOK DIAGRAM PPI 8255 .....	18
2 – 3 CONTROL WORD PPI 8255 .....	19
2 – 4 PEMBAGIAN ALAMAT RAM INTERNAL RTC DS 12887A .....	23
3 – 1 BLOK DIAGRAM.....	28
3 – 2 RANGKAIAN MIKROKONTROLLER AT89S52 .....	30
3 – 3 RANGKAIAN PEWAKTUAN .....	31
3 – 4 RANGKAIAN POWER ON RESET.....	32
3 – 5 RANGKAIAN DEKODER ALAMAT.....	37
3 – 6 RANGKAIAN EEPROM 6264.....	39
3 – 7 RANGKAIAN PPI 8255 .....	40
3 – 8 RANGKAIAN LCD.....	41
3 – 9 RANGKAIAN KEYPAD.....	43
3 – 10 RANGKAIAN SENSOR .....	44
3 – 11 PENGUAT TEGANGAN .....	46
3 – 12 PEMBANGKIT CLOCK .....	48
3 – 13 RANGKAIAN ADC 0809 .....	49
3 – 14 RANGKAIAN DRIVER BUZZER .....	50
4 – 1 PENGUJIAN PAPAN TOMBOL.....	56

4 – 2 RANGKAIAN PENGUAT SINYAL TL 074.....	59
4 – 3 PENGUJIAN PENGUATAN PENGUAT SINYAL TL 074 .....	60
4 – 4 DIAGRAM BLOK PENGUJIAN RANGKAIAN ADC .....	62
4 – 5 DIAGRAM BLOK PENGUJIAN MIKROKONTROLLER.....	65
4 – 6 DIAGRAM BLOK PENGUJIAN RANGKAIAN TAMPILAN.....	66
4 – 7 DIAGRAM BLOK PENGUJIAN RANGKAIAN PENYIMPAN DATA EEPROM.....	67
4 – 8 DIAGRAM BLOK PENGUJIAN LCD.....	68
4 – 9 DIAGRAM BLOK PENGUJIAN RTC .....	69

## DAFTAR TABEL

TABEL	HALAMAN
2-1 KELUARGA MCS 52.....	7
2-2 NAMA DAN ALAMAT REGISTER PADA REGISTER FUNGSI KHUSUS.....	11
2-3 FUNGSI KHUSUS PORT 3 .....	14
2-4 TINGKATAN PRIORITAS INTERUPSI .....	15
2-5 OPERASI DASAR PPI 8255.....	20
2-6 FUNGSI PENYEMAT RTC DS12887A.....	24
2-7 FUNGSI PENYEMAT CRISTAL CAIR TM162.....	26
3-1 PEMETAAN ALAMAT .....	35
3-2 TABEL DEKODER 74LS138 .....	36
3-3 TABEL FUNGSI LATCH 74LS573.....	38
3-4 HUBUNGAN ANTARA SUHU DAN TEGANGAN KELUARAN SENSOR.....	45
3-5 HUBUNGAN ANTARA TEGANGAN MASUKAN DAN KELUARAN PENGUAT SINYAL .....	46
4-1 HASIL PENGUJIAN PAPAN TOMBOL .....	57
4-2 HASIL PENGUJIAN PENGUAT SINYAL DENGAN MULTIMETER .....	61
4-3 HASIL DAN ANALISIS PENGUJIAN RANGKAIAN ADC.....	63



4 – 4	HASIL PENGUJIAN SISTEM MIKROKONTROLLER.....	65
4 – 5	TEGANGAN OUT PUT LM 35 DIBANDINGKAN DENGAN SUHU AKTUAL YANG DIUKUR DENGAN TERMOMETER.....	70
4 – 6	HASIL PENGUJIAN KESTABILAN SENSOR SUHU.....	71

# **BAB I**

## **PENDAHULUAN**

### **1.1. LATAR BELAKANG**

Seiring perkembangan jaman, manusia semakin dituntut untuk melakukan segala sesuatu bukan hanya secara cepat, tetapi juga harus tepat. Untuk itu, manusia senantiasa mencari akal guna mempermudah pekerjaannya sehari-hari.

Berkaitan dengan hal tersebut, manusia juga cenderung untuk mencari alternatif pemecahan masalah bagi pekerjaan yang monoton dan sering dilakukan secara manual sehingga pekerjaan tersebut dapat dikerjakan secara otomatis dengan sedikit campur tangannya.

Saat ini terdapat beberapa jenis oven baik yang manual maupun otomatis. Kekurangan pada oven yang terdapat dipasaran adalah tidak adanya fasilitas untuk penambahan menu masakan atau dengan kata lain, seting waktu telah ditentukan oleh pabrik. Pada tugas akhir ini akan didesain suatu oven yang dapat menambah menu masakan sesuai dengan keinginan kita.

### **1.2. RUMUSAN MASALAH**

Mengacu pada latar belakang masalah di atas, maka rumusan masalah dapat dijabarkan sebagai berikut:

- Bagaimana mendesain dan membuat suatu sistem berbasis mikrokontroler AT89S52.
- Bagaimana mensinkronisasi RTC sebagai basis pewaktuannya dengan mikrokontroler.

- Bagaimana mikrokontroler dapat menampilkan menu pilihan pemrograman secara interaktif pada LCD.
- Bagaimana merancang program oven elektronik terprogram.

Dengan alasan tersebut maka disusunlah suatu Tugas Akhir dengan judul “**PERANCANGAN DAN PEMBUATAN OVEN TERPROGRAM DENGAN SISTEM MENU BERBASIS MIKROKONTROLLER AT89S52**”.

### **1.3. BATASAN MASALAH**

Mengingat begitu luasnya bahan kajian mengenai oven, maka pokok bahasan dibatasi sebagai berikut:

- Pengendali utama sistem oven ini menggunakan mikrokontroler 89S52.
- Basis pewaktuannya menggunakan sebuah komponen RTC (*Real Time Clock*).
- Bila ada peralatan yang mempunyai fungsi yang sama, tetapi berbeda dalam metode, model fisik, dan rangkaianannya tidak akan dibahas.
- Suhu dan Timer sudah diset pada masing – masing menu.

### **1.4. TUJUAN PEMBAHASAN**

Tujuan penulisan tugas akhir ini adalah merencanakan dan membuat alat oven otomatis terprogram dengan menggunakan mikrokontroler AT89S52.

---

### 1.5. Metode Penelitian

Untuk mewujudkan alat oven otomatis ini, dilalui tahapan-tahapan sebagai berikut:

- Studi Literatur

Diperlukan untuk mempelajari dasar teori yang berhubungan dengan mikrokontroler, RTC (*Real Time Clock*) dan komponen pendukung sistem yang lain serta pengumpulan data-data mengenai karakteristik komponen yang akan digunakan. Data tersebut digunakan sebagai dasar perencanaan dan pembuatan alat.

- Perencanaan Alat

- Sebelum dilakukan pembuatan alat, terlebih dahulu dilakukan penentuan spesifikasi alat yang akan dirancang tersebut

Berdasarkan spesifikasi di atas, maka dilakukan perancangan alat dengan langkah-langkah sebagai berikut:

- Perancangan blok diagram keseluruhan sistem
- Perancangan rangkaian untuk masing-masing blok.
- Perancangan perangkat lunak.

- Pembuatan Alat

Dimulai dari pembuatan PCB (*Printed Circuit Board*), dilanjutkan dengan tahap perakitan komponen pada PCB, dan terakhir adalah tahap pembuatan perangkat lunak yang diperlukan untuk mengendalikan perangkat keras sistem.

- Pengujian Rangkaian

---

Pengujian dilakukan untuk mengetahui kesesuaian antara alat yang dirancang dengan spesifikasi yang telah ditentukan sebelumnya.

- **Pengambilan Kesimpulan**

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari perancangan dan pengujian alat. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang ditentukan saat dilakukan perancangan berarti pembuatan alat tersebut telah sesuai dengan harapan.

---

## **1.6. SISTEMATIKA PEMBAHASAN**

### **BAB I : Pendahuluan**

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan pembahasan, metode penelitian.

### **BAB II : TEORI PENUNJANG**

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

### **BAB III : PERANCANGAN DAN PEMBUATAN ALAT**

Dijelaskan mengenai diagram blok rangkaian , rangkaian dengan komponen – komponen pendukungnya (rangkaiian lengkap) , serta memberikan penjelasan mengenai cara kerja rangkaian .

### **BAB IV : PENGUJIAN**

Memuat hasil pengujian terhadap alat yang telah dibuat.

### **BAB V : PENUTUP**

Memuat kesimpulan dan saran-saran.

---

## BAB II

### DASAR TEORI

Dalam merencanakan dan merealisasikan sistem ini dibutuhkan pemahaman mengenai pengetahuan yang berhubungan dengan aplikasi tersebut. Pemahaman tersebut akan sangat bermanfaat dalam perancangan perangkat keras maupun perangkat lunak sistem. Adapun pengetahuan yang mendukung perencanaan dan realisasi alat antara lain sistem mikrokontroler AT89S52, *real time clock* DS12887A, serta penampil kristal cair.

#### 2.1. Minimum Sistem AT89S52

##### 2.1.1. Mikrokontroler AT89S52

Mikrokontroler AT89S52 merupakan salah satu anggota keluarga MCS-51, yaitu suatu komponen produksi ATME1 yang berorientasi kontrol (*microcontroller*). Intel mengklarifikasikan dalam kelompok *embedded microcontroller*, yang artinya adalah mikrokontroler yang dapat diprogram ulang (*reprogrammable*). Di dalam *chip* mikrokontroler AT89S52 ini sudah tersedia berbagai macam peralatan pendukung mikroprosesor seperti RAM, *serial port*, *bus-bus* data dan lainnya yang membuat pemakai *chip* ini dapat menekan penambahan komponen pendukung. Spesifikasi perangkat keras dari mikrokontroler AT89S52 adalah sebagai berikut :

- CPU (*Central processing unit*) dengan lebar data 8 bit.
- Prosesor *Boolean* untuk operasi logika 1 bit.
- Pembangkit *clock* internal.
- Dua buah *timer/counter* 16 bit.
- Dua buah saluran interupsi eksternal.
- Jalur I/O dua arah (*bidirectional*) 32 buah.
- Memori program terpisah dari memori data.
- Memori data internal 128 byte.
- Alamat memori program eksternal 64 Kilobyte.
- Alamat memori data eksternal 64 Kilobyte.

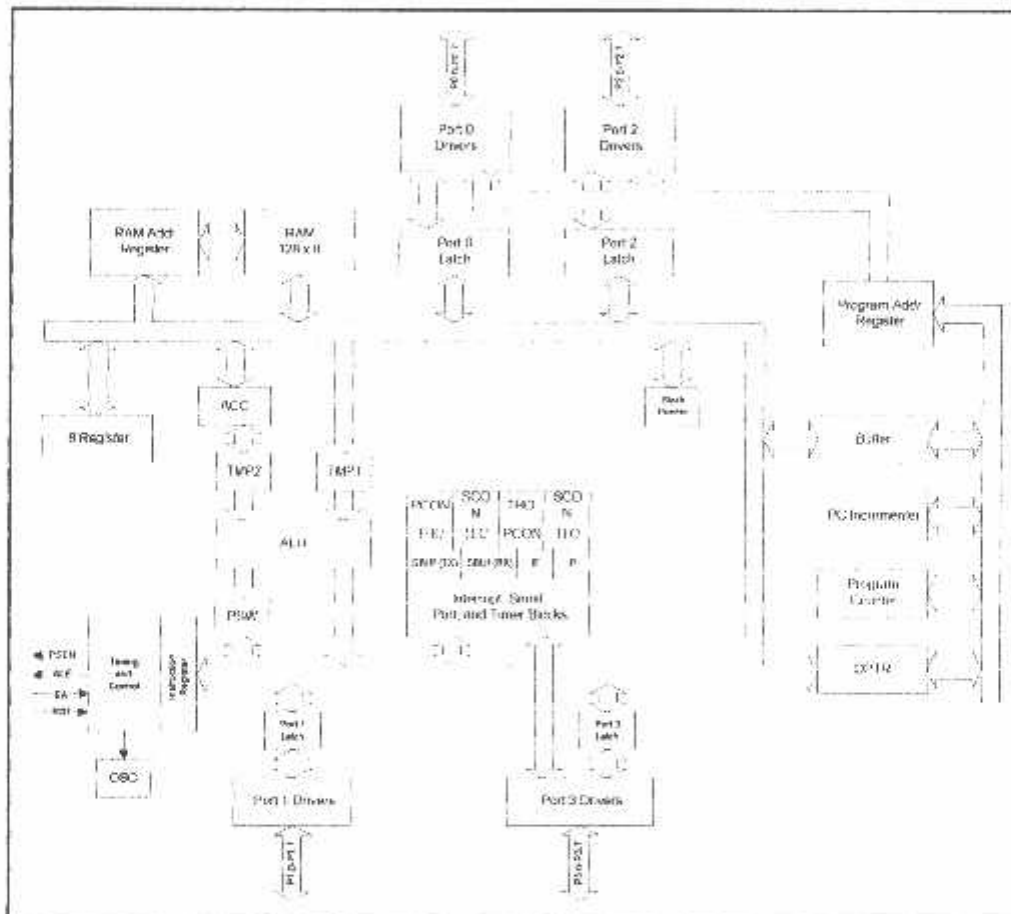
Tabel 2-1 Keluarga MCS<sup>®</sup>52

Tipe	Tipe tanpa EPROM	Tipe ber-EPROM	Kapasitas ROM	Kapasitas RAM	Port I/O	Pewaktu
8051	8031	-	4K	128	4	2
8051AH	8031AH	8751H	4K	128	4	2
8052AH	8032AH	8752BH	8K	256	4	3
80C51BH	80C31BH	87C51	4K	128	4	2
80C52	80C32	-	8K	256	4	3
83C51FA	80C51BI	8751FA	8K	256	4	3
83C51FB	80C51FB	87C51FB	16K	256	4	3
83C152	80C152	-	8K	256	5	3

Sumber : Intel, 1994:1-9



Keluarga MCS<sup>®</sup>52 yang diproduksi Intel mempunyai konfigurasi yang berbeda-beda sesuai dengan jenisnya. Masing-masing jenis saling kompatibel serta mempunyai kelebihan tersendiri. Sedangkan diagram blok MCS<sup>®</sup>52 secara umum ditunjukkan dalam Gambar 2-1.



Gambar 2-1 Diagram Blok MCS<sup>®</sup>52

Sumber : Intel,1994:3-4

#### a. Organisasi Memori

Organisasi memori pada mikrokontroler AT89S52 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program digunakan untuk menyimpan instruksi-instruksi yang akan dijalankan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat penyimpanan data yang sedang diolah mikrokontroler.

Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler AT89S52 dilengkapi dengan ROM internal namun untuk program yang besar digunakan ROM eksternal yang terpisah dari mikrokontroler. Untuk dapat menggunakan memori program eksternal ini penyemat  $\overline{EA}$  dihubungkan dengan penyemat  $V_{SS}$  (logika 0).

Memori program mikrokontroler menggunakan alamat 16 *bit* mulai  $0000_{16}$ - $FFFF_{16}$ , sehingga kapasitas penyimpanan program maksimal adalah  $2^{16}$  *byte* atau 64 Kb. Sinyal yang digunakan untuk membaca memori program eksternal adalah sinyal  $\overline{PSEN}$  (*program store enable*).

Selain memori program mikrokontroler AT89S52 juga memiliki memori data internal berkapasitas 128 *byte* dan mampu mengakses memori data eksternal sebesar 64 Kb. Semua memori data internal dapat dialamati dengan pengalamatan langsung atau tidak langsung. Ciri dari pengalamatan langsung adalah *operand* berisi alamat data yang diolah. Sedangkan ciri dari pengalamatan tidak langsung adalah *operand* adalah alamat *register* yang berisi alamat data yang akan diolah. Sebagian memori

---

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

#### b. Register Fungsi Khusus

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

Register-register ini hanya dapat diakses dengan pengalamatan langsung. Enam belas alamat pada register fungsi khusus dapat dialamati *perbit* maupun *per-byte* dan terletak pada alamat  $80_{11}-FF_{11}$ . Secara perangkat keras, register fungsi khusus ini dibedakan dengan memori data internal.

---

Tabel 2-2 Nama dan Alamat Register pada Register Fungsi Khusus

Simbol	Nama Register	Nilai pada saat reset	Alamat
ACC	<i>Accumulator</i>	0000 <sub>11</sub>	0E0 <sub>11</sub>
B	<i>Register B</i>	00 <sub>11</sub>	0F0 <sub>11</sub>
PSW	<i>Program Status Word</i>	00 <sub>11</sub>	0D0 <sub>11</sub>
SP	<i>Stack Pointer</i>	07 <sub>11</sub>	81 <sub>11</sub>
DPTR	<i>Data Pointer 2 bytes</i>		
DPL	<i>Low bytes</i>	0000 <sub>11</sub>	82 <sub>11</sub>
DPH	<i>High bytes</i>	0000 <sub>11</sub>	83 <sub>11</sub>
P0	<i>Port 0</i>	FF <sub>11</sub>	80 <sub>11</sub>
P1	<i>Port 1</i>	FF <sub>11</sub>	90 <sub>11</sub>
P2	<i>Port 2</i>	FF <sub>11</sub>	0A0 <sub>11</sub>
P3	<i>Port 3</i>	FF <sub>11</sub>	0B0 <sub>11</sub>
IP	<i>Interrupt priority control</i>	XXX00000 <sub>13</sub>	0B8 <sub>11</sub>
IE	<i>Interrupt enable control</i>	0XX00000 <sub>13</sub>	0A8 <sub>11</sub>
TMOD	<i>Timer/counter mode control</i>	00 <sub>11</sub>	89 <sub>11</sub>
TCON	<i>Timer/counter control</i>	00 <sub>11</sub>	88 <sub>11</sub>
TH0	<i>Timer/counter 0 high byte</i>	00 <sub>11</sub>	8C <sub>11</sub>
TL0	<i>Timer/counter 0 low byte</i>	00 <sub>11</sub>	8A <sub>11</sub>
TH1	<i>Timer/counter 1 high byte</i>	00 <sub>11</sub>	8D <sub>11</sub>
TL1	<i>Timer/counter 1 low byte</i>	00 <sub>11</sub>	8B <sub>11</sub>
SCON	<i>Serial control</i>	00 <sub>11</sub>	98 <sub>11</sub>
SBUF	<i>Serial data buffer</i>	Independen	99 <sub>11</sub>
PCON	<i>Power control</i>	HMOS 0XXXXXXXX <sub>B</sub> CHMOS 0XXX0000 <sub>13</sub>	87 <sub>11</sub>

Sumber : Intel, 1994:2-8

Beberapa macam register fungsi khusus yang sering digunakan, dijelaskan sebagai berikut [Intel, 1994:3-5] :

- *Accumulator (ACC)* merupakan *register* untuk penambahan dan pengurangan. Perintah *Mnemonic* untuk mengakses akumulator disederhanakan sebagai *A*.
  - *Register B* merupakan *register* khusus yang berfungsi melayani operasi perkalian dan pembagian.
  - *Program Status Word (PSW)* terdiri dari beberapa *bit* status yang menggambarkan kejadian di akumulator sebelumnya. Yaitu *carry bit*, *auxiliary carry*, dua *bit* pemilih bank, bendera *overflow*, *parity bit*, dan dua bendera yang dapat didefinisikan sendiri oleh pemakai.
  - *Stack pointer (SP)* merupakan *register 8 bit* yang dapat diletakkan di alamat manapun pada RAM internal. Isi *register* ini ditambah sebelum data disimpan, selama instruksi *PUSH* dan *CALL*. Pada saat *reset*, *register SP* diinisialisasi pada alamat  $07_{11}$ , sehingga *stack* akan dimulai pada lokasi  $08_{11}$ .
  - *Data pointer (DPTR)* terdiri dari dua *register*, yaitu untuk *byte* tinggi (*Data pointer high*, *DPH*) dan *byte* rendah (*Data pointer Low*, *DPL*) yang berfungsi untuk mengunci alamat 16 *bit*.
  - *Port 0* sampai *port 3* merupakan *register* yang berfungsi untuk membaca dan mengeluarkan data pada *port 0*, 1, 2, 3. Masing-masing *register* ini dapat dialamati per-*byte* maupun per-*bit*.
-

- *Serial data buffer* (SBUF) merupakan dua *register* yang terpisah, *register buffer* pengirim dan sebuah *register buffer* penerima. Meletakkan data pada SBUF berarti meletakkan pada *buffer* pengirim yang akan mengirimkan data melalui transmisi serial. Membaca data SBUF berarti menerima data dari *buffer* penerima.
- *Control register* terdiri dari *register* yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua register khusus, yaitu register IP (*interrupt priority*) dan register IE (*Interrupt Enable*). Untuk mengontrol pelayanan *timer/counter* terdapat *register* khusus, yaitu *register* TCON (*timer/counter control*) serta untuk pelayanan *port* serial menggunakan *register* SCON (*serial port control*)

#### c. Port Masukan dan Keluaran

Mikrokontroler AT89S52 mempunyai 4 *port* dan masing-masing *port* terdiri dari 8 saluran *bit*. Ke empat *port* ini bersifat *bidirectional* yaitu dapat digunakan sebagai masukan atau keluaran.

*Port* 0 digunakan sebagai saluran data yang dimultipleks dengan saluran alamat rendah untuk mengakses memori eksternal, baik memori program maupun memori data. *Port* 2 mengeluarkan bagian alamat tinggi untuk mode pengalamatan memori 16 *bit*. *Port* 1 dan 3 berfungsi sebagai saluran masukan dan keluaran multi

---

fungsi. Jika dibutuhkan *port 3* mempunyai fungsi khusus seperti ditunjukkan pada Tabel 2-3.

**Tabel 2-3 Fungsi khusus *port 3***

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

Sumber : Intel, 1994:3-7

#### d. Sistem Interupsi

Mikrokontroler AT89S52 mempunyai dua sumber interupsi eksternal dan sumber interupsi internal yang dapat diprogram agar sensitif terhadap perubahan level atau transisi. Interupsi *timer* aktif saat *register timer* yang bersangkutan mengalami *rollover*, interupsi serial akan aktif pada saat mikrokontroler mengirim/menerima data. Setiap sumber interupsi dapat diaktifkan/dimatikan melalui perangkat lunak.

Tabel 2-4 Tingkatan Prioritas Interupsi

Prioritas Interupsi	Sumber Interupsi	Alamat Vektor
1	IE0 (Interupsi eksternal 0)	0003 <sub>16</sub>
2	TF0 ( <i>timer overflow flag 0</i> )	000B <sub>16</sub>
3	IE1 (Interupsi eksternal 1)	0013 <sub>16</sub>
4	TF1 ( <i>timer overflow flag 1</i> )	001B <sub>16</sub>
5	RI dan T1	0023 <sub>16</sub>
6	TF2 dan EXF2	002B <sub>16</sub>

Sumber : Intel, 1994:3-25

Hirarki tingkatan prioritas interupsi dapat dilihat dalam Tabel 2-4. Interupsi yang mempunyai tingkatan prioritas lebih tinggi tidak dapat diinterupsi oleh yang lebih rendah. Meskipun demikian melalui perangkat lunak hirarki tersebut dapat diubah, yaitu dalam *register interrupt priority (IP)*.

### 2.1.2 Memori

Memori dalam suatu sistem mikrokontroler merupakan suatu piranti yang berfungsi untuk menyimpan program dan data yang dibutuhkan oleh mikrokontroler. Memori secara garis besar dibagi menjadi dua macam yaitu memori yang hanya dapat dibaca (*Read only memory*) dan memori yang dapat dibaca maupun ditulisi (*Random Access memory*).



a. Read Only Memory (ROM)

ROM adalah suatu bentuk memori yang hanya dapat dibaca isinya. Isi ROM tidak mudah dihapus atau tidak mudah hilang meskipun catu daya tidak diberikan padanya. Karena sifatnya yang tidak mudah dihapus tersebut ROM disebut juga memori *non volatile* (Tidak mudah menguap). Suatu program atau data statis yang diinginkan agar tidak mudah hilang dapat disimpan dalam ROM. Menurut sifatnya ROM dapat dibagi menjadi beberapa macam, yaitu :

- PROM (*Programmable Read Only Memory*) yaitu jenis ROM yang sekali ditulisi dan tidak dapat dihapus kembali.
- EPROM (*Erasable Programmable Read Only Memory*) yaitu jenis ROM yang dapat ditulisi maupun dihapus kembali. Menurut cara penghapusannya EPROM dapat dibagi menjadi dua yaitu UV-EPROM (*Ultra Violet EPROM*) dan EEPROM (*Electrically EPROM*).

b. RAM (*Random Access Memory*)

RAM adalah memori yang dapat dibaca maupun ditulisi. Menurut sifatnya RAM biasa disebut sebagai memori yang mudah menguap (*volatile*), yaitu bila catu daya yang diberikan pada RAM dihilangkan, maka data pada RAM akan hilang. Ada dua macam RAM yaitu :

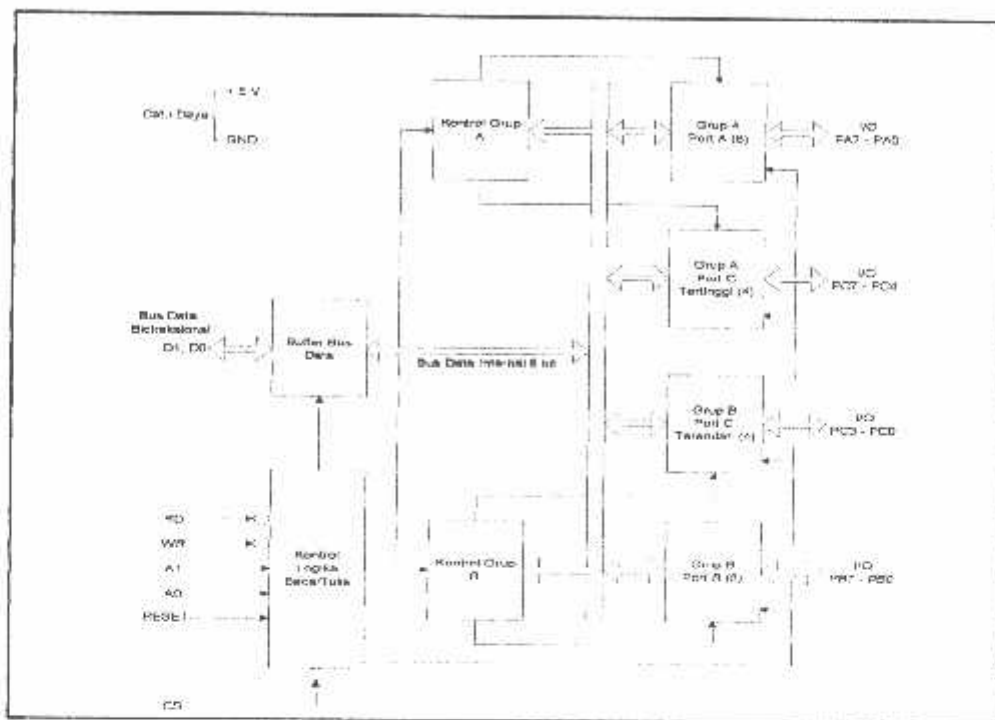
- RAM statik yaitu RAM yang tersusun atas flip-flop. Selama catu daya diberikan pada RAM, maka data akan tetap tersimpan.
-

- RAM dinamik yaitu RAM yang menggunakan kapasitor sebagai penyimpan data. RAM ini memerlukan penyegaran data karena sifat kapasitor dapat menurun muatannya.

### 2.1.3. PPI 8255

PPI 8255 merupakan suatu *chip* yang dirancang untuk keperluan antarmuka dalam suatu sistem mikroprosesor. Chip ini memiliki 24 penyemat I/O yang masing-masing dapat diprogram dalam dua grup yaitu grup A dan grup B. Grup A terdiri dari *port* A ditambah dengan *port C upper* ( $C_4$ - $C_7$ ) sedangkan grup B terdiri dari *port* dan *port C lower* ( $C_0$ - $C_3$ ). Blok diagram PPI 8255 ditunjukkan dalam Gambar 2-2.

---



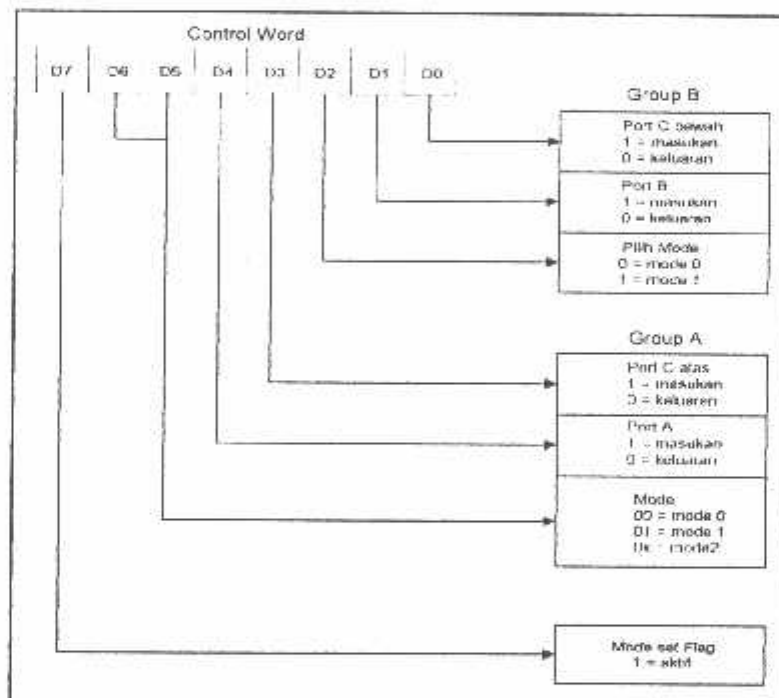
Gambar 2-2. Blok Diagram PPI 8255

Sumber : Intel Developer CD

Selain itu, pada *chip* ini terdapat tiga *port* yang dapat diprogram sebagai jalur keluaran atau masukan, yaitu :

- *Port* A, terdiri dari PA<sub>0</sub>-PA<sub>7</sub>
- *Port* B, terdiri dari PB<sub>0</sub>-PB<sub>7</sub>
- *Port* C, terdiri dari PC<sub>0</sub>-PC<sub>3</sub> (*lower*) dan PC<sub>4</sub>-PC<sub>7</sub> (*upper*)

Konfigurasi fungsi setiap *port* pada PPI ditentukan oleh *control word* yang diberikan ke *bus* data PPI pada awal program sebelum PPI digunakan. Proses ini biasa disebut inisialisasi PPI dan merupakan syarat mutlak yang harus diberikan kepada PPI. *Control word* ini mengandung berbagai informasi yang berhubungan dengan pemanfaatan PPI, seperti mode operasi, fungsi masing-masing *port* dan aktivitas PPI. *Control word* ini akan disimpan pada *control register*. Format masing-masing *bit* dari *control word* ditunjukkan dalam Gambar 2-3.



Gambar 2-3 Control Word PPI 8255

Sumber : Intel Developer CD

Sebelum memakai peralatan PIO mikrokontroler harus menginisialisasinya dengan melaksanakan dua operasi dasar yaitu :

- Penentuan modus.
- Penunjukan arah saluran.

Untuk menentukan modus atau mode operasi PPI 8255 dilaksanakan dengan mengatur struktur *bit* pada *Control word* yang dikirimkan melalui program. *Control word* tersebut disimpan dalam *register* kontrol dengan alamat  $A_0=1$  dan  $A_1=1$ . Bagian internal logic akan mengatur transfer data dan mengontrol informasi pada *bus* data internal. Mode kontrol dikirimkan ke dua buah *port* kontrol yaitu grup A dan grup B. Grup A mengatur penentuan mode dari *port* A dan 4 *bit* tertinggi dari *port* C. Grup B mengontrol *port* B dan 4 *bit* terendah dari *port* C. Pada Tabel 2-5 ditunjukkan operasi dasar PPI 8255.

**Tabel 2-5 Operasi Dasar PPI 8255**

$A_0$	$A_1$	Operasi I/O
0	0	<i>Bus Data—Port A</i>
0	1	<i>Bus Data—Port B</i>
1	0	<i>Bus Data—Port B</i>
1	1	<i>Bus Data—Kontrol</i>

Sumber : Intel Developer CD

PPI 8255 memiliki tiga mode operasi yang memiliki fungsi berbeda. Pemilihan mode ini dilakukan secara *software* dengan melakukan pengaturan *control word*.

a. Mode Operasi 0 (*Basic input/output*)

Konfigurasi ini digunakan untuk melakukan operasi masukan dan keluaran yang sederhana pada ketiga *port*. Pada mode ini tidak dibutuhkan proses jabat tangan (*handshaking*). Fungsi dasar dari mode 0 ini adalah :

- Dua *8-bit port* dan dua *4-bit port*.
- Masing-masing *port* dapat berfungsi sebagai masukan atau keluaran.
- Keluaran terkunci (*output latched*).
- Masukan tidak terkunci.

a. Mode Operasi 1 (*Strobed Input Output*)

Mode ini digunakan untuk mentransfer data pada *port* dengan menggunakan sinyal jabat tangan (*handshaking*). Dalam mode ini, port A dan port B memanfaatkan jalur-jalur pada *port C* untuk membangkitkan sinyal jabat tangan. Fungsi dasar dari mode 1 adalah :

- Dua grup (grup A dan grup B).
  - Masing-masing grup terdiri dari *8-bit* jalur data dan *4-bit* jalur kontrol.
  - Jalur data *8-bit* dapat digunakan sebagai masukan atau keluaran dan keduanya terkunci.
-

- Jalur kontrol 4-bit digunakan untuk mengendalikan dan mengetahui status dari jalur data 8-bit.

c. Mode Operasi 2 (*Strbed Bidirectional Bus I/O*)

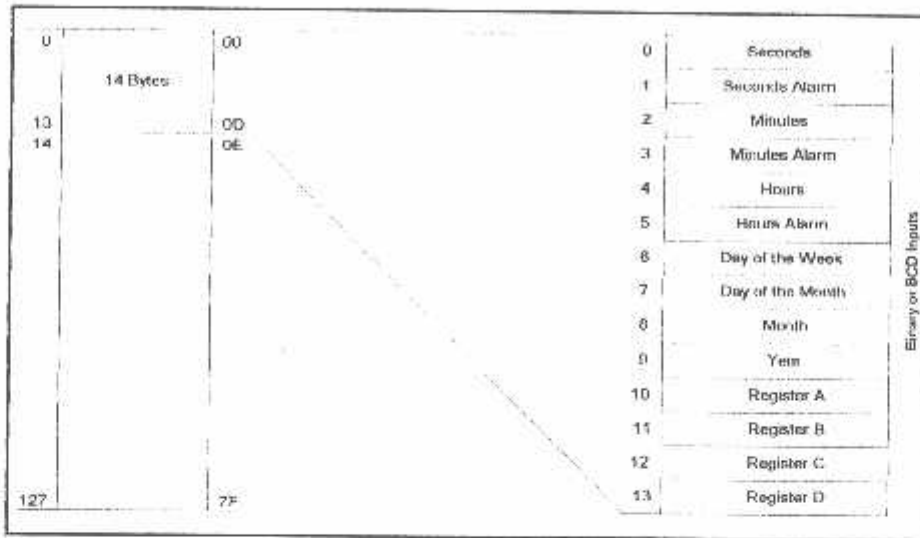
Dengan konfigurasi ini, sebuah jalur data 8-bit dapat digunakan untuk mengirim atau menerima data (*bidirectional*). Fungsi dasar mode operasi 2 ini adalah:

- Hanya digunakan pada grup A
- Terdapat sebuah 8-bit, *bidirectional bus* (*port A*) dan 5-bit jalur kontrol (*port C*).
- Masukan dan keluaran terkunci (*input and output latched*).
- 5-bit jalur kontrol pada port C digunakan untuk mengendalikan dan mengetahui status dari 8-bit *bidirectional bus* (*port A*).

## 2.2. Real Time Clock DS12887A

RTC 12887A buatan *Dallas Semiconductor* ini selain menyediakan data-data mengenai waktu yang ditampilkan dalam biner maupun BCD juga menyediakan RAM internal sebesar 128 bytes. RAM internal terdiri atas 14 bytes yang digunakan untuk *clock* dan *register* kontrol, sedangkan 114 bytes lainnya dapat digunakan oleh pemakai. Pembagian alamat RAM internal RTC DS12887A ditunjukkan dalam Gambar 2-4.

---



**Gambar 2-4** Pembagian Alamat RAM Internal RTC DS 12887A

Sumber : *Dallas Semiconductor, 1999:5-17*

RTC 12887A dapat beroperasi tanpa adanya catu daya hingga 10 tahun. Penyemat-penyemat RTC 12887A kompatibel dengan RTC tipe 146818B dan DS1287A [Dallas Semiconductor, 1999:1-17]. Fungsi penyemat-penyemat RTC DS12887A ditunjukkan pada Tabel 2-6.



