

SKRIPSI

PERANCANGAN DAN PEMBUATAN ALAT PEMBERI INFORMASI PERGANTIAN JAM PELAJARAN SEKOLAH SECARA OTOMATIS BERBASIS MIKROKONTROLER AT89S51



Disusun Oleh :

TRI ANTORO ANGGUN MAHMUDI

NIM : 97.17.026

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

LEMBAR PERSETUJUAN



**PERANCANGAN DAN PEMBUATAN ALAT PEMBERI INFORMASI
PERGANTIAN JAM PELAJARAN SEKOLAH SECARA OTOMATIS
BERBASIS MIKROKONTROLER AT89S51**

SKRIPSI

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

Disusun oleh :

TRI ANTORO ANGGUN MAHMUDI

NIM. 97-17.026

Mengetahui,

Ketua Jurusan T. Elektro S-1

Diperiksa dan Disetujui,

Dosen Pembimbing

Ir. F. Yudi Limpraptono, MT.
NIP.Y. 103 9500 247

Ir. F. Yudi Limpraptono, MT.
NIP.Y. 103 9500 247

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



BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Tri Antoro Anggun Mahmudi
NIM : 97.17.026
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : **Perencanaan dan Pembuatan Alat Pemberi Informasi
Pergantian Jam Pelajaran Sekolah Secara Otomatis
Berbasis Mikrokontroller AT89S51**

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

Hari : Selasa
Tanggal : 09 Februari 2010
Dengan Nilai : 76,26 (B+) *84*

Panitia Ujian Skripsi



Ketua Majelis Penguji
(Ir. H. Sidik Noertjahjono, MT.)
NIP.Y. 1028700163

Sekretaris Majelis Penguji

(Ir. F. Yudi Limpraptono, MT.)
NIP.Y. 1039500247

Anggota Penguji

Penguji I

(Ir. Eko Nurcahyo)
NIP.Y. 1028700172

Penguji II

(M. Ibrahim Ashari, ST. MT.)
NIP.P.1030100358

**PERANCANGAN DAN PEMBUATAN ALAT
PEMBERI INFORMASI PERGANTIAN JAM PELAJARAN
SEKOLAH
SECARA OTOMATIS BERBASIS MIKROKONTROLLER
AT89S51**

TRI ANTORO ANGGUN MAHMUDI

NIM : 97.17.026

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

ABSTRAKSI

Bel atau alarm yang terdapat di sekolah-sekolah dibunyikan secara manual, yakni menggunakan operator untuk menekan tombol bel atau alarm-nya. Hal demikian sangatlah tidak praktis, seiring jaman otomatisasi. Pada Skripsi ini akan dibahas mengenai alat pemberi informasi pergantian jam pelajaran sekolah secara otomatis berbasis mikrokontroler AT89S51. Definisi bel atau alarm sekolah secara otomatis adalah suatu penanda untuk memulai dan mengakhiri pelajaran di sekolah yang akan berbunyi pada waktu yang ditentukan tanpa harus menekan tombol.

Pertama-tama kita dapat melakukan seting waktu bel atau alarm melalui tombol push button menu, up, dan down, dan kemudian memasukkan input suara agar nantinya dihasilkan output suara sebagai alarm. Pada saat waktu telah sesuai dengan input nilai waktu alarm, maka secara otomatis mikrokontroler akan mengeluarkan suara alarm dan output suara sesuai dengan seting alarm yang telah di inputkan

Alat ini menggunakan AT89S51 sebagai pengontrol, Real Time Clock DS1307 sebagai pewaktu. Sebagai input adalah 4 tombol push-button untuk melakukan seting waktu alarm, dan untuk memasukkan input suara kedalam ISD (record). Sedangkan output-nya adalah output waktu pada LCD, output buzzer sebagai alarm, dan output suara yang terekam dalam ISD.

Kata Kunci : ISD 25120., Mikrokontroler, RTC, LCD

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, karena atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan dan menyusun laporan skripsi ini dengan baik. Penyusunan laporan skripsi ini merupakan tindak lanjut penulis sebagai syarat untuk memperoleh gelar Sarjana Teknik Elektro S-1.

Dengan terselesaikannya penyusunan laporan skripsi ini, maka dengan setulus hati penulis menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Sidik Noertjahjono, M.T. selaku Dekan Fakultas Teknik Industri Institut Teknologi Nasional Malang.
3. Bapak Ir.F.Yudi Limpraptono, M.T. selaku Ketua Jurusan Teknik Elektro Institut Teknologi Nasional Malang sekaligus selaku Dosen Pembimbing.
4. Bapak dan Ibu selaku Dosen Penguji.
5. Bapak dan Ibu Dosen Institut Teknologi Malang, selaku pembimbing saya secara tidak langsung.
6. Bapak dan Ibu Staf Institut Teknologi Malang, yang banyak membantu.
7. Kedua orang tuaku yang tak henti-hentinya mendo'akan dan mencurahkan kasih sayang yang tak mungkin terbalas, kedua kakakku dan adikku yang selalu mendukungku.
8. Istri dan anakku yang selalu mendo'akan dan memberi semangat .
9. Temanku Dayat, Ipank dan tak lupa Mas Bagong terima kasih semuanya dan teman-teman X'97 Krek (sparring partnerku),Kawor,Alis,Mieft,Sahman,Paimo atas

supportnya dan juga yang telah banyak membantu dalam penyelesaian laporan skripsi ini.

Dalam penyusunan laporan skripsi ini, penulis mengakui bahwa masih banyak terdapat kekurangan. Maka penulis mengharapkan adanya saran dan kritikan yang bersifat membangun dan menyempurnakan dari pembaca atas penyusunan laporan skripsi ini.

Semoga laporan skripsi ini dapat memberikan sedikit sumbangsih pengetahuan bagi pembaca. Dan semoga Allah SWT. senantiasa memberikan petunjuk dan bimbingannya kepada kita semua, Amin.

Malang, Pebruari 2010

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
 BAB I. PENDAHULUAN	
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan	2
1.4. Batasan Masalah	2
1.5. Metodologi Penulisan	3
1.6. Sistematika Penulisan	3
 BAB II. LANDASAN TEORI	
2.1. Mikrokontroller AT89S51	5
2.1.1. Teori Umum.....	5
2.1.2. Arsitektur AT89S51.....	6
2.1.3. Konfigurasi Pin pada Mikrikontroller AT89S51	7
2.1.4. Karakteristik Oscilator Inverting	10
2.1.5. Organisasi Memory	11
2.1.5.1. Program Memory Internal	11
2.1.5.2. Data Memory (RAM) Internal	11

2.1.5.3. SFR (Special Function Register).....	13
2.2. RTC DS1307	14
2.3. Speak Call ISD 25120	20
2.4. Tombol Akses (Push-button).....	25
2.5. Dioda.....	26
2.5.1.Dioda Penyearah.....	26
2.6. Kapasitor.....	27
2.6.1. Cara Kerja Kapasitor	29
2.7. Resistor	30
2.8. LCD (Liquid Crystal Display)	33
2.8.1. Sinyal Interface M1632	34
2.8.2. Interface Ke MCS-51.....	34

BAB III. PERENCANAAN DAN PEMBUATAN ALAT

3.1. Diagram Blok.....	41
3.2. Prinsip Kerja Alat	24
3.3. Mikrokontroller AT89S51	46
3.3.1. Pemetaan Memory	47
3.3.2. Rangkaian Clock.....	48
3.3.3. Rangkaian Reset	48
3.3.4. Hubungan Pin pada AT89S51	49
3.4. Rangkaian RTC DS1307	50
3.5. Rangkaian Pemutar Suara ISD25120	52
3.6. Perancangan Rangkaian LCD.....	54
3.7. Perangkat Lunak	57

BAB IV. PENGUJIAN ALAT

4.1. Pengujian Rangkaian Rangkaian ISD 25120.....	59
4.1.1. Langkah Pengujian	60
4.1.2. Hasil Pengujian.....	61
4.2. Pegujian Rangkaian LCD	61
4.3. Pengujian Terhadap Rangkaian RTC	62
4.4. Pengujian Terhadap Rangkaian Push Button	63

BAB V. PENUTUP

5.1. Kesimpulan	66
5.2. Saran	67

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1. Diagram Blok AT89S51	6
Gambar 2.2. Konfigurasi Pin-pin AT89S51	7
Gambar 2.3. Karakteristik Oscilator	10
Gambar 2.4. Organisasi RAM Internal	12
Gambar 2.5. Proses Transfer Data pada I2C	14
Gambar 2.6. Format Data	17
Gambar 2.7. Format Data	18
Gambar 2.8. RTC DS1307	19
Gambar 2.9. Pin-pin IC ISD 25120	21
Gambar 2.10 Peraga Tombol /saklar	25
Gambar 2.11. Simbol Dioda	26
Gambar 2.12. Simbol Kapasitor	27
Gambar 2.13. Rangkaian Pengisian Kapasitor	29
Gambar 2.14. Grafik Tegangan dan Arus Pengisian Kapasitor	30
Gambar 2.15. Mengirim/Mengambil Data Ke/Dari M1632	35
Gambar 2.16. Hubungan M1632 ke MCS'51	36
Gambar 2.17. Rangkaian LCD M1632	41
Gambar 3.1. Blok Diagram	45
Gambar 3.2. Rangkaian Clock	48
Gambar 3.3. Rangkaian Reset	49
Gambar 3.4. Hubungan Pin pada AT89S51	50
Gambar 3.5. Rangkaian Serial RTC DS1307	51

Gambar 3.6. Rangkaian ISD 25120.....	52
Gambar 3.7.Rangkaian LCD pada Mikrokontroller.....	56
Gambar 3.8. Flowchart Keseluruhan Alat.....	58
Gambar 4.1. Rangkaian Pengujian ISD 25120.....	60
Gambar 4.2. Foto Tampilan LCD Hasil Percobaan.....	62
Gambar 4.3.Rangkaian Pengujian RTC DS1307	63
Gambar 4.4. Foto Hasil Pengujian Rangkaian RTC DS1307.....	63
Gambar 4.5. Rangkaian Pengujian Push-button.....	64
Gambar 4.6. Foto Alat	66

DAFTAR TABEL

Tabel 2-1. Pengaturan RS0-RS1 Untuk Select Register Bank	11
Tabel 2-2. 128 Byte Special Function Register	13
Tabel 2-3. Setting DS1307	19
Tabel 2-4. Tabel Gelang Warna Resistor	31
Tabel 2-5. Fungsi Pin-pin LCD	42
Tabel 4-1. Hasil Pengujian ISD	61
Tabel 4-2. Hasil Pengukuran pada Rangkaian Push-button	65



BAB I

PENDAHULUAN

1.1. Latar Belakang

Kemajuan pengetahuan dan teknologi dewasa ini berkembang cepat sekali dan berpengaruh dalam pembuatan alat-alat canggih, yaitu alat yang dapat bekerja secara otomatis dan memiliki ketelitian tinggi dengan bantuan mikrokontroller. Dengan adanya perkembangan teknologi inilah manusia semakin dipermudah dalam segala aspek kehidupan dan dituntut untuk lebih jeli dalam menyikapinya.

Dalam hal ini, pada era komputerisasi dan otomatisasi sudah masuk dalam segala bidang termasuk dalam dunia pendidikan. Untuk dunia pendidikan, salah satunya adalah telah menggunakan alat untuk memberikan informasi tentang waktu pergantian jam pelajaran, jam masuk kelas, jam istirahat maupun waktu pulang atau yang biasa disebut adalah bel.

Bel yang terdapat di sekolah-sekolah dibunyikan secara manual, yakni menggunakan operator untuk menekan tombol bel tersebut. Hal demikian sangatlah tidak praktis, seiring jaman otomatisasi. Pada Skripsi ini akan dibahas mengenai alat pemberi informasi pergantian jam pelajaran sekolah secara otomatis berbasis mikrokontroler AT89S51. Definisi dari alat ini adalah suatu penanda untuk memulai dan mengakhiri pelajaran di sekolah yang akan berbunyi pada waktu yang ditentukan tanpa harus menekan tombol.

1.2. Rumusan Masalah

Dari uraian tersebut diatas maka timbul beberapa permasalahan diantaranya adalah sebagai berikut:

1. Bagaimana merancang Hardware dari keseluruhan alat?
2. Bagaimana membuat Software?

1.3. TUJUAN

Merancang dan membuat Alat Pemberi Informasi Pergantian Jam Pelajaran Sekolah Secara Otomatis Berbasis Mikrokontroler AT89S51 sehingga dapat membantu kelancaran proses belajar mengajar di sekolah.

1.4. BATASAN MASALAH

Agar permasalahan tidak meluas maka penulis membatasi permasalahan sebagai berikut:

1. Menggunakan Mikrokontroler AT89S51 sebagai pengendali utama.
2. Menggunakan RTC DS1307 sebagai pewaktu.
3. Menggunakan LCD M1632 sebagai display kerja alat.
4. Menggunakan ISD 25120 untuk merekam suara alarm agar nantinya output suara sebagai alarm dapat sesuai yang kita inginkan.
5. Alat Pemberi Informasi Pergantian Jam Pelajaran Sekolah ini ditujukan untuk sekolah dasar dan sekolah tingkat lanjut.

BAB IV : PENGUJIAN ALAT

Berisi tentang uji coba alat yang telah dibuat, pengoperasian dan spesifikasi alat.

BAB V : PENUTUP

Merupakan kesimpulan dari pembahasan pada bab-bab sebelumnya dan kemungkinan pengembangan alat.

BAB II

TEORI DASAR

Bab ini akan menguraikan tentang dasar-dasar teori dasar yang menunjang dalam perancangan dan pembuatan alat. Uraian teori dalam bab ini meliputi teori IC AT89S51, ISD 25120, LCD M1632 dan perangkat-perangkat pendukung lainnya.

2.1. Mikrokontroler AT89S51

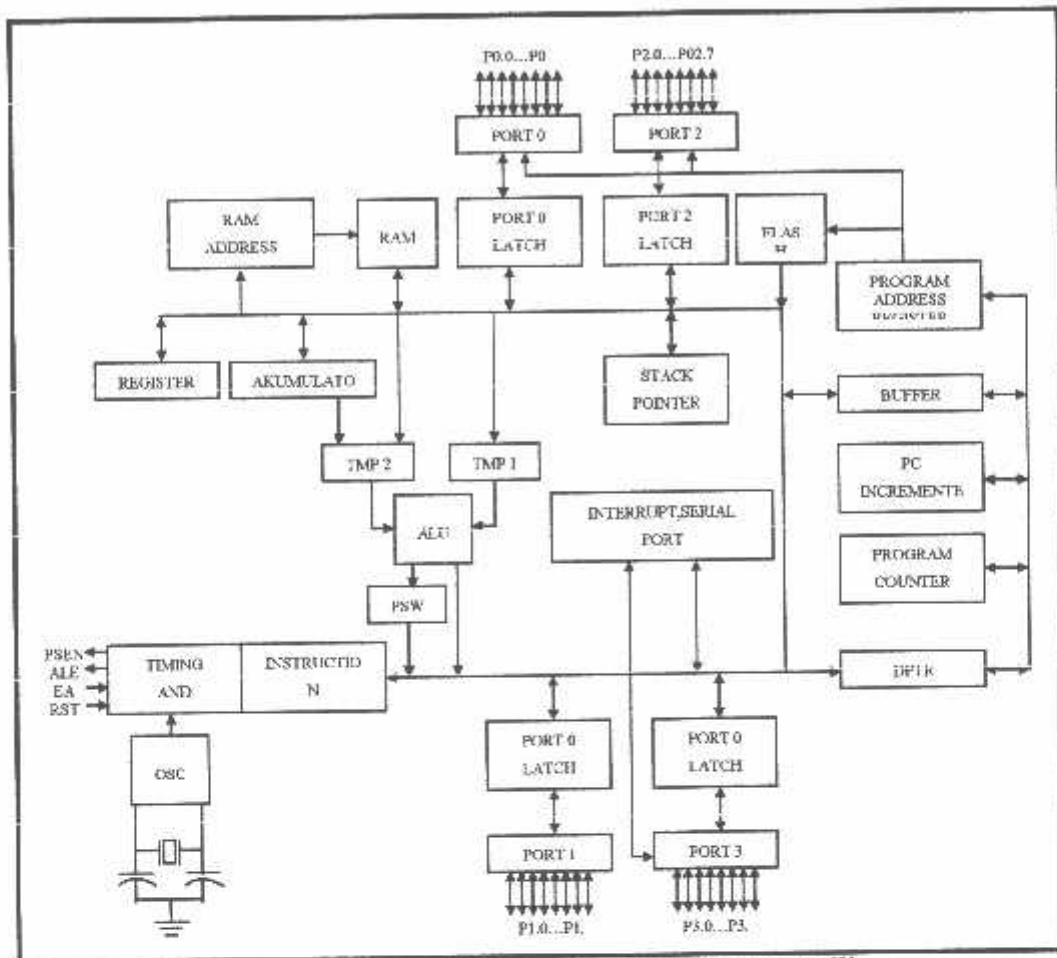
2.1.1. Teori umum.

AT89S51 merupakan sebuah mikrokontroller 8-bit CMOS, Low Power dengan 4Kb flash Programmable and Erasable Read Only Memory (PEROM). IC ini dibuat sesuai dengan standart industri konfigurasi pin dan intruction set dari MCS-51.

- 4Kb Flash Memory.
- 128 Byte Internal RAM.
- 32 I/O Lines.
- 2 Timer/Counter 16-Level.
- 1 Serial Port Full Duplex.
- On Chip Osilator.

AT89S51 mempunyai dua buah Power-Saving mode yang dapat diatur melalui software, yaitu: IDE Mode yang akan menghentikan CPU sebagai RAM, Timer/Counter, Serial Port dan Interrupsi system tetap berfungsi. Power Down Mode yang akan menyimpan ini di RAM, tetapi menahan Oscilator untuk tidak mengaktifkan chip yang lain sampai terjadi reset secara hardware.

Seperti yang terlihat pada gambar 2.1 dibawah ini yang memperlihatkan blok diagram dari Mikrokontroller AT89S51



Gambar 2.1. Diagram Blok AT89S51 ^[3]

2.1.2. Arsitektur AT89S51

Arsitektur Mikrokontroler AT89S51 adalah sebagai berikut:

- CPU (Central Processing Unit) 8-bit dengan register A (Accumulator) dan B.
- 16-Bit Program Counter (PC) dan data pointer (DPTR).
- 8-Bit Program status word (PSW).
- 4-Bit Stack pointer (SP).
- 4 Kbyte internal EPROM
- 128 byte internal RAM.
- 4 bank register, masing-masing berisi 8 register.

- INTO (P3.2) : Interrupt 0 external.
- INTI (P3.3) : Interrupt 1 external.
- TO (P3.4) : Input external timer 0.
- T1 (P3.5) : Input external timer 1.
- WR (P3.6) : Strobe tulis data memory external.
- RD (P3.7) : Strobe baca data memory external.

7. RST

Input reset.

8. ALE\PROG

Pulsa output ALE digunakan untuk proses “latching” byte address rendah (A0 - A7) selama pengaksesan ke external memory. Pin ini juga untuk memasukkan pulsa program selama pemrograman.

Pada operasi normal ALE mengeluarkan rate konstant yaitu 16 frekuensi osilasi dan boleh digunakan untuk timing external.

9. PSEN

Merupakan strobe baca ke program memory external.

10. EA\VPP

External address enable EA digroundkan jika mengakses memory external.

Untuk mengakses memory internal maka dihubungkan ke VCC.

11. XTAL 1 dan XTAL 2.

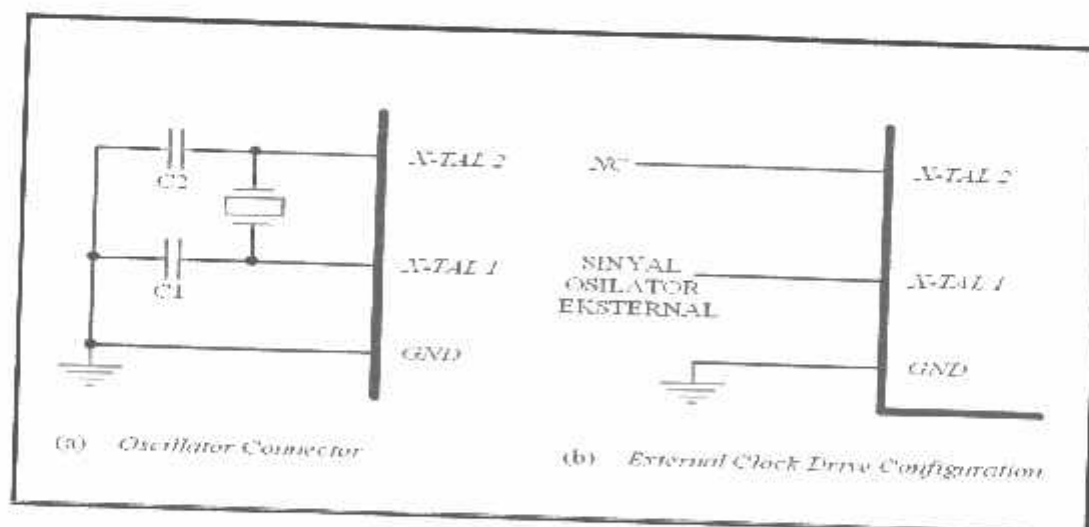
Kaki ini dihubungkan dengan kristal bila menggunakan osilator internal.

XTAL 1 merupakan input inverting osilator amplifier sedangkan XTAL 2 merupakan out-put inverting osilator amplifier.

2.1.4. Karakteristik Oscillator Inverting

X-TAL 1 dan X-TAL 2 secara berurutan merupakan input dan output dari sebuah inverting amplifier yang dapat dikonfigurasi penggunaannya sebagai on chip oscillator seperti yang ditunjukkan pada gambar 2-3a. X-TAL 1 dan X-TAL 2 ini dapat menggunakan sebuah kristal quartz maupun resonator keramik.

Untuk memberikan AT89S51 dari sumber clock external. Maka pin X-TAL 2 dibiarkan tidak berhubungan dan X-TAL 1 dihubungkan dengan sumber clock external seperti pada gambar 2-3b. Rangkaian ini tidak melakukan duty cycle dari setiap sinyal clock internal, karena input bagi masukan rangkaian clock internal dihubungkan ke flip-flop pembagi dua, tetapi spesifikasi nilai tegangan pada saat tinggi dan rendah, maksimum dan minimumnya harus diberikan.



Gambar 2.3. Karakteristik Oscilator ^[3]

2.1.5. Organisasi Memory

Didalam AT89S51 memiliki ruangan alamat telah dibedakan untuk program memory dan data memory. Pemisahan memori program dan data tersebut membolehkan memori data diakses dengan alamat 8-bit, sehingga dapat dengan

cepat dan mudah disimpan dan dimanipulasi oleh CPU 8-bit. Namun demikian, alamat memori data 16-bit bisa juga dihasilkan melalui register DPTR.

2.1.5.1. Program Memory Internal

AT89S51 memiliki program memory internal sebesar 4Kbyte dengan ruang alamat 0000H-0FFFH. Jika alamat-alamat program lebih tinggi daripada 0FFFH, yang melebihi kapasitas ROMFash memory internal menyebabkan AT89S51 secara otomatis mengambil Code Byte dari program memory external. Code Byte juga dapat diambil hanya dari external memory dengan alamat 0000H-FFFFH dengan cara menghubungkan Pin EA ke Ground.

2.1.5.2. Data Memory (RAM) Internal

Ruang alamat bawah memory data (RAM) internal dengan kapasitas 128 byte yaitu 00H-7FH yang terbagi atas 3 daerah, yaitu :

- 4 Bank Register

Setiap bank terdiri dari 8 register (R0-R7). Sehingga jumlah register untuk keempat bank register (bank 0 - bank 3) menjadi 32 buah register yang menempati ruang alamat 00H-1FH. Mengaktifkan salah satu bank register dapat dilakukan dengan mengatur RS0-RS1 pada PSW (Program Status Word).Ini bisa dilihat pada tabel 2.1 berikut ini :

Tabel 2-1. Pengaturan RS0-RS1 Untuk Select Register Bank ^[5]

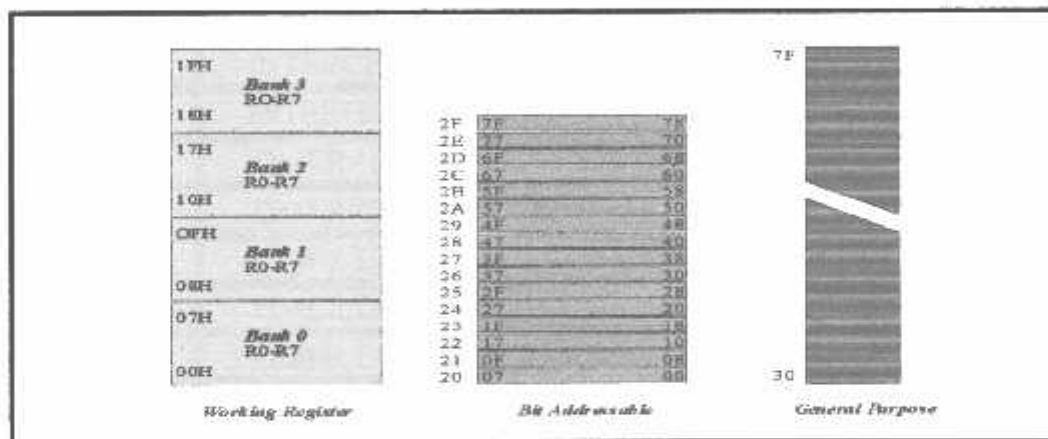
RS1	RS0	Select Register Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

- Bit Addressable.

Terdiri dari 16 byte yang berada pada alamat 20H-2FH. Masing-masing 128 bit lokasi ini dapat di alamatkan secara langsung.

- General Purpose.

Terdiri dari 80 byte yang menempati alamat 30H-7FH yang dapat dialamati secara langsung dan dapat digunakan untuk keperluan umum (General Purpose RAM). Misalkan digunakan untuk lokasi stack, seperti yang terlihat pada gambar 2.4, yaitu memory data (RAM) internal dengan kapasitas 128 byte yaitu 00H-7FH yang terbagi atas 3 daerah dibawah ini:



Gambar 2.4. Organisasi RAM Internal [3]

2.1.5.3. SFR (Special Function Register).

Untuk mengoperasikan AT89S51 yang tidak menggunakan alamat internal RAM (00-7FH) dilakukan oleh SFR yang beraddress 80H-FFH, tetapi tidak semua address tersebut digunakan sebagai SFR. Fungsi-fungsi SFR dijelaskan oleh tabel 2 - 2 sebagai berikut.

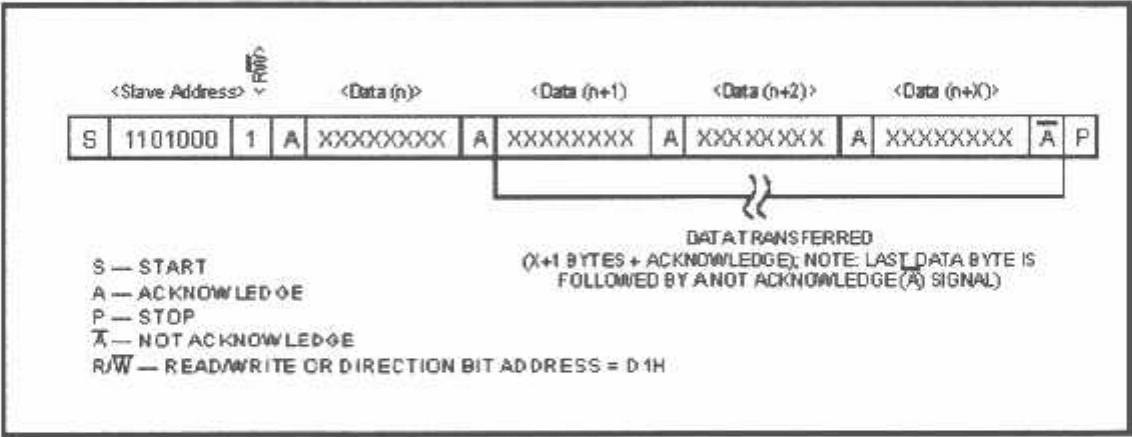
Pada sistem jalur data I2C, terdapat dua tipe arah proses pengiriman data yaitu:

1. **Data transfer dari master menuju *slave*.** *Byte* pertama yang dikirimkan oleh master menuju *slave* adalah alamat *slave*. Lalu selanjutnya adalah *byte-byte* data. *Slave* membalas dengan bit *acknowledge* setiap berhasil menerima 1 *byte* data.
2. **Data transfer dari *slave* menuju master.** *Byte* pertama (alamat *slave*) dikirimkan oleh master. Kemudian *slave* yang mempunyai alamat yang dituju oleh master membalas dengan bit *acknowledge*. Lalu diikuti dengan proses pengiriman *byte-byte* data dari *slave* menuju master. Master membalas dengan mengirimkan bit *acknowledge* setiap berhasil menerima 1 *Byte*. Untuk mengakhiri proses pengiriman data master membalas dengan mengirimkan bit "*not acknowledge*" kepada *slave*.

Pada aplikasi ini DS1307 bekerja dengan dua mode, yaitu:

1. **Mode *Slave* Penerima (Master Menulis Pada *Slave*):** Data serial dan clock diterima melalui SDA dan SCL. Setiap menerima *byte* data DS1307 akan merespon dengan membangkitkan bit *acknowledge*. Untuk mengawali proses pengiriman data dari master menuju *slave* diawali dengan kondisi START dan diakhiri dengan kondisi STOP, seperti yang terlihat pada gambar 2.6, setiap *slave* akan membaca alamat yang dituju oleh master dan memeriksa apakah alamat tersebut sama dengan alamat *Slave* tersebut. *Byte* alamat *slave* adalah *Byte* pertama yang diterima *slave* setelah master membangkitkan kondisi START. *Byte* alamat terdiri dari 7 bit data, untuk DS1307 *Byte* alamat tersebut adalah 1101000b, dan diikuti

START. *Byte* alamat DS1307 terdiri dari 7-bit alamat dan 1 bit arah. 7-bit alamat tersebut adalah 1101000 dan bit arah tersebut (R/W) adalah 0 untuk *read*. Setelah menerima dan mengolah data alamat, DS1307 akan membalas dengan membangkitkan bit *acknowledge* pada pin SDA. Kemudian DS1307 mulai mengirimkan data dimulai dari data yang terdapat pada alamat yang ditunjuk oleh register pointer. Nilai register pointer secara otomatis akan bertambah setiap terjadi proses pembacaan 1 *Byte* data. Untuk mengakhiri proses pengiriman data maka master harus mengirimkan tanda “not *acknowledge*” kepada *slave*, seperti contoh gambar dibawah ini:



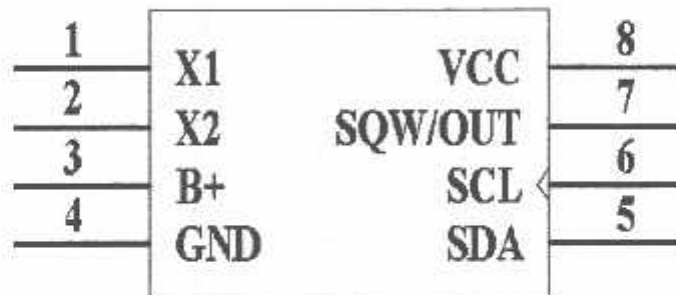
Gambar 2.7. Format Data ^[12]

Untuk dapat mengambil nilai waktu dan tanggal maka master harus melakukan proses pembacaan data (*Read*) pada *slave* (DS1307), dengan alamat register sesuai dengan tabel 2-3. Setiap nilai-nilai waktu atau tanggal disimpan pada register yang mempunyai alamat yang berbeda-beda, misalnya untuk register detik yang menyimpan nilai detik, menempati alamat register 00h.

Tabel 2-3. Setting DS1307 ^[12]

ADDRESS	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	FUNCTION	RANGE
00H	CH	10 Seconds			Seconds				Seconds	00-59
01H	0	10 Minutes			Minutes				Minutes	00-59
02H	0	12	10 Hour	10 Hour	Hours				Hours	1-12 + AM/PM 00-23
		24	PM/AM							
03H	0	0	0	0	0	DAY			Day	01-07
04H	0	0	10 Date		Date				Date	01-31
05H	0	0	0	10 Month	Month				Month	01-12
06H	10 Year				Year				Year	00-99
07H	OUT	0	0	SQWVE	0	0	RS1	RS0	Control	—
08H-3FH									RAM 56 x 8	00H-FFH

Sedangkan untuk gambar 2.8 dibawah ini menunjukkan konfigurasi pin pada RTC DS1307



Gambar 2.8. RTC DS1307 ^[12]

Konfigurasi Pin DS1307:

- **Vcc**, + 5 Volt
- **X1 dan X2**, merupakan jalur untuk koneksi ke kristal 32,768 kHz.
- **B+(V_{BAT})**, Input untuk baterai *back-up* tegangan sebesar 3Volt.
- **SQW/OUT**, *Square Wave/Output Driver*
- **SCL**, *Serial Clock*
- **SDA**, *Serial Data*
- **GND**, *Ground*

2.3. Speak Call ISD 25120(IC Data Perekam Suara)

IC penyimpanan suara yang digunakan merupakan jenis EEPROM (*Electrically Erasable Programmable Read Only Memory*) yaitu ROM yang dapat diprogram, dihapus dan diprogram ulang secara elektrik dengan arus listrik, bukan sinar *ultraviolet*. IC ISD (*Information Storage Device*), yang dipakai yaitu ISD 25120. IC ini dapat merekam pesan maksimal 120 detik dan dapat dikaskade sehingga pesan yang disimpan dapat diperpanjang sesuai dengan keinginan kita dengan 160 alamat yang berbeda.

Didalam ISD 25120 dilengkapi dengan internal *amplifier*, *internal automatic gain control (AGC)*, *filter antialiasing* (perata) dan *speaker amplifier* (penguat speaker). Secara keseluruhan seri ISD 25120 dapat melakukan sebuah perekaman atau pemutaran ulang pesan dengan komponen sederhana seperti mikropon, *speaker*, beberapa komponen penunjang, duah buah saklar dan sumber tegangan.

Rekaman akan disimpan dalam sel memori yang tidak mudah hilang (*non volztile*), memberikan tempat penyimpanan yang masih kosong. Cara unik ini yang membuat ISD disebut *Direct Analog Stroge Technology (DAST)* atau teknik penyimpanan analog langsung, dengan jalan sinyal suara (*voice*) dan bunyi disimpan secara langsung dalam bentuk analog, kedalam memori EPROM. Penyimpanan analog langsung memungkinkan reproduksi suara secara alami dalam satu chip tunggal.

mode operasi normal dan pengalamatan secara tidak langsung (*Simultaneously*).

❖ **Vssd dan Vssa (Ground) Pin 12 dan 13**

Sama seperti Vccd dan Vcca, analog input dan digital sirkuit di dalam ISD 25120 menggunakan *bus ground* yang terpisah untuk meminimalisasi *noise*. Pin ini harus dihubungkan sedekat mungkin dengan *ground*.

❖ **Speaker Output (SP +, SP -) Pin 14 dan 15**

Pin SP + dan SP – digunakan untuk mengeluarkan suara yang telah direkam ke *speaker* atau ke *device* lainnya. Output ini mempunyai impedansi sebesar 16 Ohm.

❖ **Microphone Input (mic) Pin 17**

Kaki mikropon ini terhubung dengan Vcc melalui beberapa kapasitor yang terhubung secara seri, bersamaan dengan resistor 10 K Ω yang berada didalam chip (*internal*). Harga dari kapasitor dari dalam perancangan ini menggunakan harga kapasitor sesuai dengan yang tertera dalam rangkaian data sheet ISD 25120.

❖ **Microphone Reference (MIC REF) Pin 18**

Ketika MIC REF menghubungkan antara Vcc dengan mikropon *ground*, maka tingkat *noise* selama perekaman dapat dikurangi. *Noise* itu disebabkan oleh *pre-amplifier* yang terdapat didalam *chip*. Bila pin ini tidak digunakan, maka tidak boleh dihubungkan dengan sinyal atau dengan tegangan apapun, harus dalam keadaan terbuka.

❖ ***Automatic Gain Control (AGC) Pin 19***

Kegunaan dari AGC adalah untuk menambah atau mengurangi secara otomatis penguatan (*Gain*) dari *pre-amplifier* yang juga meluaskan batas dari sinyal input yang dapat digunakan oleh mikropon tanpa terjadi distorsi. AGC ini dapat secara dinamis meluaskan batas dari suara yang terekam baik itu suara bisikan sampai suara yang keras. Untuk menggunakan fasilitas AGC ini, resistor dan kapasitor luar (eksternal) harus dihubungkan secara *parallel* antara pin AGC dengan *ground*. Harga yang direkomendasikan adalah $R = 470\text{ K}\Omega$ dan $C = 4,7\text{ }\mu\text{F}$ (Dalam perancangan ini juga dipakai harga seperti diatas sama dengan data sheet ISD).

❖ ***Analog Input (ANA IN) Pin 20***

Kapasitor eksternal (luar) menghubungkan antara ANA IN ke ANA OUT pin harga-harga dari kapasitor luar bersama dengan $3\text{ K}\Omega$ input impedansi di ANA IN dapat dipilih sendiri untuk memberikan keadaan *cut off* (terputus) pada frekuensi rendah sampai pada *pass band* suara. ANA IN juga dapat digunakan pada input sumber *alternative* dari sinyal analog pada sinyal mikropon terus ke kapasitor kopling.

❖ ***Analog Output (ANA OUT) Pin 21***

Sinyal dari mikropon dikuatkan dan dikeluarkan melalui ANA OUT pin. Penguatan tegangan dari *pre-amp* tergantung dari tingkat tegangan AGC (*Automatik Gain Control*) pin. *Pre-amplifier* ini mempunyai penguatan maksimum sekitar 24 dB untuk tingkat masukan kecil.

❖ **Record (REC) Pin 27**

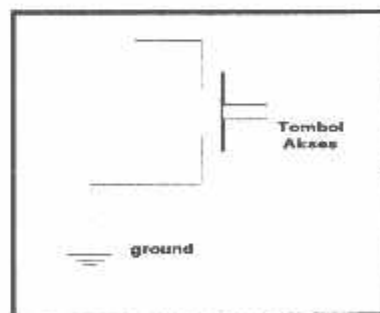
Input sinyal REC akan aktif dalam kondisi *low*. ISD 25120 akan merekam bila REC dalam keadaan *low*, dan sinyal ini harus terus dalam keadaan *low* bila ingin terus merekam. Jika input REC ini tertekan *low* dalam keadaan masih memutar ulang pesan (*playback*), maka *playback* akan berhenti dan ISD akan merekam.

❖ **VCCA Dan VCCD Pin 16 Dan 28**

Analog dan digital sirkuit yang terdapat didalam chip ISD 25120 menggunakan bus power yang terpisah untuk meminimalisasi noise. Pin power ini harus dihubungkan sedekat mungkin dengan sumber tegangan.

2.4. Tombol Akses (Push-button)

Tombol atau saklar digunakan sebagai tombol kontrol masukan untuk mikrokontroller. Tombol atau saklar berfungsi sebagai penghantar aliran arus pada suatu rangkaian. Jika kondisi saklar atau tombol terbuka (tidak ditekan), maka akan berlogika 1, sebaliknya apabila tombol ditekan maka akan berlogika 0 dikarenakan rangkaian akan terhubung dengan ground. Dibawah ini adalah gambar peraga dari Tombol Akses atau Push-button



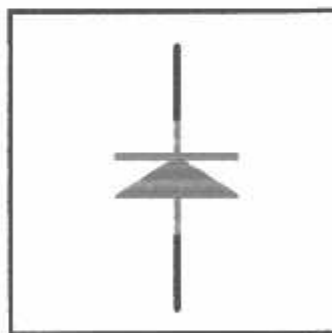
Gambar 2.10. Peraga Tombol / saklar ^[7]

2.5. Dioda

Semua dioda prinsip kerjanya adalah sebagai penyearah. Tetapi karena proses pembuatan, bahan dan penerapannya yang berbeda-beda, maka namanya pun berbeda-beda serta memiliki karakteristik yang berbeda pula. Macam-macam dioda antara lain: Dioda Penyearah, Dioda Zener, Dioda Tunnel, Dioda Varaktor dan Dioda Photo.

2.5.1. Dioda Penyearah

Kebanyakan dioda penyearah dibuat dari bahan silicon, mengingat kemampuan dan keandalannya yang tinggi. Mampu dilewati arus yang besar dan dapat bekerja pada suhu yang tinggi. Keuntungan ini dimungkinkan karena elektron tersebut baru bias terlepas dari intinya oleh kekuatan atau energi yang sangat besar. Penyearah disini berarti merubah arus bolak-balik (AC) menjadi arus searah (DC) yang berfrekuensi rendah. Gambar simbol dioda seperti dibawah ini:



Gambar 2.11. Simbol Dioda ^[7]

2.6. Kapasitor

Kapasitor adalah komponen elektronika yang terdiri dari plat konduktor yang satu disekat terhadap yang lainnya dengan isolator (dielektrik) dan berfungsi sebagai pelawan untuk arus bolak-balik, filter, tuning, kopel antar rangkaian, pembangkit gelombang sinus dan konduktor khusus. Untuk menentukan banyaknya muatan listrik yang dapat disimpan oleh kapasitor, dapat dirumuskan sebagai berikut:

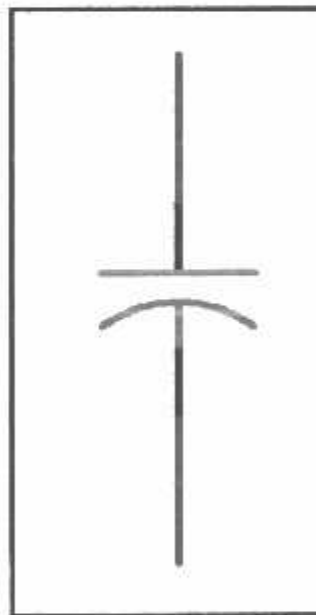
$$Q = C \times V \dots\dots\dots (2-1)$$

Dimana :

Q = Banyaknya muatan dalam Coulomb

C = Kapasitor dalam farad

V = Tegangan yang ada diantara keping keping dalam Volt



Gambar 2.12. Simbol Kapasitor ^[7]

Sebuah kapasitor dikatakan mempunyai kapasitas satu farad apabila muatan sebesar satu coulomb membuat tegangan naik sebesar satu volt, apabila farad ternyata terlampau besar sehingga digunakan mikrofard dan satuan lainnya yaitu nano farad dan pikofard. Kapasitor yang berfungsi sebagai penyimpan tenaga listrik dalam praktek dapat digunakan untuk pewaktu, untuk pewaktu kapasitor dengan resistor yang disebut sebagai rangkaian sebagai rangkaian RC.

Rumus pada rangkaian RC sebagai berikut:

$$RC = R \times C \dots\dots\dots(2-2)$$

Dimana:

R = Tahanan dalam kapasitor

C = Kapasitor dalam farad

RC = Konstanta waktu dalam detik

Dalam kapasitor terdapat reaktansi kapasitor, jika kapasitor tersebut diterapkan pada arus bolak balik sinus, tegangan akan tertinggal sejauh 90°.