Tabel 2-6 Fungsi Penyemat RTC DS12887A

Penyemat	Fungsi
V <sub>CC</sub> dan GND	Merupakan penyemat catu daya, V <sub>CC</sub> dihubungkan pada catu daya positif dan GND pada <i>ground</i> . Tegangan catu daya adalah 5 V.
MOT (Motel)	Memilih mode diagram pewaktuan. Apabila dihubungkan pada V <sub>CC</sub> berarti diagram pewaktuan Motorola yang dipakai, jika dihubungkan dengan <i>ground</i> berarti sistem pewaktuan lain yang dipakai (Intel).
SQW ( <i>Square wave output</i> )	Mengeluarkan sebuah sinyal dari 15 periode yang ada. Besar frekuensi SQW dapat diubah dengan diprogram pada register A. Untuk mengaktifkan atau menonaktifkan sinyal SQW dipilih lewat <i>bit</i> SQW pada register B.
AD <sub>0</sub> -AD <sub>7</sub> ( <i>Address data 0-7</i> )	<i>Bus</i> data dan <i>bus</i> alamat yang masih termultiplexs. Pengiriman data maupun alamat dilakukan melalui <i>bus</i> ini.
AS ( <i>Address strobe</i> )	Untuk memisahkan <i>bus</i> data dan <i>bus</i> alamat (ALE). Tepi turun dari ALE akan menyebabkan alamat ditahan 12887A secara internal.
DS ( <i>Data strobe</i> )	Berfungsi sama dengan sinyal OE ( <i>output enable</i> ) pada komponen memori. Sinyal DS dihubungkan dengan sinyal RD yang berasal dari Mikrokontroler untuk melakukan proses membaca data pada RAM internal RTC.
CS ( <i>Chip select</i> )	Mengaktifkan piranti RTC. Sinyal CS didapat dari dekoder alamat dengan alamat tertentu.
IRQ ( <i>Interrupt request</i> )	Untuk menginterupsi mikrokontroler. Untuk mereset IRQ, mikrokontroler memberikan program pada <i>register</i> C RTC. Saat tidak terdapat interupsi, penyemat ini dalam kondisi impedansi tinggi.
RESET	Untuk mereset beberapa <i>flag</i> menjadi nol tetapi tidak berpengaruh pada unjuk kerja <i>clock</i> , kalender dan fungsi RAM.
R/W ( <i>Read/write</i> )	Untuk membaca dan menulis data pada RTC.
RCLR ( <i>RAM clear</i> )	Menghapus data-data pada RAM dengan cara menghubungkan penyemat /RCLR dengan <i>ground</i> .

Sumber : Dallas Semiconductor, 1999:4-17

### 2.3. Penampil Kristal Cair

Penampil kristal cair ini terdiri atas tumpukan tipis atau sel dari dua lembar kaca yang sampingnya tertutup rapat. Antar dua lembar kaca tersebut diberi bahan kristal cair (*liquid crystal*) yang *tembus* cahaya. Permukaan luar dari masing-masing keping kaca mempunyai lapisan penghantar *tembus* cahaya seperti oksida timah (*tin oxide*) atau oksida indium (*indium oxide*). Sel mempunyai ketebalan sekitar  $1 \times 10^{-5}$  meter dan diisi dengan kristal cair.

Penampil kristal cair yang digunakan pada tugas akhir ini berupa dot matriks  $5 \times 7$ , tersusun sebanyak dua baris dan masing-masing baris terdiri atas 16 karakter. Fungsi penyemat-penyemat penampil kristal cair tipe TM162 ditunjukkan pada Tabel 2-7.

Tabel 2-7 Fungsi Penyemat Penampil Kristal Cair TM162

Penyemat	Jumlah Terminal	I/O	Tujuan	Fungsi
DB <sub>0</sub> -DB <sub>3</sub>	4	I/O	MCU	Saluran I/O 4 bit ( <i>nibble</i> ) bawah sebagai bus data dua arah tiga keadaan, berisi perintah dan data yang akan ditampilkan di penampil kristal cair.
DB <sub>4</sub> -DB <sub>7</sub>	4	I/O	MCU	Saluran I/O 4 bit ( <i>nibble</i> ) atas sebagai bus data dua arah tiga keadaan, berisi perintah dan data yang akan ditampilkan di penampil kristal cair.
E	1	Input	MCU	Merupakan tanda mulai operasi. Bila sinyal ini aktif, berarti MCU dalam keadaan menulis atau membaca data.
R/W	1	Input	MCU	Sinyal seleksi baca dan tulis. 0 = tulis 1 = baca
RS	1	Input	MCU	Sinyal pemilih register. 0 = instruksi register (tulis) 1 = data register (baca dan tulis)
V <sub>LC</sub>	1	-	Power Supply	Merupakan terminal <i>power supply</i> untuk mengendalikan tampilan LCD
V <sub>DD</sub>	1	-	Power Supply	Power supply +5V
V <sub>SS</sub>	1	-	Power Supply	Terminal pertanahan

Sumber : Seiko Instrument, Inc. 1987:7

#### 2.4. EPROM

Kerugian utama dari EPROM adalah tidak bisa diprogram ulang. Dengan demikian, kesalahan dalam program tidak bisa dibetulkan. EPROM (*Erasable Programmable Read-Only Memory*) mengatasi persoalan ini. Pengguna dapat menghapus data yang telah tersimpan dan mengubah programnya. Program yang telah tersimpan dapat dihapus dengan menyinari sel memori dengan sinar ultra ungu lewat “jendela” yang ada pada IC. Proses ini akan memakan waktu kira-kira 20 – 30 menit. Setelah proses ini IC akan berada dalam keadaan kosong kembali, dan siap untuk diprogram ulang. EPROM mempunyai dua kerugian :

1. Keseluruhan isi memori harus dihapus sebelum diprogram ulang. Hal ini berarti penghapusan sebagian isi memori tidak dimungkinkan.
2. IC harus diambil dari rangkaian apabila akan disinari. Hal ini berarti penghapusan program tidak bisa dilakukan di tempat.

#### 2.5. EEPROM

EEPROM (*Electrical Erasable Programmable Read-Only Memory*), yang juga sering disebut EAPROM (*Electrical Alterable Programmable Read-Only Memory*) mengatasi persoalan yang timbul pada EPROM. Isi EEPROM dapat dihapus dan diprogram ulang di tempat dengan menggunakan isyarat listrik yang sesuai. Lebih lanjut, lokasi-lokasi tertentu dari EEPROM dapat dihapus dan diprogram ulang tanpa mengganggu yang lain. Dalam skripsi ini yang dipakai adalah memori jenis EEPROM.

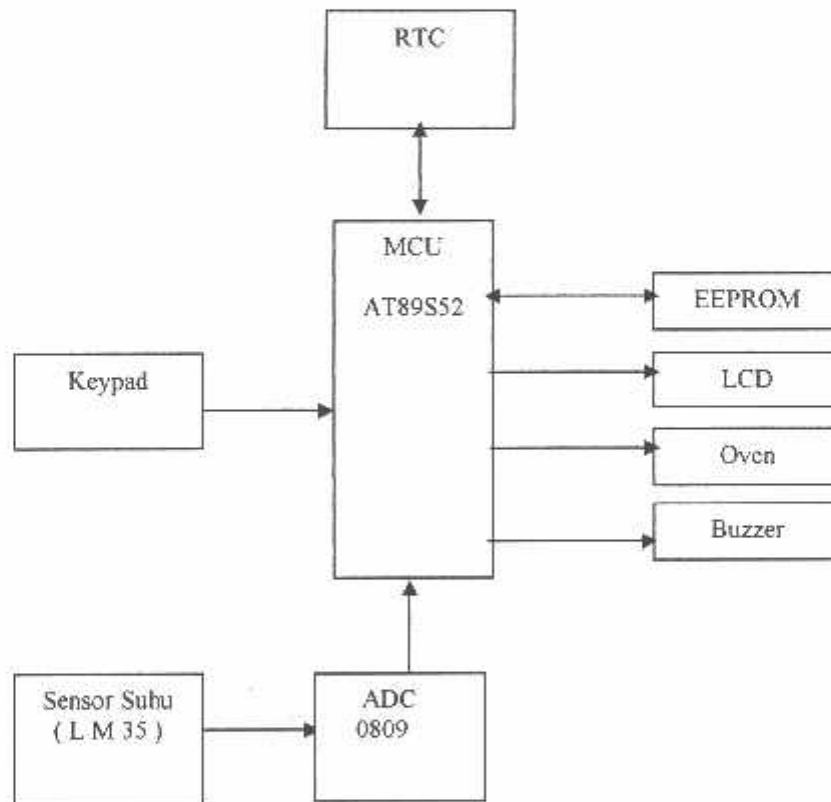
---

### BAB III

#### PERENCANAAN DAN REALISASI ALAT

Alat yang direncanakan ini berfungsi untuk mengatur suhu dan lama oven listrik dengan bantuan mikrokontroller Atmel dan antarmuka yang sengaja dirancang untuk keperluan tersebut.

Bab ini membahas tentang perencanaan dan realisasi perangkat keras dan perangkat lunak yang digunakan dalam sistem ini. Diagram skematiknya ditunjukkan dalam Gambar 3-1.



Gambar 3-1 Blok Diagram

### 3.1. Perangkat Keras

Perangkat keras sistem ini terdiri beberapa bagian , yaitu :

1. Sistem mikrokontroler 89S52.
2. Papan tombol (*keypad*).
3. Sensor
4. Rangkaian penguat tegangan, rangkaian yang menguatkan tegangan dari sensor menjadi tegangan 0V sampai 5V agar dapat dibaca oleh ADC.
5. Rangkaian pengubah data analog ke data digital dengan ADC 0809.
6. *Liquid Crystal Display*, LCD.

#### 3.1.1. Sistem Mikrokontroler

Sistem mikrokontroler terdiri atas mikrokontroler dan komponen-komponen pendukung agar sistem dapat bekerja dengan optimal. Komponen pendukung pada perancangan sistem ini adalah EEPROM, PPI 8255, latch dan dekoder.

EEPROM merupakan tempat menyimpan data yang sedang diproses. Latch berfungsi untuk memisahkan bit-bit data dan bit-bit alamat yang termultipleks. Sedangkan dekoder untuk membagi peta alamat komponen-komponen yang digunakan.

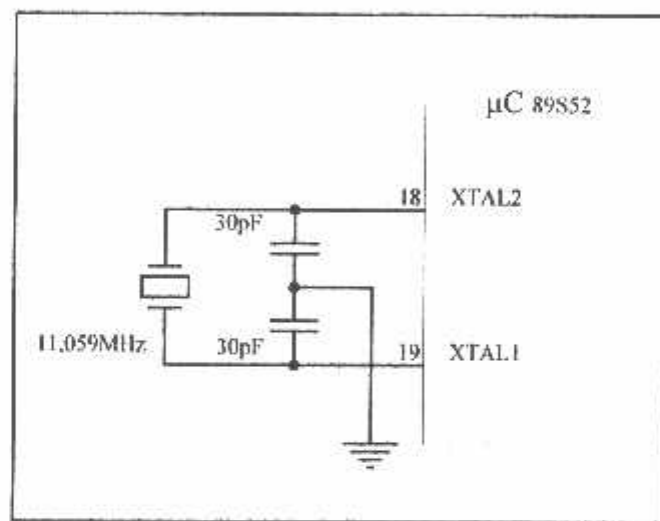
##### 3.1.1.1 Mikrokontroler AT89S52

Mikrokontroler AT89S52 harus didukung oleh beberapa rangkaian lain agar dapat melakukan prosesnya, yaitu berupa rangkaian clock dan reset. Selain

---



89S52 yaitu 30 pF. Pemilihan besar frekuensi kristal disesuaikan dengan pemilihan kecepatan yang diharapkan untuk transfer data melalui pin *serial interface* 89S52 tersebut. Gambar 3-3. memperlihatkan rangkaian pewaktu yang digunakan.



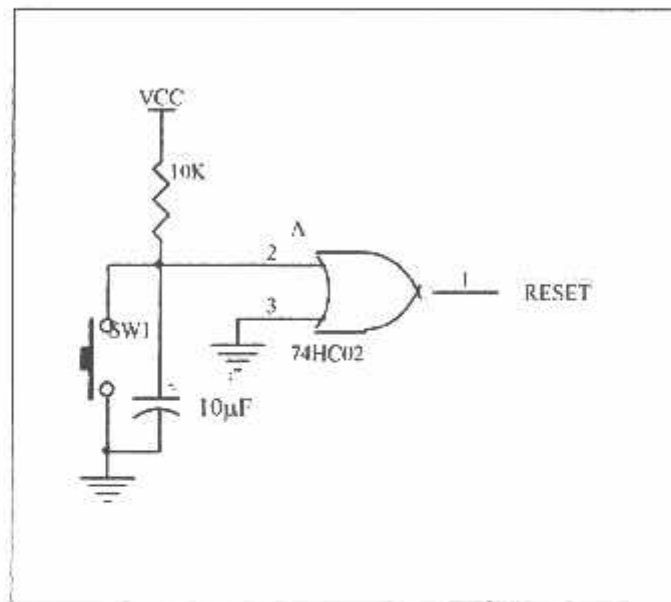
**Gambar 3-3 Rangkaian Peawaktuan**

- **Reset**

Untuk mereset mikrokontroler, pin RST harus diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset pada saat awal catu daya dihidupkan, suatu reset otomatis dapat dilakukan dengan menghubungkan pin RST ke rangkaian *Power-On Reset*, seperti dalam Gambar 3-4. Ketika catu daya dinyalakan, rangkaian akan menahan pin RST dalam kondisi logika tinggi selama selang beberapa saat tergantung nilai kapasitor dan kecepatan pengisian muatannya. Pemberian catu daya pada mikrokontroler



tanpa suatu sinyal reset dapat menyebabkan CPU memulai eksekusi instruksinya dari lokasi yang tak tertentu. Ini disebabkan karena *Program Counter* tidak terinisialisasi.



**Gambar 3-4 Rangkaian Power-On Reset**

Rangkaian Reset terbentuk oleh komponen R dan C yang sudah baku (ditetapkan oleh perusahaan pembuat IC AT 89S52). Nilai R yang dipakai adalah 10 K $\Omega$  dan C =10  $\mu$ F.

Sedangkan untuk mencari frekuensi dari reset tersebut dengan menggunakan rumus sebagai berikut :

$$F_0 = \frac{1}{1,1 \times R \times C}$$

Sehingga dengan komponen resistor dengan nilai 10 Kohm serta kapasitor dengan nilai 10 uF akan dihasilkan frekuensi.

$$f_0 = \frac{1}{1,1 \times R \times C}$$

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

$$f_0 = 9,09 \text{ Hz}$$

Maka Periode Clock :  $T = \frac{1}{f_0}$

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

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

- **Pembagian Port**

Mikrokontroler 89S52 mempunyai 4 buah port yaitu port 0 (P0), port 1 (P1), port 2 (P2) dan port 3 (P3). Mikrokontroler 89S52 tidak memiliki EPROM internal, maka port 0 digunakan untuk saluran data dan alamat bagian rendah yang dimultiplek (AD0-AD7) dan alamat bagian tinggi dikeluarkan pada port 2 (A8-A15). Karena menggunakan komponen EEPROM, PPI dan komunikasi serial pada sistem mikrokontrolernya maka beberapa bit port 3 digunakan sebagai pembangkit sinyal kontrol (Tx, Rx, RD dan WR). Dalam perancangan port 1 digunakan sebagai masukan dari *keypad*.

- **Sinyal kontrol Memori Eksternal**

Untuk mengakses memori eksternal yang digunakan pada perancangan alat ini, maka harus digunakan sinyal kontrol RD, WR, ALE, PSEN dan EA. Sinyal-sinyal tersebut digunakan untuk mengontrol proses pemindahan data mikrokontroler. Untuk mengakses memori data eksternal diperlukan sinyal RD dan WR yang dihubungkan dengan EEPROM. Sinyal WR mengendalikan proses

pembacaan data pada EEPROM, sedangkan sinyal RD mengendalikan proses penulisan pada EEPROM.

Untuk mengakses memori program diperlukan sinyal PSEN yang digunakan untuk mengeksekusi program yang tersimpan pada EPROM. Selain itu harus diperhatikan juga sinyal EA dan ALE. EA harus dihubungkan dengan VSS (logika 0) yang berarti bahwa memori eksternal diakses mulai alamat 0000H sampai FFFFH, atau dengan kata lain pengaksesan seluruh alamat dilakukan di luar chip 89S52. Sinyal ALE berfungsi untuk memisahkan bit-bit data dan bit-bit alamat rendah yang dimultiplek pada port 0.

#### **3.1.1.2. Dekoder dan Peta Alamat**

Proses aliran data yang diperlukan oleh mikrokontroler tergantung dari pengalamatan yang dipilih mikrokontroler tersebut. Untuk memilih komponen mana yang diakses maka diperlukan suatu rangkaian dekoder. Dekoder ini akan dihubungkan dengan alamat tinggi yang dikeluarkan mikrokontroler. Sistem ini mempergunakan pengalamatan 16 bit untuk mengamati EEPROM dan PPI.

Peta alamat pada sistem ini direncanakan dibagi per-delapan kilo byte, yang berarti bahwa alamat mikrokontroler yang tersedia sebesar 64 kilo byte dibagi menjadi 8 blok. Masing-masing blok dipilih dengan menggunakan 3 bit alamat tinggi yang dikeluarkan melalui port 2. Tabel 3-1, memperlihatkan pemetaan alamat pada sistem ini.

**Tabel 3-1 Pemetaan Alamat**

Chip Select	Alamat	Fungsi
CS0	0000-1FFF	RTC
CS1	2000-3FFF	-
CS2	4000-5FFF	-
CS3	6000-7FFF	-
CS4	8000-9FFF	
CS5	A000-BFFF	LCD
CS6	C000-DFFF	EEPROM
CS7	E000-FFFF	PPI 8255

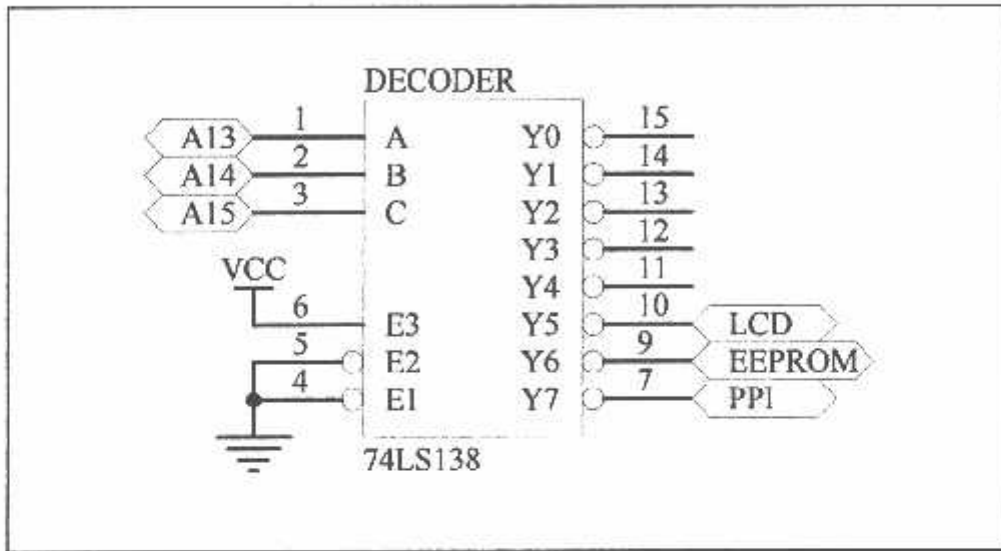
Pengalamatan EPROM diletakkan mulai alamat 0000H, karena setiap kali setelah reset mikrokontroler akan menuju alamat tersebut untuk menjalankan program yang telah diisikan. EEPROM diletakkan pada alamat C000H-DFFFH. Blok-blok yang masih kosong dapat dimanfaatkan untuk penambahan memori atau untuk keperluan lainnya. Pemilihan alamat-alamat ini disesuaikan dengan keperluan dan perencanaan yang dapat dilakukan untuk pengembangan berikutnya.

Dekoder yang dipergunakan adalah dekode 74LS138, suatu dekode 3 ke 8. Tabel fungsinya seperti terlihat dalam Tabel 3-2.

Tabel 3-2 Tabel Dekoder 74LS138

INPUT					OUTPUT							
ENABLE		SELECT										
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	H	H	H	H	H	L	H	H	H	H	H
H	L	H	H	H	H	H	H	L	H	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Dekoder ini dikondisikan selalu membaca data masukannya setiap saat, sehingga pin G1 diberi logika tinggi dan pin G2A, G2B dihubungkan dengan pentanahan. Saluran alamat yang digunakan untuk memilih alamat diambil dari mikrokontroler pada bit A13, A14 dan A15, dihubungkan dengan pin A, B dan C pada dekoder. Rangkaian dekoder ditunjukkan dalam Gambar 3-5.



Gambar 3-5 Rangkaian Dekoder Alamat

### 3.1.1.3. Latch

Latch digunakan untuk memisahkan antara bit-bit data dengan bit-bit alamat rendah yang dikeluarkan pada port 0 oleh mikrokontroler 89S52. Tipe yang akan dipakai adalah tipe 74LS573. Latch ini merupakan latch berdelapan tipe D transparan yang berarti bahwa saat enable dan C dalam keadaan logika tinggi maka keluaran Q akan mengikuti masukan data D. Bila enable diberi logika rendah maka keluaran akan dilatch pada level data yang telah diset sebelumnya (Q0). Tabel fungsi 74LS573 dapat dilihat dalam Tabel 3-3.

Tabel 3-3 Tabel Fungsi Latch 74LS573

OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	Q0
H	X	X	Z

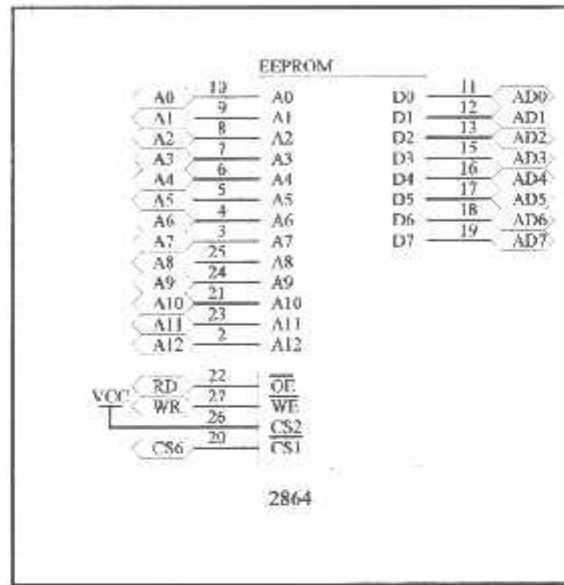
Pin enable C pada latch ini dihubungkan dengan sinyal ALE yang dikeluarkan oleh mikrokontroler. Sedangkan saluran masukan latch dihubungkan dengan port 0 yang mengeluarkan bit-bit data dan alamat rendah.

Sinyal ALE berlogika tinggi maka bit-bit yang dikeluarkan melalui port 0 adalah bit-bit alamat, dan 74LS573 akan meneruskan bit-bit ini ke keluarannya yang terhubung dengan saluran alamat komponen yang dituju. Apabila sinyal ALE dalam keadaan logika rendah maka 74LS573 menahan bit-bit tersebut sehingga bit-bit ini akan dibaca sebagai data oleh komponen yang terhubung.

#### 3.1.1.4. EEPROM

EEPROM yang digunakan untuk menyimpan data adalah jenis 2864 yang berkapasitas 8 kilobyte. Komponen ini tersusun atas 13 jalur alamat A0-A12, 8 jalur data D0-D7, dua saluran seleksi CS1 dan CS2 serta sinyal kontrol

pembacaan dan penulisan OE dan WE. Rangkaian EEPROM pada rangkaian ini ditunjukkan dalam Gambar 3-6.



**Gambar 3-6 Rangkaian EEPROM 2864**

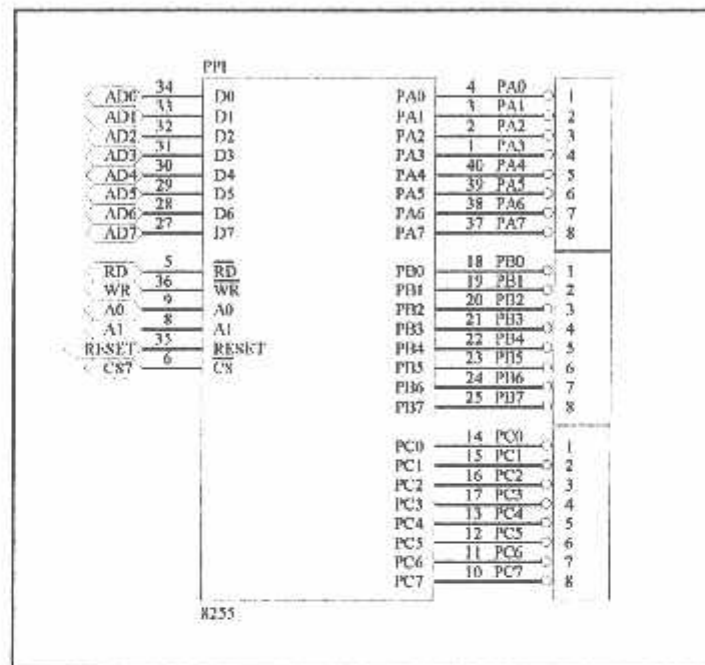
### 3.2. PPI 8255

Mikrokontroler 89S52 memiliki 4 buah port, yaitu port 0, port 1, port 2 dan port 3. Tetapi karena tidak mempunyai EPROM internal, maka untuk mengakses EPROM eksternal, port 0 dan port 2 difungsikan menjadi bus alamat dan bus data. Sementara itu beberapa bit dari port 3 juga memiliki fungsi lain yaitu sinyal RD, WR, TxD, RxD, INT0, INT1, T0 dan T1. Jadi port bebas yang tersisa adalah port 1. Untuk menambah jumlah port digunakanlah PPI 8255.

Dalam perencanaan, PPI 8255 ditempatkan pada alamat E000H-E003H. Pin A0 dan A1 PPI yang dipergunakan untuk memilih register yang diakses dihubungkan ke A0 dan A1 sistem mikrokontroler. Sementara pin CS IC 8255



dihubungkan ke CS7 (keluaran dari *address decoder* 74LS138). Rangkaian PPI 8255 seperti terlihat dalam Gambar 3-7.



**Gambar 3-7 Rangkaian PPI 8255**

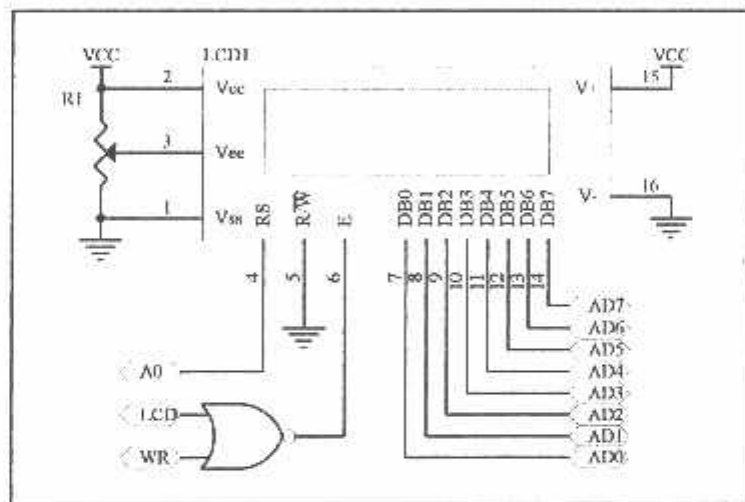
### 3.3. LCD

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

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

yang diakses. LCD TM162ABC memiliki 2 register yaitu register data dan register instruksi.

Dalam sistem ini, LCD menempati ruang alamat A000H-A001H. Pin R/W dihubungkan ke *ground* atau selalu berlogika 0 karena dalam perancangan. LCD ini hanya selalu dalam operasi tulis dan pin RS dihubungkan ke pin A0 sistem mikrokontroler. Pengaktifan LCD ini selanjutnya tergantung pada pin E. Dimana pin E ini tergantung dari CS5 dari address dekoder dan perintah write mikrokontroler. Rangkaian LCD seperti terlihat dalam Gambar 3-8.



Gambar 3-8 Rangkaian LCD

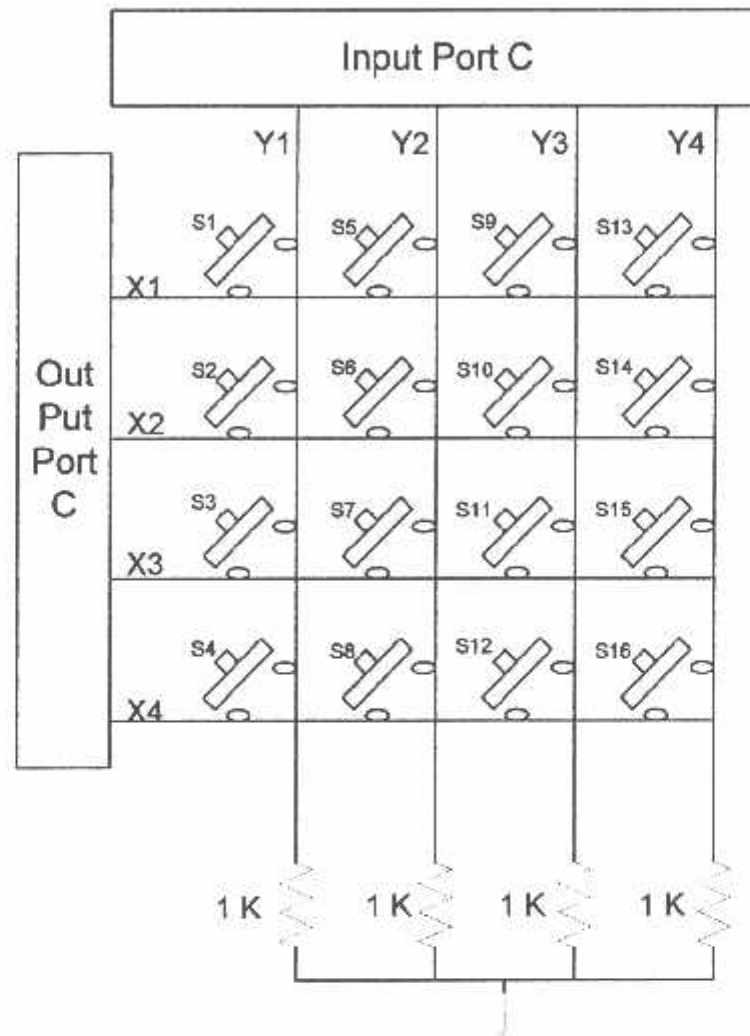
### 3.4. Papan Tombol (*Keypad*)

Papan tombol ini digunakan untuk memasukkan data referensi dan mengubah data bila diinginkan. Untuk menterjemahkan informasi yang diterima dari papan tombol, maka *keypad* dihubungkan dengan *port C* PPI 8255.

Papan tombol tersebut mempunyai matrik 4 baris dan 3 kolom. Deretan baris dan kolom dari papan tombol dihubungkan dengan *port C* PPI 8255 yang difungsikan sebagai masukan dan keluaran. Deretan kolom dihubungkan dengan

*ground* (berlogika 0) dan *port C* (PC4-PC7) yang difungsikan sebagai *input* mikrokontroler. Sedangkan deretan baris dihubungkan ke *port C* (PC0-PC3) yang telah diberi data 0001 dan secara kontinyu data tersebut bergeser satu bit ke kiri. Pergeseran data satu bit ini dimaksudkan untuk menentukan posisi tombol yang ditekan dalam satu kolom. *Port* ini difungsikan sebagai *output* dari mikrokontroler. Dengan demikian kalau tombol tidak ditekan maka masukan *port C* (PC4-PC7) di pin yang terhubung tombol tersebut berlogika 0 dan bila tombol ditekan akan berlogika 1. Rangkaian papan tombol tersebut dapat dilihat dalam Gambar 3-9.

---



Gambar 3-9 Rangkaian Keypad

### 3.5. Rangkaian Sensor

Rangkaian ini berfungsi untuk mengindera suhu ruangan yang akan diatur dan mengubah informasi tersebut menjadi tegangan analog. Untuk mengindera suhu digunakan sensor suhu LM 35 dengan pertimbangan antara lain sederhana rangkaianannya, range pengukurannya lebar, keluarannya linier terhadap suhu,



**Tabel 3-4 Hubungan Antara Suhu Dan Tegangan Keluaran Sensor**

Suhu (°C)	Vout (V)
0	0
1	0,01
⋮	⋮
25	0,25
⋮	⋮
50	0,5

### 3.6. Rangkaian Penguat Sinyal

Spesifikasi keluaran sensor yang diinginkan untuk diumpankan ke ADC 0809 dapat mengukur temperatur dari 0°C sampai 100°C. Dengan jangkauan temperatur tersebut maka sensor akan mengeluarkan tegangan dari 0 V sampai 1000 mV (dianggap bahwa karakter sensor mengikuti spesifikasi sebagaimana yang dijelaskan dalam data sheet).

$$\begin{aligned}
 V_{\text{out LM35}} &= \text{temperatur} \times 10\text{mV} \\
 &= 100 \times 10 \\
 &= 1000 \text{ mV}
 \end{aligned}$$

Keluaran sensor suhu tersebut perlu dikuatkan agar didapatkan ketelitian data yang lebih baik jika dimasukkan ke ADC. Penguatan diambil 5 kali oleh dua buah penguat TL 074. Penguat pertama berfungsi untuk menguatkan tegangan masukan dan penguat kedua berfungsi untuk membalik polaritas tegangan hasil penguatan penguat pertama. Untuk mencari nilai R2 dan R3 dapat dicari berdasarkan rumus:

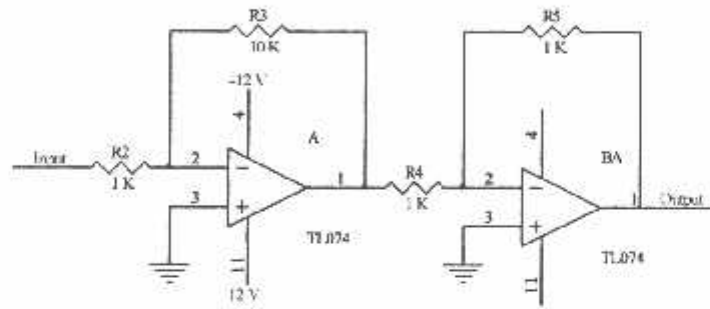
$$V_{out} = -A_v \times V_{in}$$

$$A_v = R_3/R_2$$

$$10 = R_3/R_2$$

Bila ditentukan  $R_3 = 10 \text{ K}\Omega$ , maka  $R_2 = 1 \text{ K}\Omega$ .

Gambar rangkaian penguat dapat dilihat dalam Gambar 3-11.



**Gambar 3-11 Penguat Tegangan**

**Tabel 3-5 Hubungan Antara Tegangan Masukan Dan Keluaran Penguat Sinyal**

$V_{in}$ (V)	$V_{out}$ (V)
0,00	0,00
0,05	0,25
.	.
.	.
0,1	0,5
.	.
.	.
0,25	1,25

### 3.7. Rangkaian Pengubah Data Analog ke Digital (ADC)

Data suhu yang diperoleh dari sensor adalah berupa besaran tegangan analog, maka data suhu tersebut harus diubah ke bentuk data digital 8 bit agar dapat dibaca oleh mikrokontroler. Untuk itu digunakan konverter analog ke digital (ADC). *Type* yang dipakai dalam perancangan ini adalah *type* ADC 0809 yang merupakan ADC dengan 8 masukan analog yang dimultipleks menjadi data digital 8 bit. Dalam perancangan alat ini dipakai satu buah *input* yaitu  $D_0$ . Untuk menyeleksi *input* tersebut maka alamat yang ada pada ADC 0809 dihubungkan pada mikrokontroler 89S52.