Untuk menentukan besarnya nilai reaktansi adalah:

$$X_c = \frac{1}{3,14.F.C} \dots\dots\dots(2-3)$$

Dimana:

Xc = Reaktansi kapasitor

F = Frekuensi dalam ohm

C = kapasitas dalam farad

Untuk mencari kapasitansi dari kapasitor dalam rangkaian dapat dicari dengan rumus:

$$F = \frac{1}{2\pi} = \frac{1}{1,4.C.R} \text{ Hz} \dots\dots\dots(2-4)$$

Dimana:

F = frekuensi dengan satuan Hz

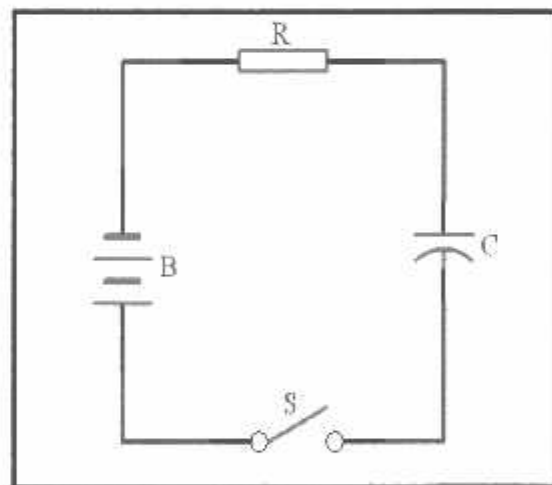
T = Amplitudo

C = besar kapasitor dengan satuan farad

R = besar hambatan dengan satuan Ω

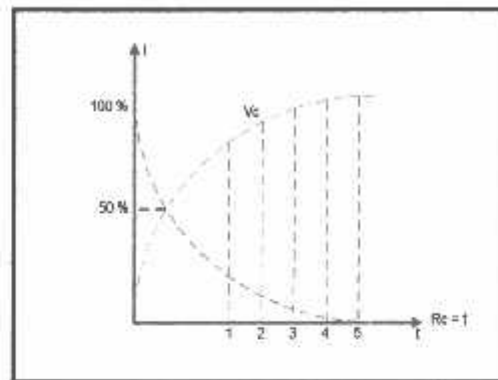
2.6.1. Cara Kerja Kapasitor

Terjadinya perubahan tegangan pada kapasitor menyebabkan perubahan arus listrik, sedangkan perubahan tersebut berbanding lurus dengan perubahan tegangan, energi akan timbul kembali pada saat tegangan berkurang menjadi nol. Pada saat saklar (S) posisi ON, arus dari tegangan V_2 mengalir ke dalam kapasitor dan beberapa saat kapasitor dapat terisi penuh. Ini dapat dilihat pada gambar 2.13 yang menunjukkan pengisian pada kapasitor. Dalam kondisi ini berarti $V_s = V_c$ dan $I = 0$.



Gambar 2.13. Rangkaian Pengisian Kapasitor ^[7]

Tegangan dan arus pengisian kapasitor dapat dilihat pada grafik 2.14 dibawah ini :



Gambar 2.14. Grafik Tegangan dan Arus Pengisian Kapasitor ^[7]

Berdasarkan grafik diatas, dapat kita ketahui bahwa setelah terjadi pengisian kapasitor secara penuh dan kapasitor akan membuang melalui R, maka C tidak seketika menjadi netral dan dalam menetralkan muatan tersebut diperlukan waktu tertentu.

2.7. Resistor

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

Warna,nilai,factor pengali dan toleransi dari sebuah resistor dapat dilihat pada tabel 2-4,yang tertera dibawah ini :

Tabel 2-4. Tabel Gelang Warna Resistor ^[7]

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

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

Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang

kode warna untuk memudahkan pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohm meter. Kode warna tersebut adalah standar manufaktur yang dikeluarkan oleh EIA (Electronic Industries Association) seperti yang ditunjukkan pada tabel 2-4.

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

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

Misalnya resistor dengan gelang kuning, violet, merah dan emas. Gelang berwarna emas adalah gelang toleransi. Dengan demikian urutan warna gelang resistor ini adalah, gelang pertama berwarna kuning, gelang kedua berwarna violet dan gelang ke tiga berwarna merah. Gelang ke empat tentu saja yang berwarna emas dan ini adalah gelang toleransi. Dari tabel 2-4 diketahui jika gelang toleransi berwarna emas, berarti resistor ini memiliki toleransi 5%. Nilai resistansinya dihitung sesuai dengan urutan warnanya. Pertama yang dilakukan

adalah menentukan nilai satuan dari resistor ini. Karena resistor ini resistor 5% (yang biasanya memiliki tiga gelang selain gelang toleransi), maka nilai satuannya ditentukan oleh gelang pertama dan gelang kedua.

Masih dari tabel 2-4 diketahui gelang kuning nilainya = 4 dan gelang violet nilainya = 7. Jadi gelang pertama dan kedua atau kuning dan violet berurutan, nilai satuannya adalah 47. Gelang ketiga adalah faktor pengali, dan jika warna gelangnya merah berarti faktor pengalinya adalah 100. Sehingga dengan ini diketahui nilai resistansi resistor tersebut adalah nilai satuan x faktor pengali atau $47 \times 100 = 4.7K \text{ Ohm}$ dan toleransinya adalah 5%.

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

Umumnya di pasar tersedia ukuran 1/8, 1/4, 1, 2, 5, 10 dan 20 watt. Resistor yang memiliki disipasi daya 5, 10 dan 20 watt umumnya berbentuk kubik memanjang persegi empat berwarna putih, namun ada juga yang berbentuk silinder. Tetapi biasanya untuk resistor ukuran jumbo ini nilai resistansi dicetak langsung dibadannya, misalnya 100 Ω 5W.

2.8. LCD (Liquid Crystal Display) M1632

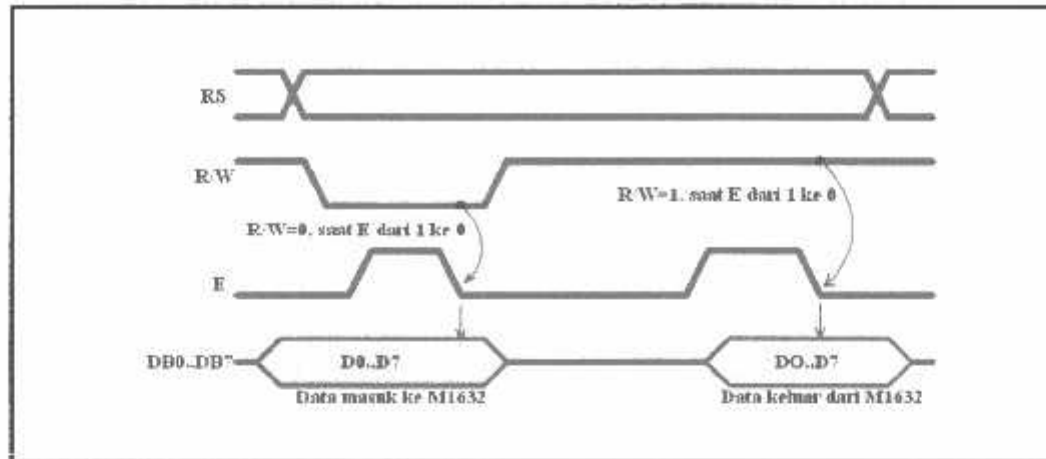
LCD Display Module M1632 buatan Seiko Instrument Inc. terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing-masing baris bisa

menampung 16 huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik pada panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi M1632 dengan mikrokontroler yang memakai tampilan LCD itu. Dengan demikian pemakaian M1632 menjadi sederhana, sistem lain yang M1632 cukup mengirimkan kode-kode ASCII dari informasi yang ditampilkan seperti layaknya memakai sebuah printer.

2.8.1. Sinyal interface M1632

Untuk berhubungan dengan mikrokontroler memakai, M1632 dilengkapi dengan 8 jalur data (**DB0..DB7**) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor. Kombinasi lainnya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas Intel dengan kombinasi sinyal **WR** dan **RD**.

RS, singkatan dari Register Select, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan. Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan. Proses mengirim/mengambil data ke/dari M1632 digambarkan dalam gambar 2.15 bisa dijabarkan sebagai berikut :



Gambar 2.15. Mengirim/Mengambil Data Ke/Dari M1632 [6]

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

2.8.2. Interface Ke MCS-51

Sinyal-sinyal M1632 yang mengikuti standar teknik interface Motorola, tidak sesuai dengan sinyal dari MC-S51, dengan demikian sinyal-sinyal itu disimulasikan melalui port MCS51. Sebagai contoh gambar 2.17 memperlihatkan hubungan AT89C2051 dengan M1632, dalam gambar tersebut **P3.7** dipakai

Sebelum pengiriman data, Akumulator A sudah terlebih dulu diisi dengan data yang akan dikirim. Data yang dikirim dengan sub-rutin **Kirim Perintah** akan diterima M1632 sebagai perintah untuk mengatur kerja M1632, dan data yang dikirim dengan sub-rutin **Kirim ASCII** akan ditampilkan di panel LCD.

Setelah M1632 menerima data, M1632 memerlukan waktu antara 40 sampai 1640 mikro-detik untuk mengolahnya, selama waktu itu M1632 untuk sementara tidak bisa menerima data, hal ini ditandai dengan bit 7 dari Register Status = '1'.

Proses pengiriman data ke M1632 dijelaskan sebagai berikut :

1. Perbedaan sub-rutin **Kirim Perintah** dan **Kirim ASCII** terletak pada nilai **RS** pada saat sub-rutin itu bekerja. Sub-rutin **Kirim Perintah** bekerja dengan **RS='0'** (baris 6), data yang dikirim AT89S51 diterima M1632 sebagai perintah untuk mengatur kerja M1632. Sub-rutin **Kirim ASCII** bekerja dengan **RS='1'** (baris 10), data yang dikirim AT89S51 diterima M1632 sebagai kode ASCII yang akan ditampilkan
2. Sinyal **RW** di-nol-kan agar M1632 siap menerima data (baris 12), setelah itu data di akumulator A diletakkan di **D0..D7** (Port 1 dari AT89S51) di baris 13, baris 14 dan 15 membangkitkan sinyal sinkronisasi **E** dengan cara membuat **P3,4** menjadi '1' dan kemudian kembali menjadi '0'. Saat sinyal **E** kembali menjadi '0' data di Port 1 akan diterima oleh M1632.
3. Selesai mengirim data, program harus menunggu sampai M1632 siap menerima data lagi. Hal ini dilakukan dengan cara mengambil Status M1632 (baris 18), dan memeriksa bit 7-nya (baris 19), selama bit 7 bernilai '1' berarti

M1632 masih sibuk mengurus diri, dan program menunggu di Tunggu Dulu.

Proses pengambilan data dari M1632 dijelaskan sebagai berikut :

1. Seperti bahasan di atas, **RS** dipakai untuk memilih Register Perintah/Status, sub-rutin **Ambil Status** bekerja dengan **RS=0** dan sub-rutin **Ambil ASCII** bekerja dengan **RS='1'**.
2. Sinyal **RW** di-satu-kan agar M1632 siap memberi data (baris 29), setelah sinyal **E** menjadi '1' (baris 30) M1632 akan meletakkan data di **D0 .. D7**, setelah data ini diambil (baris 31) sinyal **E** dikembalikan menjadi '0'

Berikut adalah Potongan Program 1 AT89S51 dengan M1632 :

```
01:      E      bit    P3.4          ; sinyal E   di P3.4
02:      RW     bit    P3.5          ; sinyal R/W di P3.5
03:      RS     bit    P3.7          ; sinyal RS   di P3.7
04:      ;Comment In Here
05:      KirimPerintah:
06:      CLR    RS                   ; RS=0 : register perintah
07:      SJMP   OutByte
08:      ;
09:      KirimASCII:
10:      SETB   RS                   ; RS=1 : Display Data RAM
11:      OutByte:
12:      CLR    RW                   ; RW = '0', kirim data
13:      MOV    P1,A                 ; siapkan data di D0..D7
14:      SETB   E                    ; buat pulsa positif
```

```

15:      CLR  E                      ; sesaat
16:      :
17:      TungguDulu:
18:      ACALL AmbilStatus
19:      JB   A.7,TungguDulu
20:      RET
21:      AmbilStatus:
22:      CLR  RS                      ; RS=0 : register status
24:      SJMP InByte
26:      AmbilASCII:
27:      SETB RS                      ; RS=1 : Display Data RAM
28:      InByte:
29:      SETB RW                      ; RW = '1', ambil data
30:      SETB E                      ; minta data pada M1632
31:      MOV  A,P1                    ; ambil data
32:      CLR  E                      ; kembalikan E ke '0'
33:      RET

```

2.7.3. Mengatur tampilan M1632

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

Setelah diberi sumber daya, ada beberapa langkah persiapan yang harus dikerjakan dulu agar M1632 bisa dipakai, langkah-langkah tersebut antara lain adalah:

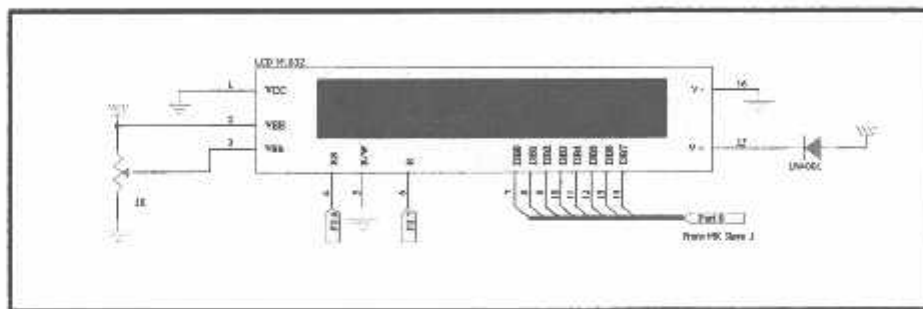
1. Tunggu dulu selama 15 mili-detik atau lebih.
2. Kirimkan perintah 30h, artinya transfer data antar M1632 dan mikrokontroler dilakukan dengan mode 8 bit
3. Tunggu selama 4.1 mili-detik
4. Kirimkan sekali lagi perintah 30h
5. Tunggu lagi selama 100 mikro-detik

Setelah langkah-langkah tersebut di atas M1632 barulah bisa menerima data dan menampilkannya dengan baik. Pada awalnya tampilan akan nampak kacau, dengan demikian perlu segera dikirim perintah menghapus tampilan dan lain sebagainya, sesuai dengan petunjuk yang ada di Lembar Data.

Di atas dipakai AT89S51 sebagai contoh, meskipun demikian semua yang dibahas di atas sepenuhnya bisa dipakai pada mikrokontroler MCS 51. Dalam pemakaiannya karena berbagai macam alasan, bisa saja sinyal **E**, **RW** dan **RS** tidak disimulasikan di **P3.4**, **P3.5** dan **P3.7**. Hal ini bisa diselesaikan dengan melakukan beberapa penyesuaian, yakni tentukan dulu perubahan rangkaian sesuai dengan keadaan yang ada, dan perubahan rangkaian itu harus di sesuaikan di baris 1 sampai 3 pada potongan program di atas.

M1632 mempunyai 8 jalur data dan memerlukan 3 jalur kontrol, dalam suatu rangkaian yang memakai banyak port dari MC-S51, bisa terjadi kekurangan port untuk menghubungkan MCS51 ke M1632. Jika sampai terjadi hal semacam ini bisa ditempuh hal hal berikut :

1. M1632 dipakai dalam mode data 4 bit, yakni hanya memakai jalur data **D0..D3**
2. Dengan sedikit tambahan rangkaian sinyal **WR** dan **RD** diubah menjadi sinyal **E** dan **R/W** gaya Motorola, sehingga tidak perlu menyediakan port untuk men-simulasikan sinyal-sinyal tersebut. Berikut adalah gambar 2.17 rangkaian LCD dengan komponen-komponen pendukung dengan pin-pin yang akan dihubungkan pada mikrokontroler MCS 51 :



Gambar 2.17. Rangkaian LCD M1632 ^[6]

LCD M1632 mempunyai spesifikasi sebagai berikut :

1. Memiliki 16 karakter dan dua baris tampilan yang terdiri dari 5 x 7 dot matrik ditambah dengan kursor.
2. Pembangkit karakter ROM untuk 192 jenis karakter.
3. Pembangkit karakter RAM untuk 8 jenis karakter.
4. 80 x 8 display data RAM (max 80 karakter).
5. Isolator didalam modul.
6. Memerlukan catu daya ± 5 volt.
7. Otomatis reset saat catu daya dinyalakan.

Adapun fungsi dari pin-pin pada LCD dapat dilihat pada tabel 2-5 dibawah ini :

Tabel 2-5. Fungsi Pin – Pin LCD ^[6]

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

Pada LCD juga terdapat instruksi – instruksi sebagai berikut :

- Display clear : membersihkan tampilan yang ada pada LCD.
- Cursor home : hanya membersihkan tampilan dan kursor kembali ke semula.
- Empty mode Set : layar beraksi sebagai tampilan tulis.

S : 1/0 = menggeser layar.

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

- Display On/Off kontrol.

D : 1 – layar on

D : 0 = layar off

C : 1 = kursor on

C : 0 = kursor off

B : 1 = kursor berkedip-kedip

B : 0 = kursor tidak berkedip – kedip

- Cursor Display Shift

S/C : 1 = LCD diidentifikasi sebagai layar

S/C : 0 = LCD diidentifikasi sebagai kursor

R/L : 1 = menggeser satu spasi ke kanan

R/L : 0 = menggeser satu spasi ke kiri

- Fuction Set

DL : 1 = panjang data LCD pada 8 bit

DL : 0 = panjang data LCD pada 4 bit

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

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

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

- CG RAM address set : menulis alamat RAM ke karakter
- DD RAM address set : menulis alamat RAM ke tampilan

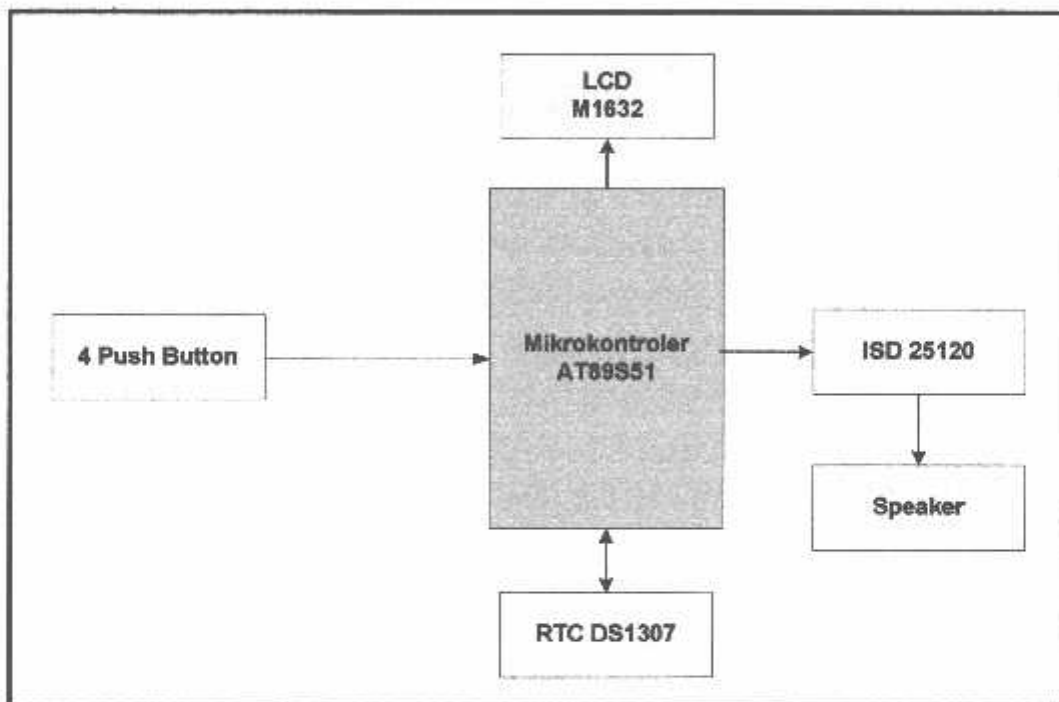


BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Diagram Blok

Secara umum Perancangan Bel/Alarm Sekolah Secara Otomatis Berbasis Mikrokontroler AT89S51 yang akan dibuat, digambarkan secara diagram blok sebagai berikut :



Gambar 3.1. Blok Diagram

- Mikrokontroler AT89S51 sebagai pengendali utama keseluruhan rangkaian.
- RTC DS1307 sebagai pewaktu yang menghitung berdasarkan tahun, tanggal, jam, menit dan detik.
- ISD 25120 digunakan untuk merekam suara sehingga dihasilkan output suara.
- Loud Speaker digunakan untuk mengeluarkan output suara.

- Mic digunakan untuk proses perekaman suara pada ISD.
- 4 Push-button digunakan sebagai input untuk seting sistem bel sekolah, yaitu tombol menu, tombol up, tombol down, dan tombol reset.
- Buzzer digunakan sebagai alarm bel sekolah.
- LCD digunakan sebagai display.
- Power Supply digunakan untuk memberikan tegangan kerja yang dibutuhkan untuk keseluruhan rangkaian.

3.2. Prinsip Kerja Alat

Pertama-tama kita dapat melakukan seting waktu bel/alarm melalui tombol push button menu, up, dan down, dan kemudian memasukkan input suara agar nantinya dihasilkan output suara sebagai alarm. Pada saat waktu telah sesuai dengan input nilai waktu alarm, maka secara otomatis mikrokontroler akan mengeluarkan suara alarm sesuai dengan seting alarm yang telah di inputkan.