Dikarenakan tegangan yang terukur cukup kecil maka tingkat resolusi dari ADC 0809 diharapkan cukup kecil, sehingga digunakan  $V_{ref} = 5$  volt, dengan tingkat resolusi ADC 0809 adalah :

$$\text{Resolusi} = V_{\text{referensi}} / (2^8 - 1)$$

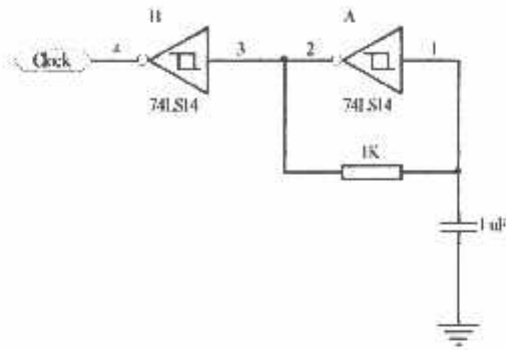
$$\text{Resolusi} = 5 / (2^8 - 1)$$

$$= 0,0196 \text{ V}$$

Jadi besarnya resolusi adalah sebesar  $0,0196 \text{ V} \approx 0,02 \text{ V}$

Untuk membuat ADC 0809 dapat bekerja, maka diperlukan sebuah clock. Pada data book yang ada, tertulis bahwa frekwensi clock pada umumnya adalah 640 KHz. Pada perancangan ini digunakan sumber pulsa dari rangkaian RC dan sebuah *inverting schmitt trigger* IC 74LS14. Rangkaian pembentuk pulsa dapat dilihat dalam Gambar 3-12.





**Gambar 3-12 Pembangkit Clock**

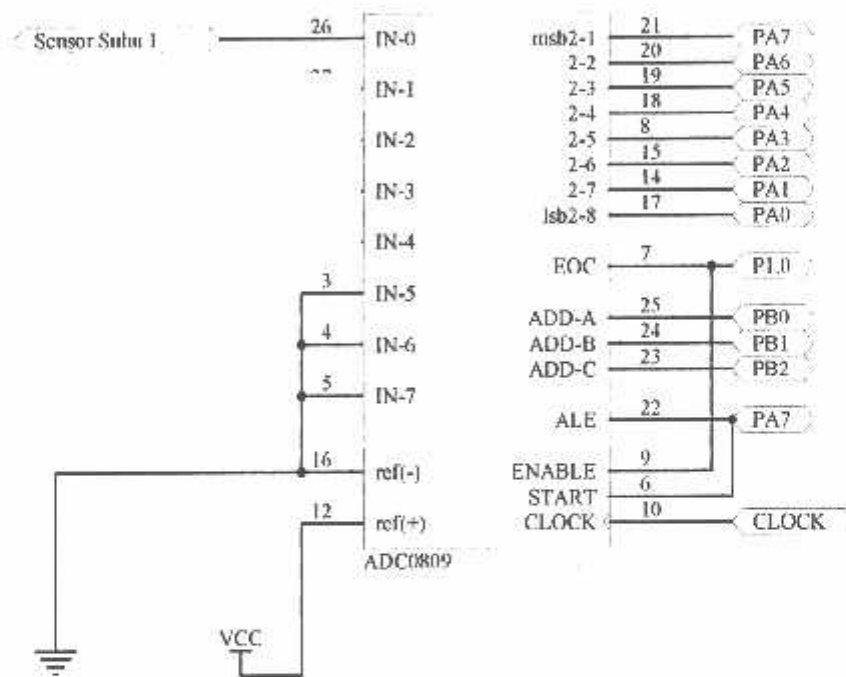
Frekuensi yang dihasilkan oleh rangkaian ini adalah sebesar :

$$\begin{aligned}
 f_{\text{count}} &= \frac{1}{12R.C} \\
 &= 1/(1,2 \times 1000 \times 1,10^{-9}) \\
 &= 833,333,33 \text{ Hz} \\
 &= 840 \text{ KHz}
 \end{aligned}$$

Agar ADC dapat bekerja secara terus menerus tanpa diperintah maka masukan START dan ALE juga diberi masukan clock dengan frekwensi yang lebih rendah, dengan periode sekitar  $100 \mu\text{S}$ . Sebuah multivibrator 74LS123 diberikan pada keluaran *counter* untuk ALE dan START agar ADC dapat langsung terpicu pada saat *counter* memberikan sinyal. Pin-pin kendali yang dihubungkan dengan mikrokontroler ialah *End of Conversion* (EOC), alamat, ALE dan *output enable*. Selain itu data disalurkan melalui kaki  $D_0$ - $D_7$ .

Saluran keluaran data digital ADC 0809 dihubungkan pada *port A* PPI 8255, sedangkan alamat di ADC 0809 yang berfungsi untuk menyeleksi data masukan ADC dihubungkan ke *port B* PPI 8255.

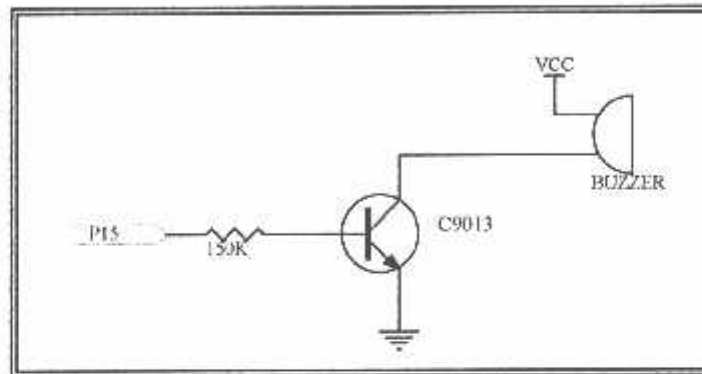
Realisasi dari rangkaian ADC tersebut terlihat dalam Gambar 3-13.



**Gambar 3-13 Rangkaian ADC 0809**

### 3.8. Driver Buzzer

Driver Buzzer diperlukan untuk mengaktifkan buzzer karena mengingat keluaran dari pengolah data yaitu mikrokontroller AT89S52 tidak mampu untuk langsung mengaktifkan buzzer.



Gambar 3-14 Rangkaian Driver Buzzer

Jika diketahui :

$$V_{cc} = 5 \text{ V}$$

$$H_{fe} \text{ C9013} = 100$$

$$R_{buzzer} = 160 \Omega$$

Maka nilai  $R_b$  dapat dicari dengan :

$$\begin{aligned} I_{buzzer} &= \frac{V_{cc}}{R_{buzzer}} \\ &= \frac{5}{160} = 3,125 \text{ mA} \end{aligned}$$

$$I_b = \frac{I_c}{H_{fe}}$$

$$= \frac{3,125}{100} = 0,03125\text{mA}$$

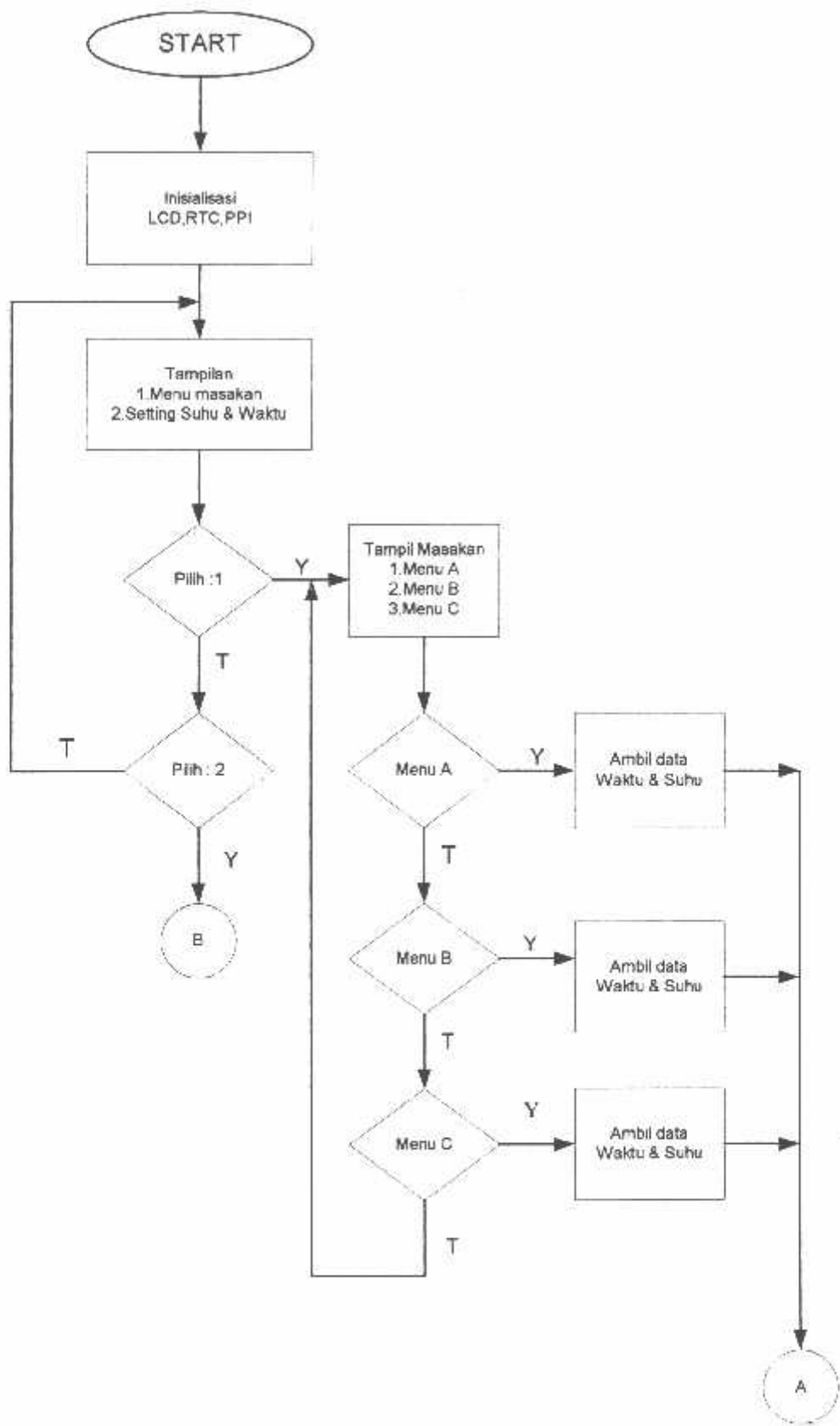
Nilai Rb Dapat dicari dengan :

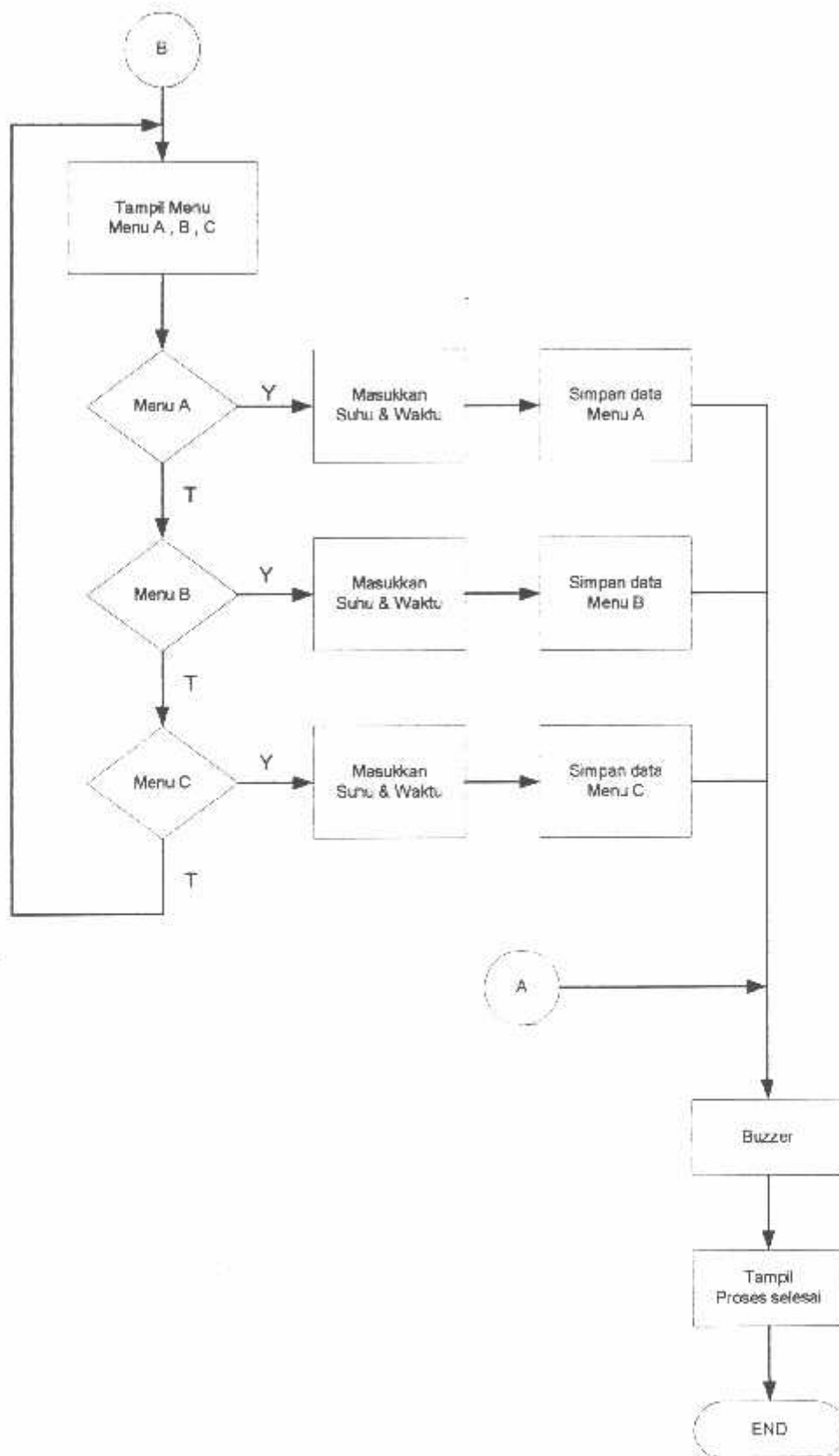
$$\begin{aligned} R_b &= \frac{V_{in} - V_{be}}{I_b} \\ &= \frac{5 - 0,7}{0,03125\text{mA}} \\ &= 137,6\text{K}\Omega \cong 150\text{K} \end{aligned}$$

Jadi besar nilai Rb yang dipasang sebesar = 150 K $\Omega$

### 3.9.Perancangan Perangkat Lunak

Perancangan perangkat lunak ( software ) didasarkan pada perancangan perangkat keras yang telah dibuat sebelumnya, guna mendapatkkan sistem kerja yang diharapkan. Berikut diagram alir dari perancangan perangkat lunak :





## **BAB IV**

### **PENGUJIAN DAN ANALISIS**

#### **4.1. Umum**

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

- ◆ Mengetahui unjuk kerja rangkaian papan tombol.
- ◆ Mengetahui unjuk kerja rangkaian ADC.
- ◆ Mengetahui unjuk kerja rangkaian RTC.
- ◆ Mengetahui unjuk kerja rangkaian LCD.
- ◆ Mengetahui unjuk kerja sistem secara keseluruhan dengan mengintegrasikan perangkat keras dan perangkat lunak.

#### 4.2. Pengujian Papan Tombol (*keypad*)

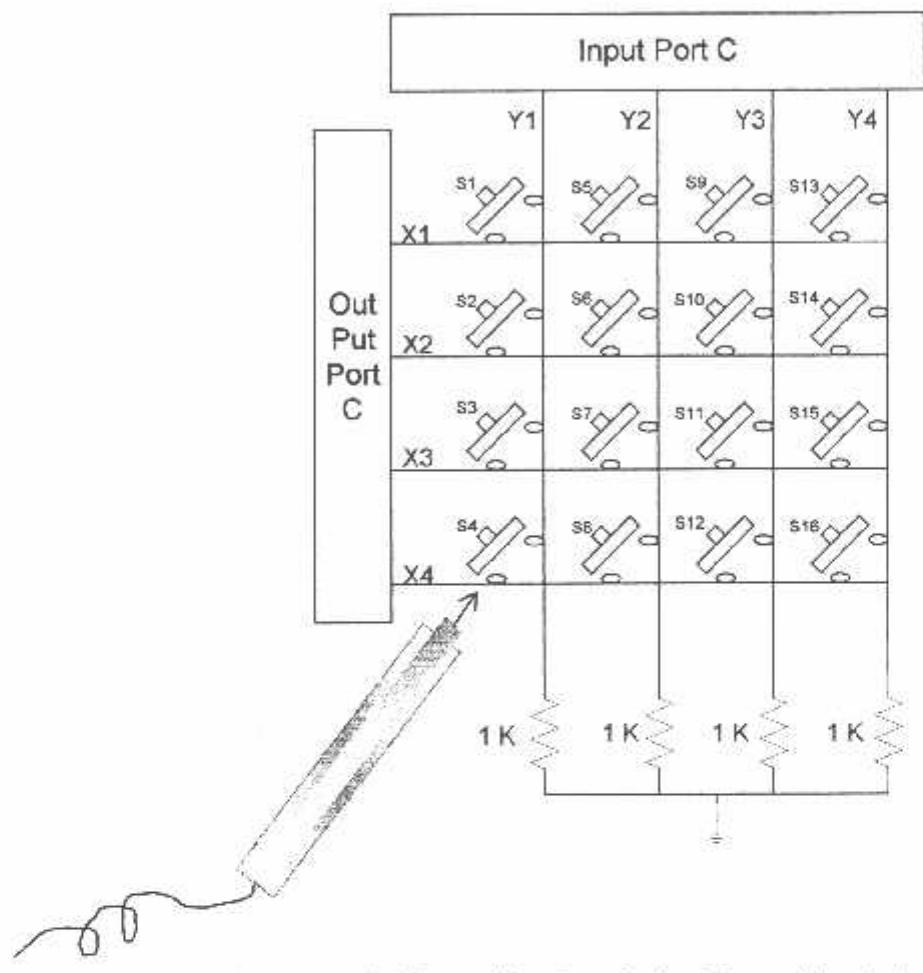
Pengujian ini bertujuan untuk mengetahui konfigurasi logika keluaran dari unit papan tombol saat tombol ditekan. Peralatan yang dipergunakan antara lain catu daya, unit papan tombol, dan *logic probe*. Dalam pengujian ini keluaran yang diamati adalah proses *scanning* yang terjadi pada lajur baris dan kolom. Lajur baris merupakan bagian output sedangkan lajur kolom merupakan bagian input. Untuk mengetahui kebenaran rangkaian keypad yang telah dibuat maka keluaran dari rangkaian keypad ini akan ditampilkan ke port 1 MCU 89S52.

Langkah-langkah pengujian papan tombol adalah sebagai berikut ;

- Menyiapkan catu daya dengan keluaran 5 V DC, rangkaian papan tombol dan *logic probe*.
- Menyusun rangkaian pengujian papan tombol seperti terlihat dalam Gambar 4-1 serta memastikan bahwa hubungan antar pin pada masing-masing saklar telah benar.
- Jalankan program pengujian keypad, amati keluaran pin Port C bagian output. Langkah ini dilakukan dalam keadaan tidak ada penekanan tombol sama sekali.
- Tekan sembarang tombol kemudian amati keluaran pin Port C bagian output.

Hasil dari pengujian rangkaian papan tombol terdapat dalam Tabel 4-1





Gambar 4-1 Pengujian Rangkaian Papan Tombol

**Tabel 4-1 Hasil Pengujian Papan Tombol**

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

Dari hasil pengujian, didapatkan data seperti dalam Tabel 4-1 maka dapat diketahui bahwa saat tombol ditekan maka keluaran port 1 mikrokontroler 89S52 akan berlogika sesuai dengan tombol yang ditekan. Hasil pengujian dalam Tabel 4-1 terlihat bahwa rangkaian papan tombol yang telah direalisasikan sesuai dengan unjuk kerja perencanaan.

### 4.3. Pengujian Penguat Sinyal

Tujuan pengujian penguat sinyal adalah untuk mengetahui tanggapan keluaran penguat sinyal TL 074 seperti dalam Gambar 4-2. apabila diberi sinyal masukan dengan penguatan ( $A_v$ ) yang telah ditentukan yaitu 10 kali penguatan. Peralatan yang dipergunakan dalam pengujian ini antara lain catu daya dan Multimeter *type* EE-02.

Pengujian penguatan  $A_v$  dapat dilakukan dengan cara menggunakan Multimeter yang berfungsi untuk mengetahui besar nilai  $V_{I_{out}}$ ,  $V_{in}$ , dan  $V_{out}$  dengan masukan dari sumber tegangan.

Spesifikasi keluaran sensor yang diinginkan untuk diumpankan ke ADC 0809 dapat mengukur temperatur dari  $0^{\circ}\text{C}$  sampai  $100^{\circ}\text{C}$ . Dengan jangkauan temperatur tersebut maka sensor akan mengeluarkan tegangan dari 0 V sampai 1000 mV (dianggap bahwa karakter sensor mengikuti spesifikasi sebagaimana yang dijelaskan dalam data sheet).

$$\begin{aligned} V_{out \text{ LM35}} &= \text{temperatur} \times 10\text{mV} \\ &= 100 \times 10 \\ &= 1000 \text{ mV} \end{aligned}$$

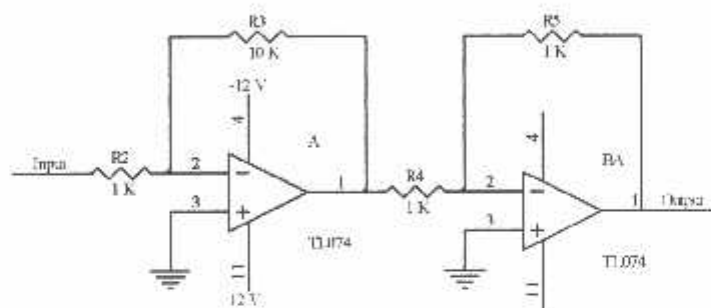
Keluaran sensor suhu tersebut perlu dikuatkan agar didapatkan ketelitian data yang lebih baik jika dimasukkan ke ADC. Penguatan diambil 20 kali oleh dua buah penguat TL 074. Penguat pertama berfungsi untuk menguatkan tegangan masukan dan penguat kedua berfungsi untuk membalik polaritas tegangan hasil penguatan penguat pertama. Untuk mencari nilai R2 dan R3 dapat dicari berdasarkan rumus:

$$V_{out} = -A_v \times V_{in}$$

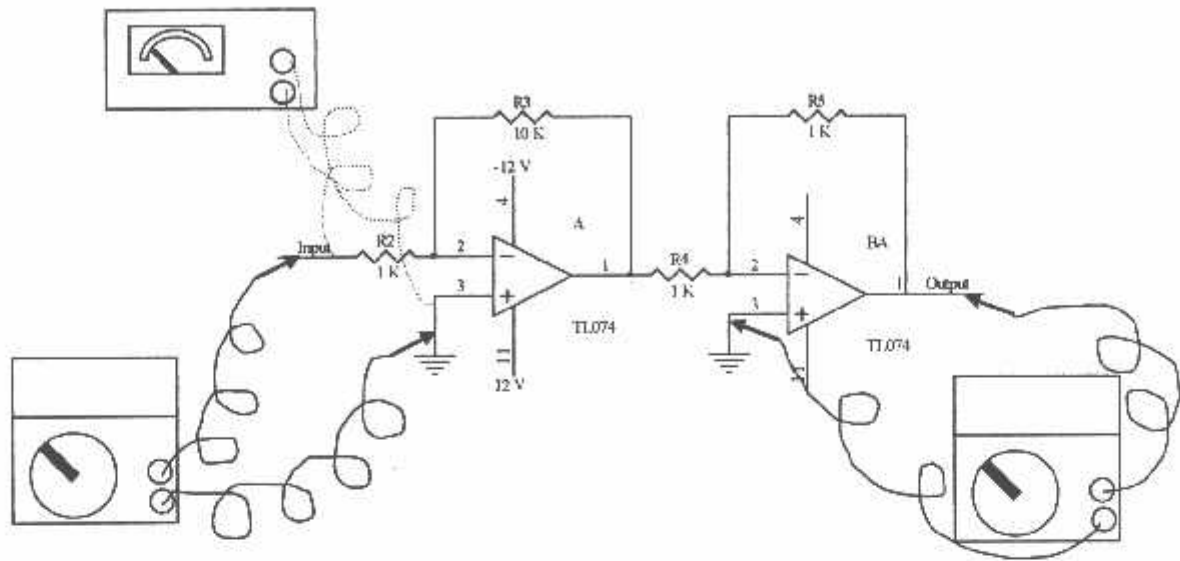
$$A_v = R_3/R_2$$

$$10 = R_3/R_2$$

Bila ditentukan  $R_3 = 10 \text{ K}\Omega$ , maka  $R_2 = 1 \text{ K}\Omega$ .



**Gambar 4-2 Rangkaian Penguat Sinyal TL 074**



**Gambar 4-3 Pengujian Penguatan Penguat Sinyal TL 074**

Langkah-langkah pengujian penguatan penguat sinyal TL 074 dengan menggunakan *Multi meter* adalah sebagai berikut :

- Menyiapkan rangkaian penguat Sinyal seperti dalam Gambar 4-3, serta memastikan bahwa catu daya 12V dan -12V DC telah terpasang dan hubungan pada rangkaian telah benar.
- Menghubungkan sumber tegangan dengan masukan input dalam Gambar 4-3.
- Setelah susunan dan hubungan antar rangkaian telah benar aktifkan sumber tegangan dan rangkaian penguat sinyal.
- Selanjutnya mengukur tegangan pada  $V_{input}$  dan  $V_{output}$ .

**Tabel 4-2 Hasil Pengujian Penguat Sinyal Dengan Multimeter**

$V_{input}$ (volt)	$V_{output}$ (volt)
0.1	0.5
0.25	1.25

Dari pengujian didapatkan hasil seperti dalam Tabel 4-2 dengan menyertakan data acuan dalam perencanaan. Dengan membandingkan hasil pengujian dan perencanaan maka didapatkan prosentase kesalahan  $V_{out}$  hasil pengujian sebesar 0.05%. Kesalahan ini dipengaruhi oleh tegangan catu rangkaian penguat sinyal.

#### 4.4. Pengujian Rangkaian ADC

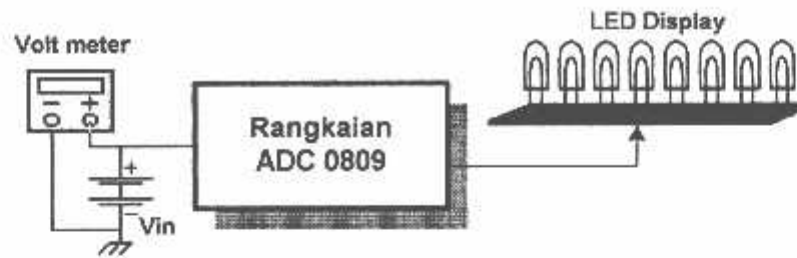
- Tujuan

Tujuan dari pengujian rangkaian ADC adalah untuk mengetahui level tegangan keluaran ADC dan kelinieran hasil konversi ADC.

- Peralatan yang diperlukan

1. Sumber tegangan DC variabel.
2. Multimeter Digital Model DT383.
3. Catu daya 5 volt.
4. Led Display.

- Prosedur Pengujian



Gambar 4-4 Diagram Blok Pengujian Rangkaian ADC

1. Rangkaian dibuat seperti Gambar 4-4.
  2. Memasang catu rangkaian sebesar 5 volt.
  3. Menghubungkan input rangkaian dengan sumber DC.
  4. Mengamati keluaran rangkaian pada LED Display dan tegangannya dengan Volt meter.
  5. Mengulangi langkah 3 untuk tegangan yang berbeda.
- Hasil Pengujian

Hasil pengujian ditunjukkan dalam tabel 4-3

Prosentase kesalahan :  $\frac{V_{out} \text{ pengujian} - V_{out} \text{ perhitungan}}{V_{out} \text{ pengujian}} \times 100\%$

$$\frac{41 - 40,8}{41} \times 100\%$$

$$= \frac{0,2}{41} \times 100\%$$

$$= 0,49\%$$

$$= 0,49\%$$

**Tabel 4-3 Hasil Dan Analisis Pengujian Rangkaian ADC**

No	V <sub>in</sub> (Volt)	V <sub>out</sub> Perhitungan (Desimal)	V <sub>out</sub> Pengujian (Desimal)	Prosentase Kesalahan (%)
1	0,00	0	0	0,000
2	0,80	40,8	41	0,490
3	1,00	51	51	0,000
4	1,50	76,5	76	0,653
5	2,00	102	101	0,980
6	2,55	130,05	130	0,038
7	3,05	155,55	156	0,289
8	3,49	177,99	178	0,005
9	4,03	205,53	206	0,228
10	4,45	226,95	227	0,022
11	5,00	255	255	0,000
				$\bar{X} = 0,277$

- Analisis Pengujian

#### 4.5. Pengujian Sistem Mikrokontroler

- Tujuan

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

- Peralatan yang dibutuhkan

1. Komputer ( PC ).
2. Led Display.



- Prosedur Pengujian

1. Membuat program yang digunakan dalam pengujian mikrokontroler.

Program yang digunakan dalam pengujian mikrokontroler ini merupakan program sederhana yang meletakkan  $0F_{11}$  dan  $F0_{11}$  pada ACC secara bergantian kemudian memindahkannya pada *Port 1* 89S52. Program yang dibuat adalah sebagai berikut :

```

ORG 0000H

                JMP    START

START:          MOV    A,#0FH

                MOV    P1,A

                CALL   TUNDA

                MOV    A,#F0H

                MOV    P1,A

                JMP    START

TUNDA:          MOV    R3,#0FFH
TUNDA1:         MOV    R2,#0FFH

                DJNZ   R2,$

                MOV    R1,#0FH

                DJNZ   R1,$

                DJNZ   R3,TUNDA1

                RET

                END

```

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



**Gambar 4-5 Diagram Blok Pengujian Mikrokontroller**

- Hasil Pengujian

Hasil pengujian ditunjukkan dalam Tabel 4-4.

**Tabel 4-4 Hasil Pengujian Sistem Mikrokontroller**

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

- Analisis Pengujian

Dari hasil pengujian dalam tabel 4-4 dapat dilihat bahwa *port 1* memberikan logika  $0F_{16}$  dan  $F0_{16}$  secara bergantian sesuai dengan isi program.

#### 4.6. Pengujian Rangkaian Tampilan

- Tujuan

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

- Peralatan yang dibutuhkan

1. Komputer ( PC ).
2. Sistem Mikrokontroler dan LCD TM1632 A.

- Prosedur Pengujian

1. Menyusun rangkaian seperti dalam Gambar 4-6.
2. Menjalankan program untuk menampilkan tulisan.
3. Mengamati keluaran pada LCD.



Gambar 4-6 Diagram Blok Pengujian Rangkaian Tampilan

#### 4.7. Pengujian Pencatatan Data pada EEPROM

- Tujuan

Untuk mengetahui apakah penyimpan data (EEPROM) dapat tetap menyimpan data saat catu daya dimatikan.

- Peralatan yang dibutuhkan

1. Komputer ( PC ).
  2. Sistem Mikrokontroler dan LCD TM202A.
- **Prosedur Pengujian**
    1. Menyusun rangkaian seperti dalam Gambar 4-7.
    2. Menjalankan program untuk menyimpan data ke EEPROM.
    3. Setelah data tersimpan, mematikan catu daya.
    4. Mengamati data pada EEPROM pada keluaran LCD setelah catu daya dihidupkan kembali.



**Gambar 4-7 Diagram Blok Pengujian Rangkaian Penyimpanan Data EEPROM**

- **Hasil Pengujian**

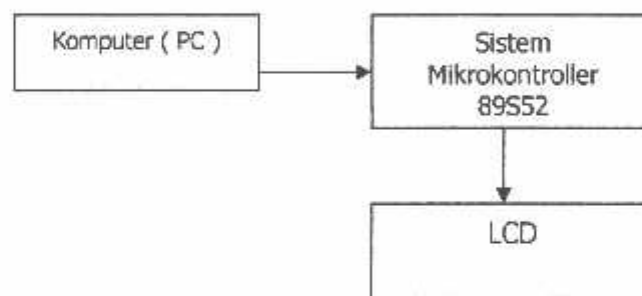
Hasil pengujian penyimpanan data menunjukkan data sebelum catu daya dimatikan adalah 55h, setelah catu dimatikan dan dihidupkan kembali data yang tersimpan pada EEPROM yaitu 55h.

- **Analisis Pengujian**

Hasil pengujian menunjukkan bahwa data yang tersimpan dalam EEPROM tidak hilang meskipun catu dimatikan. Dengan demikian penyimpanan data (EEPROM) telah bekerja dengan baik.

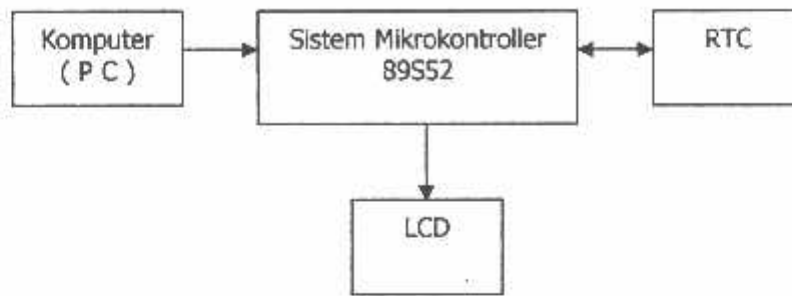
#### 4.8. Pengujian RTC

Pengujian RTC dilakukan untuk mengetahui apakah komponen yang digunakan dapat berfungsi dengan baik ataukah tidak. Untuk menguji RTC, sistem mikrokontroler dan LCD yang sudah diuji dihubungkan dengan RTC. Diagram blok pengujian RTC dapat dilihat dalam Gambar 4-8. Apabila RTC dapat berfungsi dengan baik maka tampilan waktunya akan berubah setiap satu detik.



**Gambar 4-8 Diagram Blok Pengujian LCD**

Berdasarkan hasil pengujian yang dilakukan selama 15 menit, dimana perubahan waktunya diukur dengan menggunakan *stopwatch* diperoleh bahwa RTC dapat bekerja dengan baik.



**Gambar 4-9 Diagram Blok Pengujian RTC**

#### **4.9. PENGUJIAN TRANSDUSER SUHU**

Pengujian pada transduser suhu dilakukan dengan cara :

1. Memberikan catu daya yang sesuai dengan spesifikasi.
2. Mengukur tegangan output dengan membandingkan dengan suhu aktual dengan menggunakan termometer.

Hasil pengujian terdapat dalam Tabel 4-5

**Tabel 4-5 Tegangan Output LM35 Dibandingkan Dengan Suhu Aktual Yang Diukur Dengan Termometer**

SUHU YANG TERBACA PADA TERMOMETER	TEGANGAN OUTPUT PADA LM35 (mV)	SUHU OUTPUT DARI LM35
20°C	203	20,3°C
21,5°C	217	21,7°C
22,5°C	224	22,4°C
23°C	232	23,2°C
24°C	239	23,9°C
25°C	252	25,2°C
26°C	261	26,1°C
28°C	282	28,2°C
29°C	289	28,9°C
30°C	301	30,1°C

Keterangan :

Perhitungan Suhu LM35 = ° C / 10mV

Bahwa 1° C = 10mV

$$^{\circ}C = \frac{\text{Tegangan Output}}{10mV}$$

Jika Tegangan Output 203mV

$$\begin{aligned} \text{maka } ^{\circ}C &= \frac{203}{10} \\ &= 20,3^{\circ}C \end{aligned}$$

Sedangkan pengujian kedua pada kestabilan sensor suhu dilakukan dengan cara memanaskan oven pemanas tempat dimana sensor suhu LM 35 diletakan, menggunakan pemanas listrik , pada tempat yang sama diletakan thermometer alkohol sebagai pembanding. Pengukuran dilakukan mulai pada suhu 0°C hingga 100°C. Sensor suhu LM 35 memiliki linieritas yang baik terhadap perubahan derajat celsius, dimana merespon perubahan 1 LSB nya sebesar 10 mV

**Tabel 4-6 Hasil Pengujian Kestabilan Sensor suhu**

Suhu(°C)	Tegangan Keluaran (V)		Error
	Perhitungan	Percobaan	
0	0	0.2	0.2
10	0.5	0.8	0.3
20	1	1.2	0.2
30	1.5	1.7	0.2
40	2	2.2	0.2
50	2.5	2.9	0.4
60	3	3.1	0.1
70	3.5	3.8	0.3
80	4	4.4	0.4
90	4.5	4.9	0.4
100	5	5.4	0.4



Resolusi sensor suhu LM35 =  $10\text{mV}/^{\circ}\text{C}$

Penguatan = 5 kali

$V_{\text{out}} = \text{Suhu} \times \text{Resolusi Sensor} \times \text{Penguatan}$

$$= 60 \times 0.01 \times 5 = 3\text{V}$$

Hasil pengukuran dengan teori ada sedikit perbedaan bisa disebabkan:

- Tidak tepat dalam membaca suhu pada termometer acuan.
  - Tidak presisinya voltmeter dan terjadi pembebanan oleh alat ukur.
  - Ada selisih suhu di termometer dengan transduser suhu pada waktu pengukuran.
-

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Dari hasil pengujian dan analisa alat yang telah dibuat maka dapat disimpulkan :

1. Pada pengujian mikrokontroller AT89S52, mikrokontroller dapat bekerja dengan baik sebagai pengendali sistem.
2. Pada pengujian EEPROM sebagai penyimpan data dapat tetap menyimpan data saat catu daya dimatikan.
3. RTC dapat menampilkan data pewaktuan dengan tepat.
4. Pada pengujian tampilan cristal cair, LCD dapat menampilkan informasi yang sedang berlangsung dengan baik sesuai dengan program.
5. Pada pengujian rangkaian papan tombol yang berfungsi sebagai masukan data dapat berjalan dengan baik.