3.3. Mikrokontroler AT89S51

Mikrokontroller AT89S51 dirancang untuk dapat berdiri sendiri, karena sudah terdapat *EPROM*, *RAM* serta *Port I/O internal*. Untuk berhubungan dengan peralatan luar *chip* dibutuhkan 3 *Bus* yaitu :

1. *Data Bus*

Yaitu jalur untuk *input - output* data yang lebarnya sesuai dengan data yang diolah oleh mikrokontroler, yaitu 8 *bit*.

2. *Address Bus*

Yaitu jalur *input - output* atau dari *memori* yang dihubungi, sehingga pada suatu saat hanya ada satu *device* yang berhubungan dengan *CPU*. Lebar *address bus* mikrokontroler AT89S51 adalah 16 *bit* (A0 - A15).

3. *Control Bus*

Berfungsi sebagai pengatur *sinkronisasi* hubungan antara *CPU* dengan *device* Luar.

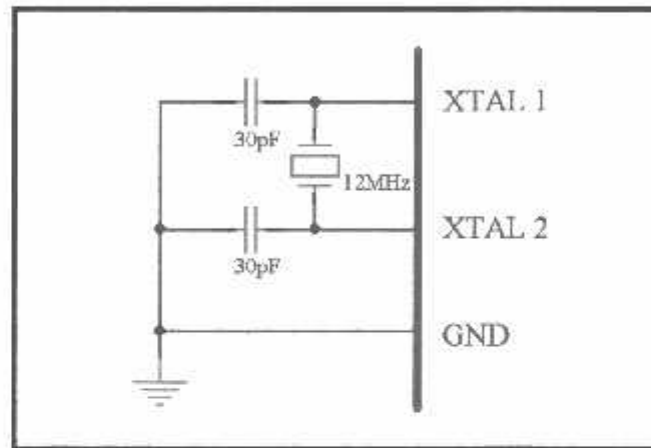
3.3.1. Pemetaan *Memori*

Mikrokontroller AT89S51 memiliki 16 *bit address* (A0 - A15) dengan demikian kapasitas maksimumnya adalah $2^{16} = 65536 \text{ byte} = 64 \text{ Kbyte}$ dengan alamat 0000H-FFFFH. Jika alamat-alamat program lebih tinggi dari 0FFFFH, yang melebihi kapasitas *RAM internal* menyebabkan mikrokontroler secara otomatis mengambil *code byte* dari program *memori* eksternal. *Code byte* juga hanya diambil dari *memori* eksternal dengan alamat 0000H-0FFFFH dengan menghubungkan *Pin* EA ke *ground*.

Dalam perancangan ini hanya menggunakan 4 *Kbyte*, karena program sudah mencukupi, sehingga *pin* EA dihubungkan ke *VCC*. Mikrokontroler AT89S51 memiliki 4 *Kbyte memori internal* yang dapat diprogram dan dihapus sesuai dengan keinginan, dan bersifat *non volatile* (tidak hilang pada saat catu daya terputus).

3.3.2. Rangkaian Clock

Mikrokontroler AT89S51 ini memiliki *internal clock*, yang berfungsi sebagai sumber *clock*, tapi masih diperlukan rangkaian tambahan untuk membangkitkan *clock* yang diperlukan. Rangkaian ini terdiri dari 2 buah kapasitor dan sebuah kristal dengan ketentuan seperti gambar 3.2 berikut :



Gambar 3.2. Rangkaian Clock AT89S51

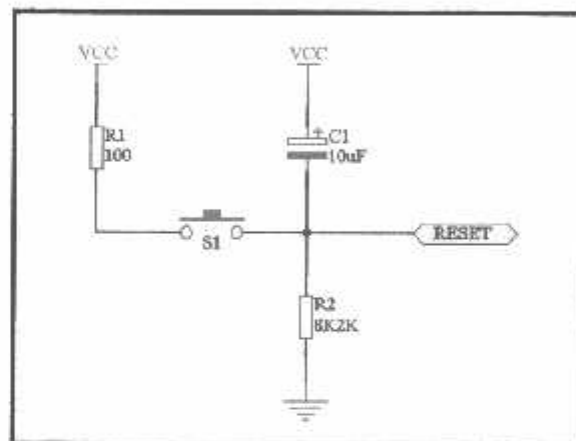
3.3.3. Rangkaian Reset

Rangkaian *reset* bertujuan agar mikrokontroler dapat menjalankan proses mulai dari alamat awal. Rangkaian *reset* untuk mikrokontroler dirancang agar mempunyai kemampuan *power on reset* yaitu *reset* yang terjadi pada saat sistem dinyalakan untuk pertama kalinya. *Reset* juga dapat dilakukan secara manual dengan menyediakan tombol yang berupa *switch*.

Jika saklar S1 ditekan , *reset* bekerja secara manual, aliran arus akan mengalir dari *VCC* melalui R1 menuju kaki RST. Tegangan di RST atau VR2 akan berubah menjadi :

$$\begin{aligned}
 VR2 &= \frac{R2 \times VCC}{R1 + R2} \\
 &= \frac{8200 \times 5}{100 + 8200} \\
 &= 4,94 \text{ V}
 \end{aligned}$$

Saat saklar dilepas, aliran arus dari VCC melalui $R1$ akan berhenti dan tegangan pada kaki RST akan turun menuju ke nol dan proses *reset* selesai. Proses ini dapat dilihat pada gambar 3.3 dibawah ini :



Gambar 3.3. Rangkaian *Reset*

3.3.4. Hubungan *Pin* Pada AT89S51

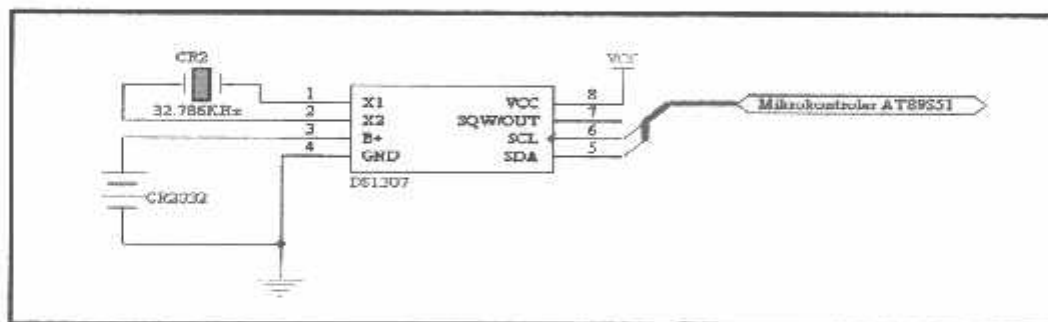
Diskripsi konfigurasi *Pin-Pin* AT89S51 pada rangkaian:

- *Pin* 1 dan 2 : dihubungkan ke *LCD* E dan RS
- *Pin* 3,4 dan 5 : dihubungkan ke rangkaian ISD PR, PD, CE, EOM
- *Pin* 6, 7 dan 8 : dihubungkan *Push-button Menu, Up, dan Down*
- *Pin* 9 : dihubungkan *Push-button reset*
- *Pin* 16 dan 17 : dihubungkan SCL dan SDA DS1307
- *Pin* 18 dan 19 : dihubungkan *X-tal* 12 MHz

Pemilihan penggunaan serial RTC DS1307 pada perancangan ini adalah dikarenakan RTC DS1307 mempunyai beberapa kelebihan sebagai berikut::

- Harga yang relative lebih murah jika dibandingkan dengan RTC lainnya
- Karena akses data dilakukan secara serial, maka hanya butuh 2 pin saja, sehingga akan menghemat port mikrokontroler
- Tidak perlu lagi ada perbaikan penanggalan untuk penggunaan ditahun 2000 ke atas
- Sudah tersedia pin untuk baterai back-up, sehingga tidak perlu lagi dibuatkan rangkaian untuk baterai back-up, apabila kehilangan supply tegangan.

Pada gambar 3.5. dibawah ini menunjukkan Rangkaian Serial RTC DS1307 :



Gambar 3.5. Rangkaian Serial RTC DS1307

Baterai back-up yang digunakan adalah baterai back-up 3V CR2032, yang dapat bertahan untuk masa operasi 10 tahun, kondisi ini amat hemat biaya. Kristal yang digunakan adalah standart quartz kristal dengan dinilai 32,768KHz. Pin SDA dan SCL dari RTC serial DS1307 ini dihubungkan ke port mikrokontroler AT89S51 P3.6 dan P3.7.

Sinyal suara merupakan bentuk sinyal analog kemudian diubah menjadi bentuk digital untuk disimpan ke dalam *memory*. Data-data digital yang sudah tersimpan yang berasal dari data analog (suara) dapat dipanggil kembali dengan memanggil alamat penyimpanan datanya. Proses perekaman pada ISD 25120 adalah sebagai berikut :

1. Pin *Chip Enable* (CE) pin 23 mendapat logika *low*.
2. Memberikan alamat dengan mengatur dip *switch*.
3. Kemudian Pin *Playback/Record* pin 27 mendapat logika *low*.
4. Perekaman dimulai dengan memasukkan data suara pada mikrofon.
5. Pin *Playback/Record* mendapat logika *high* kembali.
6. Mencari alamat terakhir yang ditandai pada *End Of Message* (EOM) pin 25 terjadi pulsa *low* sesaat kemudian *high* kembali.
7. Demikian seterusnya, untuk melanjutkan perekaman alamat terakhir dari sebelumnya diberikan spasi dan memulai perekaman kembali sampai batas waktu dari kemampuan ISD-nya.

Kemudian untuk prosedur pemanggilan data-data suara yang telah direkam adalah sebagai berikut :

1. Pin *Chip Enable* (CE) pin 23 mendapat logika *low*.
2. Memanggil alamat suara yang diinginkan, pada waktu perekaman tadi.
3. Kemudian memberikan logika *low* pada *Playback/Record* pin 27, yang berarti pemanggilan suara dimulai.
4. Menunggu pada *End Of Message* (EOM) pin 25 terdapat pulsa logika *low* sesaat.

5. Bila sudah terdapat pulsa logika *low* sesaat pada EOM, segera diberikan logika *high* pada pin *Chip Enable* CE, ini menandakan akhir dari data suara pada alamat itu.
6. Demikian seterusnya pada alamat-alamat selanjutnya, sesuai dengan data suara yang diinginkan.

3.6. Perancangan Rangkaian LCD

LCD Display Module M1632 buatan *Seiko Instrument Inc.* terdiri dari dua bagian, yang pertama merupakan panel *LCD* sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing-masing baris bisa menampung 16 huruf/angka. Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik pada panel *LCD*, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi M1632 dengan mikrokontroler yang memakai tampilan *LCD* itu.

Untuk berhubungan dengan mikrokontroler pemakai, M1632 dilengkapi dengan 8 jalur data (**DB0..DB7**) yang dipakai untuk menyalurkan kode *ASCII* maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor. Kombinasi lainnya **E** dan **R/W** merupakan sinyal standar pada komponen buatan Motorola. Sebaliknya sinyal-sinyal dari MCS51 merupakan sinyal khas *Intel* dengan kombinasi sinyal **WR** dan **RD**.

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

- *Pin 6* : dihubungkan dengan P1.1 AT89S51
- *Pin 7-14* : dihubungkan dengan AT89S51 (P0.0-P0.7)

Untuk tampilan dipergunakan *LCD* Dot Matrik 2 x 16 karakter. Sinyal-sinyal yang diperlukan oleh *LCD* adalah *RS* dan *Enable*, sinyal *RS* dan *Enable* dipergunakan sebagai *input* yang outputnya dipakai untuk mengaktifkan *LCD*. *LCD* akan aktif apabila mikrokontroler memberikan instruksi tulis pada *LCD*. Saat kondisi *RS don't care* dan *Enable 0* maka *LCD* tetap pada kondisi semula, pengiriman data ke *LCD* dilakukan saat *RS* berlogika 0 dan *enable* berlogika 1.

Instruksi dikirim pada *LCD* bila keadaan *RS 1* dan *Enable 1*. *Pin LCD* ini untuk data terkoneksi pada *Port 0* mikrokontoler AT89S51. Kemudian untuk *RS* dihubungkan pada *Port 1.0*, tulis/baca (*Read/Write*) diberikan logika *low* karena disini *LCD* bersifat menulis data, dan yang terakhir *Enable (E)* dikendalikan dengan *Port 1.1*.

3.7. Perangkat Lunak (*software*)

Perencanaan perangkat lunak sangat diperlukan untuk menjalankan *sistem* seperti yang diinginkan. Mikrokontroler AT89S51 tidak akan bisa dijalankan tanpa adanya *software*. Didalam perencanaan perangkat lunak ini di buat perintah-perintah yang berfungsi untuk mengatur urutan dan tata kerja dari keseluruhan sistem.



BAB IV

PENGUJIAN ALAT

Setelah melakukan perancangan atau pembuatan sistem kontrol ini, maka kita perlu melakukan suatu pengujian sistem. Yang mana pengujian sistem ini bertujuan antara lain :

1. Mengetahui sejauh mana Sistem Reminder Berbasis Mikrokontroler AT89S51 ini berfungsi sebagaimana yang kita harapkan.
2. Mencari dan menemukan berbagai kendala yang mungkin timbul pada saat Alat Pemberi Informasi Pergantian Jam Pelajaran Sekolah Secara Otomatis Berbasis Mikrokontroler AT89S51 beroperasi untuk kemudian diperbaiki sampai pada tingkat kesalahan sistem yang sekecil mungkin sehingga didapatkan hasil yang sebaik-baiknya.

4.1. Pengujian Rangkaian Pemutar/Perekam Suara ISD 25120

Pengujian ini bertujuan untuk mengetahui apakah rangkaian ISD25120 yang telah dibuat apakah dapat bekerja seperti yang kita harapkan atau tidak. Selain itu juga untuk mengetahui alamat data suara yang disimpan pada memori ISD 25120. Adapun peralatan yang digunakan dalam pengujian ini adalah sebagai berikut:

1. *DIP Switch 8 bit.*
2. Rangkaian ISD 25120
3. *Power Supply 5 Volt.*
4. *Loudspeaker.*

4.1.2. Hasil Pengujian

Alamat data suara dan hasil suara setelah pengujian dapat dilihat pada tabel 4-2 dibawah ini :

Tabel 4-2. Hasil Pengujian ISD

PENGUJIAN	JAM	OUTPUT SUARA	KONDISI
1	07.00	“JAM PERTAMA”	SESUAI
2	07.45	“JAM KEDUA”	SESUAI
3	08.30	“JAM KETIGA”	SESUAI
4	09.15	“JAM KEEMPAT”	SESUAI
5	10.00	“JAM KELIMA”	SESUAI
6	10.45	“JAM ISTIRAHAT”	SESUAI
7	11.00	“JAM KEENAM”	SESUAI
8	11.45	“JAM KETUJUH”	SESUAI
9	12.30	“JAM PULANG”	SESUAI

4.2. Pengujian Rangkaian Tampilan LCD

• **Tujuan**

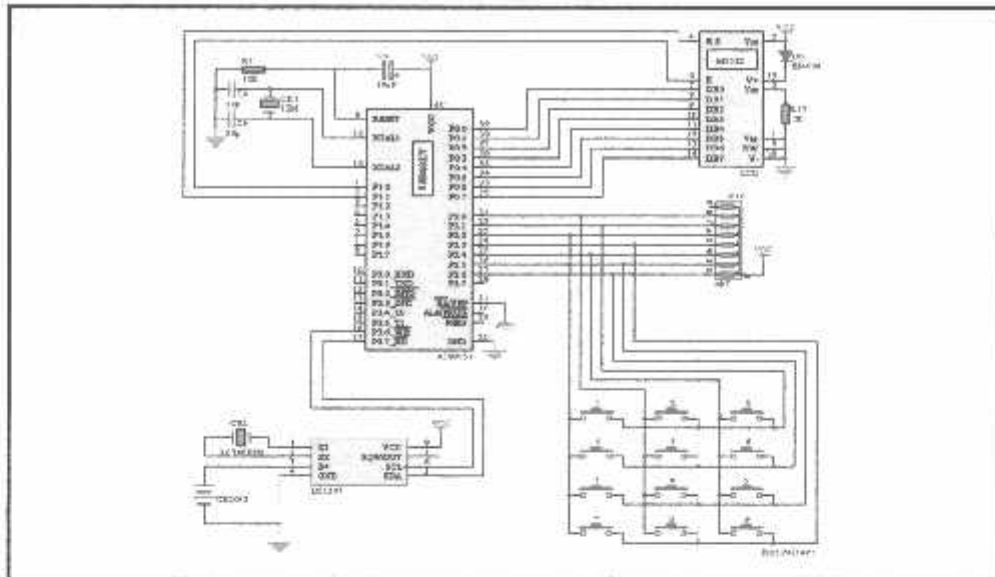
Untuk mengetahui kemampuan rangkaian tampilan yang sudah dibuat apakah dapat mendukung sistem yang direncanakan untuk memampikan data pada LCD.

• **Peralatan yang dibutuhkan**

1. Power Supply 5 Volt
2. Sistem Mikrokontroler dan LCD M1632

• **Prosedur Pengujian**

1. Menyusun rangkaian seperti dalam Gambar 4.3
2. Menjalankan program untuk menampilkan tulisan



Gambar 4.4. Rangkaian Pengujian RTC DS1307

Berikut ini adalah gambar hasil pengujian rangkaian RTC DS1307 seperti gambar

4.5:



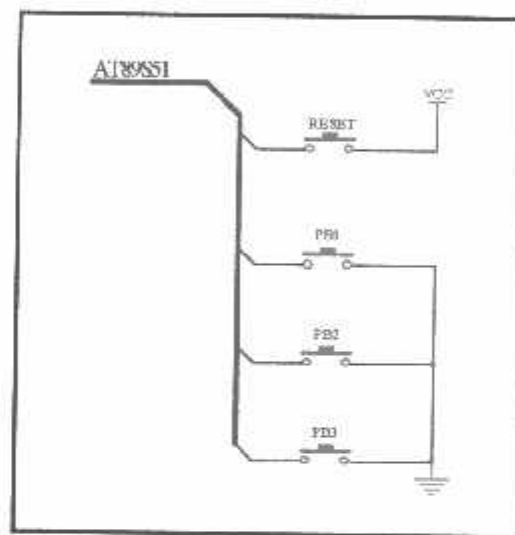
Gambar 4.5. Foto Hasil Pengujian Rangkaian RTC DS1307

4.4. Pengujian Terhadap Rangkaian *Push-button*

Rangkaian *push-button* berfungsi sebagai inputan mikrokontroler untuk seting sistem reminder. *Push-button* diberikan inputan tegangan sebesar 4,9 Volt dan dirangkai dengan menggunakan *common ground*. Sehingga apabila terjadi penekanan pada tombol *push-button*, maka akan terjadi arus hubung singkat yang menyebabkan kondisi tegangan pada pin *push-button* yang terhubung dengan

mikrokontroler berubah dari kondisi "high" 4,9 Volt menjadi kondisi "low" 0 Volt.

Pengujian rangkaian *push-button* dilakukan dengan cara mengukur tegangan pada *pin* mikrokontroler yang terhubung dengan rangkaian *push-button* dengan menggunakan multimeter DC dengan batas 20V/DC. Kondisi tegangan pada *pin* mikrokontroler yang terhubung dengan rangkaian *push-button* sebelum adanya penekanan pada tombol *push-button* adalah 4,9V. Berikut cara pengukuran tegangan pada rangkaian *push-button*, seperti pada gambar 4.6 dibawah ini :



Gambar 4.6. Rangkaian Pengujian *Push-Button*.

Sedangkan untuk hasil dari pengukuran dari tombol *push-button* seperti yang tercantum pada tabel 4-2 dibawah ini :

BAB V

PENUTUP

5.1. Kesimpulan

Pada bab ini akan dibahas mengenai kesimpulan dari hasil laporan Tugas Akhir ini. Kesimpulan yang dibuat tentu saja berdasarkan dari hasil perencanaan dan pembuatan alat seperti yang dibahas pada bab-bab sebelumnya. Kesimpulan yang didapat adalah sebagai berikut:

1. Alat ini mudah dalam pengoperasiannya di lengkapi *push button menu, up, dan down* untuk seting alat, dan LCD sebagai display.
2. Dengan menggunakan RTC DS1307 maka data waktu yang tersimpan tidak hilang meskipun power supply dimatikan dikarenakan dilengkapi baterai back-up tegangan.
3. Dikarenakan Sistem yang telah dibuat dilengkapi dengan output suara menggunakan IC perekam suara ISD 25120, sehingga batas durasi maksimum outputan suara yang dapat direkam pada sistem ini adalah 120 detik..

5.2. Saran

1. Diharapkan alat ini dapat dikembangkan dengan menambahkan penguat suara.
2. Diharapkan alat ini bisa digunakan pada jam pelajaran dibulan puasa,karena alat ini sudah terprogram dengan setting waktu yang sesuai pada hari efektif jam pelajaran,yakni 1 jam pelajaran = 45 menit.