#### **5.2. Saran**

Dalam pembuatan skripsi ini masih terdapat beberapa kekurangan, sehingga untuk mencapai hasil yang lebih baik dan pengembangan lebih lanjut maka dapat diberikan saran – saran sebagai berikut :

1. Sebaiknya untuk pengembangan lebih lanjut, pemasukan menu bisa dikirim atau diakses melalui remote. Dalam hal ini lebih mengotomatisasi cara kerja oven.

## DAFTAR PUSTAKA

1. Dallas Semiconductor Corp. 1995. *Dallas Semiconductor Data Book*
2. Fairchild Semiconductor. 1999. *Fairchild Semiconductor Data Book*.
3. Intel, 1989, *MCS<sup>®</sup>-51 Family of Microcontroller*
4. Kenneth J. Ayala, "The 8051 Microcontroller Architecture, Programming, and Applications," Carolina, 1991.
5. Malvino. 1985. *Prinsip-prinsip Elektronika*. Edisi Ketiga. Alih Bahasa Barmawi, Tjia. Jakarta : Erlangga.
6. Moh. Ibnu Malik & Anistardi, 1997. *Bereksperimen dengan Mikrokontroler 8031*. Jakarta : Penerbit PT Elex Media Komputindo.

## LAMPIRAN





## FORMULIR BIMBINGAN SKRIPSI

Nama : Mochamad Yusuf  
Nim : 9917181  
Masa Bimbingan : 18-Mar-2005 s/d 18-Sep-2005  
Judul Skripsi : Perancangan dan pembuatan oven terprogram dengan sistem menu berbasis Mikrokontroler AT89S52

NO	Tanggal	Uraian	Paraf Pembimbing
1.	18-02-2005	Konsultasi Rancangan Sistem	
2.	23-02-2005	Revisi Rancangan EEPROM	
3.	25-02-2005	Revisi Rancangan Kelembaban	
4.	19-03-2005	Konsultasi Bab I, II, III	
5.	21-03-2005	Konsultasi Flowchart Program	
6.	22-03-2005	Revisi Penulisan	
7.	23-03-2005	Konsultasi melalui seminar hasil	
8.	26-03-2005	AC Chapter, Diagram Lampu	
9.			
10.			

Malang, 26-03-2005  
Dosen Pembimbing

Joseph Dedy Irawan, ST, MT

Form. S-4a



### 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 : *Mochamad Yusuf*  
NIM : *9917181*  
Perbaikan meliputi :

- ① Diagram sambungan elemen memori.
- ② Flow Chart untuk proses pengukuran suhu
- ③ Pengujian untuk kestabilan suhu ?

Malang,

200



### 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 : M. Yusuf  
N I M :  
Perbaikan meliputi :

- Perbaiki penyajian.
- KDS sesuai ref.
- hasil pemecahan
- ~~MATRIK~~

Malang, 30/5 2005  
(Eff M)



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

### FORMULIR PERBAIKAN SKRIPSI

Dari hasil ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro  
Konsentrasi Elektronika yang diselenggarakan pada :

Hari : Selasa

Tanggal : 29 Maret 2005

Telah dilakukan Perbaikan Skripsi Oleh :

1. Nama Mahasiswa : Mochamad Yusuf
2. NIM : 99.17.181
3. Jurusan : Teknik Elektro S-1
4. Konsentrasi : Teknik Elektronika
5. Judul Skripsi : Perancangan Dan Pembuatan Oven Terprogram Dengan Sistem Menu Berbasis Mikrokontroler AT89S52

Perbaikan meliputi :

No	Materi Perbaikan	Keterangan
1	DIAGRAM TAMBAHAN MEMORI	
2	FLOWCHART PENGATURAN SUHU	
3	PENGUJIAN UNTUK KESTABILAN SUHU	
4	MEMPERBAIKI PENGUJIAN AGAR SESUAI PERENCANAAN	
5	TATA TULIS	

Dosen Pembimbing

( Joseph Dedy Irawan, ST, MT )  
NIP. 1039800324

Anggota Penguji

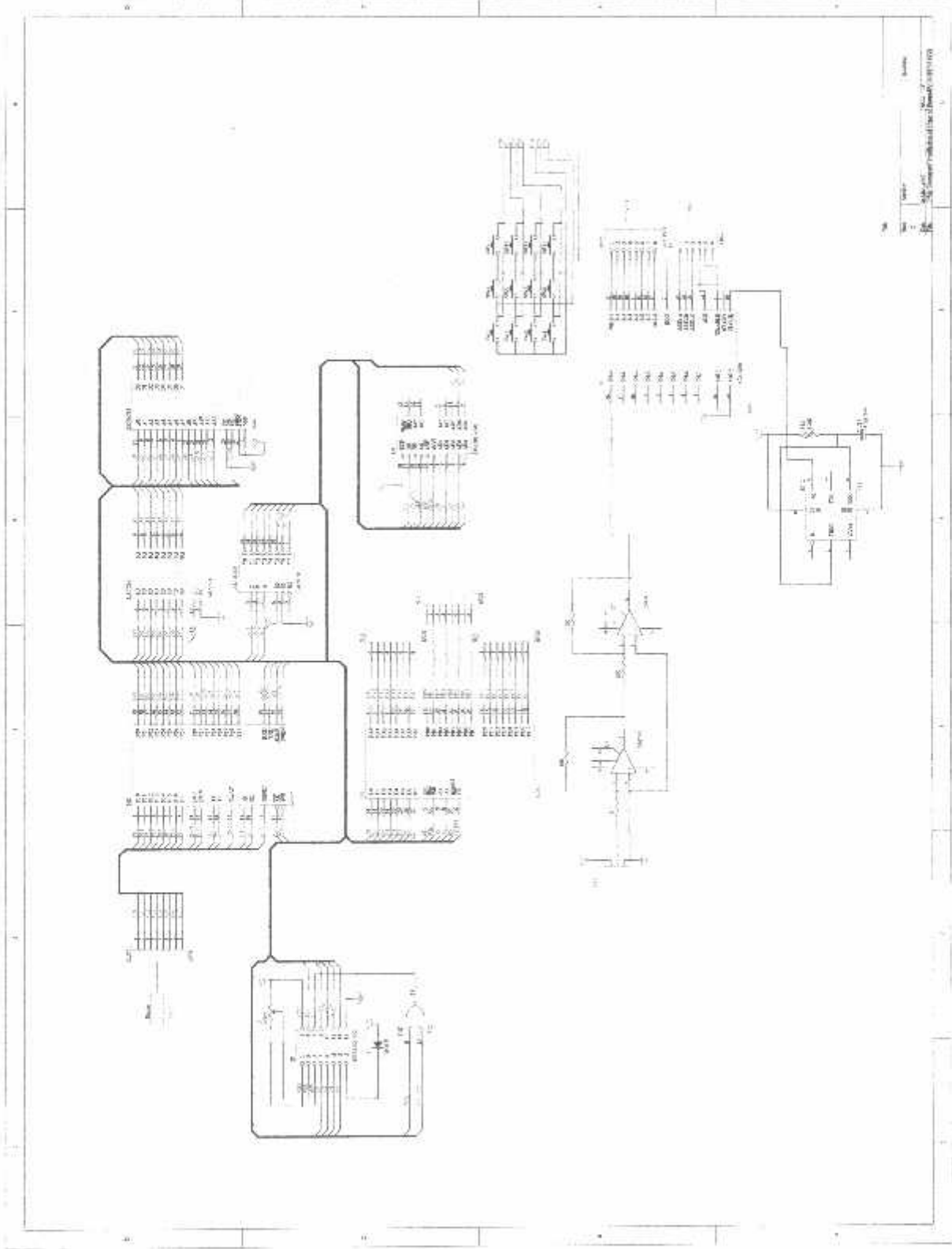
Penguji Pertama

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

Penguji Kedua

( Ir. H. Erfan Ahmad Dahlan )  
NIP. 131124663





```

;===== MODULE RTC =====
detrtc   Equ    0000h    ;alamat detik rtc
mntrtc   Equ    0002h
jamrtc   Equ    0004h
harirtc  Equ    0006h
tglrtc   Equ    0007h
blnrtc   Equ    0008h
thnrtc   Equ    0009h
regartc  Equ    000ah
regbrtc  Equ    000bh
regcrtc  Equ    000ch

Masakan1   Equ    0c000h
Masakan2   Equ    0c100h
Masakan3   Equ    0c200h
Masakan4   Equ    0c300h

EE_Jadwal1   Equ    detrtc+30h
EE_Jadwal2   Equ    detrtc+40h
EE_Jadwal1Off Equ    detrtc+50h
EE_Jadwal2Off Equ    detrtc+60h

StatusRelay1 Equ    detrtc+70h
StatusRelay2 Equ    detrtc+71h

Relay       Bit    P1.0
Buzzer      Bit    P1.1

Suhu        Data    27h
ADC          Data    28h
Cahaya      Data    29h
Tmp1        Data    2Ah
Tmp2        Data    2Bh
Tmp3        Data    2Ch
Tmp4        Data    2Dh

TmpJam      Data    2Eh
TmpMenit    Data    2Fh
TmpDetik    Data    30h
dphh       Data    31h
dpll       Data    32h
TmpSuhu     Data    33h
TmpWaktu    Data    34h
TmpSuhu1    Data    35h
TmpSuhu2    Data    36h
MenitStop   Data    37h
DetikStop   Data    38h
FlagStop    Data    39h
FlagCounter Data    3Ah

Booting:
    Mov     SP,#80h
    Clr     Relay
    Setb    Buzzer

```

---

```

        Clr     EA
        Call   LCD_Initialization
        Call   LCD_Blink_Off

        Mov    R0,#20h
        Mov    R1,#02h
        Call   RTC_Initialization

        Mov    R0,#20h
        Mov    R1,#02h
        Call   RTC_Initialization

        Call   RTC_Run

        Mov    A,#91h
        Call   PPI_Initialization

Menu_Awal:
        Setb   Buzzer
        Call   LCD_Blink_Off
        Call   LCD_Clrscr
        Call   LCD_Line1
        Mov    DPTR,#txt_menu1
        Call   LCD_String
        Call   LCD_Line2
        Mov    DPTR,#txt_menu2
        Call   LCD_String

AmbilData:
        Mov    R0,#2
        Call   Keypad_Get
        Cjne  A,#'1',AmbilData1
        Jmp   OvenSekarang

AmbilData1:
        Cjne  A,#'2',AmbilData
        Jmp   EditSuhu

;-----
OvenSekarang:
        Call   LCD_Clrscr
        Call   LCD_Line1
        Mov    DPTR,#txt_masakan1
        Call   LCD_String
        Call   LCD_Line2
        Mov    DPTR,#txt_masakan2
        Call   LCD_String

AmbilMasakan:
        Mov    R0,#2
        Call   Keypad_Get
        Cjne  A,#'1',AmbilMasakan1
        Call   LCD_Clrscr
        Call   LCD_Line1
        Mov    DPTR,#txt_telur
        Call   LCD_String
        Mov    DPTR,#Masakan1

```

---

```

    Mov     dphh,DPH
    Mov     dp11,DPL
    Jmp     ProsesOven
AmbilMasakan1:
    Cjne   A,#'2',AmbilMasakan2
    Call   LCD_Clrscr
    Call   LCD_Line1
    Mov    DPTR,#txt_donat
    Call   LCD_String
    Mov    DPTR,#Masakan2
    Mov    dphh,DPH
    Mov    dp11,DPL
    Jmp    ProsesOven
AmbilMasakan2:
    Cjne   A,#'3',AmbilMasakan3
    Call   LCD_Clrscr
    Call   LCD_Line1
    Mov    DPTR,#txt_Pizza
    Call   LCD_String
    Mov    DPTR,#Masakan3
    Mov    dphh,DPH
    Mov    dp11,DPL
    Jmp    ProsesOven
AmbilMasakan3:
    Cjne   A,#'4',AmbilMasakan
    Call   LCD_Clrscr
    Call   LCD_Line1
    Mov    DPTR,#txt_Kentang
    Call   LCD_String
    Mov    DPTR,#Masakan4
    Mov    dphh,DPH
    Mov    dp11,DPL
    Jmp    ProsesOven
ProsesOven:
    ;Call  LCD_Line2
    Mov    DPH,dphh
    Mov    DPL,dp11
    Movx   A,@DPTR
    Mov    TmpSuhu,A
    Inc    DPTR
    Movx   A,@DPTR
    Mov    TmpWaktu,A

    Call   LCD_Line2
    Mov    DPTR,#txt_Waktu
    Call   LCD_String

    Mov    R0,#00h
    Mov    R1,#00h
    Mov    R2,#00h
    Call   RTC_SetTime

    Mov    MenitStop,#0
    Mov    DetikStop,#0

    Mov    RC,#20h

```

---

```

    Mov     R1, #02h
    Call    RTC_Initialization

    Call    RTC_Stop
    Mov     FlagStop, #0
AmbilSuhuWaktuOven:
    Call    AmbilSuhu

    Mov     R0, #18h
    Call    LCD_Cursor_Position
    Mov     R0, #18h
    Call    LCD_Cursor_Position
    Mov     A, Suhu
    Call    ShowSuhu

;-----

    Mov     A, Suhu
    Cjne   A, TmpSuhu1, CekKecilOven
RelayOn:
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     A, #'*'
    Call    LCD_Data
    Mov     FlagCounter, #0FFh
    Setb   Relay
    Jmp    MulaiHitung
RelayOn12:
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     A, #'*'
    Call    LCD_Data
    Mov     FlagCounter, #0h
    Setb   Relay
    Jmp    MulaiHitung
CekKecilOven:
    Jc     RelayOn12
    Mov     A, Suhu
    Cjne   A, TmpSuhu2, CekBesarOven
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     R0, #2Fh
    Call    LCD_Cursor_Position
    Mov     A, #' '
    Call    LCD_Data
    Mov     FlagCounter, #0FFh
    Clr    Relay
    Jmp    MulaiHitung
RelayOn11:
    Mov     FlagCounter, #0FFh
    Jmp    MulaiHitung

```

---

```

CekBesarOven:
    Jc     RelayOn1
    Mov    R0,#2Fh
    Call   LCD_Cursor_Position
    Mov    R0,#2Fh
    Call   LCD_Cursor_Position
    Mov    A,#' '
    Call   LCD_Data
    Mov    FlagCounter,#0FFh
    Clr    Relay
;-----
MulaiHitung:
    Mov    A,FlagCounter
    Cjne   A,#00,RunRTC
    Mov    A,FlagStop
    Cjne   A,#00,Stoopp
    Mov    FlagStop,#0FFh
    Mov    DetikStop,TmpDetik
Stoopp:
    Call   RTC_Stop
    Jmp    BandingWaktu
RunRTC:
    Mov    A,FlagStop
    Cjne   A,#0FFh,RunRTC1
    Mov    FlagStop,#0
    Call   RTC_Stop
    Mov    R0,#00h
    Call   RTC_SetTime
RunRTC1:
    Call   RTC_Run
    Jmp    BandingWaktu
BandingWaktu:
    Call   RTC_GetTime
    Mov    A,TmpMenit
    Cjne   A,TmpWaktu,TerusAmbilSuhu
    Call   LCD_Clrscr
    Call   LCD_Line1
    Mov    DPTR,#txt_matang
    Call   LCD_String
    Clr    Buzzer
    Clr    Relay
asdfasdfsdf:
    Mov    R0,#2
    Call   Keypad_Get
    Cjne   A,#'#',asdfasdfsdf
    Setb   Buzzer
    Jmp    Menu_Awal
TerusAmbilSuhu:
    Mov    R0,#27h
    Call   LCD_Cursor_Position
    Mov    R0,#27h
    Call   LCD_Cursor_Position
    Call   RTC_ShowTime
    Call   Delay_Fix_100ms
    Call   Delay_Fix_100ms

```

---

```

        Call    Delay_Fix_100ms
        Jmp     AmbilSuhuWaktuOven
;=====
EditSuhu:

        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov     DPTR,#txt_masakan1
        Call    LCD_String
        Call    LCD_Line2
        Mov     DPTR,#txt_masakan2
        Call    LCD_String
eAmbilMasakan:
        Mov     R0,#2
        Call    Keypad_Get
        Cjne   A,#'1',eAmbilMasakan1
        Mov     DPTR,#Masakan1
        Mov     dphh,DPH
        Mov     dpll,DPL
        Jmp     ProsesEdit
eAmbilMasakan1:
        Cjne   A,#'2',eAmbilMasakan2
        Mov     DPTR,#Masakan2
        Mov     dphh,DPH
        Mov     dpll,DPL
        Jmp     ProsesEdit
eAmbilMasakan2:
        Cjne   A,#'3',eAmbilMasakan3
        Mov     DPTR,#Masakan3
        Mov     dphh,DPH
        Mov     dpll,DPL
        Jmp     ProsesEdit
eAmbilMasakan3:
        Cjne   A,#'4',eAmbilMasakan4
        Mov     DPTR,#Masakan4
        Mov     dphh,DPH
        Mov     dpll,DPL
        Jmp     ProsesEdit
ProsesEdit:

        Call    LCD_Clrscr
        Call    LCD_Line1
        Mov     DPTR,#txt_Suhu1
        Call    LCD_String
        Call    LCD_Line2
        Mov     DPTR,#txt_waktu1
        Call    LCD_String

        Call    LCD_Blink_On
        Mov     R0,#16h
        Call    LCD_Cursor_Position
        Mov     R0,#16h
        Call    LCD_Cursor_Position

        Mov     DPH,dphh
        Mov     DPL,dpll

```

```

Movx    A, @DPTR
Call    Conversion_Hexa_to_Bcd
Mov     A, R0
Anl    A, #0fh
Orl    A, #30h
Call    LCD_Data
Mov     A, R1
Call    LCD_Hexa

Mov     R0, #26h
Call    LCD_Cursor_Position
Mov     R0, #26h
Call    LCD_Cursor_Position
Mov     DPH, dphh
Mov     DPL, dpll
Inc     DPTR
Movx    A, @DPTR
Call    Conversion_Hexa_to_Bcd
Mov     A, R1
Call    LCD_Hexa

;-----
Mov     R0, #16h
Call    LCD_Cursor_Position
Mov     R0, #16h
Call    LCD_Cursor_Position
Mov     R0, #0
Call    Keypad_Get
Cjne   A, #'#', Nextttt1
Call    LCD_Blink_Off
Jump   Menu_Awal
Nextttt1:
Anl    A, #0Fh
Mov     70h, A

Mov     R0, #0
Call    Keypad_Get
Cjne   A, #'#', Nextttt2
Call    LCD_Blink_Off
Jump   Menu_Awal
Nextttt2:
Anl    A, #0Fh
Swap   A
Mov     71h, A

Mov     R0, #0
Call    Keypad_Get
Cjne   A, #'#', Nextttt3
Call    LCD_Blink_Off
Jump   Menu_Awal
Nextttt3:
Anl    A, #0Fh
Orl    A, 71h
Mov     71h, A

;-----

```

---



```

    Mov     R0,#26h
    Call    LCD_Cursor_Position
    Mov     R0,#26h
    Call    LCD_Cursor_Position

    Mov     R0,#0
    Call    Keypad_Get
    Cjne   A,##',Nextttt4
    Call    LCD_Blink_Off
    Jump   Menu_Awal
Nextttt4:
    Anl    A,#0Fh
    Swap   A
    Mov     72h,A

    Mov     R0,#0
    Call    Keypad_Get
    Cjne   A,##',Nextttt5
    Call    LCD_Blink_Off
    Jump   Menu_Awal
Nextttt5:
    Anl    A,#0Fh
    Orl    A,72h
    Mov     72h,A

    Mov     6,70h
    Mov     7,71h
    Call    Kon_Des_Hexa
    Mov     60h,A

    Mov     6,#0
    Mov     7,72h
    Call    Kon_Des_Hexa
    Mov     61h,A

;----- proses -----
Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR,#txt_isian
Call    LCD_String
Call    LCD_Line2
Mov     DPTR,#txt_yakin1
Call    LCD_String

    Mov     R0,#15h
    Call    LCD_Cursor_Position
    Mov     R0,#15h
    Call    LCD_Cursor_Position

    Mov     A,70h
    Orl    A,#30h
    Call    LCD_Data
    Mov     A,71h
    Call    LCD_Hexa

```

---

```

    Mov     R0,#1Dh
    Call    LCD_Cursor_Position
    Mov     R0,#1Dh
    Call    LCD_Cursor_Position

    Mov     A,72h
    Call    LCD_Hexa

    Call    LCD_Blink_Off

Keputusan:
    Mov     R0,#2
    Call    Keypad_Get
    Cjne   A,#'1',Keputusan1
    Jmp     SimpanEdit
Keputusan1:
    Cjne   A,#'2',Keputusan
    Jmp     Menu_Awal
SimpanEdit:
    Mov     DPH,dphh
    Mov     DPL,dpll
    Mov     A,60h
    Call    EEPROM_Write
    Inc     DPTR
    Mov     A,61h
    Call    EEPROM_Write
    Jmp     Menu_Awal
    Sjmp   $
;-----

```

Subroutine Conversion\_Hexa\_to\_Bcd:

```

    Push   B
    Mov     B,#100
    Div    AB
    Mov     R0,A
    Mov     A,#10
    Xch    A,B
    Div    AB
    Swap   A
    Add    A,B
    Mov     R1,A
    Pop    B

```

EndSub

Subroutine Conversion\_bcd\_to\_hexa:

```

    Push   B
    Mov     B,#100
    Div    AB
    Mov     R6,A
    Mov     A,#10
    Xch    A,B
    Div    AB
    Swap   A
    Add    A,B
    Mov     R7,A
    Pop    B

```

---



```

Nop
Nop
Mov     A,#00110000b ;ena
Call    PPI_Send_Port_B
Nop

Nop
Nop
Nop
Lcall   PPI_Get_Port_A
Mov     Suhu,A
Mov     A,#00000000b ;ena+alamat
Call    PPI_Send_Port_B
Nop

Nop
Nop

```

EndSub

```

;-----
;----- MODUL KEYPAD -----
keypad_id_keypad      Equ    02h    ;register r2
keypad_cek_key_B8     Equ    03h    ;register r3
keypad_temp_baris     Equ    04h    ;register r4
keypad_temp_data      Equ    05h    ;register r5
keylx                 Equ    06h
keydelay              Equ    07h
keydat1               Equ    08h
keydat2               Equ    09h
keypad_curpos         Equ    10h
keypad_port_a         Equ    0eC00h
keypad_port_b         Equ    0eC01h
keypad_port_c         Equ    0eC02h
keypad_port_dw        Equ    0eC03h
keypad_shift          Equ    06h
Keypad_cons_1:        Db      '1^',0
Keypad_cons_2:        Db      'ABC2',0
Keypad_cons_3:        Db      'DEF3',0
Keypad_cons_4:        Db      'GHI4',0
Keypad_cons_5:        Db      'JKL5',0
Keypad_cons_6:        Db      'MNO6',0
Keypad_cons_7:        Db      'PQRS7',0
Keypad_cons_8:        Db      'TUV8',0
Keypad_cons_9:        Db      'WXYZ9',0
Keypad_cons_0:        Db      '0 ',0
Keypad_cons_10:       Db      ',;?1()"',0
Keypad_dphh           Equ    0Ah
Keypad_dp11           Equ    0Bh
Keypad_dphh1          Equ    0Ch
Keypad_dp111          Equ    0Dh
Keypad_dphh2          Equ    0Eh
Keypad_dp112          Equ    0Fh
Subroutine Keypad_Initialization:
    Push    DPL
    Push    DPH
    Mov     A,R0

```

```

        Cjne    A,#1,keypad_38_init
keypad_end_initx:
    Mov    A,#81h ;pa:out pb:out pcl:in pch:out
    Mov    DPTR,#keypad_port_cw
    Movx   @DPTR,A
    Jmp    keypad_end_init
keypad_8B_init:
    Cjne    A,#2,keypad_end_initx
    Mov    A,#82h ;pa:out pb:in pc:out
    Mov    DPTR,#keypad_port_cw
    Movx   @DPTR,A
keypad_end_init:
    Pop    DPH
    Pop    DPL
EndSub
Subroutine Keypad_Alpha:
    Mov    keypad_shift,#0
keypad_alpha:
    Mov    R0,#2
    Call   Keypad_Get
    Cjne   A,'#',tampilkan_data_keypad
    Inc    keypad_shift
    Mov    A,keypad_shift
    Cjne   A,#5,keypad_alpha
    Mov    keypad_shift,#0
    Jmp    keypad_alpha
tampilkan_data_keypad:
    Cjne   A,'#1',proses_key_2
    Mov    DPTR,#Keypad_cons_1
    Jmp    proses_keypad_alpha
proses_key_2:
    Cjne   A,'#2',proses_key_3
    Mov    DPTR,#Keypad_cons_2
    Jmp    proses_keypad_alpha
proses_key_3:
    Cjne   A,'#3',proses_key_4
    Mov    DPTR,#Keypad_cons_3
    Jmp    proses_keypad_alpha
proses_key_4:
    Cjne   A,'#4',proses_key_5
    Mov    DPTR,#Keypad_cons_4
    Jmp    proses_keypad_alpha
proses_key_5:
    Cjne   A,'#5',proses_key_6
    Mov    DPTR,#Keypad_cons_5
    Jmp    proses_keypad_alpha
proses_key_6:
    Cjne   A,'#6',proses_key_7
    Mov    DPTR,#Keypad_cons_6
    Jmp    proses_keypad_alpha
proses_key_7:
    Cjne   A,'#7',proses_key_8
    Mov    DPTR,#Keypad_cons_7
    Jmp    proses_keypad_alpha
proses_key_8:
    Cjne   A,'#8',proses_key_9

```

---

```

        Mov     DPTR,#Keypad_cons_8
        Jmp     proses_keypad_alpha
proses_key_9:
        Cjne   A,#'9',proses_key_0
        Mov     DPTR,#Keypad_cons_9
        Jmp     proses_keypad_alpha
proses_key_0:
        Cjne   A,#'0',proses_key_a
        Mov     DPTR,#Keypad_cons_0
        Jmp     proses_keypad_alpha
proses_key_a:
        Mov     B,A
        Call   LCD_Data
        Mov     A,B
        Ret
proses_keypad_alpha:
        Mov     A,keypad_shift
        Movc   A,@A+DPTR
        Cjne   A,#0,proses_keypad_alphaxx
        Ret
proses_keypad_alphaxx:
        Call   Delay_Fix_100ms
        Mov     B,A
        Call   LCD_Data
        Mov     A,B
        Ret
EndSub
Subroutine Keypad_Get:
        Call   keypad_get_44
        Mov     B,A
        Mov     A,R2
        Cjne   A,#01,Keypad_Get
        Call   Delay_Fix_100ms
        Call   Delay_Fix_100ms
        Call   Delay_Fix_100ms
        Mov     A,B
EndSub
Subroutine Keypad_Get_Matrix_44:
;-----
keypad_get_matrik_44:
keypad_loop_awal_44:
keypad_get_44:
        Mov     A,#80h                ;kolom satu
        Mov     keypad_temp_baris,A   ;kolom
keypad_loopstart_44:
        Call   keypad_get_port_c1
        Anl     A,#00001111b
        Cjne   A,#0,keypad_loopstart_44 ;cek keypad
        Mov     A,keypad_temp_baris   ;kolom
        Mov     A,keypad_temp_baris   ;kolom
        Rr     A                      ;naikan kolom
        Cjne   A,#0Fh,keypad_loop_44 ;cek jika > 4 baris
        Ret
;-----
;                               BARIS SATU

```

```

;-----
keypad_baris1_44:
    Mov     keypad_id_keypad,#1      ;simpan id keypad
;(<>0:ditekar)
    Cjne   A,#00000001b,keypad_baris2_44 ;baris 1
    Mov    A,keypad_temp_baris
    Cjne   A,#10000000b,keypad_kolom21_44 ;kolom 1
    Mov    A,#'1'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom21_44:
    Cjne   A,#01000000b,keypad_kolom31_44 ;kolom 2
    Mov    A,#'2'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom31_44:
    Cjne   A,#00100000b,keypad_kolom41_44 ;kolom 3
    Mov    A,#'3'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom41_44:
    Cjne   A,#00010000b,keypad_end_baris1_44 ;kolom 4
    Mov    A,#'A'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_end_baris1_44:
    Jmp    keypad_end_matrik1_44
;-----
;                BARIS DUA
;-----
keypad_baris2_44:
    Cjne   A,#00000010b,keypad_baris3_44 ;baris 2
    Mov    A,keypad_temp_baris
    Cjne   A,#10000000b,keypad_kolom22_44 ;kolom 1
    Mov    A,#'4'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom22_44:
    Cjne   A,#01000000b,keypad_kolom32_44 ;kolom 2
    Mov    A,#'5'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom32_44:
    Cjne   A,#00100000b,keypad_kolom42_44 ;kolom 3
    Mov    A,#'6'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_kolom42_44:
    Cjne   A,#00010000b,keypad_end_baris2_44 ;kolom 4
    Mov    A,#'B'
    Mov    keypad_temp_data,A
    Jmp    keypad_end_matrik_44
keypad_end_baris2_44:
    Jmp    keypad_end_matrik1_44
;-----

```

```

;          BARIS TIGA
;-----
keypad_baris3_44:
    Cjne    A,#00000100b,keypad_baris4_44    ;baris 3
    Mov     A,keypad_temp_baris
    Cjne    A,#10000000b,keypad_kolom23_44   ;kolom 1
    Mov     A,#'7'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom23_44:
    Cjne    A,#01000000b,keypad_kolom33_44   ;kolom 2
    Mov     A,#'8'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom33_44:
    Cjne    A,#00100000b,keypad_kolom43_44   ;kolom 3
    Mov     A,#'9'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom43_44:
    Cjne    A,#00010000b,keypad_end_baris3_44 ;kolom 5
    Mov     A,#'C'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_end_baris3_44:
    Jmp     keypad_end_matrik1_44

;-----
;          BARIS EMPAT
;-----
keypad_baris4_44:
    Cjne    A,#00001000b,keypad_baris4_44    ;baris 4
    Mov     A,keypad_temp_baris
    Cjne    A,#10000000b,keypad_kolom24_44   ;kolom 1
    Mov     A,#'+'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom24_44:
    Cjne    A,#01000000b,keypad_kolom34_44   ;kolom 2
    Mov     A,#'0'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom34_44:
    Cjne    A,#00100000b,keypad_kolom44_44   ;kolom 3
    Mov     A,#'#'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_kolom44_44:
    Cjne    A,#00010000b,keypad_end_baris4_44 ;kolom 4
    Mov     A,#'D'
    Mov     keypad_temp_data,A
    Jmp     keypad_end_matrik_44
keypad_end_baris4_44:
    Jmp     keypad_end_matrik1_44
;----- end deteksi keypad -----

```



```

keypad_end_matrik_44:
    Call    keypad_syarat_key_88
    Jmp     keypad_end_matrik2_44
keypad_end_matrik1_44:
    Jmp     keypad_loop_awal_44
keypad_end_matrik2_44:
    Mov     A, keypad_temp_data
    Ret

;-----
; send port c high
;-----
keypad_send_port_ch:
    Mov     DPTR, #keypad_port_c
    Movx    @DPTR, A
    Ret

;---- end of send port ch ----

;-----
; get port c low
;-----
keypad_get_port_cl:
    Mov     DPTR, #keypad_port_c
    Movx    A, @DPTR
    Ret

;--- end of get port cl

; Syarat tampilkan data
;-----
keypad_syarat_key_88:
    Mov     A, R0      ;syarat tampilkan karakter
    Cjne   A, #0, keypad_sandi_data_88
    Mov     A, keypad_temp_data
    Jmp     keypad_tampil_data_88
keypad_sandi_data_88:
    Cjne   A, #1, keypad_hide_data_88
    Mov     A, R1      ;sandi
    Jmp     keypad_tampil_data_88
keypad_hide_data_88:
    Jmp     keypad_ora_tampil_88
keypad_tampil_data_88:
    Lcall   LCD_Data
keypad_ora_tampil_88:
    Ret

key1:
    Push   DPH
    Push   DPL
    Mov    key1x, #2
    Mov    keydelay, #50

keyin1:
    Mov    R0, key1x
    Call   keypad_get_44
    Mov    keydat1, A
    Mov    A, R2
    Cjne  A, #01, keyin1
    Mov    A, keydat1
    Jmp   keyin12

```

```

keyin12:
    Cjne    A,#'1',keyin13
    Mov     DPTR,#Keypad_cons_1
    Jump    keyin_proses
keyin13:
    Cjne    A,#'2',keyin14
    Mov     DPTR,#Keypad_cons_2
    Jump    keyin_proses
keyin14:
    Cjne    A,#'3',keyin15
    Mov     DPTR,#Keypad_cons_3
    Jump    keyin_proses
keyin15:
    Cjne    A,#'4',keyin16
    Mov     DPTR,#Keypad_cons_4
    Jump    keyin_proses
keyin16:
    Cjne    A,#'5',keyin17
    Mov     DPTR,#Keypad_cons_5
    Jump    keyin_proses
keyin17:
    Cjne    A,#'6',keyin18
    Mov     DPTR,#Keypad_cons_6
    Jump    keyin_proses
keyin18:
    Cjne    A,#'7',keyin19
    Mov     DPTR,#Keypad_cons_7
    Jump    keyin_proses
keyin19:
    Cjne    A,#'8',keyin20
    Mov     DPTR,#Keypad_cons_8
    Jump    keyin_proses
keyin20:
    Cjne    A,#'9',keyin21
    Mov     DPTR,#Keypad_cons_9
    Jump    keyin_proses
keyin21:
    Cjne    A,#'0',keyin22
    Mov     DPTR,#Keypad_cons_0
    Jump    keyin_proses
keyin22:
    Cjne    A,#'+',keyin23
    Mov     DPTR,#Keypad_cons_10
    Jump    keyin_proses

keyin23:
    Jump    keyin_end
;-----
keyin_proses:
    Mov     Keypad_dphh,DPH
    Mov     Keypad_dp11,DPL
    Mov     Keypad_dp111,DPL
    Mov     Keypad_dp111,DPL
    Mov     DPH,Keypad_dphh
    Mov     DPL,Keypad_dp11
    Clr     A

```

```

        Movc    A,@A+DPTR
        Mov     B,A
        Mov     RC, keypad_curpos
        Call    LCD_Cursor_Position
        Mov     A,B
        Call    LCD_Data
        Mov     A,#0
        Call    PPI_Send_Port_C
        Mov     R0, keydelay
        Call    Delay_Var_10ms
keyin100:
        Mov     R0,#2
        Call    keypad_get_44
        Cjne   A,#01, keyin101a
        Jump   keyin_proses1
keyin101a:
        Mov     A,#0
        Call    PPI_Send_Port_C
        Mov     DPH,Keypad_dphh
        Mov     DPL,Keypad_dp11
        Clr     A
        Movc   A,@A+DPTR
        Cjne   A,#0, keyin_end
        Mov     DPH,Keypad_dphh1
        Mov     DPL,Keypad_dp111
        Clr     A
        Movc   A,@A+DPTR
        Jump   keyin_end
keyin_proses1:
        Mov     A,#0
        Call    PPI_Send_Port_C
        Mov     A, keydat1
        Cjne   A, keydat2, keyin101a
        Mov     DPH,Keypad_dphh
        Mov     DPL,Keypad_dp11
        Inc     DPTR
        Clr     A
        Movc   A,@A+DPTR
        Cjne   A,#0, keyin_200
        Mov     Keypad_dphh,Keypad_dphh1
        Mov     Keypad_dp11,Keypad_dp111
        Mov     R0, keypad_curpos
        Call    LCD_Cursor_Position
        Mov     DPH,Keypad_dphh
        Mov     DPL,Keypad_dp11
        Clr     A
        Movc   A,@A+DPTR
        Call    LCD_Data
        Call    LCD_Blink_On
        Mov     R0, keydelay
        Call    Delay_Var_10ms
        Jump   keyin100
keyin_200:
        Mov     Keypad_dphh,DPH
        Mov     Keypad_dp11,DPL
        Mov     B,A