DAFTAR PUSTAKA

1. Andi Nalwan, Paulus. 2003. *Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler 89C51*. Jakarta: PT Elex Media Komputindo.
 2. Budiharto, Widodo. 2007. *12 Proyek Mikrokontroler untuk Pemula*. Jakarta: PT Elex Media Komputindo.
 3. Eko Putra, Agfianto. 2004. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta: Penerbit Gava Media.
 4. www.atmel.com
 5. www.beyondlogic.com
 6. www.delta-electronic.com
 7. www.electroniclab.com
 8. www.fairchildsemi.com
 9. www.isd.com
 10. www.joker-robotics.com
 11. www.laipac.com
 12. www.maxim-ic.com
 13. www.parallax.com
 14. [www.seiko instruments inc.com](http://www.seiko-instruments-inc.com)
-

L A M P I R A N



INSTITUT TEKNOLOGI NASIONAL
JL. RAYA KARANGLO, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : TRI ANTORO ANGGUN MAHMUDI
Nim : 91 17 026
Masa Bimbingan : 16 Agustus 2009 s/d 16 Februari 2010
Judul Skripsi : PERENCANAAN DAN PEMBUATAN ALAT
PEMBERI INFORMASI PERGANTIAN JAM
PELAJARAN SEKOLAH SECARA OTOMATIS
BERBASIS MIKROKONTROLLER AT89S51

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	2 sept '09	Bab I	
2	4 Nop '09	Bab II	
3	15 Nop '09	Bab III	
4	1 Des '09	Bab IV	
5	15 Jan '10	Penutup	
6	15 Jan '10	See Pelar	
7			
8			
9			
10			

Malang,
Dosen Pembimbing,

Ir. F. Yudi Ampraptono, MT
NIP. 195103 9500 247



FORMULIR PERBAIKAN SKRIPSI

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

Hari : Selasa
Tanggal : 09 - 02 - 2010

Telah dilakukan perbaikan Skripsi oleh mahasiswa,

Nama : TRI ANTORO ANGGUN MAHMUDI
NIM : 97.17.026
Jurusan : T. ELEKTRO
Masa Bimbingan : 16 AGUSTUS 2009 s/d 16 FEBRUARI 2010
Judul : PERANCANGAN DAN PEMBUATAN ALAT PEMBERI INFORMASI PERGANTIAN JAM PELAJARAN SEKOLAH SECARA OTOMATIS BERBASIS MIKROKONTROLER AT89S51

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1	09-03-2010	Pengantar pada gambar	
2	09-03-2010	Cocokkan daftar pustaka dengan gambar	
3	09-03-2010	Pengujian tabel 4.2.	
4	09-03-2010	Foto LCD tampilkan tanggal, bulan, tahun	
5	09-03-2010	Data sheet ISD menyatakan 120 menit	

Malang, 10 - 03 - 2010

Disetujui,
Penguji II

Penguji I

(Ir. Eko Nurcahyo)
NIP.Y. 102 8700 172

(M. Ibrahim Ashari, ST. MT.)
NIP.P.103 0100 358

Mengetahui,
Dosen Pembimbing

(Ir. F. Yudi Limpraptono, MT.)
NIP.Y. 103 9500 247

ISD2560/75/90/120

**SINGLE-CHIP, MULTIPLE-MESSAGES,
VOICE RECORD/PLAYBACK DEVICE
60-, 75-, 90-, AND 120-SECOND DURATION**



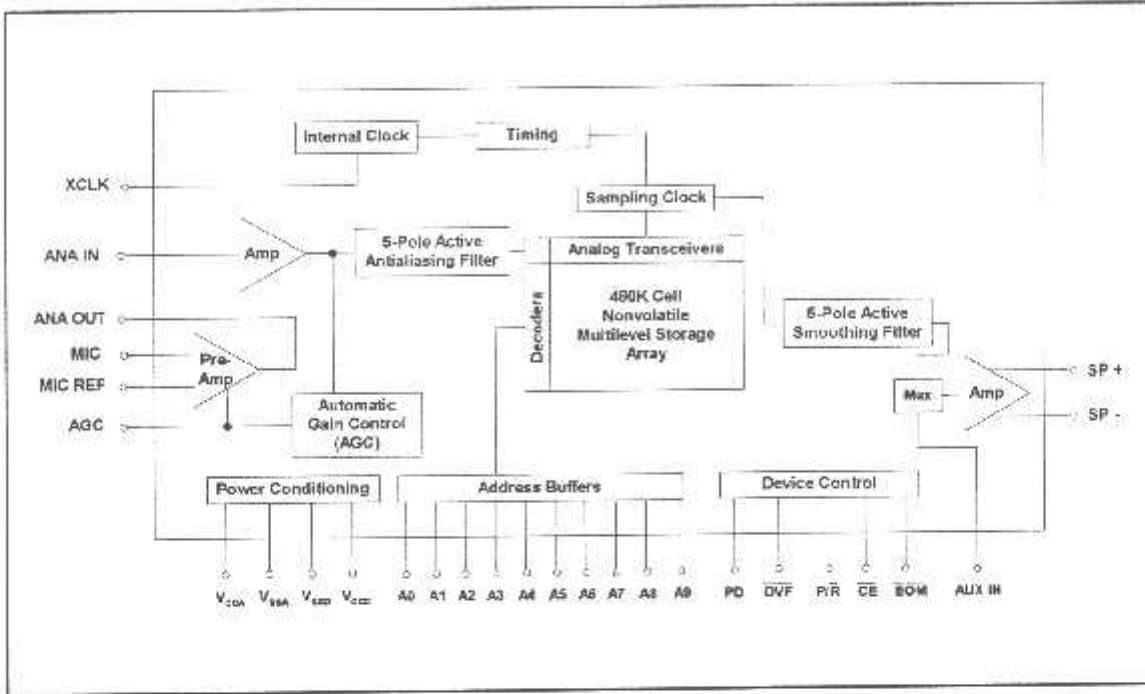
1. GENERAL DESCRIPTION

Winbond's ISD2500 ChipCorder[®] Series provide high-quality, single-chip, Record/Playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multi-level storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved. Recordings are stored into on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through Winbond's patented multilevel storage technology. Voice and audio signals are stored directly into memory in their natural form, providing high-quality, solid-state voice reproduction.

2. FEATURES

- Easy-to-use single-chip, voice record/playback solution
- High-quality, natural voice/audio reproduction
- Single-chip with duration of 60, 75, 90, or 120 seconds.
- Manual switch or microcontroller compatible
- Playback can be edge- or level-activated
- Directly cascadable for longer durations
- Automatic power-down (push-button mode)
 - Standby current 1 μ A (typical)
- Zero-power message storage
 - Eliminates battery backup circuits
- Fully addressable to handle multiple messages
- 100-year message retention (typical)
- 100,000 record cycles (typical)
- On-chip clock source
- Programmer support for play-only applications
- Single +5 volt power supply
- Available in die form, PDIP, SOIC and TSOP packaging
- Temperature = die (0°C to +50°C) and package (0°C to +70°C)

3. BLOCK DIAGRAM

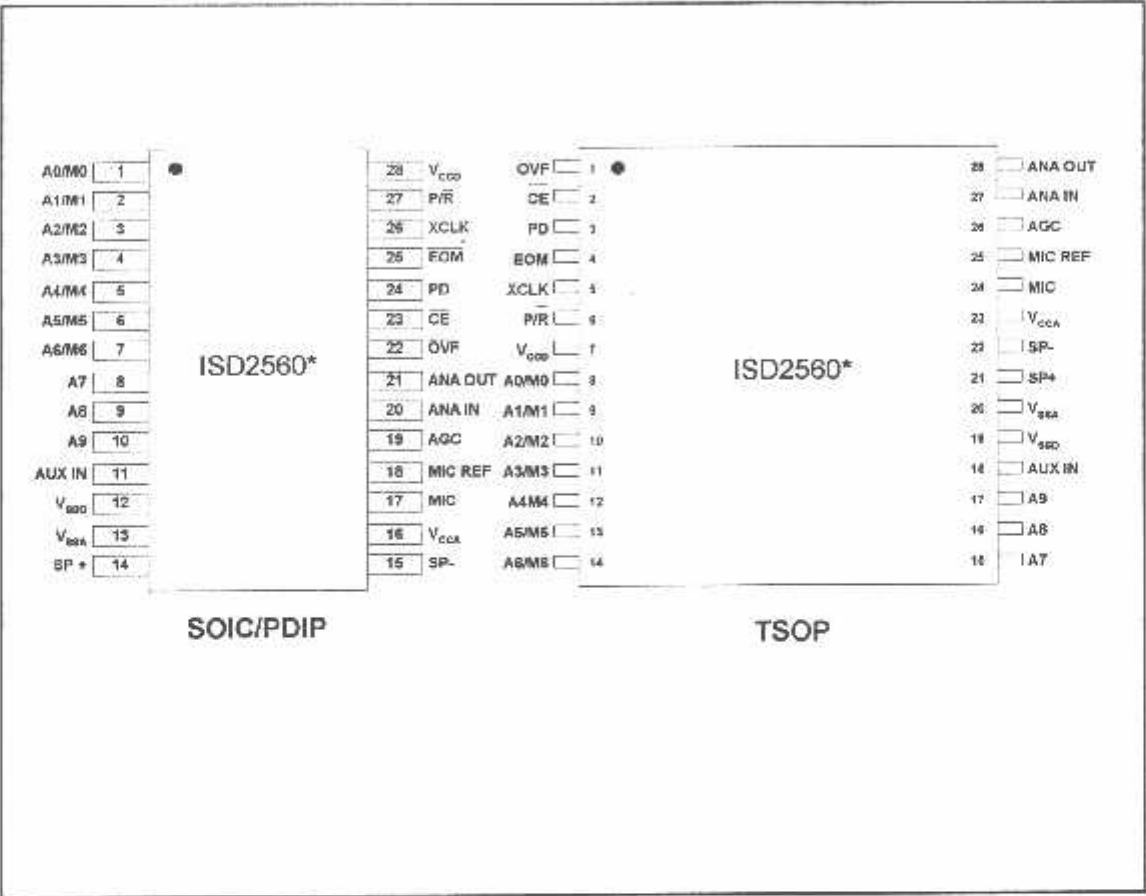


4. TABLE OF CONTENTS

1. GENERAL DESCRIPTION.....	2
2. FEATURES	2
3. BLOCK DIAGRAM	3
4. TABLE OF CONTENTS	4
5. PIN CONFIGURATION.....	5
6. PIN DESCRIPTION.....	6
7. FUNCTIONAL DESCRIPTION.....	10
7.1. Detailed Description.....	10
7.2. Operational Modes	11
7.2.1. Operational Modes Description.....	12
8. TIMING DIAGRAMS.....	16
9. ABSOLUTE MAXIMUM RATINGS.....	19
9.1 Operating Conditions	20
10. ELECTRICAL CHARACTERISTICS	21
10.1. Parameters For Packaged Parts	21
10.1.1. Typical Parameter Variation with Voltage and Temperature	24
10.2. Parameters For Die	25
10.2.1. Typical Parameter Variation with Voltage and Temperature	28
10.3. Parameters For Push-Button Mode.....	29
11. TYPICAL APPLICATION CIRCUIT	30
12. PACKAGE DRAWING AND DIMENSIONS	35
12.1. 28-Lead 300-Mil Plastic Small Outline IC (SOIC).....	35
12.2. 28-Lead 600-Mil Plastic Dual Inline Package (PDIP).....	36
12.3. 28-Lead 8x13.4MM Plastic Thin Small Outline Package (TSOP) Type 1	37
12.4. ISD2560/75/95/120 Product Bonding Physical Layout (Die) ^[1]	38
14. VERSION HISTORY	41

ISD2560/75/90/120

5. PIN CONFIGURATION



* Same pinouts for ISD2575 / 2590 / 25120 products

6. PIN DESCRIPTION

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
Ax/Mx	1-10/ 1-7	8-17/ 8-14	Address/Mode Inputs: The Address/Mode Inputs have two functions depending on the level of the two Most Significant Bits (MSB) of the address pins (A8 and A9). If either or both of the two MSBs are LOW, the inputs are all interpreted as address bits and are used as the start address for the current record or playback cycle. The address pins are inputs only and do not output any internal address information during the operation. Address inputs are latched by the falling edge of $\overline{CE}$. If both MSBs are HIGH, the Address/Mode inputs are interpreted as Mode bits according to the Operational Mode table on page 12. There are six operational modes (M0...M6) available as indicated in the table. It is possible to use multiple operational modes simultaneously. Operational Modes are sampled on each falling edge of $\overline{CE}$, and thus Operational Modes and direct addressing are mutually exclusive.
AUX IN	11	18	Auxiliary Input: The Auxiliary Input is multiplexed through to the output amplifier and speaker output pins when $\overline{CE}$ is HIGH, P/R is HIGH, and playback is currently not active or if the device is in playback overflow. When cascading multiple ISD2500 devices, the AUX IN pin is used to connect a playback signal from a following device to the previous output speaker drivers. For noise considerations, it is suggested that the auxiliary input not be driven when the storage array is active.
V _{SSA} , V _{SSD}	13, 12	20, 19	Ground: The ISD2500 series of devices utilizes separate analog and digital ground busses. These pins should be connected separately through a low-impedance path to power supply ground.
SP+/SP-	14/15	21/22	Speaker Outputs: All devices in the ISD2500 series include an on-chip differential speaker driver, capable of driving 50 mW into 16 Ω from AUX IN (12.2mW from memory). ^[1] The speaker outputs are held at V_{SSA} levels during record and power down. It is therefore not possible to parallel speaker outputs of multiple ISD2500 devices or the outputs of other speaker drivers. ^[2] A single-end output may be used (including a coupling capacitor between the SP pin and the speaker). These outputs may be used individually with the output signal taken from either pin. However, the use of single-end output results in a 1 to 4 reduction in its output power.

^[1] Connection of speaker outputs in parallel may cause damage to the device.

^[2] Never ground or drive an unused speaker output.

ISD2560/75/90/120

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
V _{CCA} , V _{CCD}	16, 28	23, 7	Supply Voltage: To minimize noise, the analog and digital circuits in the ISD2500 series devices use separate power busses. These voltage busses are brought out to separate pins and should be tied together as close to the supply as possible. In addition, these supplies should be decoupled as close to the package as possible.
MIC	17	24	Microphone: The microphone pin transfers input signal to the on-chip preamplifier. A built-in Automatic Gain Control (AGC) circuit controls the gain of this preamplifier from -15 to 24dB. An external microphone should be AC coupled to this pin via a series capacitor. The capacitor value, together with the internal 10 K Ω resistance on this pin, determines the low-frequency cutoff for the ISD2500 series passband. See Winbond's Application Information for additional information on low-frequency cutoff calculation.
MIC REF	18	25	Microphone Reference: The MIC REF input is the inverting input to the microphone preamplifier. This provides a noise-canceling or common-mode rejection input to the device when connected to a differential microphone.
AGC	19	26	Automatic Gain Control: The AGC dynamically adjusts the gain of the preamplifier to compensate for the wide range of microphone input levels. The AGC allows the full range of whispers to loud sounds to be recorded with minimal distortion. The "attack" time is determined by the time constant of a 5 K Ω internal resistance and an external capacitor (C2 on the schematic of Figure 5 in section 11) connected from the AGC pin to V _{SSA} analog ground. The "release" time is determined by the time constant of an external resistor (R2) and an external capacitor (C2) connected in parallel between the AGC pin and V _{SSA} analog ground. Nominal values of 470 K Ω and 4.7 μ F give satisfactory results in most cases.
ANA IN	20	27	Analog Input: The analog input transfers analog signal to the chip for recording. For microphone inputs, the ANA OUT pin should be connected via an external capacitor to the ANA IN pin. This capacitor value, together with the 3.0 K Ω input impedance of ANA IN, is selected to give additional cutoff at the low-frequency end of the voice passband. If the desired input is derived from a source other than a microphone, the signal can be fed, capacitively coupled, into the ANA IN pin directly.
ANA OUT	21	28	Analog Output: This pin provides the preamplifier output to the user. The voltage gain of the preamplifier is determined by the voltage level at the AGC pin.

ISD2560/75/90/120

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
$\overline{\text{OVF}}$	22	1	Overflow: This signal pulses LOW at the end of memory array, indicating the device has been filled and the message has overflowed. The $\overline{\text{OVF}}$ output then follows the $\overline{\text{CE}}$ input until a PD pulse has reset the device. This pin can be used to cascade several ISD2500 devices together to increase record/playback durations.
$\overline{\text{CE}}$	23	2	Chip Enable: The $\overline{\text{CE}}$ input pin is taken LOW to enable all playback and record operations. The address pins and playback/record pin (P/R) are latched by the falling edge of $\overline{\text{CE}}$. $\overline{\text{CE}}$ has additional functionality in the M6 (Push-Button) Operational Mode as described in the Operational Mode section.
PD	24	3	Power Down: When neither record nor playback operation, the PD pin should be pulled HIGH to place the part in standby mode (see I_{SB} specification). When overflow ($\overline{\text{OVF}}$) pulses LOW for an overflow condition, PD should be brought HIGH to reset the address pointer back to the beginning of the memory array. The PD pin has additional functionality in the M6 (Push-Button) Operation Mode as described in the Operational Mode section.
$\overline{\text{EOM}}$	25	4	End-Of-Message: A nonvolatile marker is automatically inserted at the end of each recorded message. It remains there until the message is recorded over. The $\overline{\text{EOM}}$ output pulses LOW for a period of T_{EOM} at the end of each message. In addition, the ISD2500 series has an internal V_{CC} detect circuit to maintain message integrity should V_{CC} fall below 3.5V. In this case, $\overline{\text{EOM}}$ goes LOW and the device is fixed in Playback-only mode. When the device is configured in Operational Mode M6 (Push-Button Mode), this pin provides an active-HIGH signal, indicating the device is currently recording or playing. This signal can conveniently drive an LED for visual indicator of a record or playback operation in process.

ISD2560/75/90/120

PIN NAME	PIN NO.		FUNCTION															
	SOIC/ PDIP	TSOP																
XCLK	26	5	External Clock: The external clock input has an internal pull-down device. The device is configured at the factory with an internal sampling clock frequency centered to ± 1 percent of specification. The frequency is then maintained to a variation of ± 2.25 percent over the entire commercial temperature and operating voltage ranges. If greater precision is required, the device can be clocked through the XCLK pin as follows: <table><tr><th>Part Number</th><th>Sample Rate</th><th>Required Clock</th></tr><tr><td>ISD2580</td><td>8.0 kHz</td><td>1024 kHz</td></tr><tr><td>ISD2575</td><td>6.4 kHz</td><td>819.2 kHz</td></tr><tr><td>ISD2590</td><td>5.3 kHz</td><td>582.7 kHz</td></tr><tr><td>ISD25120</td><td>4.0 kHz</td><td>512 kHz</td></tr></table> These recommended clock rates should not be varied because the antialiasing and smoothing filters are fixed, and aliasing problems can occur if the sample rate differs from the one recommended. The duty cycle on the input clock is not critical, as the clock is immediately divided by two. If the XCLK is not used, this input must be connected to ground.	Part Number	Sample Rate	Required Clock	ISD2580	8.0 kHz	1024 kHz	ISD2575	6.4 kHz	819.2 kHz	ISD2590	5.3 kHz	582.7 kHz	ISD25120	4.0 kHz	512 kHz
Part Number	Sample Rate	Required Clock																
ISD2580	8.0 kHz	1024 kHz																
ISD2575	6.4 kHz	819.2 kHz																
ISD2590	5.3 kHz	582.7 kHz																
ISD25120	4.0 kHz	512 kHz																
P/R	27	6	Playback/Record: The P/R input pin is latched by the falling edge of the $\overline{CE}$ pin. A HIGH level selects a playback cycle while a LOW level selects a record cycle. For a record cycle, the address pins provide the starting address and recording continues until PD or $\overline{CE}$ is pulled HIGH or an overflow is detected (i.e. the chip is full). When a record cycle is terminated by pulling PD or $\overline{CE}$ HIGH, then End-Of-Message (EOM) marker is stored at the current address in memory. For a playback cycle, the address inputs provide the starting address and the device will play until an EOM marker is encountered. The device can continue to pass an EOM marker if $\overline{CE}$ is held LOW in address mode, or in an Operational Mode. (See Operational Modes section)															

M6 – Push-Button Mode

The ISD2500 series contain a Push-Button Operational Mode. The Push-Button Mode is used primarily in very low-cost applications and is designed to minimize external circuitry and components, thereby reducing system cost. In order to configure the device in Push-Button Operational Mode, the two most significant address bits must be HIGH, and the M6 mode pin must also be HIGH. A device in this mode always powers down at the end of each playback or record cycle after $\overline{\text{CE}}$ goes HIGH.

When this operational mode is implemented, three of the pins on the device have alternate functionality as described in the table below.

TABLE 3: ALTERNATE FUNCTIONALITY IN PINS

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{\text{CE}}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse-activated)
$\overline{\text{EOM}}$	Active-HIGH Run Indicator

 $\overline{\text{CE}}$ (START/PAUSE)

In Push-Button Operational Mode, $\overline{\text{CE}}$ acts as a LOW-going pulse-activated START/PAUSE signal. If no operation is currently in progress, a LOW-going pulse on this signal will initiate a playback or record cycle according to the level on the $\overline{\text{P/R}}$ pin. A subsequent pulse on the $\overline{\text{CE}}$ pin, before an $\overline{\text{EOM}}$ is reached in playback or an overflow condition occurs, will pause the current operation, and the address counter is not reset. Another $\overline{\text{CE}}$ pulse will cause the device to continue the operation from the place where it is paused.

PD (STOP/RESET)

In Push-Button Operational Mode, PD acts as a HIGH-going pulse-activated STOP/RESET signal. When a playback or record cycle is in progress and a HIGH-going pulse is observed on PD, the current cycle is terminated and the address pointer is reset to address 0, the beginning of the message space.

 $\overline{\text{EOM}}$ (RUN)

In Push-Button Operational Mode, $\overline{\text{EOM}}$ becomes an active-HIGH RUN signal which can be used to drive an LED or other external device. It is HIGH whenever a record or playback operation is in progress.

Recording in Push-Button Mode

1. The PD pin should be LOW, usually using a pull-down resistor.

2. The $\overline{P/R}$ pin is taken LOW.
3. The $\overline{CE}$ pin is pulsed LOW. Recording starts, $\overline{EOM}$ goes HIGH to indicate an operation in progress.
4. When the $\overline{CE}$ pin is pulsed LOW. Recording pauses, $\overline{EOM}$ goes back LOW. The internal address pointers are not cleared, but the $\overline{EOM}$ marker is stored in memory to indicate as the message end. The $\overline{P/R}$ pin may be taken HIGH at this time. Any subsequent $\overline{CE}$ would start a playback at address 0.
5. The $\overline{CE}$ pin is pulsed LOW. Recording starts at the next address after the previous set $\overline{EOM}$ marker. $\overline{EOM}$ goes back HIGH.^[3]
6. When the recording sequences are finished, the final $\overline{CE}$ pulse LOW will end the last record cycle, leaving a set $\overline{EOM}$ marker at the message end. Recording may also be terminated by a HIGH level on PD, which will leave a set $\overline{EOM}$ marker.