```

---

```

    Mov     R0, keypad_curpos
    Call    LCD_Cursor_Position
    Mov     A, B
    Call    LCD_Data
    Call    LCD_Blink_On
    Mov     R0, keydelay
    Call    Delay_Var_10ms
    Jmp     keyin100

;-----
keyin_end:

    Call    LCD_Blink_On
    Pop     DPL
    Pop     DPH
    Ret

EndSub
Subroutine LCD_Clrscr:
    Push    ACC
    Mov     A, #01h
    Call    LCD_Command
    Pop     ACC
EndSub
;*****
; subrutin menempatkan kursor
;*****
Subroutine LCD_Cursor_Position:
    Push    ACC
    Mov     A, R0
    Anl     A, #0F0h
    Cjne    A, #10h, lcd_cursor_position1
    Mov     A, R0
    Anl     A, #0Fh
    Orl     A, #80h
    Call    LCD_Command
    Jmp     lcd_cursor_position_end
lcd_cursor_position1:
    Cjne    A, #20h, lcd_cursor_position_end
    Mov     A, R0
    Anl     A, #0Fh
    Orl     A, #0C0h
    Call    LCD_Command
lcd_cursor_position_end:
    Pop     ACC
EndSub
;*****
; kirim data ke lcd1
;*****
Subroutine LCD_Data:
    Push    DPL
    Push    DPH
    Push    ACC
    Mov     DPTR, #lcd1
    Movx    @DPTR, A

```

```

        Call    LCD_Delay
        Lcall   LCD_Delay
        Lcall   LCD_Delay
        Lcall   LCD_Delay
        Pop     ACC
        Pop     DPH
        Pop     DPL
EndSub
;*****
; kirim instruksi ke lcd1
;*****
lcd0    Equ    0a000h    ;lcd control operation
lcd1    Equ    lcd0+1    ;lcd data operation
Subroutine LCD_Command:
        Push   DPL
        Push   DPH
        Mov    DPTR,#lcd0
        Movx   @DPTR,A
        Lcall  LCD_Delay
        Lcall  LCD_Delay
        Pop    DPH
        Pop    DPL
        Ret
EndSub
Subroutine LCD_Delay:
        Push   01h
        Push   02h
        Push   03h
        Mov    3,#5
tunda1qqqa:
        Mov    1,#30
        Djnz   1,$
        Mov    2,#40
        Djnz   2,$
        Mov    2,#1
        Djnz   2,$
        Djnz   3,tunda1qqqa
        Pop    03h
        Pop    02h
        Pop    01h
EndSub
Subroutine LCD_Delete_Line1:
        Push   ACC
        Push   DPH
        Push   DPL
        Call   LCD_Line1
        Mov    DPTR,#txt_blank
        Call   LCD_String
        Call   LCD_Line1
        Pop    DPL
        Pop    DPH
        Pop    ACC
EndSub
Subroutine LCD_Delete_Line2:
        Push   ACC
        Push   DPH

```

```

        Push    DPL
        Call    LCD_Line2
        Mov     DPTR,#txt_blank
        Call    LCD_String
        Call    LCD_Line2
        Pop     DPL
        Pop     DPH
        Pop     ACC
        Ret

txt_blank:   Db      '          ',0
EndSub

Subroutine LCD_Initialization:
    Mov     A,#38h      ;function set 8 bit
    Lcall   LCD_Command
    Mov     A,#0Ch      ;display on, cursor off, blink off
    Lcall   LCD_Command
    Mov     A,#06h      ;increment, no display shift
    Lcall   LCD_Command
    Mov     A,#01h      ;clear display
    Lcall   LCD_Command
EndSub

Subroutine LCD_Line1:
    Push    ACC
    Mov     A,#R0h
    Lcall   LCD_Command
    Pop     ACC
EndSub

Subroutine LCD_Line2:
    Push    ACC
    Mov     A,#0C0h
    Lcall   LCD_Command
    Pop     ACC
EndSub

Subroutine LCD_Off:
    Push    ACC
    Mov     A,#00001000b
    Lcall   LCD_Command
    Pop     ACC
EndSub

Subroutine LCD_On:
    Push    ACC
    Mov     A,#00001100b
    Lcall   LCD_Command
    Pop     ACC
EndSub

;-----
Subroutine LCD_String:
    Push    ACC
    Push    DPL
    Push    DPH

getcar1:
    Clr     A           ; mengambil data dari epron
    Movc    A,@A+DPTR
    Cjne    A,#0,tampill ; tes apakah data habis?
    Ljmp    mettul

tampill:   ; keluarkan data ke lcd

```

```

        Call    LCD_Data
        Inc     DPTR          ; naikkan dptr
        Ljmp   getcarl
mettul:
        Pop    DPH
        Pop    DPL
        Pop    ACC
EndSub
;*****
; subrutin tampil text pada lcd1
; dari data di ram
;*****
Subroutine LCD_String_Ram:
        Push   ACC
        Push   DPL
        Push   DPH
        Push   00h
getcarlz:
        Movx   A,@DPTR
        Cjne   A,#0,tampilz   ; tes apakah data habis?
        Ljmp   mettulz
tampilz:
        ; keluarkan data ke lcd
        Call   LCD_Data
        Inc    DPTR          ; naikkan dptr
        Ljmp   getcarlz

mettulz:
        Pop    00h
        Pop    DPH
        Pop    DPL
        Pop    ACC
EndSub
Subroutine LCD_Blink_Off:
        Push   ACC
        Mov    A,#00001100b   ;
        Lcall  LCD_Command
        Pop    ACC
EndSub
Subroutine LCD_Blink_On:
        Push   ACC
        Mov    A,#00001101b   ;
        Lcall  LCD_Command
        Pop    ACC
EndSub
Subroutine LCD_Cursor_Off:
        Push   ACC
        Mov    A,#00001100b   ; cursor on
        Lcall  LCD_Command
        Pop    ACC
EndSub
Subroutine LCD_Cursor_On:
        Push   ACC
        Mov    A,#00001110b   ; cursor on
        Lcall  LCD_Command
        Pop    ACC

```

```

EndSub
Subroutine LCD_Cursor_Left:
    Push    ACC
    Mov     A,#00010000b    ; cursor left
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Cursor_Right:
    Push    ACC
    Mov     A,#00010100b    ; cursor left
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Display_Left:
    Push    ACC
    Mov     A,#00011000b    ; display left
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Display_Right:
    Push    ACC
    Mov     A,#00011100b    ; display right
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Cursor_Home:
    Push    ACC
    Mov     A,#00000010b    ; display right
    Lcall   LCD_Command
    Pop     ACC
EndSub
Subroutine LCD_Hexa:
    Push    07h
    Push    ACC

    Mov     7,A
    Anl     A,#0F0h
    Swap    A
    Orl     A,#30h
    Lcall   Tes_Huruf_
    Call    LCD_Data
    Mov     A,7
    Anl     A,#0Fh
    Orl     A,#30h
    Lcall   Tes_Huruf_
    Lcall   LCD_Data

    Pop     ACC
    Pop     07h
    Ret

Tes_Huruf_:
    Cjne    A,#3Ah,tes_huruf_1
    Mov     A,#'A'
    Ret
tes_huruf_1:

```

---



```

        Cjne    A,#3Bh,tes_huruf_2
        Mov     A,#'B'
        Ret
tes_huruf_2:
        Cjne    A,#3Ch,tes_huruf_3
        Mov     A,#'C'
        Ret
tes_huruf_3:
        Cjne    A,#3Dh,tes_huruf_4
        Mov     A,#'D'
        Ret
tes_huruf_4:
        Cjne    A,#3Eh,tes_huruf_5
        Mov     A,#'E'
        Ret
tes_huruf_5:
        Cjne    A,#3Fh,tes_huruf_6
        Mov     A,#'F'
        Ret
tes_huruf_6:
        Ret
EndSub
ppi_porta    Equ    0e000h
ppi_portb    Equ    0e001h
ppi_portc    Equ    0e002h
ppi_portcw   Equ    0e003h

Subroutine PPI_Get_Port_A:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_porta
    Movx    A,@DPTR
    Pop     DPL
    Pop     DPH
EndSub
Subroutine PPI_Get_Port_B:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_portb
    Movx    A,@DPTR
    Pop     DPL
    Pop     DPH
EndSub
Subroutine PPI_Get_Port_C:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_portc
    Movx    A,@DPTR
    Pop     DPL
    Pop     DPH
EndSub
Subroutine PPI_Initialization:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_portcw
    Movx    @DPTR,A        ; kirim ke port control ppi 0255

```

```

        Pop    DPL
        Pop    DPH
EndSub
Subroutine PPI_Send_Port_A:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_porta
    Movx   @DPTR,A
    Pop     DPL
    Pop     DPH
EndSub
Subroutine PPI_Send_Port_B:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_portb
    Movx   @DPTR,A
    Pop     DPL
    Pop     DPH
EndSub
Subroutine PPI_Send_Port_C:
    Push    DPH
    Push    DPL
    Mov     DPTR,#ppi_portc
    Movx   @DPTR,A
    Pop     DPL
    Pop     DPH
EndSub
Subroutine Serial_Receive:
    Jnb    RI,$
    Clr    RI
    Mov    A,SBUF
    Clr    RI
EndSub
Subroutine Serial_Transmit:
    Mov    SBUF,A
    Jnb    TI,$
    Clr    TI
EndSub
Subroutine KonversiKey:
KonversiKey1:
    Cjne   A,#0C,KonversiKey2
    Mov    A,#'1'
    Ret
KonversiKey2:
    Cjne   A,#01,KonversiKey3
    Mov    A,#'2'
    Ret
KonversiKey3:
    Cjne   A,#02,KonversiKey4
    Mov    A,#'3'
    Ret
KonversiKey4:
    Cjne   A,#04,KonversiKey5
    Mov    A,#'4'
    Ret
KonversiKey5:

```

---

```

        Cjne    A,#05,KonversiKey6
        Mov     A,#'5'
        Ret
KonversiKey6:
        Cjne    A,#06,KonversiKey7
        Mov     A,#'6'
        Ret
KonversiKey7:
        Cjne    A,#08,KonversiKey8
        Mov     A,#'7'
        Ret
KonversiKey8:
        Cjne    A,#09,KonversiKey9
        Mov     A,#'8'
        Ret
KonversiKey9:
        Cjne    A,#10,KonversiKey10
        Mov     A,#'9'
        Ret
KonversiKey10:
        Cjne    A,#00001101b,KonversiKey11
        Mov     A,#'0'
        Ret
KonversiKey11:
        Cjne    A,#00001110b,KonversiKeyxx
        Mov     A,#'E'
        Ret
KonversiKeyxx:
        Mov     A,#0
EndSub

```

Subroutine EEPROM\_Write:

```

RAM_Write:
        Push    01
        Mov     01,A
save_03:
        Movx   @DPTR,A
save_01:
        Movx   A,@DPTR
        Cjne   A,01h,save_01
        Pop    01
EndSub

```

Subroutine EEPROM\_Read:

```

RAM_Read:
        Movx   A,@DPTR
EndSub

```

```

rtc_buf1gl      Equ  0010h
rtc_buf1bln     Equ  0011h
rtc_buf1thn     Equ  0012h
rtc_buf1det     Equ  0013h
rtc_buf1min     Equ  0014h
rtc_buf1jam     Equ  0015h
rtc_buf1har     Equ  0016h

```

```

;*****
; subrutin cek uip rtc
;*****
Subroutine RTC_Cek_UIP:
    Mov     DPTR,#regartc
rtc_uip_ssibuk:
    Movx    A,@DPTR
    Jb     ACC.7,rtc_uip_ssibuk
EndSub

Subroutine RTC_Delay0:
    Push    03
    Push    02
    Push    01

    Mov     3,#1
rtc_tundalo:
    Mov     1,#01h
    Djnz   1,$
    Mov     2,#01h
    Djnz   2,$
    Djnz   3,rtc_tundalo
    Pop     01
    Pop     02
    Pop     03
EndSub
;*****
;*****
; rutin baca rtc (mengambil data dari rtc & menyimpannya di
; accumulator)
;*****
;*****
Subroutine RTC_GetTime:
    Push    DPL
    Push    DPH
    Push    ACC

bacartc:
bacadet:
    Lcall   RTC_Cek_UIP
    Lcall   RTC_Delay0
    Mov     DPTR,#detrtc           ;baca detik rtc
    Movx    A,@DPTR
    Mov     DPTR,#rtc_bufdet      ; pindahkan ke buffer
sementara
    Mov     TmpDetik,A
    Call    RAM_Write
bacamnt:
    Lcall   RTC_Cek_UIP
    Lcall   RTC_Delay0
    Mov     DPTR,#mntrtc
    Movx    A,@DPTR
    Mov     DPTR,#rtc_bufmin
    Mov     TmpMenit,A
    Call    RAM_Write
bacaJam:

```

```

        Lcall    RTC_Cek_UI?
        Lcall    RTC_Delay0
        Mov     DPTR,#jamrtc
        Movx    A,@DPTR
        Mov     DPTR,#rtc_bufjam
        Mov     ImpJam,A
        Call    RAM_Write

        Pop     ACC
        Pop     DPH
        Pop     DPL
        Ret

EndSub
;*****
; inisialisasi rtc
;*****
Subroutine RTC_Initialization:
        Push    DPL
        Push    DPH
        Push    ACC

        Mov     DPTR,#regartc
        Mov     A,R0
        Movx    @DPTR,A
        Mov     DPTR,#regbrtc
        Mov     A,R1
        Movx    @DPTR,A

        Pop     ACC
        Pop     DPH
        Pop     DPL
        Ret

EndSub
;*****
; menjalankan rtc
;-----
Subroutine RTC_Run:
        Push    DPL
        Push    DPH
        Push    ACC

        Mov     DPTR,#regbrtc
        Mov     A,#02h           ;run count
        Movx    @DPTR,A

        Pop     ACC
        Pop     DPH
        Pop     DPL

EndSub
Subroutine RTC_Stop:
        Push    DPL
        Push    DPH
        Push    ACC

        Mov     DPTR,#regbrtc
        Mov     A,#82h           ;stop count

```

```

        Movx    @DPTR,A

        Pop     ACC
        Pop     DPH
        Pop     DPL
        Ret

EndSub
;*****
; subrutin pindah new setting jam
;*****
Subroutine RTC_SetTime:
    Push     DPL
    Push     DPH
    Push     ACC

    Mov      DPTR,#regbrtc
    Mov      A,#82h           ;stop count

    Mov      DPTR,#detrtc
    Mov      A,R2
    Movx     @DPTR,A

    Mov      DPTR,#mntrtc
    Mov      A,R1
    Movx     @DPTR,A

    Mov      DPTR,#jamrtc
    Mov      A,R0
    Movx     @DPTR,A

    Mov      DPTR,#regbrtc
    Mov      A,#02h         ;run count
    Movx     @DPTR,A

    Pop      ACC
    Pop      DPH
    Pop      DPL
    Ret

EndSub
;*****
; subrutin tampil jam:menit:detik lcd1
;*****
Subroutine RTC_ShowTime:
    Push     DPL
    Push     DPH
    Push     ACC

rtc_sacajam1:
    ;Mov     DPTR,#rtc_bufjam
    ;Call    RAM_Read
    ;Mov     06h,A
    ;Anl    A,#0F0h
    ;Swap   A
    ;Add    A,#30h
    ;Call   LCD_Data
    ;Mov    A,06h
    ;Anl   A,#0Fh

```

```

;Add    A,#30h
;Call   LCD_Data
;Mov    A,#': '
;Call   LCD_Data
rtc_sacamnt1:
Mov     DPTR,#rtc_bufmir
Call    RAM_Read
Mov     06h,A
Anl    A,#0F0h
Swap   A
Add    A,#30h
Call   LCD_Data
Mov    A,06h
Anl    A,#0Fh
Add    A,#30h
Call   LCD_Data
Mov    A,#': '
Call   LCD_Data
rtc_sacadet1:
Mov     DPTR,#rtc_bufdet
Call    RAM_Read
Mov     06h,A
Anl    A,#0F0h
Swap   A
Add    A,#30h
Call   LCD_Data
Mov    A,06h
Anl    A,#0Fh
Add    A,#30h
Call   LCD_Data
Pop    ACC
Pop    DPH
Pop    DPL
Ret
EndSub

```

```

;*****
; hari-hari
;*****
txt_minggu:      Db    'Min',0
txt_senin:       Db    'Sen',0
txt_selasa:      Db    'Sel',0
txt_rabu:        Db    'Rab',0
txt_kamis:       Db    'Kam',0
txt_jumat:       Db    'Jum',0
txt_sabtu:       Db    'Sab',0
txt_januari:     Db    'Jan',0
txt_februari:    Db    'Feb',0
txt_maret:       Db    'Mar',0
txt_april:       Db    'Apr',0
txt_mei:         Db    'Mei',0
txt_juni:        Db    'Jun',0
txt_juli:        Db    'Jul',0
txt_agustus:     Db    'Agt',0
txt_september:   Db    'Sep',0
txt_oktober:     Db    'Okt',0

```

---

```
txt_nopember:    Db    'Nop',0
txt_dezember:   Db    'Des',0
```

```
;----- DELAY -----
Subroutine Delay_Var_1ms:
    Call    Delay_Fix_1ms
    Djnz   R0,Delay_Var_1ms
EndSub
Subroutine Delay_Var_10ms:
    Call    Delay_Fix_10ms
    Djnz   R0,Delay_Var_10ms
EndSub
Subroutine Delay_Var_100ms:
    Call    Delay_Fix_1ms
    Djnz   R0,Delay_Var_100ms
EndSub
Subroutine Delay_Var_1s:
    Call    Delay_Fix_1s
    Djnz   R0,Delay_Var_1s
EndSub
Subroutine Delay_Var_10s:
    Call    Delay_Fix_10s
    Djnz   R0,Delay_Var_10s
EndSub
Subroutine Delay_Var_10us:
    Call    Delay_Fix_10us
    Djnz   R0,Delay_Var_10us
EndSub
Subroutine Delay_Fix_10us:
    Push    1
    Mov     1,#20
    Djnz   1,$
    Pop     1
EndSub
Subroutine Delay_Fix_10s:
    Push    1
    Mov     1,#100
delay_fix_10s_1:
    Call    Delay_Fix_100ms
    Djnz   1,delay_fix_10s_1
    Pop     1
EndSub
Subroutine Delay_Fix_1s:
    Push    1
    Mov     1,#100
delay_fix_1000ms_1:
    Call    Delay_Fix_10ms
    Djnz   1,delay_fix_1000ms_1
    Pop     1
EndSub
Subroutine Delay_Fix_100ms:
    Push    1
    Mov     1,#10
delay_fix_100ms_1:
```



```

    Call    Delay_Fix_10ms
    Cjnz   1,delay_fix_100ms_1
    Pop    1
EndSub
Subroutine Delay_Fix_10ms:
    Mov    TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
    Mov    TLO,#3Dh    ; siapkan waktu tunda 50 mili-detik
    Mov    TH0,#0B0h
    Clr    TFO    ; me-nol-kan bit limpahan
    Setb   TR0    ; timer mulai bekerja
    Jnb    TFO,$    ; tunggu di sini sampai melimpah
    Clr    TR0    ; timer berhenti kerja
    Ret
EndSub
Subroutine Delay_Fix_1ms:
    Mov    TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
    Mov    TLO,#0EDh    ; siapkan waktu tunda 50 mili-detik
    Mov    TH0,#70h
    Clr    TFO    ; me-nol-kan bit limpahan
    Setb   TR0    ; timer mulai bekerja
    Jnb    TFO,$    ; tunggu di sini sampai melimpah
    Clr    TR0    ; timer berhenti kerja
    Ret
EndSub
Subroutine Get_DPTR_Suhu:
    Push   B
    Mov    DPTR,#suhu0
    Mov    B,#6    ; jumlah antar lokasi nama
variabel = 12
    Mul    AB    ; kalikan a dengan b
    Add    A,DPL    ; jumlahkan a dengan dpl
    Mov    DPL,A    ; simpan ke dpl
    Mov    A,B    ; ambil b
    Adde   A,DPH
    Mov    DPH,A
    Pop    B
    Ret
EndSub

suhu0:    Db    0h,'0',0
suhu1:    Db    0h,'0,4',0
suhu2:    Db    1h,'0,8',0
suhu3:    Db    1h,'1,2',0
suhu4:    Db    2h,'1,6',0
suhu5:    Db    2h,'2',0
suhu6:    Db    2h,'2,4',0
suhu7:    Db    3h,'2,7',0
suhu8:    Db    3h,'3,1',0
suhu9:    Db    4h,'3,5',0
suhu10:   Db    4h,'3,9',0
suhu11:   Db    4h,'4,3',0
suhu12:   Db    5h,'4,7',0
suhu13:   Db    5h,'5,1',0
suhu14:   Db    6h,'5,5',0

```

---

suhu15:	Db	6h, '5,9',0
suhu16:	Db	6h, '6,3',0
suhu17:	Db	7h, '6,7',0
suhu18:	Db	7h, '7,1',0
suhu19:	Db	8h, '7,5',0
suhu20:	Db	8h, '7,8',0
suhu21:	Db	8h, '8,2',0
suhu22:	Db	9h, '8,6',0
suhu23:	Db	9h, '9',0
suhu24:	Db	9h, '9,4',0
suhu25:	Db	0Ah, '9,8',0
suhu26:	Db	0Ah, '10,2',0
suhu27:	Db	0Bh, '10,6',0
suhu28:	Db	0Bh, '11',0
suhu29:	Db	0Bh, '11,4',0
suhu30:	Db	0Ch, '11,8',0
suhu31:	Db	0Ch, '12,2',0
suhu32:	Db	0Ch, '12,5',0
suhu33:	Db	0Dh, '12,9',0
suhu34:	Db	0Dh, '13,3',0
suhu35:	Db	0Eh, '13,7',0
suhu36:	Db	0Eh, '14,1',0
suhu37:	Db	0Eh, '14,5',0
suhu38:	Db	0Fh, '14,9',0
suhu39:	Db	0Fh, '15,3',0
suhu40:	Db	10h, '15,7',0
suhu41:	Db	10h, '16,1',0
suhu42:	Db	10h, '16,5',0
suhu43:	Db	11h, '16,9',0
suhu44:	Db	11h, '17,3',0
suhu45:	Db	12h, '17,6',0
suhu46:	Db	12h, '18',0
suhu47:	Db	12h, '18,4',0
suhu48:	Db	13h, '18,8',0
suhu49:	Db	13h, '19,2',0
suhu50:	Db	14h, '19,6',0
suhu51:	Db	14h, '20',0
suhu52:	Db	14h, '20,4',0
suhu53:	Db	15h, '20,8',0
suhu54:	Db	15h, '21,2',0
suhu55:	Db	16h, '21,6',0
suhu56:	Db	16h, '22',0
suhu57:	Db	16h, '22,4',0
suhu58:	Db	17h, '22,7',0
suhu59:	Db	17h, '23,1',0
suhu60:	Db	18h, '23,5',0
suhu61:	Db	18h, '23,9',0
suhu62:	Db	18h, '24,3',0
suhu63:	Db	19h, '24,7',0
suhu64:	Db	19h, '25,1',0
suhu65:	Db	1Ah, '25,5',0
suhu66:	Db	1Ah, '25,9',0
suhu67:	Db	1Ah, '26,3',0
suhu68:	Db	1Bh, '26,7',0
suhu69:	Db	1Bh, '27,1',0
suhu70:	Db	1Ch, '27,5',0

---

suhu71:	Db	1Ch, '27,8',0
suhu72:	Db	1Ch, '28,2',0
suhu73:	Db	1Dh, '28,6',0
suhu74:	Db	1Dh, '29',0
suhu75:	Db	1Dh, '29,4',0
suhu76:	Db	1Eh, '29,8',0
suhu77:	Db	1Eh, '30,2',0
suhu78:	Db	1Fh, '30,6',0
suhu79:	Db	1Fh, '31',0
suhu80:	Db	1Fh, '31,4',0
suhu81:	Db	20h, '31,8',0
suhu82:	Db	20h, '32,2',0
suhu83:	Db	20h, '32,5',0
suhu84:	Db	21h, '32,9',0
suhu85:	Db	21h, '33,3',0
suhu86:	Db	22h, '33,7',0
suhu87:	Db	22h, '34,1',0
suhu88:	Db	22h, '34,5',0
suhu89:	Db	23h, '34,9',0
suhu90:	Db	23h, '35,3',0
suhu91:	Db	24h, '35,7',0
suhu92:	Db	24h, '36,1',0
suhu93:	Db	24h, '36,5',0
suhu94:	Db	25h, '36,9',0
suhu95:	Db	25h, '37,3',0
suhu96:	Db	26h, '37,6',0
suhu97:	Db	26h, '38',0
suhu98:	Db	26h, '38,4',0
suhu99:	Db	27h, '38,8',0
suhu100:	Db	27h, '39,2',0
suhu101:	Db	28h, '39,6',0
suhu102:	Db	28h, '40',0
suhu103:	Db	28h, '40,4',0
suhu104:	Db	29h, '40,8',0
suhu105:	Db	29h, '41,2',0
suhu106:	Db	2Ah, '41,6',0
suhu107:	Db	2Ah, '42',0
suhu108:	Db	2Ah, '42,4',0
suhu109:	Db	2Bh, '42,7',0
suhu110:	Db	2Bh, '43,1',0
suhu111:	Db	2Ch, '43,5',0
suhu112:	Db	2Ch, '43,9',0
suhu113:	Db	2Ch, '44,3',0
suhu114:	Db	2Dh, '44,7',0
suhu115:	Db	2Dh, '45,1',0
suhu116:	Db	2Eh, '45,5',0
suhu117:	Db	2Eh, '45,9',0
suhu118:	Db	2Eh, '46,3',0
suhu119:	Db	2Fh, '46,7',0
suhu120:	Db	2Fh, '47,1',0
suhu121:	Db	30h, '47,5',0
suhu122:	Db	30h, '47,8',0
suhu123:	Db	30h, '48,2',0
suhu124:	Db	31h, '48,5',0
suhu125:	Db	31h, '49',0
suhu126:	Db	31h, '49,4',0

---

suhu127:	Db	32h, '49, 8', 0
suhu128:	Db	32h, '50, 2', 0
suhu129:	Db	33h, '50, 6', 0
suhu130:	Db	33h, '51, ', 0
suhu131:	Db	33h, '51, 4', 0
suhu132:	Db	34h, '51, 8', 0
suhu133:	Db	34h, '52, 2', 0
suhu134:	Db	34h, '52, 6', 0
suhu135:	Db	35h, '52, 9', 0
suhu136:	Db	35h, '53, 3', 0
suhu137:	Db	36h, '53, 7', 0
suhu138:	Db	36h, '54, 1', 0
suhu139:	Db	36h, '54, 5', 0
suhu140:	Db	37h, '54, 9', 0
suhu141:	Db	37h, '55, 3', 0
suhu142:	Db	38h, '55, 7', 0
suhu143:	Db	38h, '56, 1', 0
suhu144:	Db	38h, '56, 5', 0
suhu145:	Db	39h, '56, 9', 0
suhu146:	Db	39h, '57, 3', 0
suhu147:	Db	3Ah, '57, 6', 0
suhu148:	Db	3Ah, '58, ', 0
suhu149:	Db	3Ah, '58, 4', 0
suhu150:	Db	3Bh, '58, 8', 0
suhu151:	Db	3Bh, '59, 2', 0
suhu152:	Db	3Ch, '59, 6', 0
suhu153:	Db	3Ch, '60, ', 0
suhu154:	Db	3Ch, '60, 4', 0
suhu155:	Db	3Dh, '60, 8', 0
suhu156:	Db	3Dh, '61, 2', 0
suhu157:	Db	3Eh, '61, 6', 0
suhu158:	Db	3Eh, '62, ', 0
suhu159:	Db	3Eh, '62, 4', 0
suhu160:	Db	3Fh, '62, 7', 0
suhu161:	Db	3Fh, '63, 1', 0
suhu162:	Db	40h, '63, 5', 0
suhu163:	Db	40h, '63, 9', 0
suhu164:	Db	40h, '64, 3', 0
suhu165:	Db	41h, '64, 7', 0
suhu166:	Db	41h, '65, 1', 0
suhu167:	Db	42h, '65, 5', 0
suhu168:	Db	42h, '65, 9', 0
suhu169:	Db	42h, '66, 3', 0
suhu170:	Db	43h, '66, 7', 0
suhu171:	Db	43h, '67, 1', 0
suhu172:	Db	44h, '67, 5', 0
suhu173:	Db	44h, '67, 9', 0
suhu174:	Db	44h, '68, 2', 0
suhu175:	Db	45h, '68, 6', 0
suhu176:	Db	45h, '69, ', 0
suhu177:	Db	45h, '69, 4', 0
suhu178:	Db	46h, '69, 8', 0
suhu179:	Db	46h, '70, 2', 0
suhu180:	Db	47h, '70, 6', 0
suhu181:	Db	47h, '71, ', 0
suhu182:	Db	47h, '71, 4', 0

---

suhu183:	Db	48h, '71, 8', 0
suhu184:	Db	48h, '72, 2', 0
suhu185:	Db	48h, '72, 5', 0
suhu186:	Db	49h, '72, 9', 0
suhu187:	Db	49h, '73, 3', 0
suhu188:	Db	4Ah, '73, 7', 0
suhu189:	Db	4Ah, '74, 1', 0
suhu190:	Db	4Ah, '74, 5', 0
suhu191:	Db	4Bh, '74, 9', 0
suhu192:	Db	4Bh, '75, 3', 0
suhu193:	Db	4Ch, '75, 7', 0
suhu194:	Db	4Ch, '76, 1', 0
suhu195:	Db	4Ch, '76, 5', 0
suhu196:	Db	4Dh, '76, 9', 0
suhu197:	Db	4Dh, '77, 3', 0
suhu198:	Db	4Eh, '77, 6', 0
suhu199:	Db	4Eh, '78', 0
suhu200:	Db	4Eh, '78, 4', 0
suhu201:	Db	4Fh, '78, 8', 0
suhu202:	Db	4Fh, '79, 2', 0
suhu203:	Db	50h, '79, 6', 0
suhu204:	Db	50h, '80', 0
suhu205:	Db	50h, '80, 4', 0
suhu206:	Db	51h, '80, 8', 0
suhu207:	Db	51h, '81, 2', 0
suhu208:	Db	52h, '81, 6', 0
suhu209:	Db	52h, '82', 0
suhu210:	Db	52h, '82, 4', 0
suhu211:	Db	53h, '82, 7', 0
suhu212:	Db	53h, '83, 1', 0
suhu213:	Db	54h, '83, 5', 0
suhu214:	Db	54h, '83, 9', 0
suhu215:	Db	54h, '84, 3', 0
suhu216:	Db	55h, '84, 7', 0
suhu217:	Db	55h, '85, 1', 0
suhu218:	Db	56h, '85, 5', 0
suhu219:	Db	56h, '85, 9', 0
suhu220:	Db	56h, '86, 3', 0
suhu221:	Db	57h, '86, 7', 0
suhu222:	Db	57h, '87, 1', 0
suhu223:	Db	58h, '87, 5', 0
suhu224:	Db	58h, '87, 8', 0
suhu225:	Db	58h, '88, 2', 0
suhu226:	Db	59h, '88, 6', 0
suhu227:	Db	59h, '89', 0
suhu228:	Db	59h, '89, 4', 0
suhu229:	Db	5Ah, '89, 8', 0
suhu230:	Db	5Ah, '90, 2', 0
suhu231:	Db	5Bh, '90, 6', 0
suhu232:	Db	5Bh, '91', 0
suhu233:	Db	5Bh, '91, 4', 0
suhu234:	Db	5Ch, '91, 8', 0
suhu235:	Db	5Ch, '92, 2', 0
suhu236:	Db	5Ch, '92, 5', 0
suhu237:	Db	5Dh, '92, 9', 0
suhu238:	Db	5Dh, '93, 3', 0

---

```

suhu239: Db 5Eh,'93,7',0
suhu240: Db 5Eh,'94,1',0
suhu241: Db 5Eh,'94,5',0
suhu242: Db 5Fh,'94,9',0
suhu243: Db 5Fh,'95,3',0
suhu244: Db 60h,'95,7',0
suhu245: Db 60h,'96,1',0
suhu246: Db 60h,'96,5',0
suhu247: Db 61h,'96,9',0
suhu248: Db 61h,'97,3',0
suhu249: Db 62h,'97,6',0
suhu250: Db 62h,'98 ',0
suhu251: Db 62h,'98,4',0
suhu252: Db 63h,'98,8',0
suhu253: Db 63h,'99,2',0
suhu254: Db 64h,'99,6',0
suhu255: Db 64h,'100 ',0

```

```

txt_menu1: Db '1.Oven masakan',0
txt_menu2: Db '2.Suhu & Waktu',0

; 0123456789abcdef
txt_suhu1: Db 'Suhu :---',0
txt_waktu1: Db 'Menit:--',0

; 0123456789abcdef
txt_isian: Db 'Suhu:--- Mnt:--',0
txt_yakin1: Db 'Yakin? 1-Y/2-T',0

txt_simpan: Db ' Data disimpan!',0

txt_masakan1: Db '1.Telur 2.Donat',0
txt_masakan2: Db '2.Piza 4.Kentang',0

txt_telur: Db 'Telur [----',0DFh,'C]',0
txt_donat: Db 'Donat [----',0DFh,'C]',0
txt_Pizza: Db 'Pizza [----',0DFh,'C]',0
txt_Kentang: Db 'Kentang[----',0DFh,'C]',0
; 0123456789abcdef
txt_Waktu: Db 'Waktu->00:00',0
txt_matang: Db 'Masakan Matang!',C
txt_Rendah: Db '----',0

```

---

## Features

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

## Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, 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 AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.