Playback in Push-Button Mode

1. The PD pin should be LOW.
2. The $\overline{P/R}$ pin is taken HIGH.
3. The $\overline{CE}$ pin is pulsed LOW. Playback starts, $\overline{EOM}$ goes HIGH to indicate an operation in progress.
4. If the $\overline{CE}$ pin is pulsed LOW or an $\overline{EOM}$ marker is encountered during an operation, the part will pause. The internal address pointers are not cleared, and $\overline{EOM}$ goes back LOW. The $\overline{P/R}$ pin may be changed at this time. A subsequent record operation would not reset the address pointers and the recording would begin where playback ended.
5. $\overline{CE}$ is again pulsed LOW. Playback starts where it left off, with $\overline{EOM}$ going HIGH to indicate an operation in progress.
6. Playback continues as in steps 4 and 5 until PD is pulsed HIGH or overflow occurs.
7. If in overflow, pulling $\overline{CE}$ LOW will reset the address pointer and start playback from the beginning. After a PD pulse, the part is reset to address 0.

Note: Push-Button Mode can be used in conjunction with modes M0, M1, and M3.

^[3] If the M1 Operational Mode pin is also HIGH, the just previously written $\overline{EOM}$ bit is erased, and recording starts at that address.

ISD2560/75/90/120

Good Audio Design Practices

Winbond products are very high-quality single-chip voice recording and playback systems. To ensure the highest quality voice reproduction, it is important that good audio design practices on layout and power supply decoupling be followed. See Application Information or below links for details.

Good Audio Design Practices

http://www.winbond-usa.com/products/isd_products/chipcoorder/applicationinfo/ac111.pdf

Single-Chip Board Layout Diagrams

http://www.winbond-usa.com/products/isd_products/chipcoorder/applicationinfo/ac112.pdf

8. TIMING DIAGRAMS

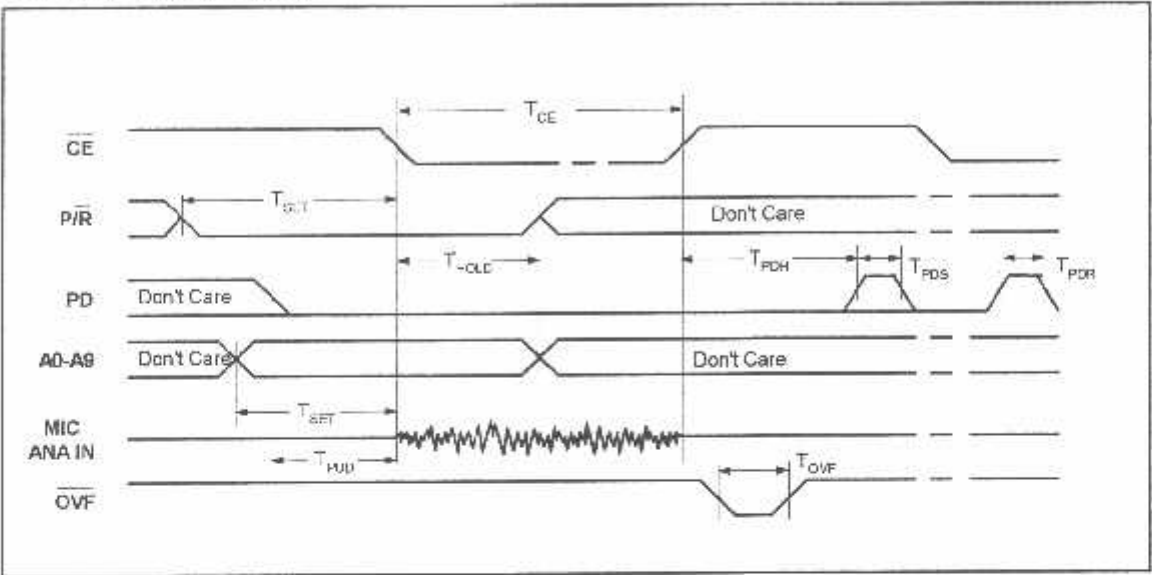


FIGURE 1: RECORD

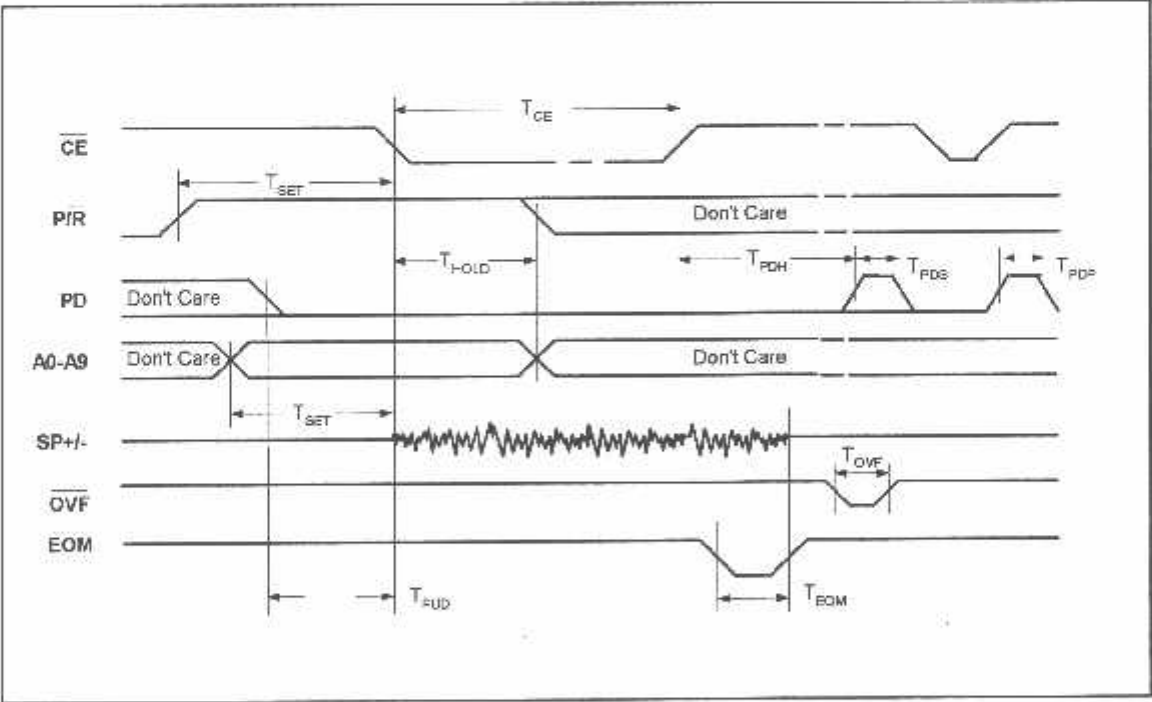


FIGURE 2: PLAYBACK

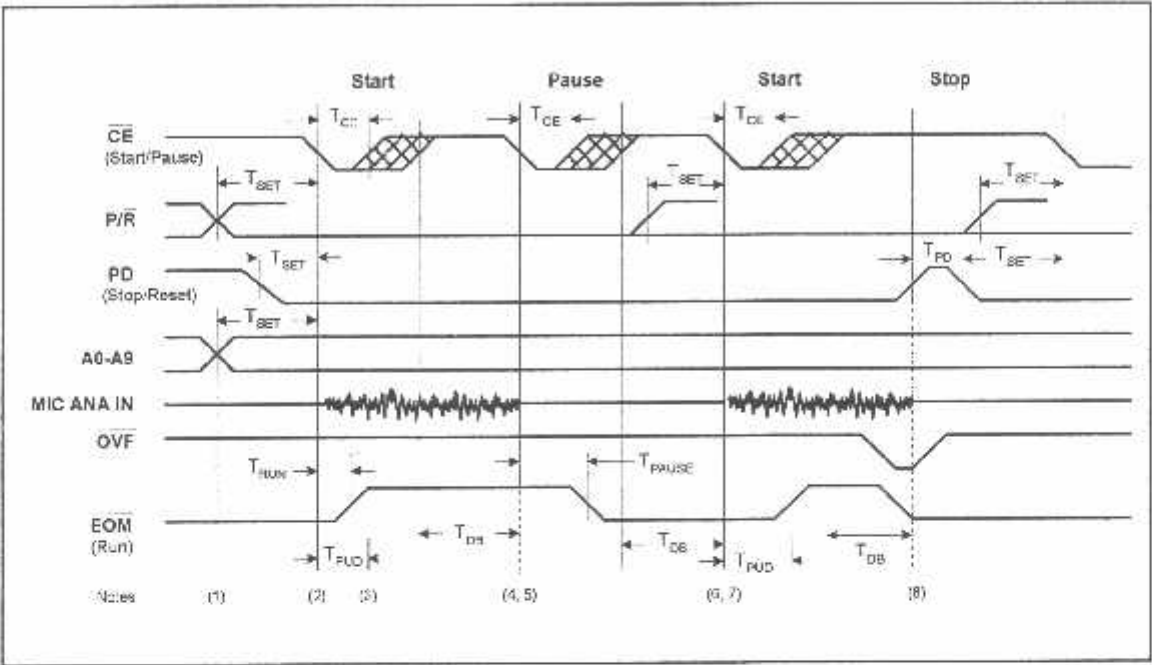


FIGURE 3: PUSH-BUTTON MODE RECORD

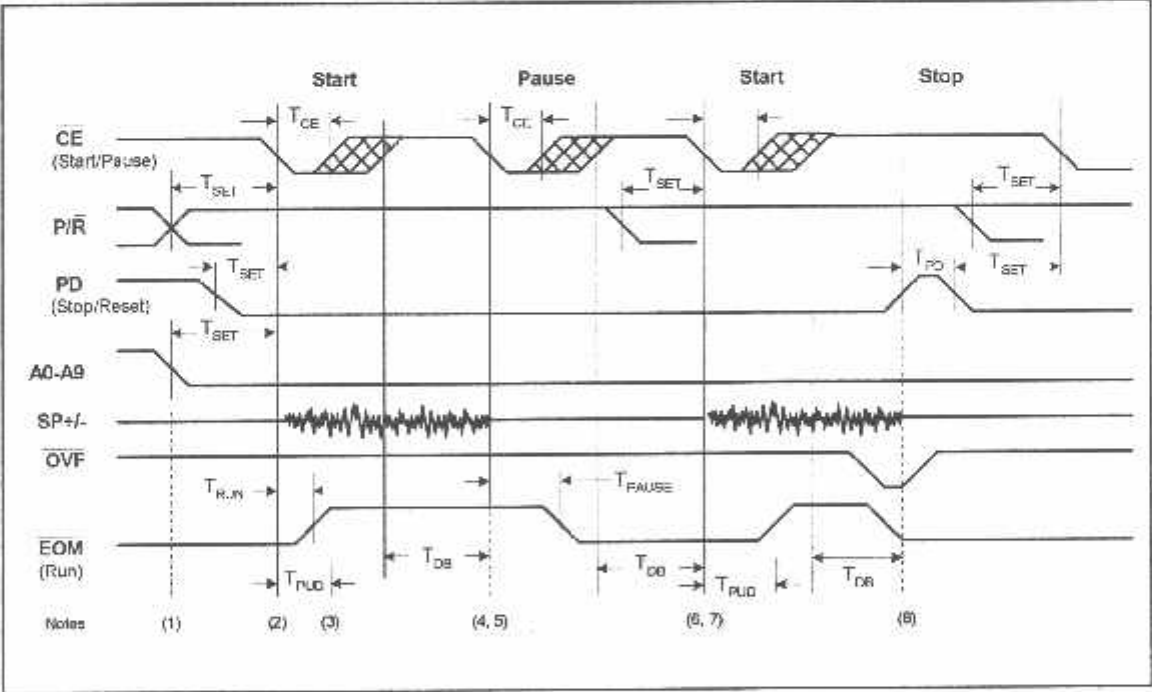


FIGURE 4: PUSH-BUTTON MODE PLAYBACK

9. ABSOLUTE MAXIMUM RATINGS

TABLE 4: ABSOLUTE MAXIMUM RATINGS (DIE)

CONDITION	VALUE
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pad	(V _{SS} -0.3V) to (V _{CC} +0.3V)
Voltage applied to any pad (Input current limited to ±20mA)	(V _{SS} -1.0V) to (V _{CC} +1.0V)
V _{CC} - V _{SS}	-0.3V to +7.0V

TABLE 5: ABSOLUTE MAXIMUM RATINGS (PACKAGED PARTS)

CONDITION	VALUE
Junction temperature	150°C
Storage temperature range	-65°C to +150°C
Voltage applied to any pin	(V _{SS} -0.3V) to (V _{CC} +0.3V)
Voltage applied to any pin (Input current limited to ±20 mA)	(V _{SS} -1.0V) to (V _{CC} +1.0V)
Lead temperature (Soldering - 10sec)	300°C
V _{CC} - V _{SS}	-0.3V to +7.0V

Note: Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability and performance. Functional operation is not implied at these conditions.

10. ELECTRICAL CHARACTERISTICS

10.1. PARAMETERS FOR PACKAGED PARTS

TABLE 8: DC PARAMETERS – Packaged Parts

PARAMETER	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNITS	CONDITIONS
Input Low Voltage	V_{IL}			0.8	V	
Input High Voltage	V_{IH}	2.0			V	
Output Low Voltage	V_{OL}			0.4	V	$I_{OL} = 4.0 \text{ mA}$
Output High Voltage	V_{OH}	$V_{CC} - 0.4$			V	$I_{OH} = -10 \mu\text{A}$
$\overline{\text{OVF}}$ Output High Voltage	V_{OH1}	2.4			V	$I_{OH} = -1.6 \text{ mA}$
$\overline{\text{EOM}}$ Output High Voltage	V_{OH2}	$V_{CC} - 1.0$	$V_{CC} - 0.8$		V	$I_{OH} = -3.2 \text{ mA}$
V_{CC} Current (Operating)	I_{CC}		25	30	mA	$R_{EXT} = \infty$ ^[3]
V_{CC} Current (Standby)	I_{SB}		1	10	μA	^[3]
Input Leakage Current	I_{IL}			± 1	μA	
Input Current HIGH w/Pull Down	I_{ILPD}			130	μA	Force V_{CC} ^[4]
Output Load Impedance	R_{EXT}	16			Ω	Speaker Load
Preamplifier Input Resistance	R_{MIC}	4	9	15	K Ω	MIC and MIC REF Pins
AUX IN Input Resistance	R_{AUX}	5	11	20	K Ω	
ANA IN Input Resistance	$R_{ANA IN}$	2.3	3	5	K Ω	
Preamplifier Gain 1	A_{PRE1}	21	24	26	dB	AGC = 0.0V
Preamplifier Gain 2	A_{PRE2}		-15	5	dB	AGC = 2.5V
AUX IN/SP+ Gain	A_{AUX}		0.98	1.0	V/V	
ANA IN to SP+/- Gain	A_{ARP}	21	23	26	dB	
AGC Output Resistance	R_{AGC}	2.5	5	9.5	K Ω	

Notes:

^[1] Typical values @ $T_A = 25^\circ$ and $V_{CC} = 5.0\text{V}$

^[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

^[3] V_{CCA} and V_{CCB} connected together.

^[4] XCLK pin only.

ISD2560/75/90/120

TABLE 9: AC PARAMETERS – Packaged Parts

CHARACTERISTIC	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNITS	CONDITIONS
Sampling Frequency	F_s					
ISD2560			8.0		kHz	[7]
ISD2575			6.4		kHz	[7]
ISD2590			5.3		kHz	[7]
ISD25120			4.0		kHz	[7]
Filter Pass Band	F_{CF}					
ISD2560			3.4		kHz	3 dB Roll-Off Point ^{[3][8]}
ISD2575			2.7		kHz	3 dB Roll-Off Point ^{[3][8]}
ISD2590			2.3		kHz	3 dB Roll-Off Point ^{[3][8]}
ISD25120			1.7		kHz	3 dB Roll-Off Point ^{[3][8]}
Record Duration	T_{REC}					
ISD2560		58.1	60.0	62.0	sec	Commercial Operation ^[7]
ISD2575		72.6	75.0	77.5	sec	Commercial Operation ^[7]
ISD2590		87.1	90.0	93.0	sec	Commercial Operation ^[7]
ISD25120		116.1	120.0	123.9	sec	Commercial Operation ^[7]
Playback Duration	T_{PLAY}					
ISD2560		58.1	60.0	62.0	sec	Commercial Operation
ISD2575		72.6	75.0	77.5	sec	Commercial Operation
ISD2590		87.1	90.0	93.0	sec	Commercial Operation
ISD25120		116.1	120.0	123.9	sec	Commercial Operation
CE Pulse Width	T_{CE}		100		nsec	
Control/Address Setup Time	T_{SET}		300		nsec	
Control/Address Hold Time	T_{HOLD}		0		nsec	
Power-Up Delay	T_{PUD}					
ISD2560		24.1	25.0	27.8	msec	Commercial Operation
ISD2575		30.2	31.3	34.3	msec	Commercial Operation
ISD2590		36.2	37.5	40.8	msec	Commercial Operation
ISD25120		48.2	50.0	53.6	msec	Commercial Operation
PD Pulse Width (record)	T_{PDR}					
ISD2560			25.0		msec	
ISD2575			31.25		msec	
ISD2590			37.5		msec	
ISD25120			50.0		msec	

ISD2560/75/90/120

TABLE 9: AC PARAMETERS – Packaged Parts (Cont'd)

CHARACTERISTIC	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNITS	CONDITIONS
PD Pulse Width (Play)	T_{PDP}		12.5		msec	
ISD2560			15.625		msec	
ISD2575			18.75		msec	
ISD2590			25.0		msec	
ISD25120					msec	
PD Pulse Width (Static)	T_{PDS}		100		nsec	^[6]
Power Down Hold	T_{PDH}		0		nsec	
EOM Pulse Width	T_{EOM}		12.5		msec	
ISD2560			15.625		msec	
ISD2575			18.75		msec	
ISD2590			25.0		msec	
ISD25120					msec	
Overflow Pulse Width	T_{OVF}		6.5		µsec	
Total Harmonic Distortion	THD		1	2	%	@ 1 kHz
Speaker Output Power	P_{OUT}		12.2	50	mW	$R_{EXT} = 16 \Omega$ ^[4]
Voltage Across Speaker Pins	V_{OUT}			2.5	V p-p	$R_{EXT} = 600 \Omega$
MIC Input Voltage	V_{IN1}			20	mV	Peak-to-Peak ^[5]
ANA IN Input Voltage	V_{IN2}			50	mV	Peak-to-Peak
AUX Input Voltage	V_{IN3}			1.25	V	Peak-to-Peak; $R_{EXT} = 16 \Omega$

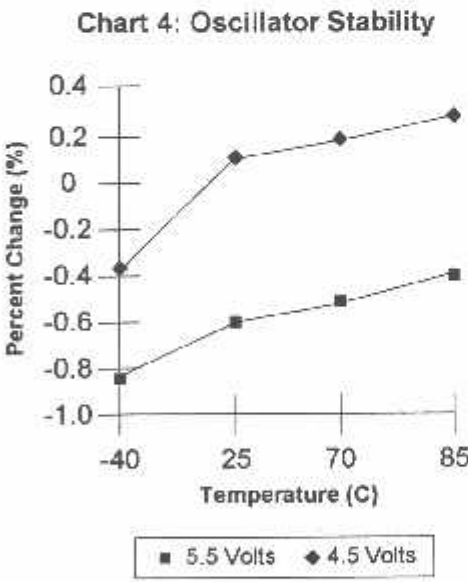
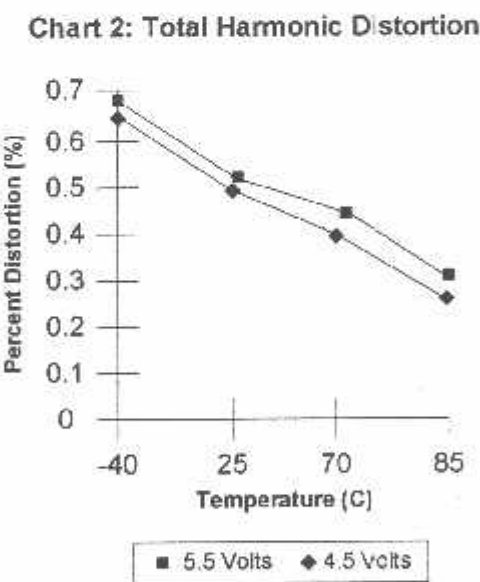
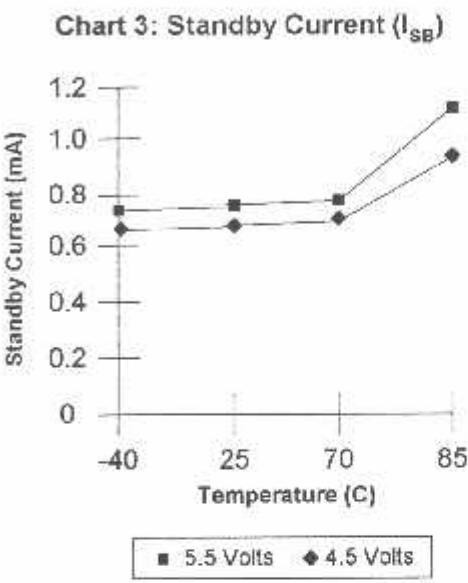
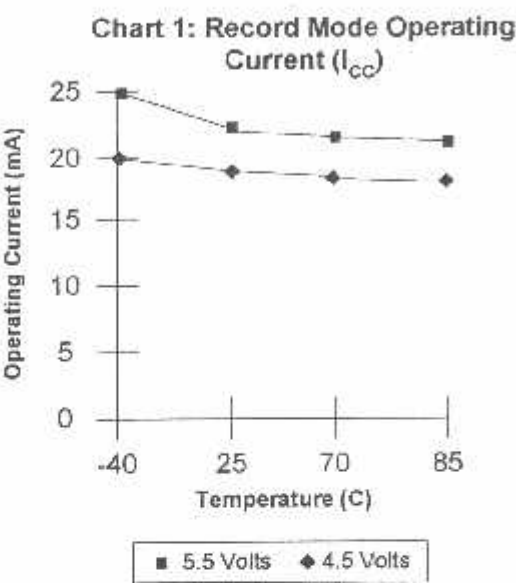
Notes:

- ^[1] Typical values @ $T_A = 25^\circ\text{C}$ and $V_{CC} = 5.0\text{V}$.
- ^[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.
- ^[3] Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).
- ^[4] From AUX IN; if ANA IN is driven at 50 mV p-p, the $P_{OUT} = 12.2 \text{ mW}$, typical.
- ^[5] With 5.1 K Ω series resistor at ANA IN.
- ^[6] T_{PDS} is required during a static condition, typically overflow.
- ^[7] Sampling Frequency and playback Duration can vary as much as +2.25 percent over the commercial temperature range. For greater stability, an external clock can be utilized (see Pin Descriptions).
- ^[8] Filter specification applies to the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 8 dB drop by nature of passing through both filters.