---

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

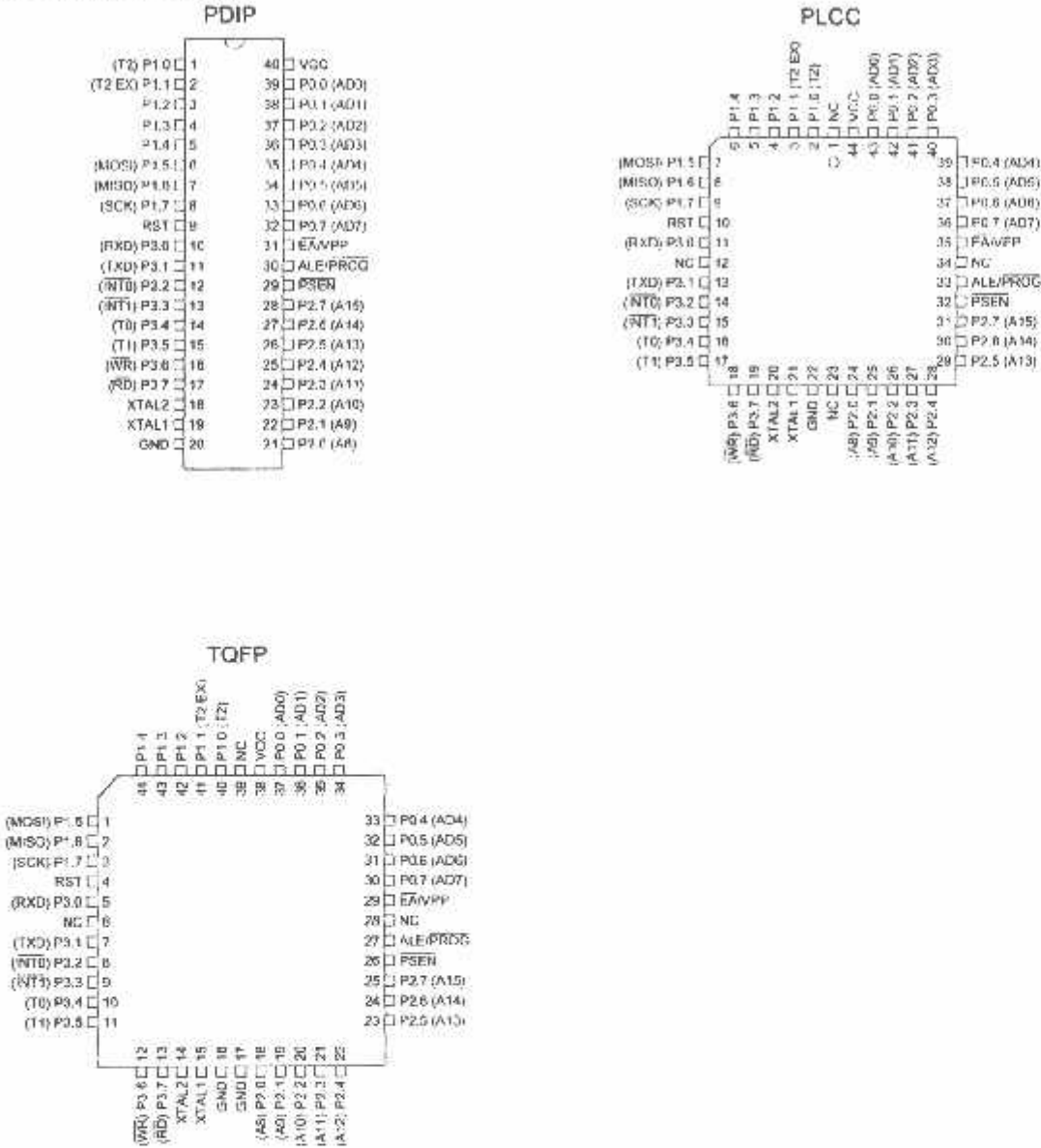
---

**AT89S52**

Rev. 1919A-07/01

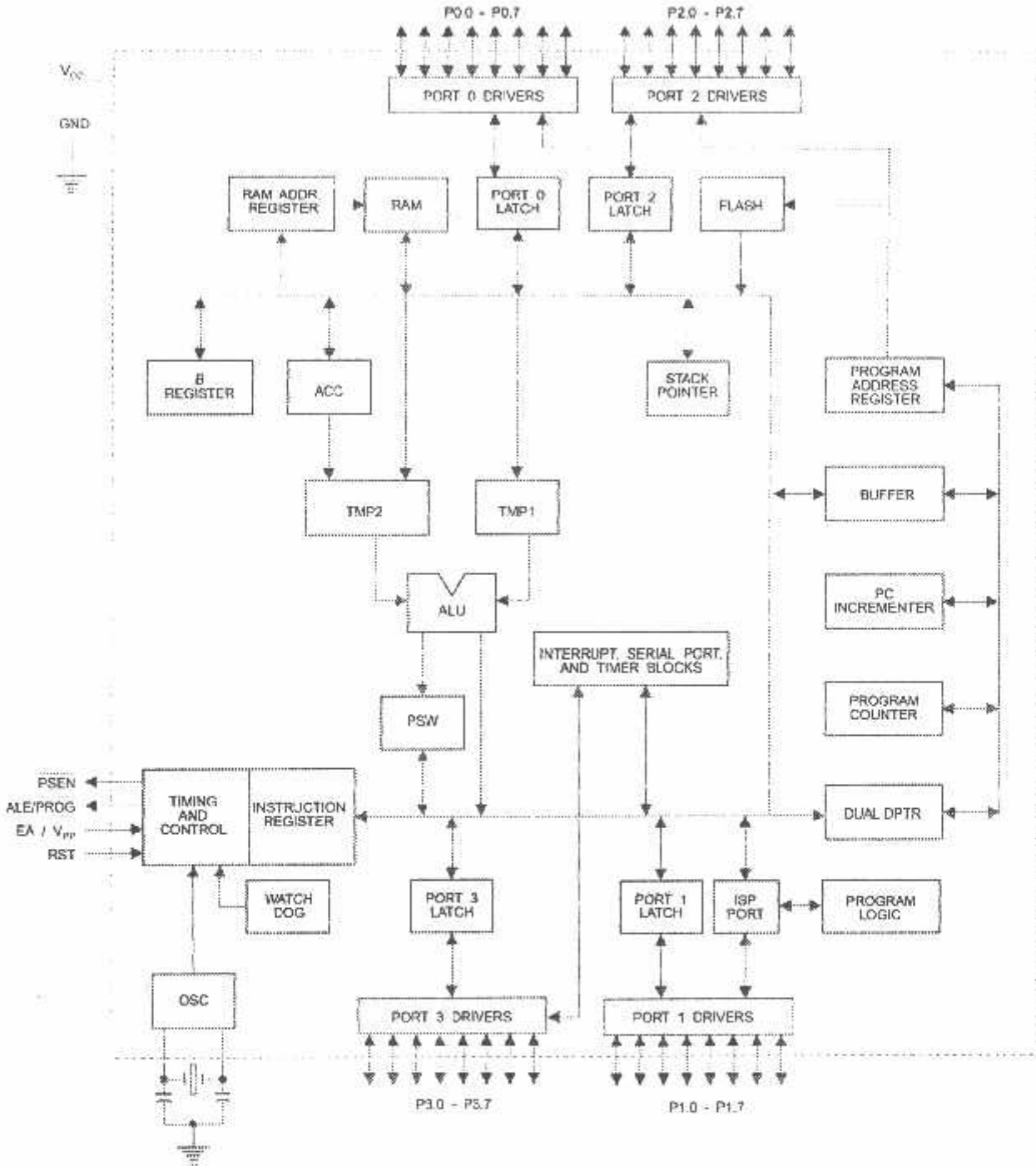


## Pin Configurations





Block Diagram



## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

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

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

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

### Port 1

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

In addition, 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, as shown in the following table.

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

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

### Port 2

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

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to

external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

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

### Port 3

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

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

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

### ALE/PROG

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

## 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 3) 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.

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

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

T2CON Address = 0C8H					Reset Value = 0000 0000B			
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T $\bar{2}$	CP/RL $\bar{2}$
	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 overflow 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/T $\bar{2}$	Timer or counter select for Timer 2. C/T $\bar{2}$ = 0 for timer function. C/T $\bar{2}$ = 1 for external event counter (falling edge triggered).
CP/RL $\bar{2}$	Capture/Reload select. CP/RL $\bar{2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL $\bar{2}$ = 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.

weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

### PSEN

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

When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

### EA/VPP

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH.

Note, however, that if lock bit 1 is programmed,  $\overline{\text{EA}}$  will be internally latched on reset.

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

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

### XTAL1

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

### XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00003000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXXX00X0		8FH
80H	P0 11111111	SP 000011	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XX0000	87H

Table 3a. AUXR: Auxillary Register

AUXR	Address = 8EH					Reset Value = XXX0XX0B			
	Not Bit Addressable								
	Bit	7	6	5	WDIDLE	DISRTO	2	1	DISALE
		-	-	-	4	3	-	-	0
-	Reserved for future expansion								
DISALE	Disable/Enable ALE								
	DISALE	Operating Mode							
	0	ALE is emitted at a constant rate of 1/6 the oscillator frequency							
	1	ALE is active only during a MOVX or MOVC instruction							
DISRTO	Disable/Enable Reset out								
	DISRTO								
	0	Reset pin is driven High after WDT times out							
	1	Reset pin is input only							
WDIDLE	Disable/Enable WDT in IDLE mode								
	WDIDLE								
	0	WDT continues to count in IDLE mode							
	1	WDT halts counting in IDLE mode							

**Dual Data Pointer Registers:** To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag:** The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by reset.

Table 3b. AUXR1: Auxillary Register 1

AUXR1	Address = A2H					Reset Value = XXXXXXX0B			
	Not Bit Addressable								
	Bit	7	6	5	4	3	2	1	DPS
		-	-	-	-	-	-	-	0
-	Reserved for future expansion								
DPS	Data Pointer Register Select								
	DPS								
	0	Selects DPTR Registers DP0L, DP0H							
	1	Selects DPTR Registers DP1L, DP1H							



## Memory Organization

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

### Program Memory

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

On the AT89S52, if EA is connected to  $V_{CC}$ , program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

### Data Memory

The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that 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 which use direct addressing access of the 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.



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

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 13-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

## Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 13-bit counter overflows when it reaches 8191 (1FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 8191 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is  $96 \times TOSC$ , where  $TOSC = 1/FOSC$ . To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

## WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

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

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

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

## UART

The UART in the AT89S52 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 '8051-Architecture Flash Microcontroller', then 'Product Overview'.

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 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 '8051-Architecture Flash Microcontroller', then 'Product Overview'.

## Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. 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.

Table 3. 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)

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.

### 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.

Figure 5. Timer in Capture Mode

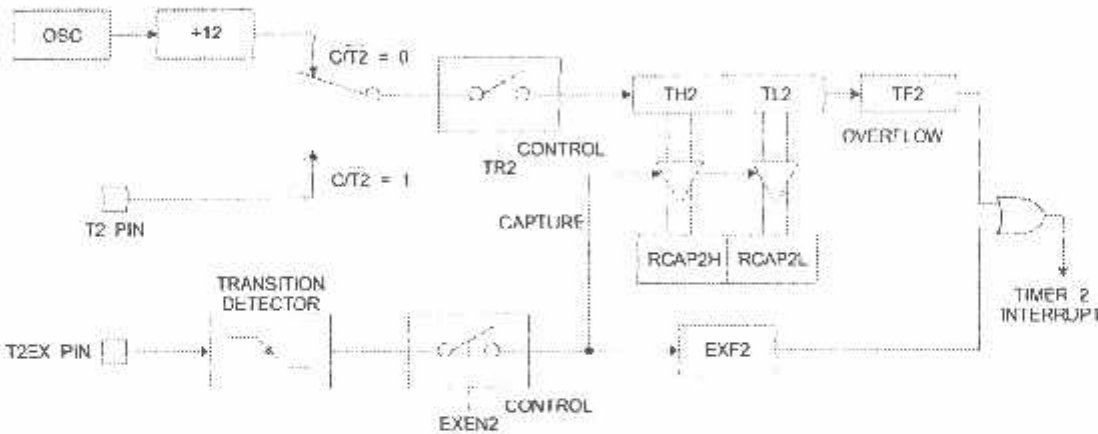


Figure 5 shows Timer 2 automatically counting up when DCEN=0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture Mode 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 6. In this mode, the T2EX pin controls

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 5.

### 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 4). 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.

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 6. Timer 2 Auto Reload Mode (DCEN = 0)

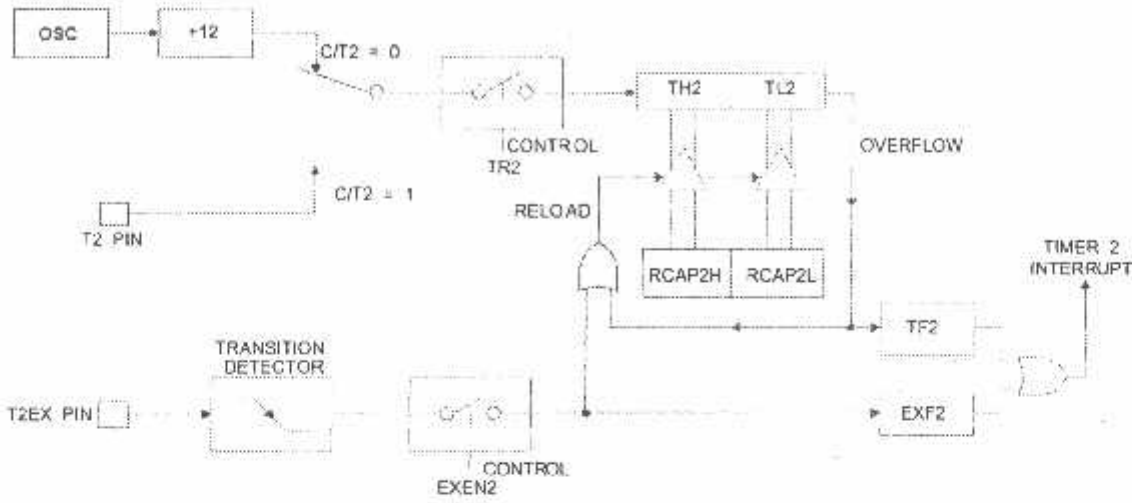


Table 4. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H							Reset Value = XXXX XX0CB	
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN
	-	-	-	-	-	-	1	0
Symbol	Function							
-	Not implemented, reserved for future							
T2OE	Timer 2 Output Enable bit							
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter							

Figure 7. Timer 2 Auto Reload Mode (DCEN = 1)

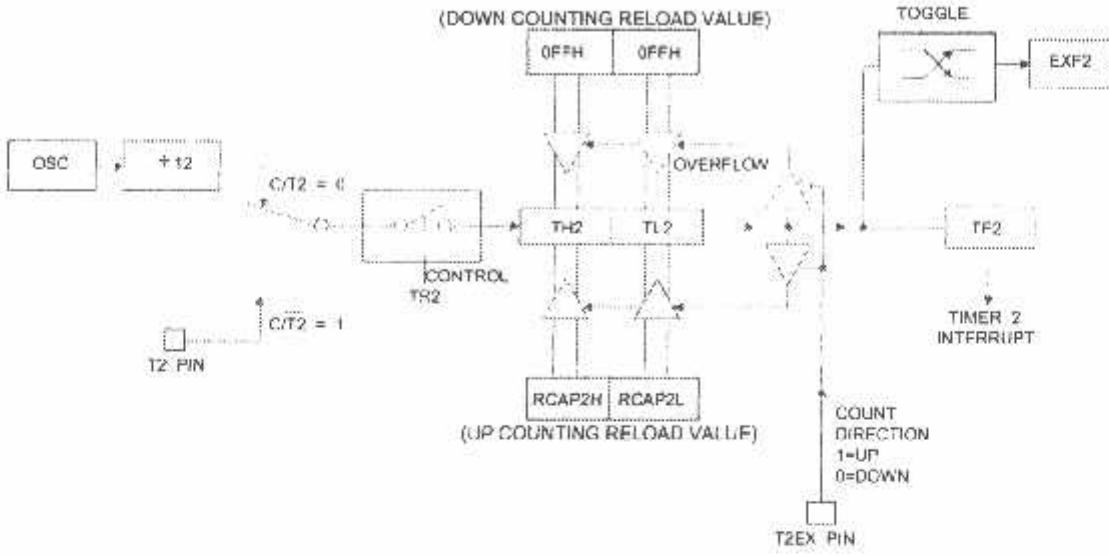
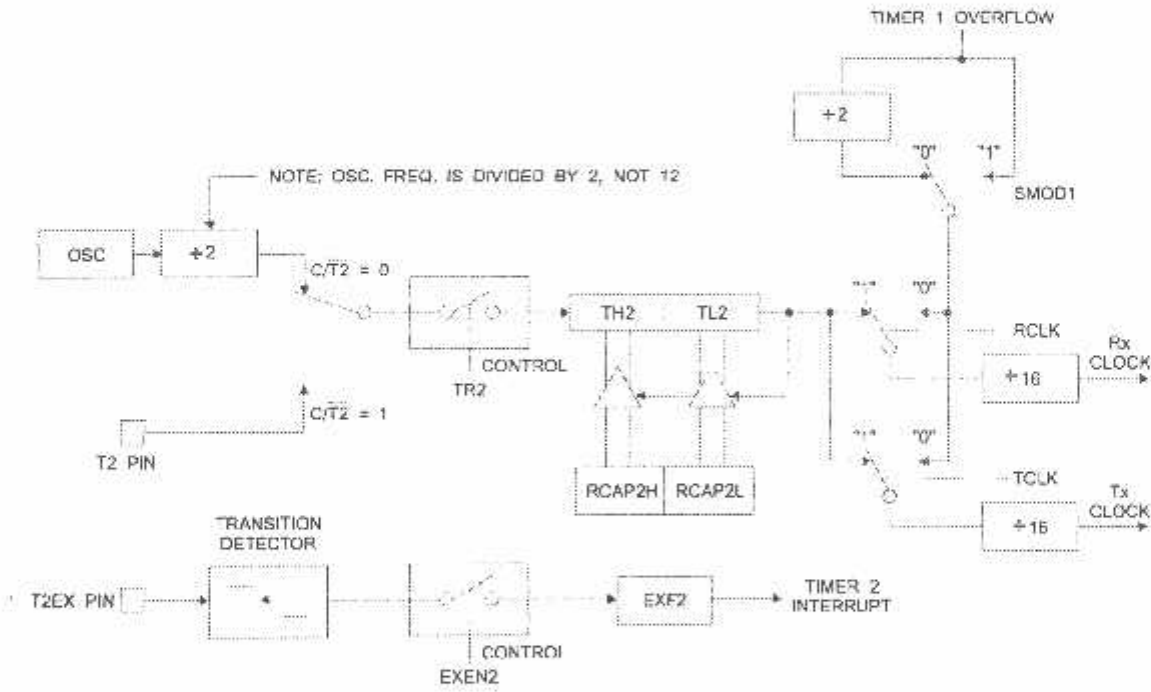


Figure 8. Timer 2 In Baud Rate Generator Mode





## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 9. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

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

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

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

In the clock-out mode, Timer 2 roll-overs 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.

## Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

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

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89S52, 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.

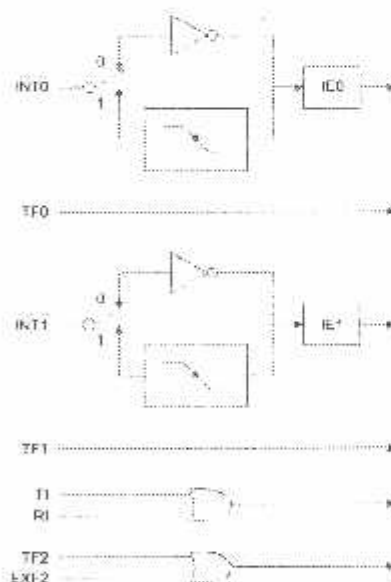
The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

**Table 5. Interrupt Enable (IE) Register**

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	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to 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.

**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.

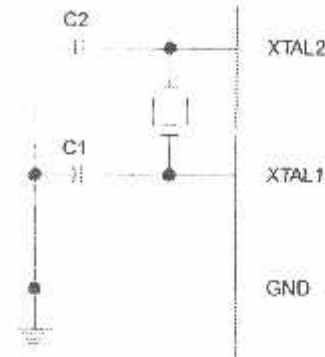
Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is 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.

**Power-down Mode**

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by 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

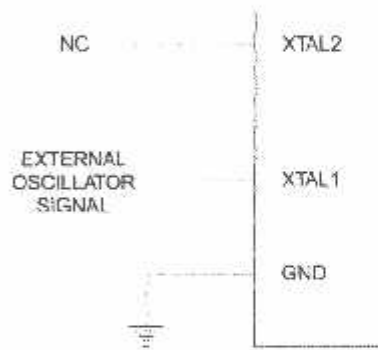
active long enough to allow the oscillator to restart and stabilize.

**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**



**Table 6. Status of External Pins During Idle and Power-down Modes**

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

## Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 7. Lock Bit Protection Modes

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

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

## Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

**Programming Algorithm:** Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise  $\overline{EA}/V_{pp}$  to 12V.
5. Pulse  $\overline{ALE}/\overline{PROG}$  once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50  $\mu$ s.

Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

**Data Polling:** The AT89S52 features  $\overline{Data}$  Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin.  $\overline{Data}$  Polling may begin any time after a write cycle has been initiated.

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

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

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

(000H) = 1EH indicates manufactured by Atmel  
 (100H) = 52H indicates 89S52  
 (200H) = 06H

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

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

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

## Programming the Flash – Serial Mode

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

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

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK)

frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

### Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
  - Apply power between VCC and GND pins.
  - Set RST pin to "H".
  - If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time by supplying the address and data together with the

appropriate Write instruction. The write cycle is self-timed and typically takes less than 1 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 device operation.

Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V<sub>CC</sub> power off.

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

### Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 10.



## Programming Interface – Parallel Mode

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

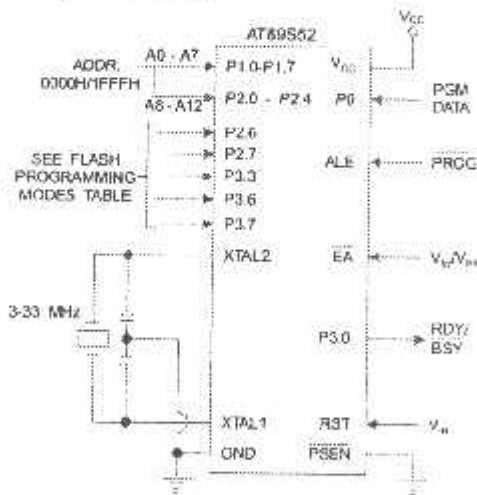
All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

**Table 8.** Flash Programming Modes

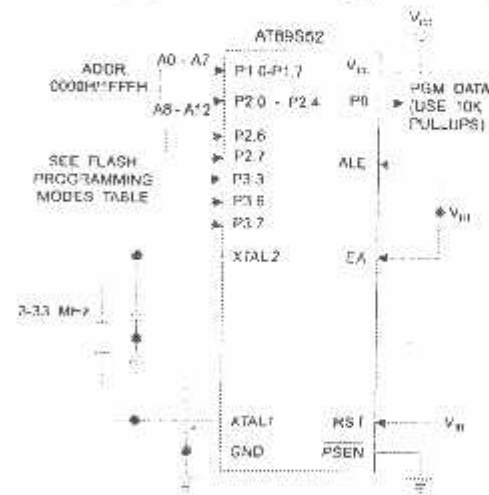
Mode	V <sub>CC</sub>	RST	PSEN	ALE/ PROG	EA/ V <sub>PP</sub>	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.4-0 P1.7-0	
												Address	
Write Code Data	5V	H	L		12V	L	H	H	H	H	D <sub>in</sub>	A12-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D <sub>out</sub>	A12-8	A7-0
Write Lock Bit 1	5V	H	L		12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L		12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L		12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	-	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L		12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	X 0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	52H	X 0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	X 0010	00H

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

**Figure 13.** Programming the Flash Memory (Parallel Mode)



**Figure 14.** Verifying the Flash Memory (Parallel Mode)





Flash Programming and Verification Characteristics (Parallel Mode)

T<sub>A</sub> = 20°C to 30°C, V<sub>CC</sub> = 4.5 to 5.5V

Symbol	Parameter	Min	Max	Units
V <sub>PP</sub>	Programming Supply Voltage	11.5	12.5	V
I <sub>PP</sub>	Programming Supply Current		10	mA
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		30	mA
f <sub>OSC(CL)</sub>	Oscillator Frequency	3	33	MHz
t <sub>AVGL</sub>	Address Setup to PROG Low	48t <sub>CLCL</sub>		
t <sub>GHAX</sub>	Address Hold After PROG	48t <sub>CLCL</sub>		
t <sub>AVGL</sub>	Data Setup to PROG Low	48t <sub>CLCL</sub>		
t <sub>GHDX</sub>	Data Hold After PROG	48t <sub>CLCL</sub>		
t <sub>EHS</sub>	P2.7 (ENABLE) High to V <sub>PP</sub>	48t <sub>CLCL</sub>		
t <sub>SHGL</sub>	V <sub>PP</sub> Setup to PROG Low	10		µs
t <sub>GHSL</sub>	V <sub>PP</sub> Hold After PROG	10		µs
t <sub>GLGH</sub>	PROG Width	0.2	1	µs
t <sub>AVDV</sub>	Address to Data Valid		48t <sub>CLCL</sub>	
t <sub>ELDV</sub>	ENABLE Low to Data Valid		48t <sub>CLCL</sub>	
t <sub>EQZ</sub>	Data Float After ENABLE	0	48t <sub>CLCL</sub>	
t <sub>CHRL</sub>	PROG High to BUSY Low		1.0	µs
t <sub>WC</sub>	Byte Write Cycle Time		50	µs

Figure 15. Flash Programming and Verification Waveforms – Parallel Mode

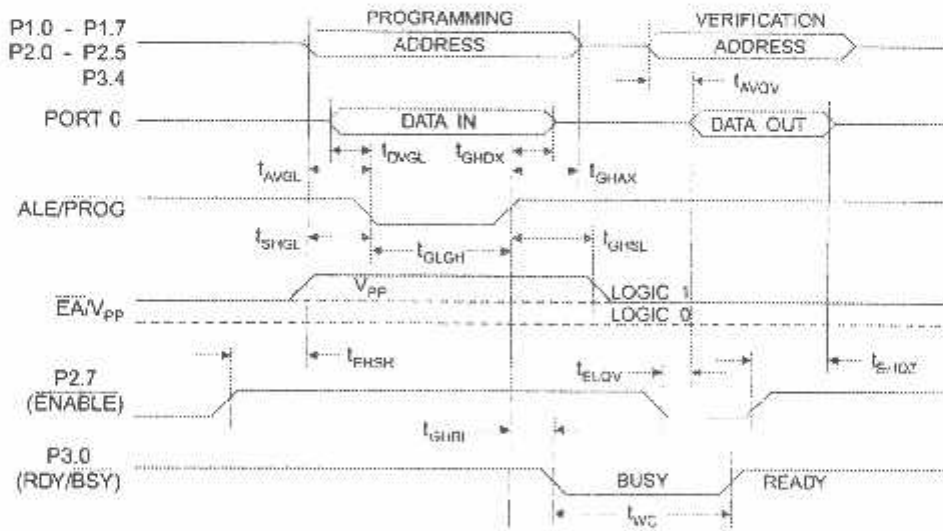
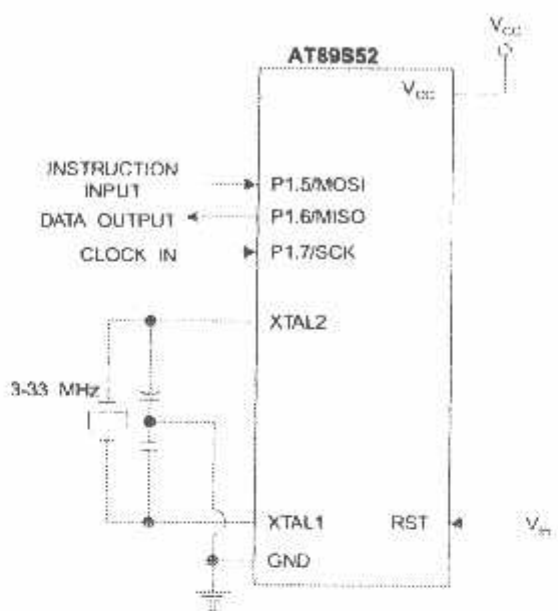


Figure 16. Flash Memory Serial Downloading



## Flash Programming and Verification Waveforms – Serial Mode

Figure 17. Serial Programming Waveforms

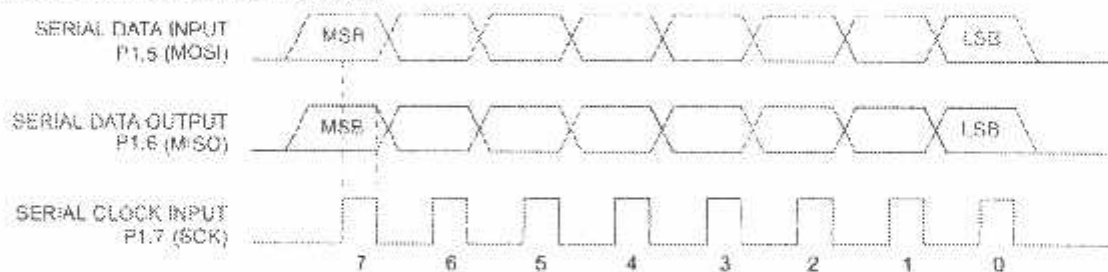


Table 9. Serial Programming Instruction Set

Instruction	Instruction Format		Byte 3	Byte 4	Operation
	Byte 1	Byte 2			
Programming Enable	1010 1100	0101 0011	XXXX XXXX	XXXX XXXX 0110 1001 (Output)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	XXXX XXXX	XXXX XXXX	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D0D0D0 D0D0D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D0D0D0 D0D0D0	Write data to Program memory in the byte mode
Write Lock Bits <sup>(2)</sup>	1010 1100	1110 00 B1 B0	XXXX XXXX	XXXX XXXX	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	XXXX XXXX	XXXX XXXX	XX 10 10 XX	Read back current status of the lock bits (a programmed lock bit reads back as a '1')
Read Signature Bytes <sup>(1)</sup>	0010 1000	xxx A5 A4 A3 A2 A1 A0	xxx xxxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

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

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

Each of the lock bits needs to be activated sequentially before Mode 4 can be executed.

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

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

## Serial Programming Characteristics

Figure 18. Serial Programming Timing

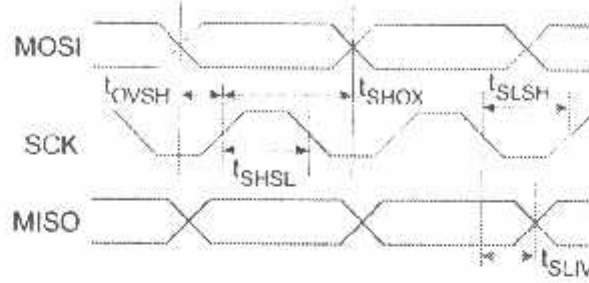


Table 10. Serial Programming Characteristics,  $T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{\text{CC}} = 4.0 - 5.5\text{V}$  (Unless otherwise noted)

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

**Absolute Maximum Ratings\***

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

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

**DC Characteristics**

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

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low Voltage	(Except EA)	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{LE1}$	Input Low Voltage (EA)		-0.5	$0.2 V_{CC} - 0.3$	V
$V_{IH}$	Input High Voltage	(Except XTAL1, RST)	$0.7 V_{CC} + 0.5$	$V_{CC} + 0.5$	V
$V_{IH1}$	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.5 \text{ mA}$		0.45	V
$V_{OL1}$	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
$V_{OH}$	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
$V_{OH1}$	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	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	$\mu\text{A}$
$I_{TI}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{L1}$	Input Leakage Current (Port 0, EA)	$0.45 < V_{IN} < V_{CC}$		110	$\mu\text{A}$
RRST	Reset Pulldown Resistor		10	30	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 <sup>(1)</sup>	$V_{CC} = 5.5\text{V}$		50

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





## AC Characteristics

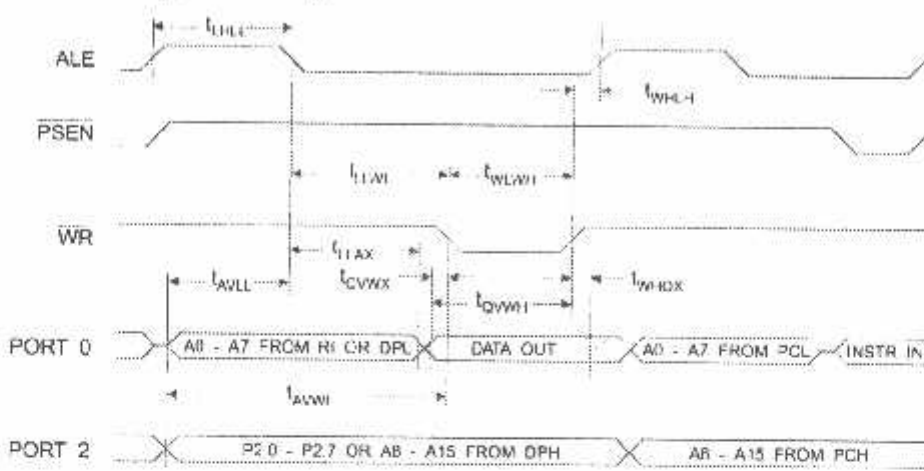
Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$ , and  $\overline{\text{PSEN}}$  = 100 pF; load capacitance for all other outputs = 80 pF.

### External Program and Data Memory Characteristics

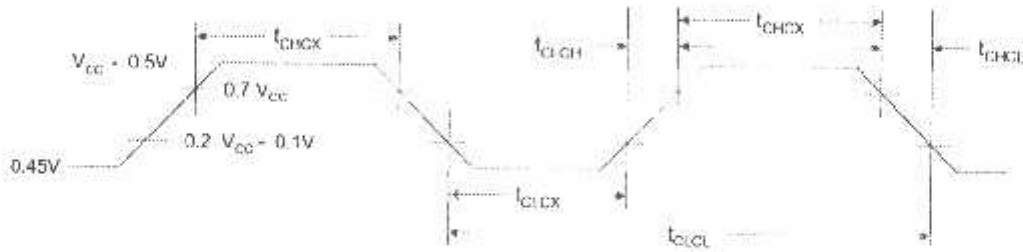
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$f_{\text{MCLK}}$	Oscillator Frequency			0	33	MHz
$t_{\text{HHL}}$	ALE Pulse Width	127		$2t_{\text{CLK}}-40$		ns
$t_{\text{AVL}}$	Address Valid to ALE Low	43		$t_{\text{CLK}}-25$		ns
$t_{\text{LLAX}}$	Address Hold After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



### External Data Memory Write Cycle



### External Clock Drive Waveforms



### External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
$t_{CLCL}$	Clock Period	30		ns
$t_{CHCX}$	High Time	12		ns
$t_{CLCX}$	Low Time	12		ns
$t_{CHCL}$	Rise Time		5	ns
$t_{CHCL}$	Fall Time		5	ns

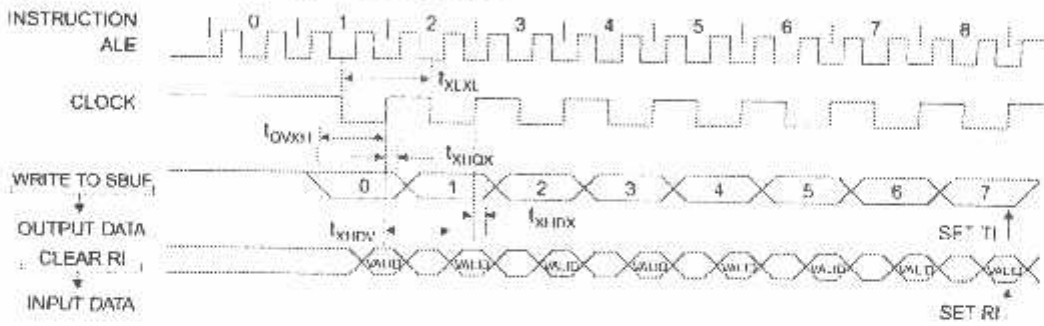


**Serial Port Timing: Shift Register Mode Test Conditions**

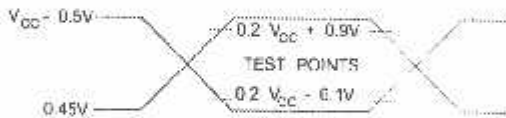
The values in this table are valid for  $V_{CC} = 4.0V$  to  $5.5V$  and Load Capacitance =  $80$  pF.

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

**Shift Register Mode Timing Waveforms**

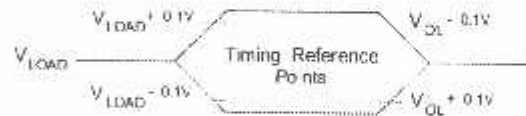


**AC Testing Input/Output Waveforms<sup>(1)</sup>**



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

**Float Waveforms<sup>(1)</sup>**



Note: 1. For timing purposes, a port pin is no longer floating when a  $100$  mV change from load voltage occurs. A port pin begins to float when a  $100$  mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.