Publication Release Date: May 2003
Revision 1.0

ISD2560/75/90/120

10.1.1. Typical Parameter Variation with Voltage and Temperature (Packaged Parts)



10.2. PARAMETERS FOR DIE

TABLE 10: DC PARAMETERS – Die

PARAMETER	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNITS	CONDITIONS
Input Low Voltage	V _{IL}			0.8	V	
Input High Voltage	V _{IH}	2.0			V	
Output Low Voltage	V _{OL}			0.4	V	I _{OL} = 4.0 mA
Output High Voltage	V _{OH}	V _{CC} - 0.4			V	I _{OH} = -10 µA
$\overline{\text{OVF}}$ Output High Voltage	V _{O-H1}	2.4			V	I _{OH} = -1.6 mA
EOM Output High Voltage	V _{O-H2}	V _{CC} - 1.0	V _{CC} - 0.8		V	I _{OH} = -3.2 mA
V _{CC} Current (Operating)	I _{CC}		25	30	mA	R _{EXT} = ∞ ^[3]
V _{CC} Current (Standby)	I _{SD}		1	10	µA	[2]
Input Leakage Current	I _{IL}			±1	µA	
Input Current HIGH w/Pull Down	I _{ILPD}			130	µA	Force V _{CC} ^[4]
Output Load Impedance	R _{EXT}	16			Ω	Speaker Load
Preamp IN Input Resistance	R _{MIC}	4	9	15	KΩ	MIC and MIC REF Pads
AUX IN Input Resistance	R _{AUX}	5	11	20	KΩ	
ANA IN Input Resistance	R _{ANA IN}	2.3	3	5	KΩ	
Preamp Gain 1	A _{PRE1}	21	24	26	dB	AGC = 0.0V
Preamp Gain 2	A _{PRE2}		-15	5	dB	AGC = 2.5V
AUX IN/SP+ Gain	A _{AUX}		0.98	1.0	V/V	
ANA IN to SP+/- Gain	A _{ARP}	21	23	26	dB	
AGC Output Resistance	R _{AGC}	2.5	5	9.5	KΩ	

Notes:

[1] Typical values @ T_A = 25°C and V_{CC} = 5.0V.

[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

[3] V_{OH1} and V_{OH2} connected together.

[4] XCLK pad only.

ISD2560/75/90/120

TABLE 11: AC PARAMETERS – Die (Cont'd)

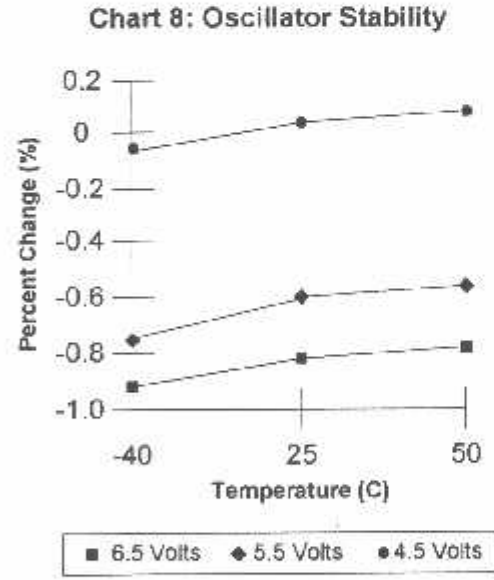
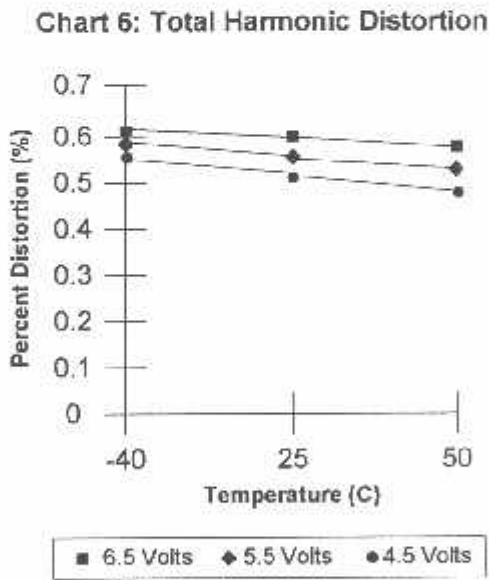
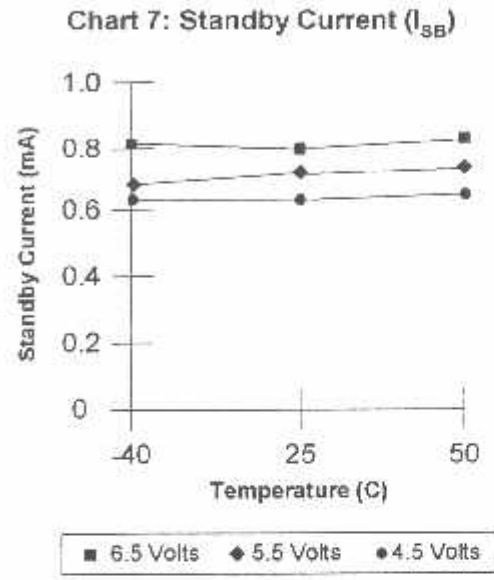
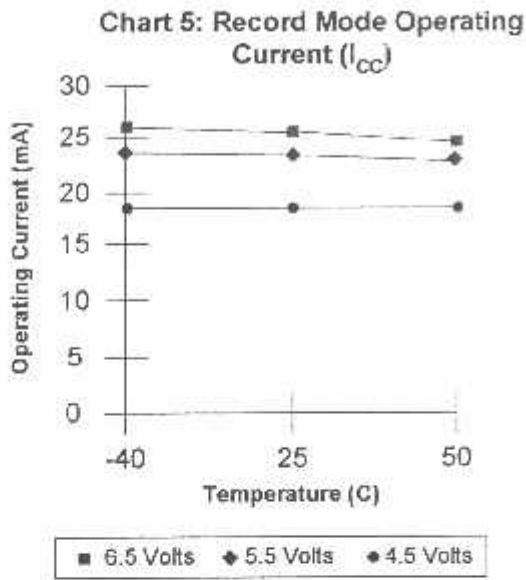
CHARACTERISTIC	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNITS	CONDITIONS
PD Pulse Width (Play)	T_{PDP}		12.5		msec	
ISD2560			15.625		msec	
ISD2575			16.75		msec	
ISD2590			25.0		msec	
ISD25120						
PD Pulse Width (Static)	T_{PDS}		100		nsec	[6]
Power Down Hold	T_{PDH}		0		nsec	
EOM Pulse Width	T_{EOM}		12.5		msec	
ISD2560			15.625		msec	
ISD2575			16.75		msec	
ISD2590			25.0		msec	
ISD25120						
Overflow Pulse Width	T_{OVF}		6.5		μ sec	
Total Harmonic Distortion	THD		1	3	%	@ 1 kHz
Speaker Output Power	P_{OUT}		12.2	50	mW	$R_{EXT} = 16 \Omega$ [4]
Voltage Across Speaker Pins	V_{OUT}			2.5	V p-p	$R_{EXT} = 600 \Omega$
MIC Input Voltage	V_{IN1}			20	mV	Peak-to-Peak [5]
ANA IN Input Voltage	V_{IN2}			50	mV	Peak-to-Peak
AUX Input Voltage	V_{IN3}			1.25	V	Peak-to-Peak; $R_{EXT} = 16 \Omega$

Notes:

- [1] Typical values @ $T_A = 25^\circ\text{C}$ and $V_{CC} = 5.0\text{V}$.
- [2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.
- [3] Low-frequency cutoff depends upon the value of external capacitors (see Pin Descriptions).
- [4] From AUX IN; if ANA IN is driven at 50 mV p-p, the $P_{OUT} = 12.2$ mW, typical.
- [5] With 5.1 K Ω series resistor at ANA IN.
- [6] T_{PDS} is required during a static condition, typically overflow.
- [7] Sampling Frequency and playback Duration can vary as much as ± 2.25 percent over the commercial temperature range. For greater stability, an external clock can be utilized (see Pin Descriptions).
- [8] Filter specification applies to the antialiasing filter and the smoothing filter. Therefore, from input to output, expect a 6 dB drop by nature of passing through both filters.

Publication Release Date: May 2003
Revision 1.0

10.2.1. Typical Parameter Variation with Voltage and Temperature (Die)



10.3. PARAMETERS FOR PUSH-BUTTON MODE

TABLE 12: PARAMETERS FOR PUSH-BUTTON MODE

PARAMETER	SYMBOL	MIN ^[2]	TYP ^[1]	MAX ^[2]	UNIT S	CONDITIONS
$\overline{\text{CE}}$ Pulse Width (Start/Pause)	T_{CE}		300		nsec	
Control/Address Setup Time	T_{SET}		300		nsec	
Power-Up Delay	T_{PUD}					
ISD2560			25.0		msec	
ISD2575			31.25		msec	
ISD2590			37.25		msec	
ISD25120			50.0		msec	
PD Pulse Width (Stop/Restart)	T_{PD}		300		nsec	
$\overline{\text{CE}}$ to EOM HIGH	T_{RUN}	25		400	nsec	
$\overline{\text{CE}}$ to EOM LOW	T_{PAUSE}	50		400	nsec	
$\overline{\text{CE}}$ HIGH Debounce	T_{DB}					
ISD2560		70		105	msec	
ISD2575		85		135	msec	
ISD2590		105		160	msec	
ISD25120		135		215	msec	

Notes:

^[1] Typical values @ $T_A = 25^\circ\text{C}$ and $V_{\text{CC}} = 5.0\text{V}$.

^[2] All Min/Max limits are guaranteed by Winbond via electrical testing or characterization. Not all specifications are 100 percent tested.

11. TYPICAL APPLICATION CIRCUIT

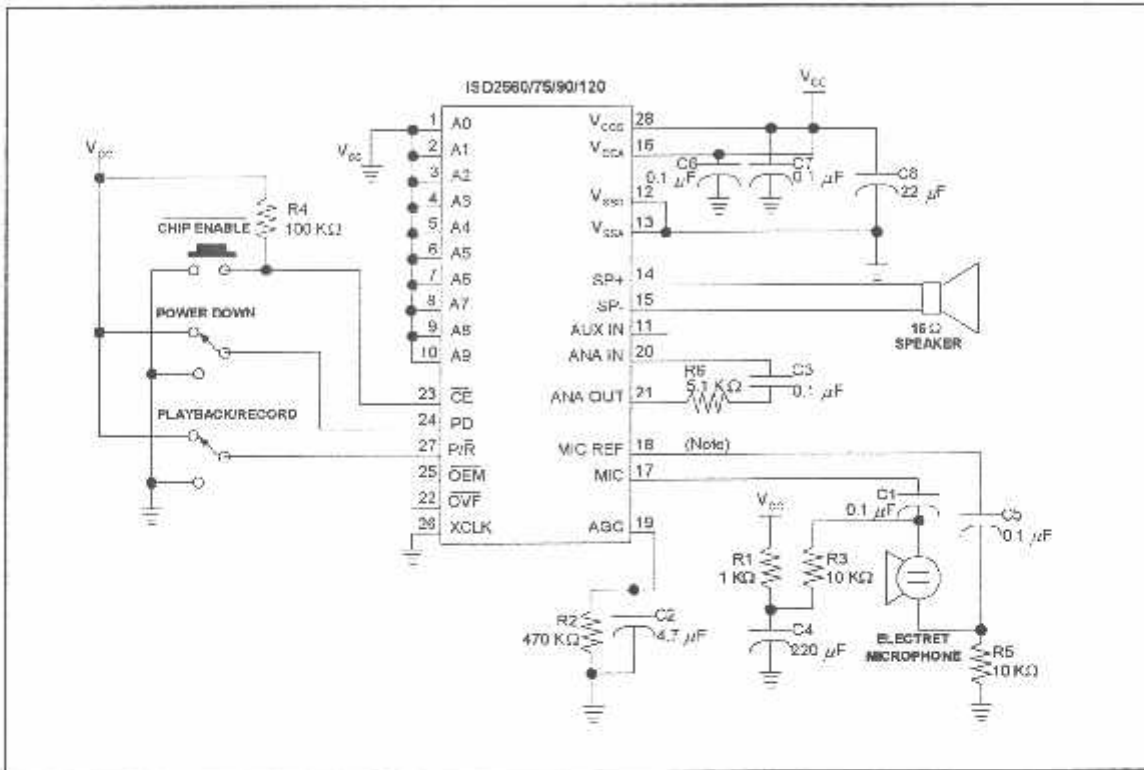


FIGURE 5: DESIGN SCHEMATIC

Note: If desired, pin 18 (PDIP package) may be left unconnected (microphone preamplifier noise will be higher). In this case, pin 18 must not be tied to any other signal or voltage. Additional design example schematics are provided below.

TABLE 13: APPLICATION EXAMPLE – BASIC DEVICE CONTROL

Control Step	Function	Action
1	Power up chip and select Record/Playback Mode	1. PD = LOW, 2. P/R = As desired
2	Set message address for record/playback	Set addresses A0-A9
3A	Begin playback	P/R = HIGH, $\overline{\text{CE}}$ = Pulse LOW
3B	Begin record	P/R = LOW, $\overline{\text{CE}}$ = LOW
4A	End playback	Automatic
4B	End record	PD or $\overline{\text{CE}}$ = HIGH

TABLE 14: APPLICATION EXAMPLE – PASSIVE COMPONENT FUNCTIONS

Part	Function	Comments
R1	Microphone power supply decoupling	Reduces power supply noise
R2	Release time constant	Sets release time for AGC
R3, R5	Microphone biasing resistors	Provides biasing for microphone operation
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages
R6	Series limiting resistor	Reduces level to high supply voltages
C1, C5	Microphone DC-blocking capacitor Low-frequency cutoff	Decouples microphone bias from chip. Provides single-pole low-frequency cutoff and command mode noise rejection.
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff
C4	Microphone power supply decoupling	Reduces power supply noise
C6, C7, C8	Power supply capacitors	Filter and bypass of power supply



FIGURE 6: ISD2560/75/90/120 APPLICATION EXAMPLE – MICROCONTROLLER/ISD2560 INTERFACE

In this simplified block diagram of a microcontroller application, the Push-Button Mode and message cueing are used. The microcontroller is a 16-pin version with enough port pins for buttons, an LED, and the ISD2500 series device. The software can be written to use three buttons: one each for play and record, and one for message selection. Because the microcontroller is interpreting the buttons and commanding the ISD2500 device, software can be written for any function desired in a particular application.

Note: Winbond does not recommend connecting address lines directly to a microprocessor bus. Address lines should be externally latched.

ISD2560/75/90/120

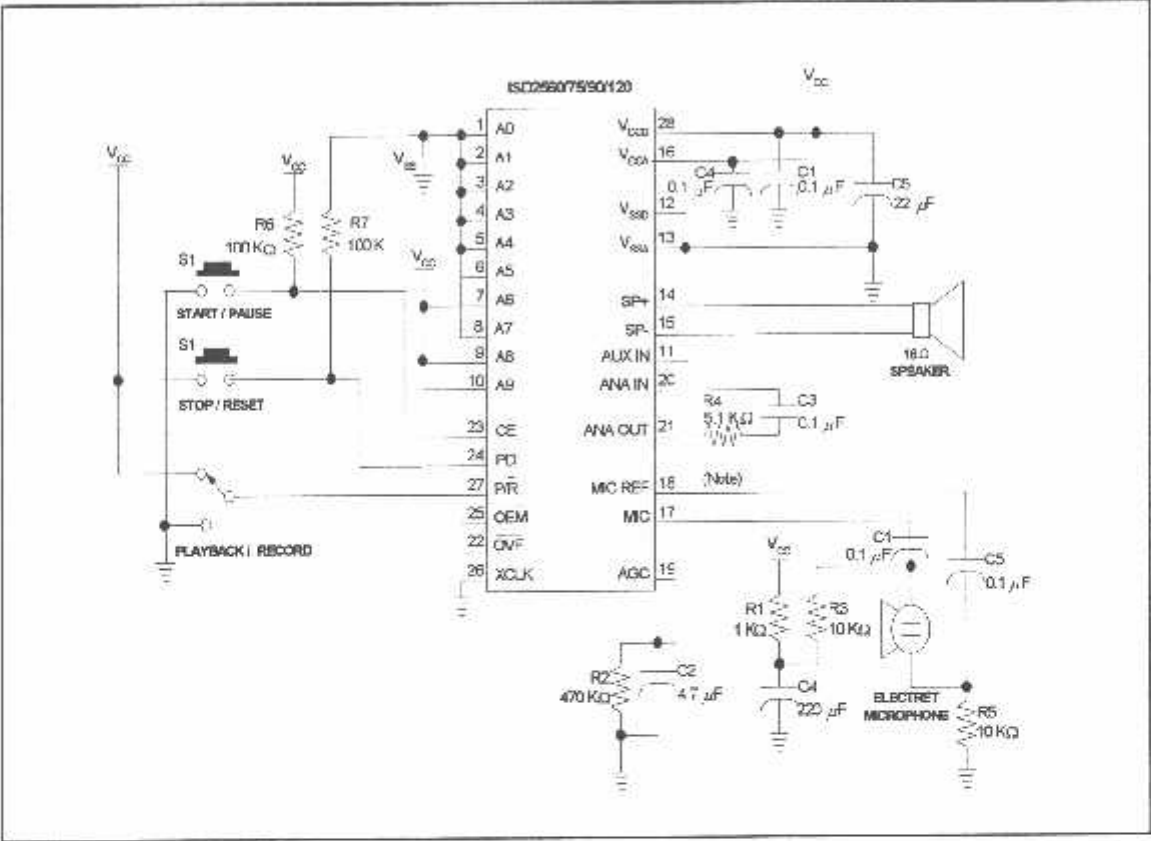


FIGURE 7: ISD2560/75/90/120 APPLICATION EXAMPLE – PUSH-BUTTON

Note: Please refer to page 13 for more details.