## Ordering Information

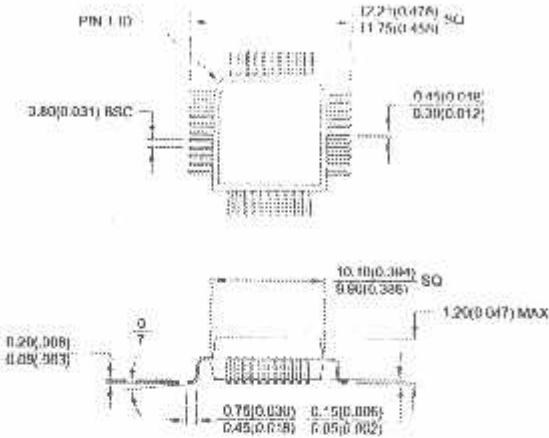
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC	44A	Commercial (0° C to 70° C)
		AT89S52-24JC	44J	
		AT89S52-24PC	40P6	
		AT89S52-24AI	44A	Industrial (-40° C to 85° C)
		AT89S52-24JI	44J	
		AT89S52-24PI	40P6	
33	4.5V to 5.5V	AT89S52-33AC	44A	Commercial (0° C to 70° C)
		AT89S52-33JC	44J	
		AT89S52-33PC	40P6	

= Preliminary Availability

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

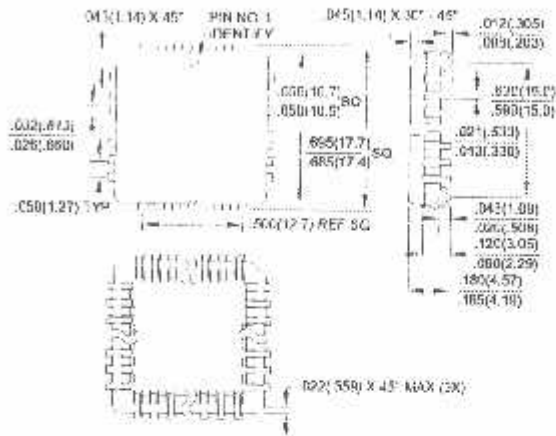
## Packaging Information

**44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)**  
Dimensions in Millimeters and (Inches)\*

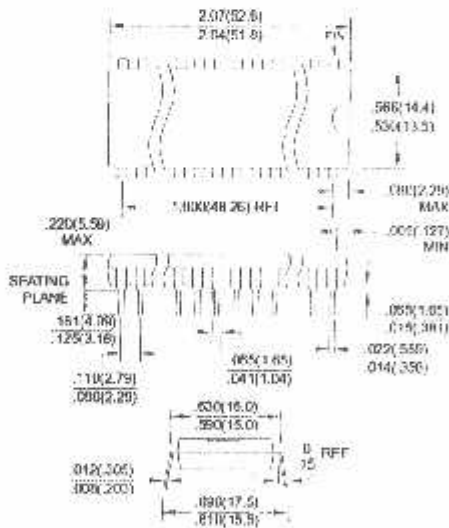


\*Controlling dimension: millimeters

**44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)**  
Dimensions in Inches and (Millimeters)



**40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)**  
Dimensions in Inches and (Millimeters)  
JEDEC STANDARD MS-011 AC





## Atmel Headquarters

**Corporate Headquarters**  
2325 Orchard Parkway  
San Jose, CA 95131  
TEL (408) 441-0311  
FAX (408) 487-2600

### Europe

Atmel SarL  
Route des Arsenaux 41  
Casa Postale 80  
CH-1705 Fribourg  
Switzerland  
TEL (41) 26-426-5555  
FAX (41) 26-426-5500

### Asia

Atmel Asia, Ltd.  
Room 1219  
Chinachem Golden Plaza  
77 Mody Road Tsimhatsui  
East Kowloon  
Hong Kong  
TEL (852) 2721-9778  
FAX (852) 2722-1369

### Japan

Atmel Japan K.K.  
9F, Tonetsu Shinkawa Bldg.  
1-24-8 Shinkawa  
Chuo-ku, Tokyo 104-0033  
Japan  
TEL (81) 3-3523-3551  
FAX (81) 3-3523-7581

## Atmel Product Operations

**Atmel Colorado Springs**  
1150 E. Cheyenne Mtn. Blvd.  
Colorado Springs, CO 80906  
TEL (719) 576-3300  
FAX (719) 540-1759

### Atmel Grenoble

Avenue de Rochepleine  
BP 123  
38521 Saint-Egreve Cedex, France  
TEL (33) 4-7658-3000  
FAX (33) 4-7658-3480

### Atmel Heilbronn

Theresienstrasse 2  
POB 3535  
D-74025 Heilbronn, Germany  
TEL (49) 71 31 67 25 94  
FAX (49) 71 31 67 24 23

### Atmel Nantes

La Chantrerie  
BP 70602  
44300 Nantes Cedex 3, France  
TEL (33) 0 2 40 18 18 18  
FAX (33) 0 2 40 18 19 60

### Atmel Rousset

Zone Industrielle  
13106 Rousset Cedex, France  
TEL (33) 4-4253-6000  
FAX (33) 4-4253-6001

### Atmel Smart Card ICs

Scottish Enterprise Technology Park  
East Kilbride, Scotland G75 0QR  
TEL (44) 1355-357-000  
FAX (44) 1355-242-743

---

### Fax-on-Demand

North America:  
1-(800) 292-8635  
International:  
1-(408) 441-0732

### e-mail

[literature@atmel.com](mailto:literature@atmel.com)

### Web Site:

<http://www.atmel.com>

### BBS

1-(408) 436-4309

### © Atmel Corporation 2001.

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

ATMEL® is the registered trademark of Atmel.

MCS-51® is the registered trademark of Intel Corporation. Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

Rev 1915A-07/01/04

## LM35 Precision Centigrade Temperature Sensors

### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 1/4^\circ\text{C}$  at room temperature and  $\pm 1/2^\circ\text{C}$  over a full  $-55$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60  $\mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55^\circ$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40^\circ$  to  $+110^\circ\text{C}$  range ( $-10^\circ$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear  $+10.0\text{ mV}/^\circ\text{C}$  scale factor
- $0.5^\circ\text{C}$  accuracy guaranteeable (at  $+25^\circ\text{C}$ )
- Rated for full  $-55^\circ$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60  $\mu\text{A}$  current drain
- Low self-heating,  $0.08^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 1/4^\circ\text{C}$  typical
- Low impedance output,  $0.1\ \Omega$  for 1 mA load

### Typical Applications

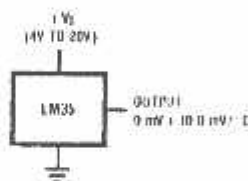
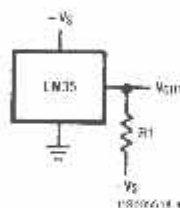


FIGURE 1. Basic Centigrade Temperature Sensor ( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )



Choose  $R_1 = -V_S/50\ \mu\text{A}$   
 $V_{OUT} = +1,500\text{ mV}$  at  $+150^\circ\text{C}$   
 $= +250\text{ mV}$  at  $+25^\circ\text{C}$   
 $= -550\text{ mV}$  at  $-55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

**Absolute Maximum Ratings** (Note 10)

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

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package,	
(Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V
Specified Operating Temperature Range: $T_{MIN}$ to $T_{MAX}$ (Note 2)	
LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

**Electrical Characteristics**

(Notes 1, B)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	$\pm 0.2$	$\pm 0.5$		$\pm 0.2$	$\pm 0.5$		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	$\pm 0.3$			$\pm 0.3$		$\pm 1.0$	$^\circ\text{C}$
	$T_A = T_{MAX}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		$^\circ\text{C}$
	$T_A = T_{MIN}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$		$\pm 1.5$	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{MIN} < T_A < T_{MAX}$	$\pm 0.18$		$\pm 0.35$	$\pm 0.15$		$\pm 0.3$	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} < T_A < T_{MAX}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 < I_L < 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		mV/mA
	$T_{MIN} < T_A < T_{MAX}$	$\pm 0.5$		$\pm 3.0$	$\pm 0.5$		$\pm 3.0$	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	$\pm 0.01$	$\pm 0.05$		$\pm 0.01$	$\pm 0.05$		mV/V
	$4\text{V} < V_S < 30\text{V}$	$\pm 0.02$		$\pm 0.1$	$\pm 0.02$		$\pm 0.1$	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^\circ\text{C}$	56	67		56	67		$\mu\text{A}$
	$V_S = +5\text{V}$	105		131	91		114	$\mu\text{A}$
	$V_S = +30\text{V}, +25^\circ\text{C}$	56.2	68		56.2	68		$\mu\text{A}$
	$V_S = +30\text{V}$	105.5		133	91.5		116	$\mu\text{A}$
Change of Quiescent Current (Note 3)	$4\text{V} < V_S < 30\text{V}, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		$\mu\text{A}$
	$4\text{V} < V_S < 30\text{V}$	0.5		2.0	0.5		2.0	$\mu\text{A}$
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	$^\circ\text{C}$
Long Term Stability	$T_A = T_{MAX}$ , for 1000 hours	$\pm 0.08$			$\pm 0.08$			$^\circ\text{C}$

## Connection Diagrams

**TO-46  
Metal Can Package\***



BOTTOM VIEW  
D800019-1

\*Case is connected to negative pin (GND)

Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH

See NS Package Number H03H

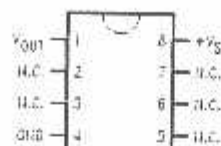
**TO-92  
Plastic Package**



BOTTOM VIEW  
D800019-2

Order Number LM35CZ,  
LM35CAZ or LM35DZ  
See NS Package Number Z03A

**SO-8  
Small Outline Molded Package**

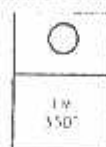


D800019-3

N.C. = No Connection

Top View  
Order Number LM35DM  
See NS Package Number M08A

**TO-220  
Plastic Package\***



D800019-4

\*Tab is connected to the negative pin (GND).

Note: The LM35DT pinout is different than the discontinued LM35DP.

Order Number LM35DT  
See NS Package Number TA03F

**Absolute Maximum Ratings** (Note 10)

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

Supply Voltage	+3.5V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp.:	
TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-220 Package,	-65°C to +150°C
Lead Temp.:	
TO-46 Package,	
(Soldering, 10 seconds)	300°C

TO-92 and TO-220 Package, (Soldering, 10 seconds)	260°C
SO Package (Note 12)	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 11)	2500V

Specified Operating Temperature Range:  $T_{MIN}$  to  $T_{MAX}$   
(Note 2)

LM35, LM35A	-55°C to +150°C
LM35C, LM35CA	-40°C to +110°C
LM35D	0°C to +100°C

**Electrical Characteristics**

(Notes 1, 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	$\pm 0.2$	$\pm 0.5$		$\pm 0.2$	$\pm 0.5$		°C
	$T_A = -10^\circ\text{C}$	$\pm 0.3$			$\pm 0.3$		$\pm 1.0$	°C
	$T_A = T_{MAX}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		°C
	$T_A = T_{MIN}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$		$\pm 1.5$	°C
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	$\pm 0.18$		$\pm 0.35$	$\pm 0.15$		$\pm 0.3$	°C
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	+10.0	+9.9, +10.1		+10.0		+9.9, +10.1	mV/°C
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$	$\pm 0.5$		$\pm 3.0$	$\pm 0.5$		$\pm 3.0$	mV/mA
Line Regulation (Note 3) $4V \leq V_S \leq 30V$	$T_A = +25^\circ\text{C}$	$\pm 0.01$	$\pm 0.05$		$\pm 0.01$	$\pm 0.05$		mV/V
		$\pm 0.02$		$\pm 0.1$	$\pm 0.02$		$\pm 0.1$	mV/V
Quiescent Current (Note 9)	$V_S = +5V, +25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5V$	105		131	91		114	μA
	$V_S = +30V, +25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30V$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V, +25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4V \leq V_S \leq 30V$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		+0.39		+0.5	+0.39		+0.5	μA/°C
Minimum Temperature or Rated Accuracy	In circuit of Figure 1, $I_L = 0$	+1.5		+2.0	+1.5		+2.0	°C
Long Term Stability	$T_J = T_{MAX}$ , for 1000 hours	$\pm 0.08$			$\pm 0.08$			°C



## Electrical Characteristics

(Notes 1, 6)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 1.0$		$\pm 0.4$	$\pm 1.0$		$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	$\pm 0.5$			$\pm 0.5$		$\pm 1.5$	$^\circ\text{C}$
	$T_A = T_{MAX}$	$\pm 0.8$	$\pm 1.5$		$\pm 0.8$		$\pm 1.5$	$^\circ\text{C}$
	$T_A = T_{MIN}$	$\pm 0.8$		$\pm 1.5$	$\pm 0.8$		$\pm 2.0$	$^\circ\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^\circ\text{C}$				$\pm 0.6$	$\pm 1.5$		$^\circ\text{C}$
	$T_A = T_{MAX}$				$\pm 0.9$		$\pm 2.0$	$^\circ\text{C}$
	$T_A = T_{MIN}$				$\pm 0.9$		$\pm 2.0$	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{MIN} < T_A < T_{MAX}$	$\pm 0.3$		$\pm 0.5$	$\pm 0.2$		$\pm 0.5$	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} < T_A < T_{MAX}$	$+10.0$	$+9.8,$ $+10.2$		$+10.0$		$+9.8,$ $+10.2$	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^\circ\text{C}$	$\pm 0.4$	$\pm 2.0$		$\pm 0.4$	$\pm 2.0$		mV/mA
	$T_{MIN} < T_A < T_{MAX}$	$\pm 0.5$		$\pm 5.0$	$\pm 0.5$		$\pm 5.0$	mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$	$\pm 0.01$	$\pm 0.1$		$\pm 0.01$	$\pm 0.1$		mV/V
	$4 \text{ V} \leq V_S \leq 30 \text{ V}$	$\pm 0.02$		$\pm 0.2$	$\pm 0.02$		$\pm 0.2$	mV/V
Quiescent Current (Note 9)	$V_S = +5 \text{ V}, +25^\circ\text{C}$	56	80		56	80		$\mu\text{A}$
	$V_S = +5 \text{ V}$	<b>105</b>		<b>158</b>	<b>91</b>		<b>138</b>	$\mu\text{A}$
	$V_S = +30 \text{ V}, +25^\circ\text{C}$	56.2	82		56.2	82		$\mu\text{A}$
	$V_S = +30 \text{ V}$	<b>105.5</b>		<b>161</b>	<b>91.5</b>		<b>141</b>	$\mu\text{A}$
Change of Quiescent Current (Note 3)	$4 \text{ V} \leq V_S \leq 30 \text{ V}, +25^\circ\text{C}$	0.2	2.0		0.2	2.0		$\mu\text{A}$
	$4 \text{ V} \leq V_S \leq 30 \text{ V}$	0.5		3.0	0.5		3.0	$\mu\text{A}$
Temperature Coefficient of Quiescent Current		$+0.39$		$+0.7$	$+0.39$		$+0.7$	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^\circ\text{C}$
Long Term Stability	$T_A = T_{MAX}, 10^\circ\text{C}$ 1000 hours	$\pm 0.08$			$\pm 0.08$			$^\circ\text{C}$

**Note 1:** Unless otherwise noted, these specifications apply:  $-55^\circ\text{C} \leq T_{stg} \leq +150^\circ\text{C}$  for the LM35 and LM35A;  $-40^\circ\text{C} \leq T_{stg} \leq +110^\circ\text{C}$  for the LM35C and LM35CA; and  $0^\circ\text{C} \leq T_{stg} \leq +100^\circ\text{C}$  for the LM35D.  $V_S = +5 \text{ Vdc}$  and  $I_{LOAD} = 50 \mu\text{A}$ , in the circuit of Figure 1. These specifications also apply from  $+2^\circ\text{C}$  to  $T_{MAX}$  in the circuit of Figure 1. Specifications in boldface apply over the full rated temperature range.

**Note 2:** Thermal resistance of the TO-46 package is  $400^\circ\text{C/W}$ , junction to ambient, and  $24^\circ\text{C/W}$  junction to case. Thermal resistance of the TO-92 package is  $100^\circ\text{C/W}$  junction to ambient. Thermal resistance of the small outline molded package is  $220^\circ\text{C/W}$  junction to ambient. Thermal resistance of the TO-220 package is  $90^\circ\text{C/W}$  junction to ambient. For additional thermal resistance information see table in the Applications section.

**Note 3:** Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to freewheeling effects can be computed by multiplying the internal dissipation by the thermal resistance.

**Note 4:** Tested Limits are guaranteed and 100% tested in production.

**Note 5:** Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

**Note 6:** Specifications in boldface apply over the full rated temperature range.

**Note 7:** Accuracy is defined as the error between the output voltage and  $10 \text{ mV}/^\circ\text{C}$  times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in  $^\circ\text{C}$ ).

**Note 8:** Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best fit straight line, over the device's rated temperature range.

**Note 9:** Quiescent current is defined in the circuit of Figure 1.

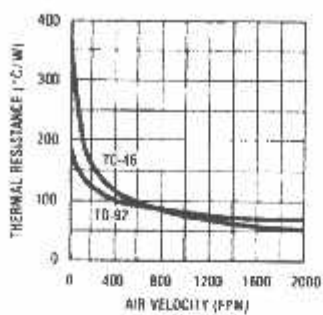
**Note 10:** Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

**Note 11:** Human body model,  $100 \text{ pF}$  discharged through a  $1.5 \text{ k}\Omega$  resistor.

**Note 12:** See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

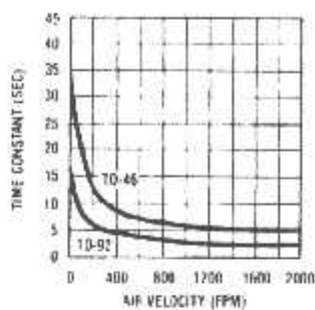
## Typical Performance Characteristics

**Thermal Resistance Junction to Air**



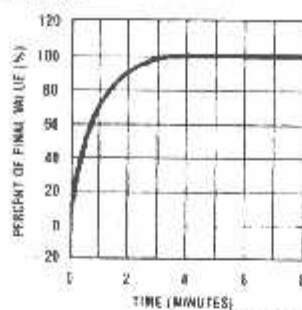
US005516-25

**Thermal Time Constant**



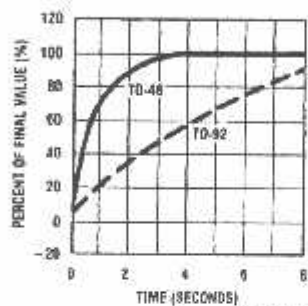
US005516-26

**Thermal Response in Still Air**



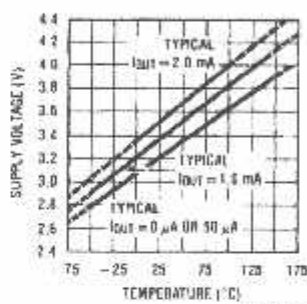
US005516-27

**Thermal Response in Stirred Oil Bath**



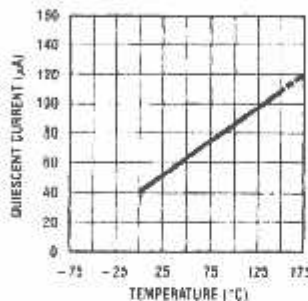
US005516-28

**Minimum Supply Voltage vs. Temperature**



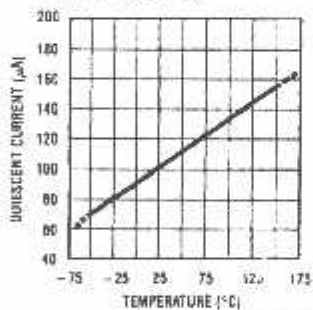
US005516-29

**Quiescent Current vs. Temperature (In Circuit of Figure 1.)**



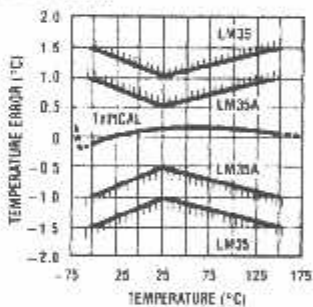
US005516-30

**Quiescent Current vs. Temperature (In Circuit of Figure 2.)**



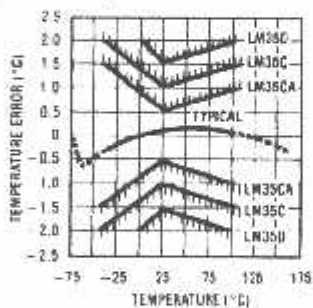
US005516-31

**Accuracy vs. Temperature (Guaranteed)**



US005516-12

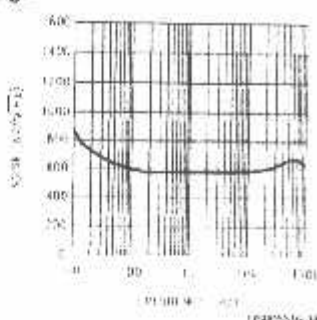
**Accuracy vs. Temperature (Guaranteed)**



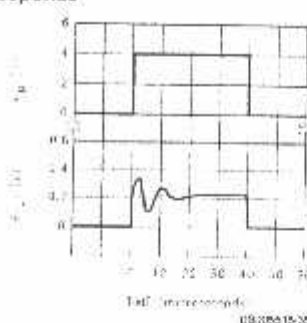
US005516-13

## Typical Performance Characteristics (Continued)

Noise Voltage



Start-Up Response



### Applications

The LM35 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface and its temperature will be within about 0.01°C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 die would be at an intermediate temperature between the surface temperature and the air temperature. This is especially true for the TO-92 plastic package, where the copper leads are the principal thermal path to carry heat into the device, so its temperature might be closer to the air temperature than to the surface temperature.

To minimize this problem, be sure that the wiring to the LM35, as it leaves the device, is held at the same temperature as the surface of interest. The easiest way to do this is to cover up these wires with a bead of epoxy which will insure that the leads and wires are all at the same temperature as the surface, and that the LM35 die's temperature will not be affected by the air temperature.

The TO-46 metal package can also be soldered to a metal surface or pipe without damage. Of course, in that case the V- terminal of the circuit will be grounded to that metal. Alternatively, the LM35 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM35 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to insure that moisture cannot corrode the LM35 or its connections.

These devices are sometimes soldered to a small light-weight heat fin, to decrease the thermal time constant and speed up the response in slowly-moving air. On the other hand, a small thermal mass may be added to the sensor to give the steadiest reading despite small deviations in the air temperature.

### Temperature Rise of LM35 Due To Self-heating (Thermal Resistance, $\theta_{JA}$ )

	TO-46 no heat sink	TO-46* small heat fin	TO-92 no heat sink	TO-92* small heat fin	SO-8 no heat sink	SO-8** small heat fin	TO-220 no heat sink
Still air	400°C/W	100°C/W	180°C/W	140°C/W	220°C/W	110°C/W	90°C/W
Moving air	100°C/W	40°C/W	90°C/W	70°C/W	105°C/W	90°C/W	28°C/W
Still oil	100°C/W	40°C/W	90°C/W	70°C/W			
Straight oil	50°C/W	30°C/W	45°C/W	40°C/W			
(Clamped to metal)							
(with heat sink)		(24°C/W)				(55°C/W)	

\*Wickel type 201, or 1" disc of 0.020" sheet brass, soldered to case, or similar.

\*\*TO-92 and SO-8 packages glued and leads soldered to 1" square of 1/16" printed circuit board with 2 oz. lot or similar.

## Typical Applications

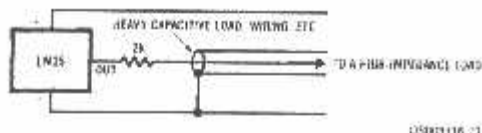


FIGURE 3. LM35 with Decoupling from Capacitive Load

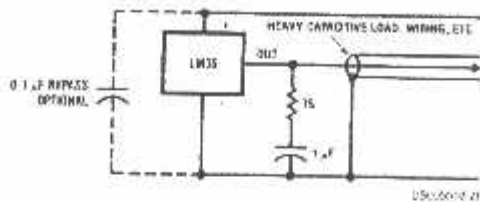


FIGURE 4. LM35 with R-C Damper

### CAPACITIVE LOADS

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see Figure 4.

When the LM35 is applied with a 200Ω load resistor as shown in Figure 5, Figure 6 or Figure 8 it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc., as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from  $V_{IN}$  to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 μF from output to ground are often useful. These are shown in Figure 13, Figure 14, and Figure 16.

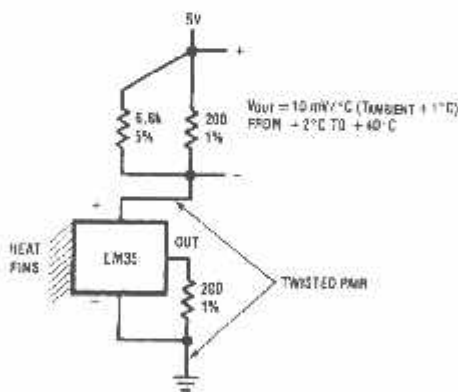


FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)

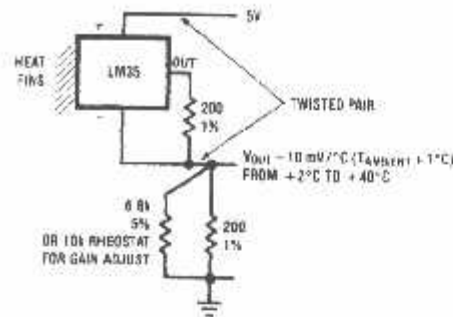


FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

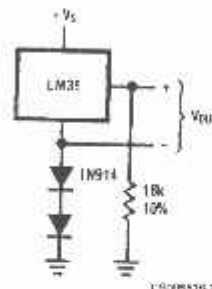


FIGURE 7. Temperature Sensor, Single Supply, -55° to +150°C

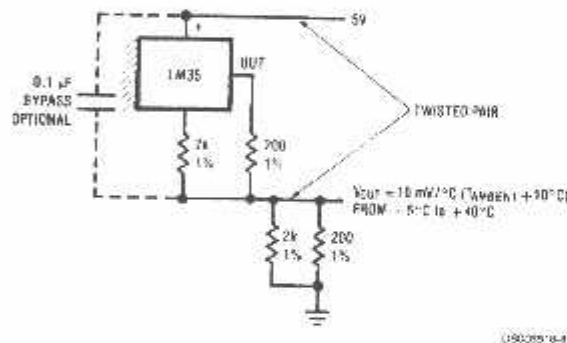


FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)

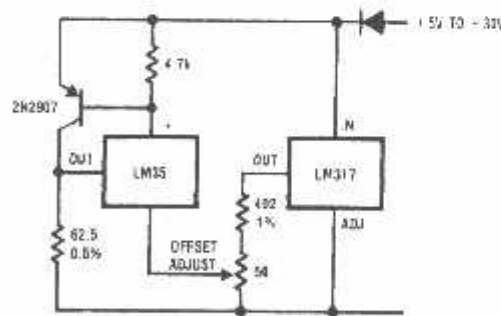


FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)

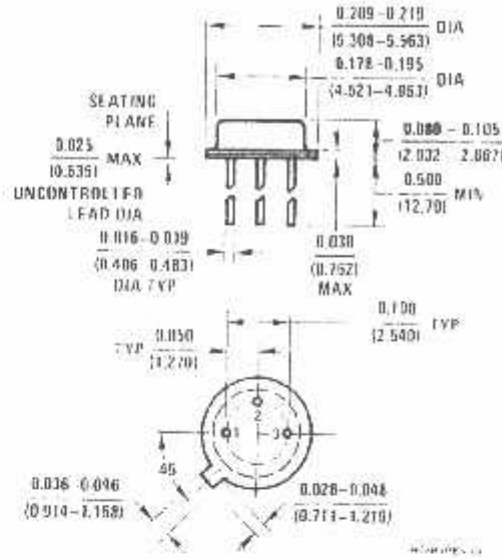




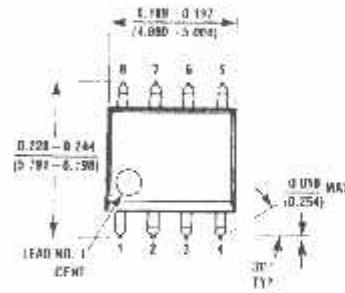


**Physical Dimensions** inches (millimeters) unless otherwise noted

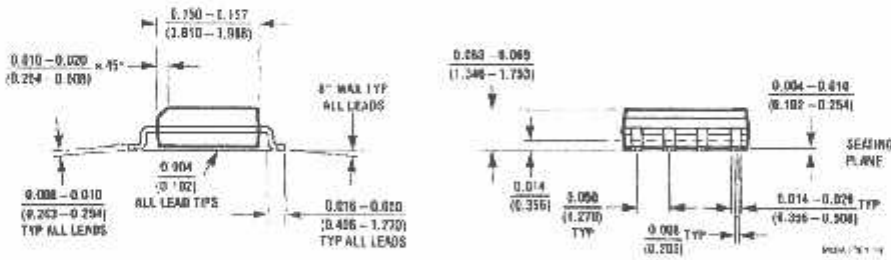
LM35



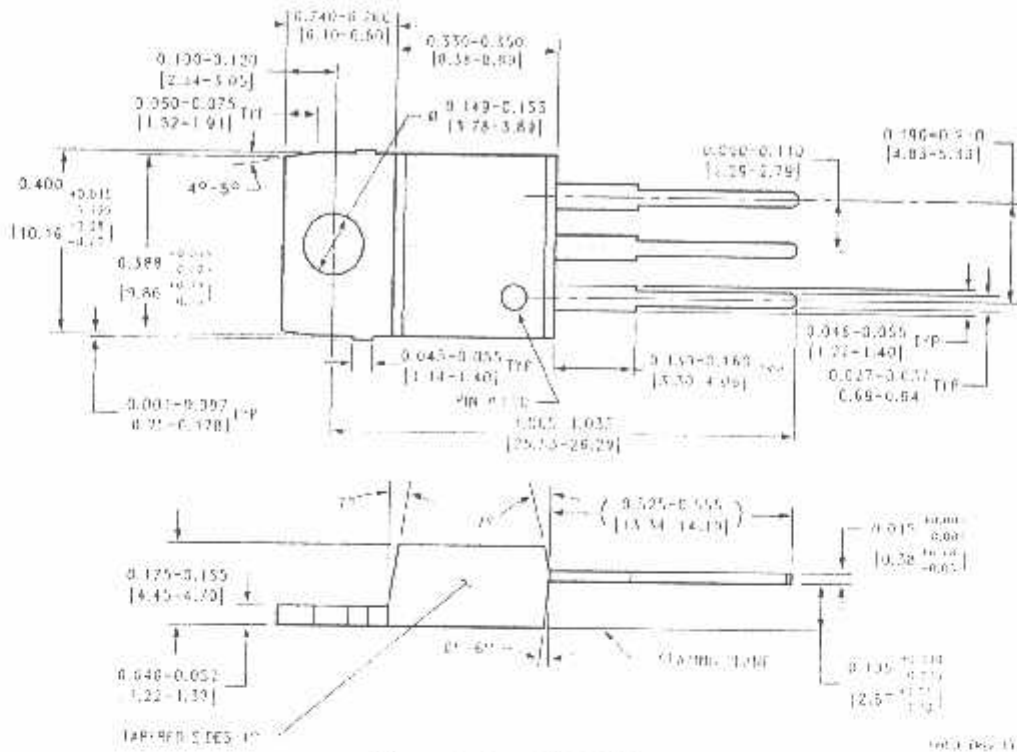
**TO-46 Metal Can Package (H)**  
 Order Number LM35H, LM35AH, LM35CH,  
 LM35CAH, or LM35DH  
 NS Package Number H03H



**SO-8 Molded Small Outline Package (M)**  
 Order Number LM35DM  
 NS Package Number M08A

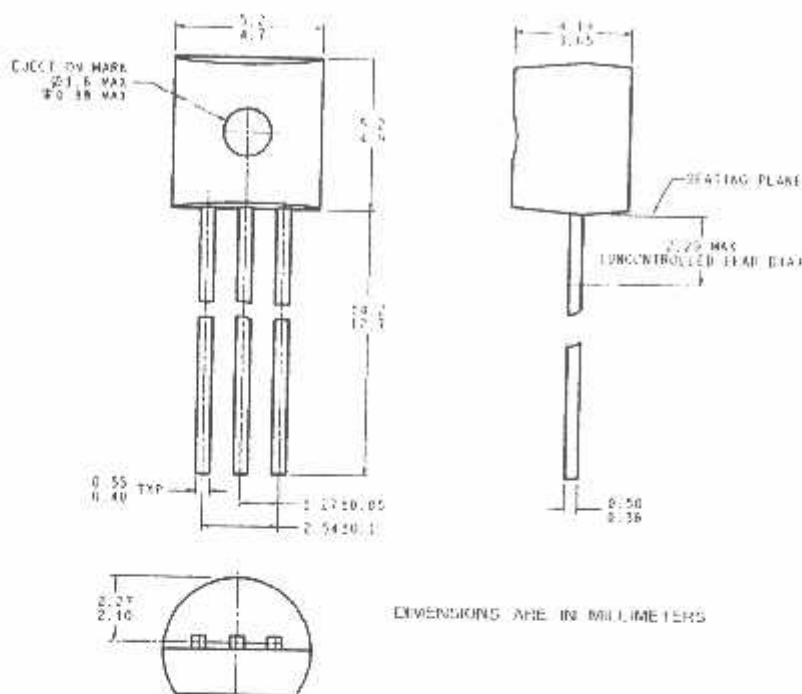




**Physical Dimensions** inches (millimeters) unless otherwise noted (Continued)


Power Package TO-220 (T)  
Order Number LM35DT  
NS Package Number TA03F

**Physical Dimensions** inches (millimeters) unless otherwise noted (Continued)



TO-92 Plastic Package (Z)  
Order Number LM35CZ, LM35CAZ or LM35DZ  
NS Package Number Z03A

LM35 Precision Centigrade Temperature Sensors

**LIFE SUPPORT POLICY**

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

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

**National Semiconductor Corporation**  
Americas  
Tel: 1-800-272-6989  
Fax: 1-800-737-7018  
Email: support@nsc.com  
www.national.com

**National Semiconductor Europe**  
Fax: +49 (0) 180-533 85 86  
Email: europe.support@nsc.com  
Deutsch: Tel: +49 (0) 89 9508 6200  
English: Tel: +44 (0) 1753 474711  
France: Tel: +33 (0) 1 41 91 0700

**National Semiconductor Asia Pacific Customer Response Group**  
Tel: 65 2544466  
Fax: 65-2504466  
Email: ap.support@nsc.com

**National Semiconductor Japan Ltd.**  
Tel: 81-3-5639-7500  
Fax: 81-3-5639-7501

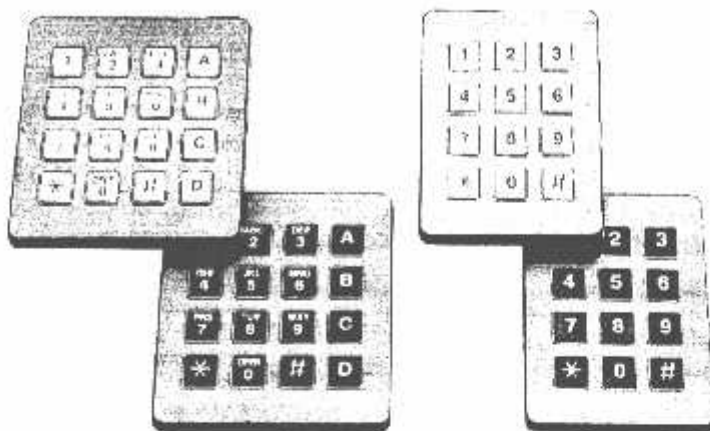
National does not assume any responsibility for use or any circuit described, no circuit patent licenses are implied and National reserves the right at any time without notice to change any circuitry and specifications.

**SERIES 96**  
Conductive Rubber

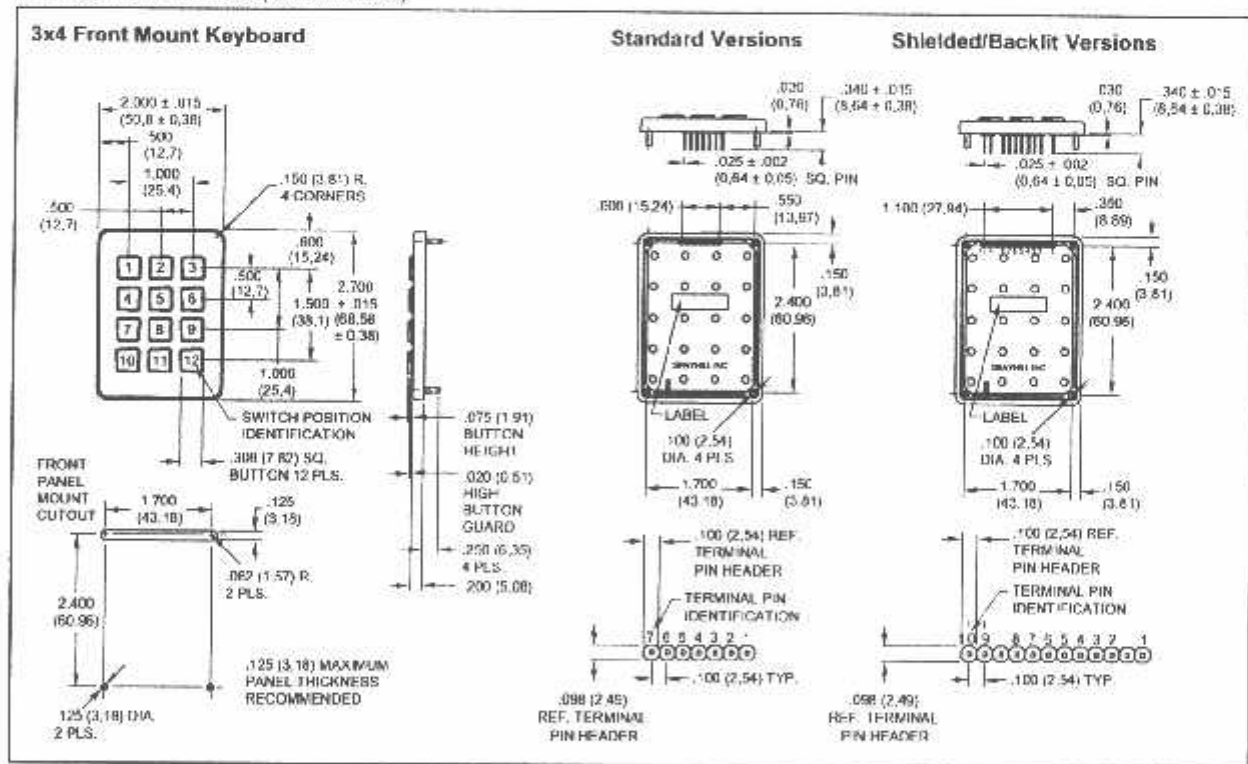
**FEATURES**

- Quality, Economical Keyboards
- Easily Customized Legends
- Matrix Circuitry
- Backlit and Shielded Options Available
- Termination Mates With Standard Connectors
- Tactile Feedback to Operator
- 1,000,000 Operations per Button
- Compatible With High Resistance Logic Inputs

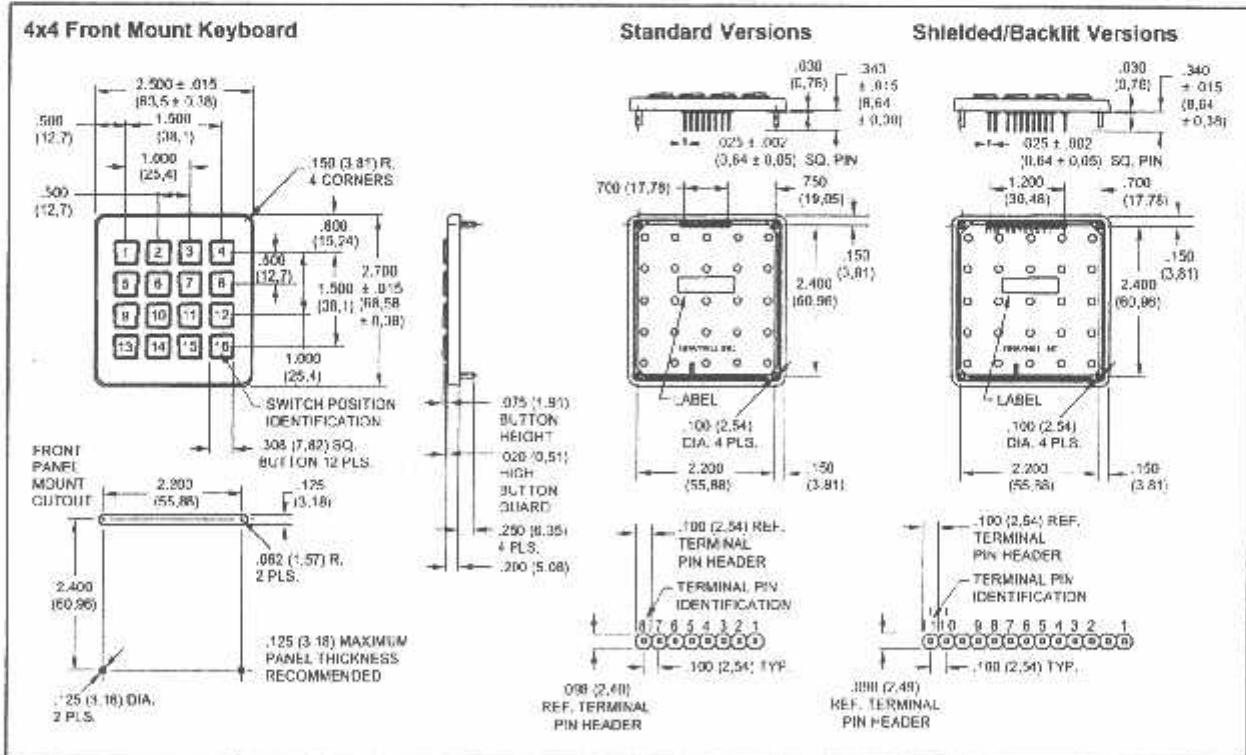
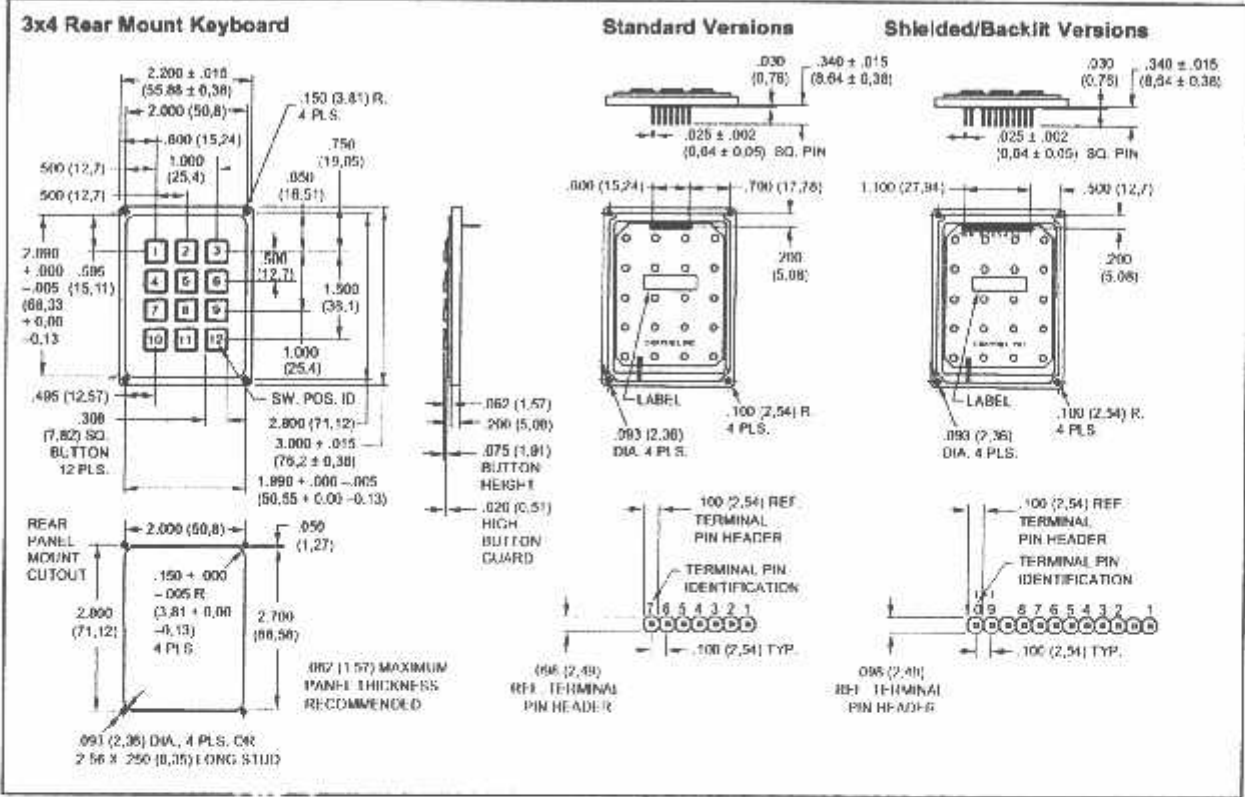
The Series 96 is Grayhill's most economical 3x4 and 4x4 keypad family. The contact system utilizes conductive rubber to mate the appropriate PC board traces. Offered in matrix circuitry, with shielded and backlit options. Built with quality component parts, the Series 96 is subjected to our rigid statistical process control to insure that it meets our reliability standards.



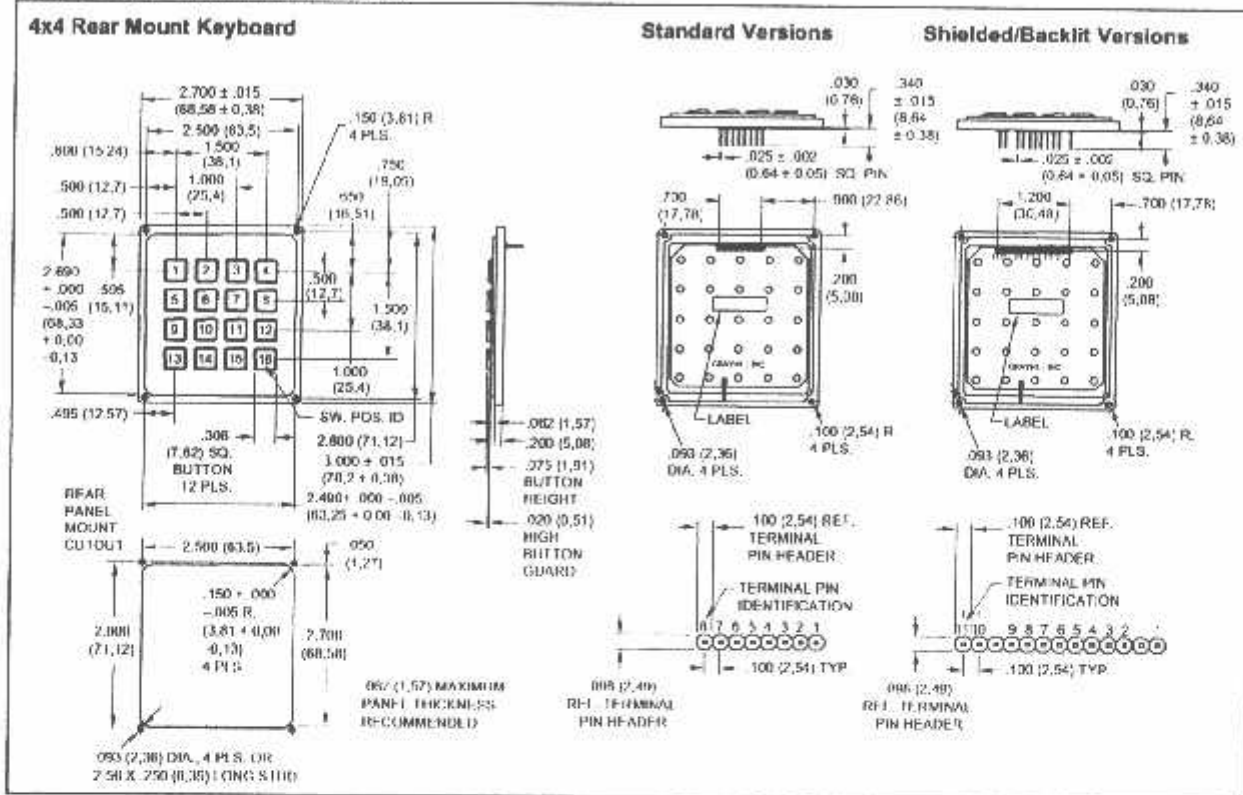
**DIMENSIONS** In inches (and millimeters)



## DIMENSIONS In inches (and millimeters)



**DIMENSIONS** In inches (and millimeters)



**CODE AND TRUTH TABLES**

Dots in the chart indicate connected terminals when switch is closed. Terminals are identified on the keyboard.

**12 Button Keypads**

BUTTON LOCATION	MATRIX CODES	
	Standard	Shielded/Backlit
1	•	•
2	•	•
3	•	•
4	•	•
5	•	•
6	•	•
7	•	•
8	•	•
9	•	•
10	•	•
11	•	•
12	•	•
	5 6 7 11 2 3 4	6 7 8 2 3 4 5 1 9 10

Shielded keypad = Shielded Backlit keypad = NC  
 Shielded and backlit keypad = Shielded  
 Shielded keypad = NC  
 Backlit keypad = EL Panel 1  
 Shielded and backlit keypad = EL Panel 1  
 Shielded keypad = NC  
 Backlit keypad = EL Panel 2  
 Shielded and backlit keypad = EL Panel 2

**16 Button Keypads**

BUTTON LOCATION	MATRIX CODES	
	Standard	Shielded/Backlit
1	•	•
2	•	•
3	•	•
4	•	•
5	•	•
6	•	•
7	•	•
8	•	•
9	•	•
10	•	•
11	•	•
12	•	•
13	•	•
14	•	•
15	•	•
16	•	•
	5 6 7 8 1 2 3 4	6 7 8 9 2 3 4 5 1 10 11

Shielded keypad = Shielded Backlit keypad = NC  
 Shielded and backlit keypad = Shielded  
 Shielded keypad = NC  
 Backlit keypad = EL Panel 1  
 Shielded and backlit keypad = EL Panel 1  
 Shielded keypad = NC  
 Backlit keypad = EL Panel 2  
 Shielded and backlit keypad = EL Panel 2

## SPECIFICATIONS

### Rating Criteria

**Rating at 12 Vdc:** 5 millamps for .5 seconds  
**Contact Bounce:** < 12 milliseconds  
**Contact Resistance:** < 100 ohms (at stated operating force)  
**Voltage Breakdown:** 250 Vac between components  
**Mechanical Operation Life:** 1,000,000 operations per key  
**Insulation Resistance:** > 10<sup>12</sup> ohms @ 500 Vdc  
**Push Out Force Per Pin:** 5 lbs.

### Operating Features

**Travel:** .040 minimum  
**Operating Force:** 175 ± 40 grams  
**Operating Temperature:** -30°C to +80°C

### Material and Finishes

**Terminal Pin:** Phosphor bronze, solder-plated  
**PC Board:** FR-4 glass cloth epoxy  
**Keypad:** Silicone rubber, durometer 50 ± 5  
**Housing:** ABS, cycloac "KJW"  
**Housing Color:** Black

### Shielding Effectiveness

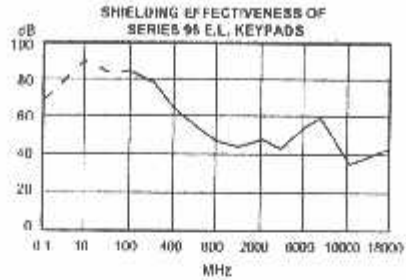
Results shown are typical for a standard Grayhill Series 84S keyboard. A conductive gasket will generally increase the shielding, depending on the size and shape of the gasket and its material. Data derived for E-Field Radiation.

### Test Method:

Measurements were made with the keyboard mounted to a brass plate, which in turn was mounted to a shielded enclosure containing the receiving equipment. A signal generator provided the frequency source that was radiated from the transmitting antenna to the enclosed receiving antenna. The spacing between antennas was maintained constant throughout the frequency range. The effectiveness rating is determined by establishing a reference reading without obstruction between the two antennas and determining the difference between that reading and the test setup reading.

### Note:

When measured in actual equipment, shielding effectiveness is determined by many factors. This method accurately represents the shielding effectiveness of the Grayhill Series 84S under ideal test conditions.



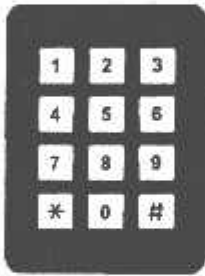
— — Represents shielding effectiveness greater than or equal to line.

Frequency MHz	Rating in dB
0.1	≥ 85.2
10	≥ 94.8
100	90.5
400	64.2
800	42.3
2,000	40.5
6,000	33.1
10,000	34.4
18,000	37.0

## STANDARD LEGENDS

Available through Grayhill Distributors

To order one of the configurations below, use the dash number shown here; select the keypad size and code, and order the part number with the appropriate legend dash number.



-102



-006

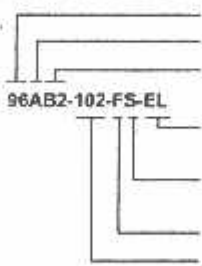


-152



-056

## ORDERING INFORMATION



**Grayhill Series Number**

**Keyboard Size:** A = 3x4, B = 4x4

**Circuitry:** B2 = Matrix (terminal pin header)

**E.L. Panel Backlighting Option**

EL = Backlit, Blank = Non-backlit

**EM/RFI Shielding Option**

S = Shielded, Blank = Non-shielded

**Mounting Option:** F = Front panel mount, R = Rear panel mount

**Standard Legend Choices**

*12 Position legends*

102 = Black legends on a white button

152 = White legends on a black button

*16 Position legends*

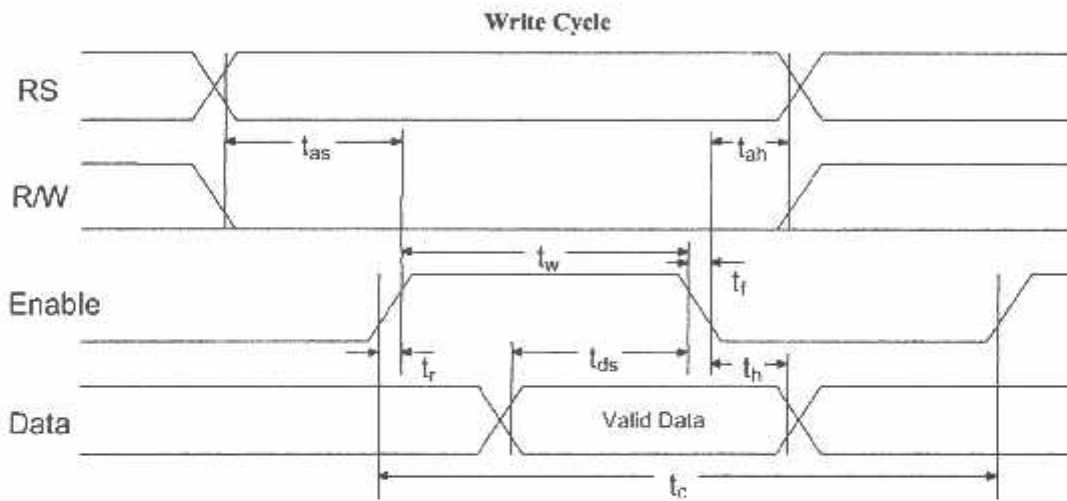
006 = Black legends on a white button

056 = White legends on a black button

Available from your local Grayhill Distributor.

For prices and discounts, contact a local Sales Office, an authorized local Distributor or Grayhill.

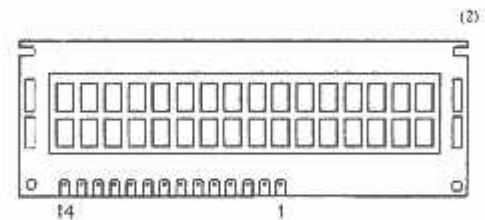
Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
OP	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero, returns shifted display to original position. DDRAM contents remains unchanged.	3
Cursor Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction, and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking(B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width(DL), number of display lines (N,M) and voltage generator control (G).	3
Set CGRAM Addr	0	0	0	1	Character Generator RAM					Sets CGRAM Address	3	
Set DDRAM Addr	0	0	1	Display Data RAM Address					Sets DDRAM Address	3		
Read Busy Flag & Addr	0	1	BF	Address Counter					Reads Busy Flag & Address Counter	0		
Read Data	1	0	Read Data					Reads data from CGRAM or DDRAM	3			
Write Data	1	1	Write Data					Writes data from CGRAM or DDRAM	3			



Parameter	Symbol	Min <sup>(1)</sup>	Typ <sup>(1)</sup>	Max <sup>(1)</sup>	Unit
Write Cycle Time	$t_c$	500	-	-	ns
Write Pulse Width (High)	$t_w$	230	-	-	ns
Write Rise/Fall Time	$t_r, t_f$	-	-	20	ns
Address Setup Time	$t_{as}$	40	-	-	ns
Address Hold Time	$t_{ah}$	10	-	-	ns
Data Setup Time	$t_{ds}$	80	-	-	ns
Data Hold Time	$t_h$	10	-	-	ns

Note<sup>1</sup> The above specifications are a indication only. Timing will vary from manufacturer manufacturer.

Note<sup>2</sup> A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x 16 character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x 16 character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD module.



Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	V0	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Strube)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

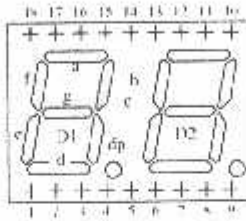


# Numeric Display

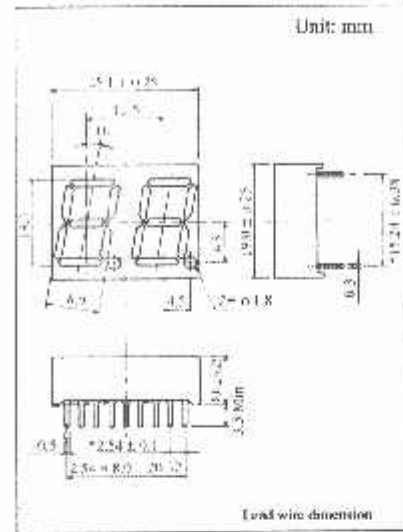
## 2 Digit 14.4mm (.6") Series

Conventional Part No.	Global Part No.	Lighting Color
LN526RA	LN226AA01	Red
LN526RK	LN226KA01	Red
LN526GA	LN326AA01	Green
LN526GK	LN326KA01	Green

### Terminal Connection



Pin No.	Assignment	Assignment
1	Cathode c1	Anode a1
2	Cathode d1	Anode d1
3	Cathode e1	Anode e1
4	Cathode f1	Anode f1
5	Cathode c2	Anode c2
6	Cathode d2	Anode d2
7	Cathode e2	Anode e2
8	Cathode f2	Anode f2
9	Cathode dcz	Anode dcz
10	Cathode h2	Anode h2
11	Cathode az	Anode az
12	Cathode cz	Anode cz
13	Common Anode D2	Common Cathode D2
14	Common Anode D1	Common Cathode D1
15	Cathode b1	Anode b1
16	Cathode a1	Anode a1
17	Cathode g1	Anode g1
18	Cathode h1	Anode h1



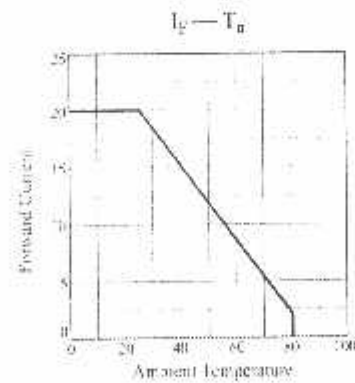
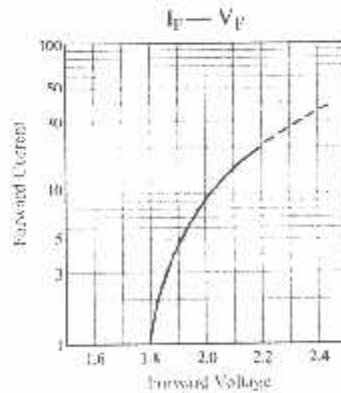
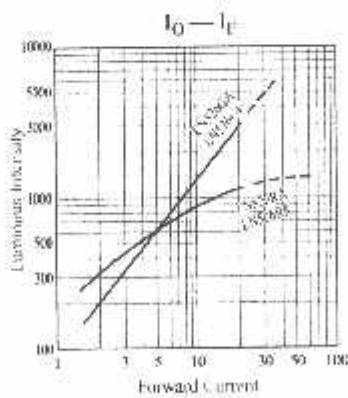
### Absolute Maximum Ratings ( $T_a = 25^\circ\text{C}$ )

Lighting Color	$P_D$ (mW)	$I_F$ (mA)	$I_{FP}$ (mA)*	$V_R$ (V)	$T_{opr}$ ( $^\circ\text{C}$ )	$T_{stg}$ ( $^\circ\text{C}$ )
Red	60	20	100	5	-25 - +80	-30 - +85
Green	60	20	100	5	-25 - +80	-30 - +85

\*  $I_{FP}$  : ... edge IFF. Pulse width 1 msec. The condition of  $I_F$  is duty 10%, Pulse width 1 msec.

### Electro-Optical Characteristics ( $T_a = 25^\circ\text{C}$ )

Conventional Part No.	Lighting Color	Common	$I_O$				$V_F$		$\lambda_D$		$I_F$	$I_R$	
			Typ	Min	Typ	$I_F$	Typ	Max	Typ	Typ		Max	$V_R$
LN526RA	Red	Anode	600	250	250	5	2.2	2.8	700	100	20	10	5
LN526RK	Red	Cathode	600	250	250	5	2.2	2.8	700	100	20	10	5
LN526GA	Green	Anode	1500	500	500	10	2.2	2.8	565	30	20	10	5
LN526GK	Green	Cathode	1500	500	500	10	2.2	2.8	565	30	20	10	5
Unit	—	—	μcc	μcd	μcd	mA	V	V	nm	nm	mA	μA	V



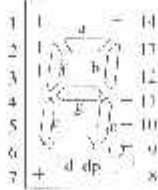


# Numeric Display

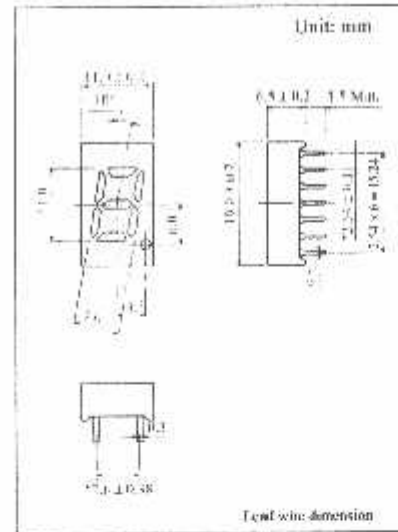
## 1 Digit 11.0 mm (.4") Series

Conventional Part No.	Citibal Part No.	Lighting Color
LN514RA	LNM214AA01	Red
LN514RK	LNM214KA01	Red
LN514GA	LNM314AA01	Green
LN514GK	LNM314KA01	Green

### Terminal Connection



Pin No.	Assignment	Assignment
1	Cathode a	Anode a
2	Cathode f	Anode f
3	Common Anode	Common Cathode
4		
5		
6		
7	Cathode c	Anode c
8	Cathode d	Anode d
9	Cathode dp	Anode dp
10	Cathode e	Anode e
11	Cathode p	Anode p
12		
13	Cathode h	Anode h
14	Common Anode	Common Cathode



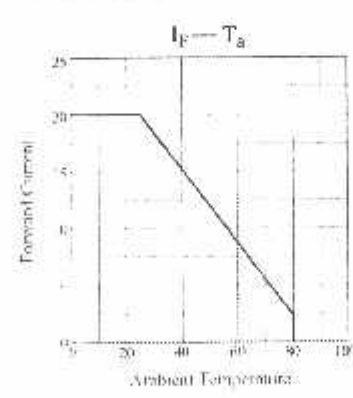
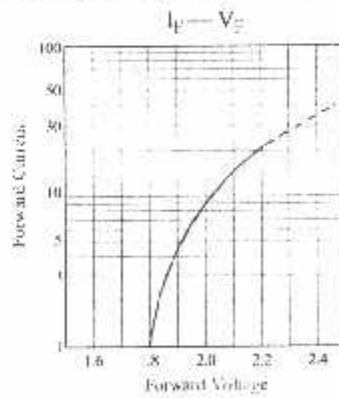
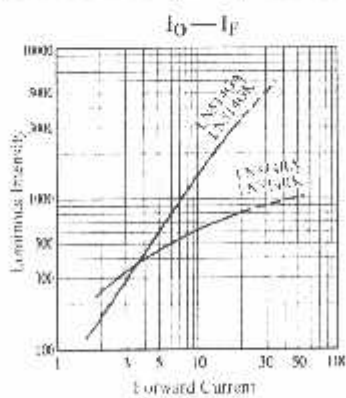
### Absolute Maximum Ratings ( $T_a = 25^\circ\text{C}$ )

Lighting Color	$P_D$ (mW)	$I_F$ (mA)	$I_{FP}$ (mA)*	$V_F$ (V)	$T_{opr}$ ( $^\circ\text{C}$ )	$T_{sig}$ ( $^\circ\text{C}$ )
Red	60	20	100	5	-25 ~ +80	-30 ~ +85
Green	60	20	100	5	-25 ~ +80	-30 ~ +85

\*  $I_{FP}$  : Pulse width 100  $\mu\text{s}$ . Pulse width 1 msec. The resolution of  $I_{FP}$  is duty 10%. Pulse width 1 msec.

### Electro-Optical Characteristics ( $T_a = 25^\circ\text{C}$ )

Conventional Part No.	Lighting Color	Common	$I_O$ / seg		$I_O$ / d.p.		$V_F$		$\lambda_p$	$\Delta\lambda$	$I_F$	$I_R$	
			Typ	Min	Typ	$I_F$	Typ	Max	Typ	Typ		Max	$V_R$
LN514RA	Red	Anode	450	150	150	5	2.2	2.8	700	100	20	10	5
LN514RK	Red	Cathode	450	150	150	5	2.2	2.8	700	100	20	10	5
LN514GA	Green	Anode	1500	500	500	10	2.2	2.8	565	30	20	10	5
LN514GK	Green	Cathode	1500	500	500	10	2.2	2.8	565	30	20	10	5
Unit			$\mu\text{cd}$	$\mu\text{cd}$	$\mu\text{cd}$	$\text{mA}$	V	V	nm	nm	$\text{mA}$	$\mu\text{A}$	V



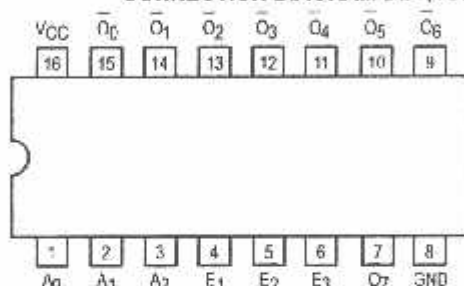


# 1-OF-8 DECODER/ DEMULTIPLEXER

The LSTTL/MSI SN54/74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- Demultiplexing Capability
- Multiple Input Enable for Easy Expansion
- Typical Power Dissipation of 32 mW
- Active Low Mutually Exclusive Outputs
- Input Clamp Diodes Limit High Speed Termination Effects

### CONNECTION DIAGRAM DIP (TOP VIEW)



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

### PIN NAMES

A <sub>0</sub> -A <sub>2</sub>	Address Inputs
E <sub>1</sub> , E <sub>2</sub>	Enable (Active LOW) Inputs
E <sub>3</sub>	Enable (Active HIGH) Input
O <sub>0</sub> -O <sub>7</sub>	Active LOW Outputs (Note b)

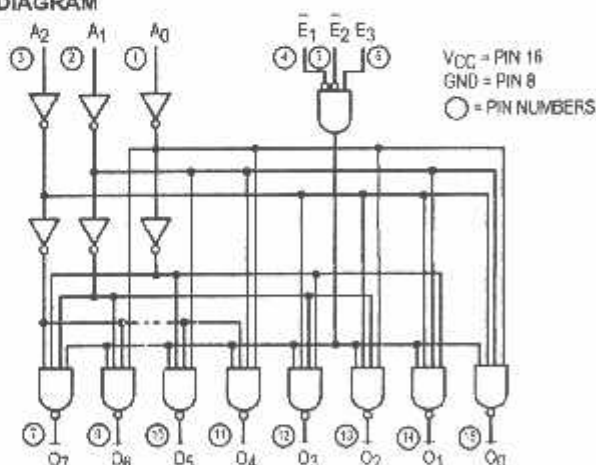
### LOADING (Note a)

	HIGH	LOW
A <sub>0</sub> -A <sub>2</sub>	0.5 U.L.	0.25 U.L.
E <sub>1</sub> , E <sub>2</sub>	0.5 U.L.	0.25 U.L.
E <sub>3</sub>	0.5 U.L.	0.25 U.L.
O <sub>0</sub> -O <sub>7</sub>	10 U.L.	5 (2.5) U.L.

### NOTES:

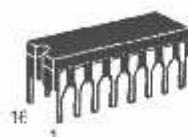
- a) 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.  
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

### LOGIC DIAGRAM

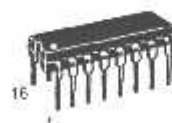


## SN54/74LS138

### 1-OF-8 DECODER/ DEMULTIPLEXER LOW POWER SCHOTTKY



J SUFFIX  
CERAMIC  
CASE 620-09



N SUFFIX  
PLASTIC  
CASE 648-08

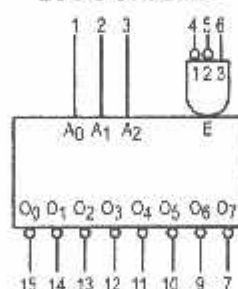


D SUFFIX  
SOIC  
CASE 751B-03

### ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

### LOGIC SYMBOL



VCC = PIN 16  
GND = PIN 8

## SN54/74LS138

### FUNCTIONAL DESCRIPTION

The LS138 is a high speed 1-of-8 Decoder/Demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs ( $A_0, A_1, A_2$ ) and when enabled provides eight mutually exclusive active LOW Outputs ( $O_0 - O_7$ ). The LS138 features three Enable inputs, two active LOW ( $E_1, E_2$ ) and one active HIGH ( $E_3$ ). All outputs will be HIGH unless  $E_1$  and  $E_2$  are LOW and  $E_3$  is HIGH. This multiple enable function allows easy parallel ex-

pansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four LS138s and one inverter. (See Figure a.)

The LS138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
$E_1$	$E_2$	$E_3$	$A_0$	$A_1$	$A_2$	$O_0$	$O_1$	$O_2$	$O_3$	$O_4$	$O_5$	$O_6$	$O_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Don't Care

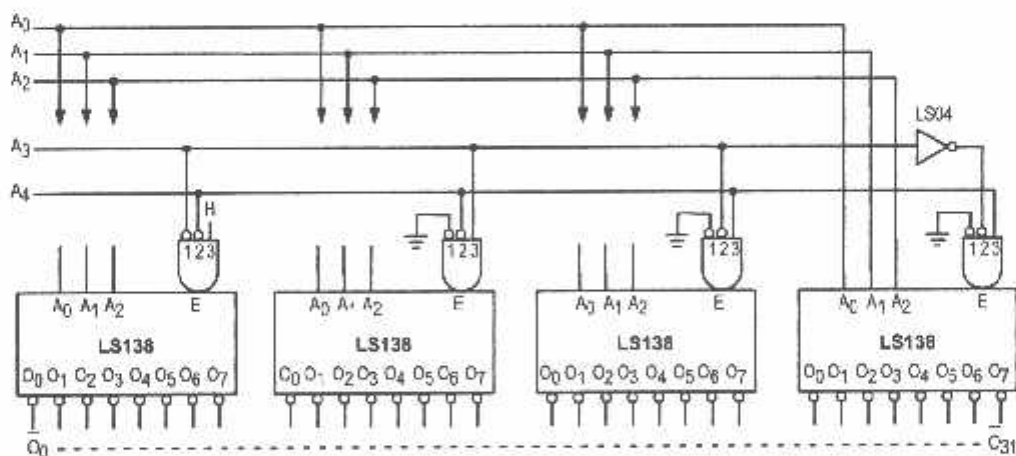


Figure a

# SN54/74LS138

## GUARANTEED OPERATING RANGES

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

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

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
V <sub>IH</sub>	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V <sub>IL</sub>	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
V <sub>IK</sub>	Input Clamp Diode Voltage		-0.65	-1.5	V	V <sub>CC</sub> = MIN, I <sub>IN</sub> = -18 mA	
V <sub>OH</sub>	Output HIGH Voltage	54	2.5	3.5	V	V <sub>CC</sub> = MIN, I <sub>OH</sub> = MAX, V <sub>IN</sub> = V <sub>IH</sub> or V <sub>IL</sub> per Truth Table	
		74	2.7	3.5	V		
V <sub>OL</sub>	Output LOW Voltage	54, 74		0.25	0.4	V	I <sub>OL</sub> = 4.0 mA
		74		0.35	0.5	V	I <sub>OL</sub> = 8.0 mA
I <sub>IH</sub>	Input HIGH Current			20	μA	V <sub>CC</sub> = MAX, V <sub>IN</sub> = 2.7 V	
				0.1	mA	V <sub>CC</sub> = MAX, V <sub>IN</sub> = 7.0 V	
I <sub>IL</sub>	Input LOW Current			-0.4	mA	V <sub>CC</sub> = MAX, V <sub>IN</sub> = 0.4 V	
I <sub>OS</sub>	Short Circuit Current (Note 1)	-20		-100	mA	V <sub>CC</sub> = MAX	
I <sub>CC</sub>	Power Supply Current			10	mA	V <sub>CC</sub> = MAX	

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

## AC CHARACTERISTICS (T<sub>A</sub> = 25°C)

Symbol	Parameter	Levels of Delay	Limits			Unit	Test Conditions
			Min	Typ	Max		
t <sub>PLH</sub> t <sub>PHL</sub>	Propagation Delay Address to Output	2		13 27	20 41	ns	V <sub>CC</sub> = 5.0 V C <sub>L</sub> = 15 pF
t <sub>PLH</sub> t <sub>PHL</sub>	Propagation Delay Address to Output	3		18 26	27 39		
t <sub>PLH</sub> t <sub>PHL</sub>	Propagation Delay E <sub>1</sub> or E <sub>2</sub> Enable to Output	2		12 21	18 32		
t <sub>PLH</sub> t <sub>PHL</sub>	Propagation Delay E <sub>3</sub> Enable to Output	3		17 25	26 38		

## AC WAVEFORMS

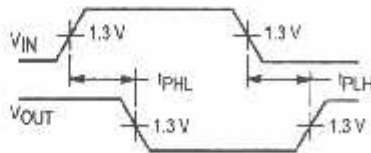


Figure 1

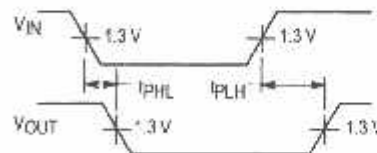


Figure 2

## FAST AND LS TTL DATA



---