TABLE 15: APPLICATION EXAMPLE – PUSH-BUTTON CONTROL

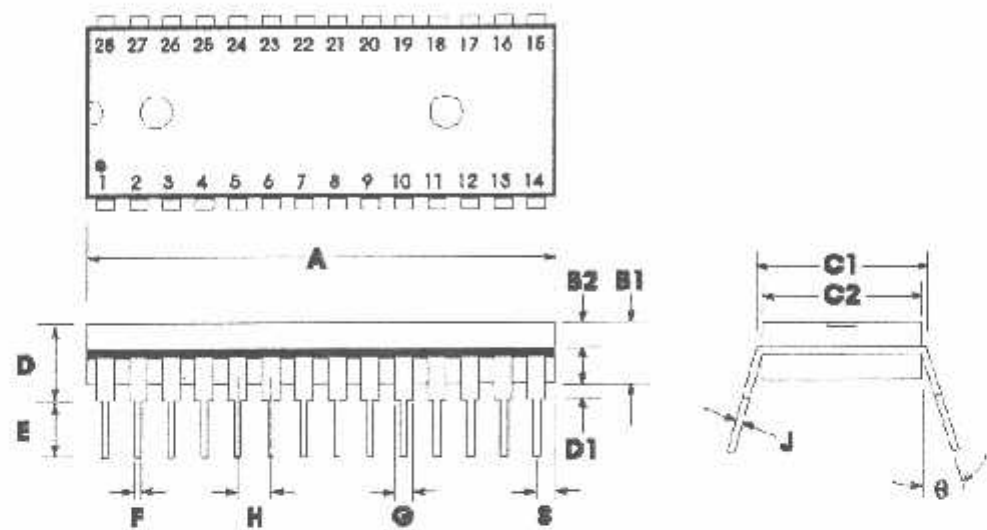
Control Step	Function	Action
1	Select Record/Playback Mode	$P/\bar{R}$ = As desired
2A	Begin playback	$P/\bar{R}$ = HIGH, $\overline{CE}$ = Pulse LOW
2B	Begin record	$P/\bar{R}$ = LOW, $\overline{CE}$ = Pulse LOW
3	Pause record or playback	$\overline{CE}$ = Pulsed LOW
4A	End playback	Automatic at EOM marker or PD = Pulsed HIGH
4B	End record	PD = Pulsed HIGH

TABLE 16: APPLICATION EXAMPLE – PASSIVE COMPONENT FUNCTIONS

Part	Function	Comments
R2	Release time constant	Sets release time for AGC
R4	Series limiting resistor	Reduces level to prevent distortion at higher supply voltages
R6, R7	Pull-up and pull-down resistors	Defines static state of inputs
C1, C4, C5	Power supply capacitors	Filters and bypass of power supply
C2	Attack/Release time constant	Sets attack/release time for AGC
C3	Low-frequency cutoff capacitor	Provides additional pole for low-frequency cutoff

ISD2560/75/90/120

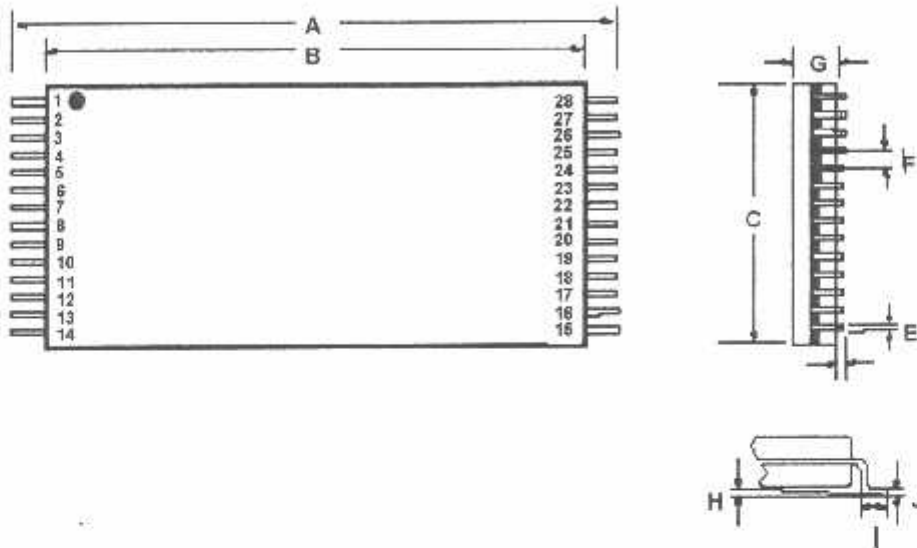
12.2. 28-LEAD 600-MIL PLASTIC DUAL INLINE PACKAGE (PDIP)



	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	1.445	1.450	1.455	36.70	36.83	36.96
B1		0.150			3.81	
B2	0.065	0.070	0.075	1.65	1.78	1.91
C1	0.600		0.625	15.24		15.88
C2	0.530	0.540	0.550	13.46	13.72	13.97
D			0.19			4.83
D1	0.015			0.38		
E	0.125		0.135	3.18		3.43
F	0.015	0.018	0.022	0.38	0.46	0.56
G	0.055	0.060	0.065	1.40	1.52	1.62
H		0.100			2.54	
J	0.008	0.010	0.012	0.20	0.25	0.30
S	0.070	0.075	0.080	1.78	1.91	2.03
q	0°		15°	0°		15°

ISD2560/75/90/120

12.3. 28-LEAD 8X13.4MM PLASTIC THIN SMALL OUTLINE PACKAGE (TSOP) TYPE 1



Plastic Thin Small Outline Package (TSOP) Type 1 Dimensions

	INCHES			MILLIMETERS		
	Min	Nom	Max	Min	Nom	Max
A	0.520	0.528	0.535	13.20	13.40	13.60
B	0.461	0.465	0.469	11.70	11.80	11.90
C	0.311	0.315	0.319	7.90	8.00	8.10
D	0.002		0.006	0.05		0.15
E	0.007	0.009	0.011	0.17	0.22	0.27
F		0.0217			0.55	
G	0.037	0.039	0.041	0.95	1.00	1.05
H	0°	3°	6°	0°	3°	6°
I	0.020	0.022	0.028	0.50	0.55	0.70
J	0.004		0.008	0.10		0.21

Note: Lead coplanarity to be within 0.004 inches.

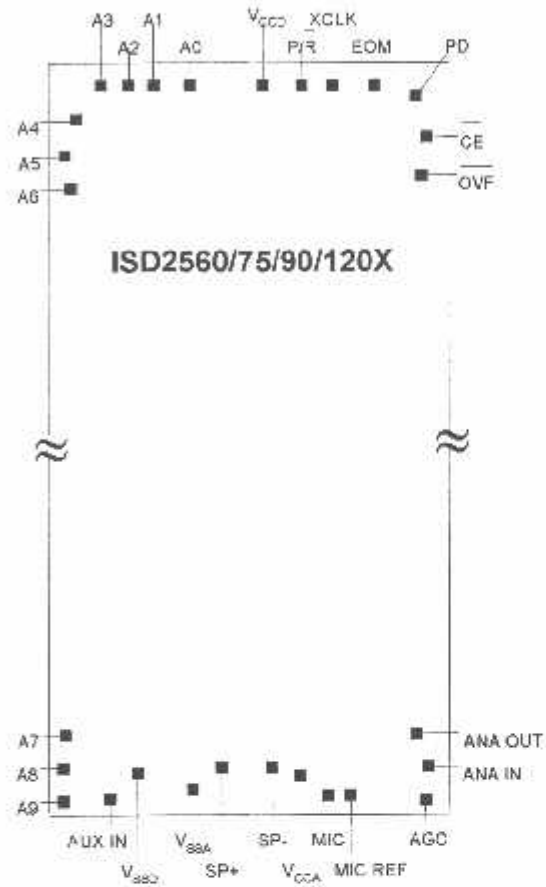
Publication Release Date: May 2003
Revision 1.0

ISD2560/75/90/120

12.4. ISD2560/75/95/120 PRODUCT BONDING PHYSICAL LAYOUT (DIE) ^[1]

ISD2560/75/95/120

- Die Dimensions
 - X: 149.5 ± 1 mils
 - Y: 262.0 ± 1 mils
- Die Thickness ^[2]
 - $11.8 \pm .4$ mils
- Pad Opening
 - 111 x 111 microns
 - 4.4 x 4.4 mils



Notes:

- ^[1] The backside of die is internally connected to V_{SS} . It **MUST NOT** be connected to any other potential or damage may occur.
- ^[2] Die thickness is subject to change, please contact Winbond factory for status and availability.

ISD2560/75/90/120

ISD2560/75/90/120 PRODUCT PAD DESIGNATIONS

(with respect to die center)

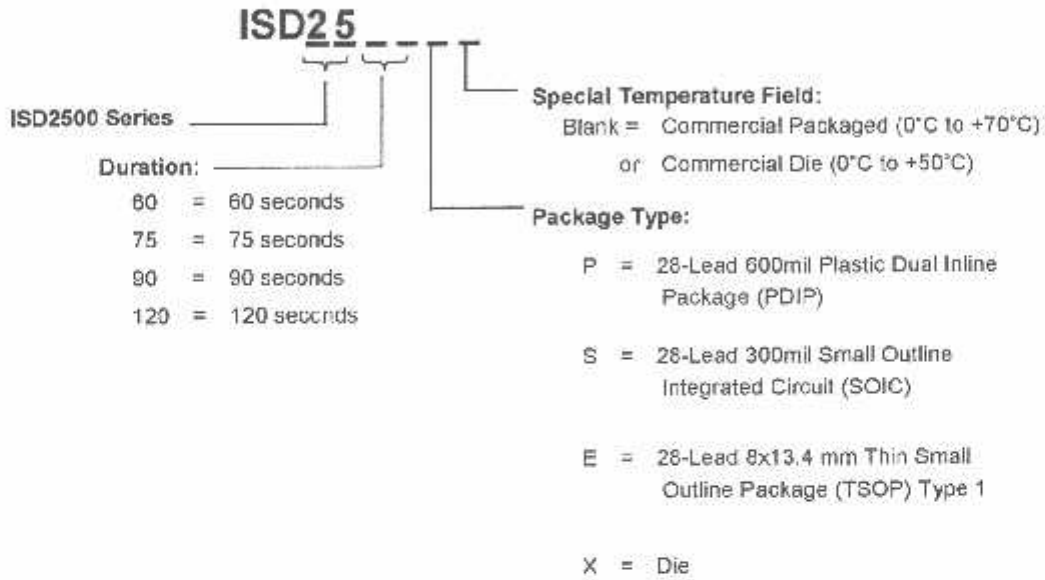
Pad	Pad Name	X Axis (μm)	Y Axis (μm)
A0	Address 0	-897.9	3135.2
A1	Address 1	-1115.4	3135.2
A2	Address 2	-1331.0	3135.2
A3	Address 3	-1544.0	3135.2
A4	Address 4	-1640.4	2888.9
A5	Address 5	-1698.2	2671.0
A6	Address 6	-1698.2	2441.5
A7	Address 7	-1731.2	-2583.2
A8	Address 8	-1731.2	-2768.4
A9	Address 9	-1731.2	-3050.8
AUX IN	Auxiliary Input	-1410.2	-3115.7
V _{SSD}	V _{SS} Digital Power Supply	-1112.4	-3096.5
V _{SSA}	V _{SS} Analog Power Supply	-408.2	-3138.9
SP+	Speaker Output +	-46.65	-3068.4
SP-	Speaker Output -	386.1	-3068.4
V _{CCA}	V _{CC} Analog Power Supply	746.9	-3110.8
MIC	Microphone Input	1101.2	-3146.0
MIC REF	Microphone Reference	1294.7	-3146.0
AGC	Automatic Gain Control	1666.4	-3130.3
ANA IN	Analog Input	1728.6	-2654.0
ANA OUT	Analog Output	1700.9	-2411.0
$\overline{\text{OVF}}$	Overflow Output	1674.6	2489.5
$\overline{\text{CE}}$	Chip Enable Input	1726.7	2824.4
PD	Power Down Input	1730.5	3094.0
$\overline{\text{EOM}}$	End of Message	1341.2	3122.1
XCLK	No Connect (optional)	986.5	3160.7
P/R	Playback/Record	807.2	3163.4
V _{CCD}	V _{CC} Digital Power Supply	544.4	3159.6

Publication Release Date: May 2003
Revision: 1.0

ISD2560/75/90/120

13. ORDERING INFORMATION

Product Number Descriptor Key



When ordering ISD2560/75/90/120 products refer to the following part numbers which are supported in volume for this product series. Consult the local Winbond Sales Representative or Distributor for availability information.

Part Number	Part Number	Part Number	Part Number
ISD2560P	ISD2575P	ISD2590P	ISD25120P
ISD2560S	ISD2575S	ISD2590S	ISD25120S
ISD2560E	ISD2575E	ISD2590E	
ISD2560X	ISD2575X	ISD2590X	ISD25120X

For the latest product information, access Winbond's worldwide website at <http://www.winbond-usa.com>

14. VERSION HISTORY

VERSION	DATE	PAGE	DESCRIPTION
0	Apr. 1998	All	Preliminary Specifications
1.0	May 2003	All	Re-format the document. Update TSOP pin configuration. Revise Overflow pad designation

The contents of this document are provided only as a guide for the applications of Winbond products. Winbond makes no representation or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to discontinue or make changes to specifications and product descriptions at any time without notice. No license, whether express or implied, to any intellectual property or other right of Winbond or others is granted by this publication. Except as set forth in Winbond's Standard Terms and Conditions of Sale, Winbond assumes no liability whatsoever and disclaims any express or implied warranty of merchantability, fitness for a particular purpose or infringement of any intellectual property.

Winbond products are not designed, intended, authorized or warranted for use as components in systems or equipments intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Winbond products are not intended for applications wherein failure of Winbond products could result or lead to a situation wherein personal injury, death or severe property or environmental injury could occur.

Application examples and alternative uses of any integrated circuit contained in this publication are for illustration only and Winbond makes no representation or warranty that such applications shall be suitable for the use specified.

ISD® and ChipCorder® are trademarks of Winbond Electronics Corporation.

The 100-year retention and 100K record cycle projections are based upon accelerated reliability tests, as published in the Winbond Reliability Report, and are neither warranted nor guaranteed by Winbond.

Information contained in this ISD® ChipCorder® data sheet supersedes all data for the ISD ChipCorder products published by ISD® prior to August, 1998.

This data sheet and any future addendum to this data sheet is(are) the complete and controlling ISD® ChipCorder® product specifications. In the event any inconsistencies exist between the information in this and other product documentation, or in the event that other product documentation contains information in addition to the information in this, the information contained herein supersedes and governs such other information in its entirety.

Copyright© 2003, Winbond Electronics Corporation. All rights reserved. ISD® is a registered trademark of Winbond. ChipCorder® is a trademark of Winbond. All other trademarks are properties of their respective owners.



Headquarters
No. 36, Chungshan Rd. II,
Science-Based Industrial Park,
Hsinchu, Taiwan
Tel: 886-3-5749100
Fax: 886-3-5749101
E-mail: info@winbond.com

Winbond Electronics Corp. (USA Corporation) America
3000 Central Expressway, Suite 200,
Folsom, CA 95630, USA
Tel: 916-977-6000
Fax: 916-977-6001
E-mail: info@winbond.com

Winbond Electronics (Shanghai) Ltd.
No. 3688, Zhongyuan Rd, Shanghai
201203, China
Tel: 86-21-52882999
Fax: 86-21-52882990

Taiwan Office
No. 36, Chungshan Rd. II,
Hsinchu, Taiwan
Tel: 886-3-5749100
Fax: 886-3-5749101
E-mail: info@winbond.com

Winbond Electronics Corporation (Japan)
2-2-1, Higashi 2-chome, 4-chome,
Koblenz-shi, Tokyo 145,
Japan
Tel: 81-3-5561-1111
Fax: 81-3-5561-1112

Winbond Electronics (HK) Ltd.
2/F, 225, Victoria Road,
Kowloon, Hong Kong
Tel: 852-250-0000
Fax: 852-250-0001
E-mail: info@winbond.com

© 2003 Winbond Electronics Corporation. All rights reserved. ISD® is a registered trademark of Winbond. ChipCorder® is a trademark of Winbond. All other trademarks are properties of their respective owners.

MCS-51 Family Macro Assembler A S E M - 5 1 V 1.3
=====

Source File: E:_AT_PROG\ITN PK 2008-2009\bel
sekolah\BELS_SEKOLAH.ASM
Object File: E:_AT_PROG\ITN PK 2008-2009\bel
sekolah\BELS_SEKOLAH.hex
List File: E:_AT_PROG\ITN PK 2008-2009\bel sekolah\BELS_SEKOLAH.lst

Line	I	Addr	Code	Source
1:		N	0000	ORG 000H
2:		0000	02 00 03	JMP START
3:				
4:				
5:				;=====
6:				;LCD KONSTANTA
7:				;=====
8:	N	0001		DISPCLR EQU 00000001B
9:	N	0008		FUNCSET EQU 0011100CB
10:	N	0006		ENTRMOD EQU 0000011CB
11:	N	000C		DISPON EQU 0000110CB
12:				;=====
13:				;PORT LCD
14:				;=====
15:	B	90	LCDE BIT	P1.0
16:	B	91	LCDRS BIT	P1.1
17:	D	0080	PLCD EQU	P0
18:	B	B6	BUZZ BIT	P3.6
19:				;=====
20:				;RTC ADDRES
21:				;=====
22:	B	B5	SCL BIT	P3.5
23:	B	B4	SDA BIT	P3.4
24:				;=====
25:				;=====
26:				;PORT ISD
27:				;=====
28:	B	93	ISD_PD BIT	P1.3
29:	B	92	ISD_CE BIT	P1.2
30:	B	94	ISD_EOM BIT	P1.4
31:	B	B1	ISD_PR BIT	P3.1
32:	D	00A0	PISD EQU	P2
33:				
34:				;=====
35:				;PUSH BUTTON ADDRES
36:				;=====

```

37:      B      97  TBDOWN  BIT      P1.7
38:      B      96  TBUP    BIT      P1.6
39:      B      95  TBMENU  BIT      P1.5
40:
41:      ;=====
42:      ;=====
43:      ;ALAMAT VARIABLE

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 2

Line	I	Addr	Code	Source
44:				;
45:	N	0030		PUTR EQU 030H
46:	N	0031		RTC_DAT EQU 031H
47:	N	0032		RTC_ADR EQU 032H
48:	N	0033		SEC EQU 033H
49:	N	0034		MIN EQU 034H
50:	N	0035		HOUR EQU 035H
51:	N	0036		DATE EQU 036H
52:	N	0037		MONTH EQU 037H
53:	N	0038		YEAR EQU 038H
54:	N	0039		JMLHR EQU 039H
55:	N	003A		TEMP EQU 03AH
56:	N	003B		MIN_P EQU 03BH
57:	N	003C		HOUR_P EQU 03CH
58:	N	003D		DATE_P EQU 03DH
59:	N	003E		MONTH_P EQU 03EH
60:	N	003F		YEAR_P EQU 03FH
61:	N	0040		M EQU 040H
62:	N	0041		R EQU 041H
63:	N	0042		REC_NUM EQU 042H
64:	N	0022		C_5mS EQU 022H
65:				
66:				;
67:				;MAIN PROGRAM
68:				;
69:	0003			START:
70:	0003	12 07 DC		LCALL INITLCD
71:	0006	12 0B 6D		CALL DELAY1D
72:	0009	75 42 00		MOV REC_NUM, #0
73:	000C	D2 93		SETB ISD_PD
74:	000E	D2 92		SETB ISD_CE
75:	0010	D2 B1		SETB ISD_PR
76:	0012	D2 94		SETB ISD_EOM
77:	0014	D2 B6		SETB BUZZ

```

78:      0016      12 0A FB      LCALL IKLAN
79:                                     ;JMP RING_AGAIN
80:
81:      0019
82:      0019      12 02 08      START_A:
83:                                     CALL DIS_BACK
84:      001C      12 00 F9      CALL DEF_ALARM
85:      001F      20 95 F7      JB      TBMENU, START_A
86:      0022      30 95 FD      JNB     TBMENU, $
87:      0025      74 C1      MOV     A, #DISPCLR
88:      0027      12 07 9C      CALL    LCDINS
89:      002A      12 02 08      CALL    DIS_BACK
90:      002D      12 04 3F      CALL SET_WAKTU      ;SET WAKTU
91:      0030      74 01      MOV     A, #DISPCLR
92:      0032      12 07 9C      CALL    LCDINS
93:      0035      80 E2      JMP     START_A
94:
95:      0037
96:      0037      12 02 08      TEST_ISD:
97:      003A      20 95 08      CALL    DIS_BACK
98:      003D      30 95 FD      JB      TBMENU, TEST_ISD_OUT
99:      0040      12 00 77      JNB     TBMENU, $
                                     CALL INPUT_SUARA

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 3

Line	I	Addr	Code	Source
100:		0043	80 F2	JMP TEST_ISD
101:				
102:		0045		TEST_ISD_OUT:
103:		0045	12 02 08	CALL DIS_BACK
104:		0048	20 96 08	JB TBUP, ISI_RECNUM
105:		004B	30 96 FD	JNB TBUP, \$
106:		004E	12 00 A4	CALL RING_AGAIN
107:		0051	80 E4	JMP TEST_ISD
108:				
109:		0053		ISI_RECNUM:
110:		0053	20 97 E1	JB TBCDOWN, TEST_ISD
111:		0056	30 97 FD	JNB TBCDOWN, \$
112:		0059	74 01	MOV A, #DISPCLR
113:		005B	12 07 9C	CALL LCDINS
114:		005E	12 0B 6D	CALL DELAY1D
115:		0061	12 38 28	CALL DSP_ISI_REC
116:		0064	20 97 FD	JB TBCDOWN, \$
117:		0067	30 97 FD	JNB TBCDOWN, \$

```

118:      006A      75 42 00      MOV  REC_NUM,#0
119:      006D      74 01      MOV  A,#DISPCLR
120:      006F      12 07 9C      CALL  LCDINS
121:      0072      12 0B 5D      CALL  DELAY1D
122:      0075      80 C0      JMP  TEST_ISD
123:
124:      ;=====PROSEDUR MENU SUMTING INPUT SUARA=====
125:      0077      INPUT_SUARA:
126:      0077      IS1:
127:      0077      74 01      MOV  A,#DISPCLR
128:      0079      12 07 9C      CALL  LCDINS
129:      007C      12 08 11      CALL  DSP_RECORD
130:      007F      E5 42      MOV  A,REC_NUM
131:      0081      F5 A0      MOV  P1SD,A
132:      0083      C2 B1      CLR  ISD_PR
133:      0085      C2 93      CLR  ISD_PD
134:      0087      C2 92      CLR  ISD_CE
135:      0089      IS2:
136:      0089      2C 95 FD      JB   TMENU,IS2
137:      008C      3C 95 FD      JNB  TMENU,S
138:      008F      D2 B1      SETB ISD_PR
139:      0091      D2 93      SETB ISD_PD
140:      0093      D2 92      SETB ISD_CE
141:      0095      E5 42      MOV  A,REC_NUM
142:      0097      24 1C      ADD  A,#28
143:      0099      F5 42      MOV  REC_NUM,A
144:      009B      74 01      MOV  A,#DISPCLR
145:      009D      12 07 9C      CALL  LCDINS
146:      00A0      12 0B 5D      CALL  DELAY1D
147:      00A3      22      RET
148:
149:      00A4      RING_AGAIN:
150:      00A4      74 01      MOV  A,#DISPCLR
151:      00A6      12 07 9C      CALL  LCDINS
152:      00A9      12 07 FB      CALL  DSP_PLAY
153:      00AC      74 1C      MOV  A,#28
154:      00AE      F5 A0      MOV  P1SD,A
155:      00B0      D2 B1      SETB ISD_PR

```

Line	I	Addr	Code	Source
156:		00B2	C2 93	CLR ISD_PD
157:		00B4	C2 92	CLR ISD_CE

```

158:      00B6      20 94 FD      JB      ISD_EOM,$
159:      00B9      D2 94      SETB     ISD_EOM
160:      00BB      D2 93      SETB     ISD_PD
161:      00BD      D2 92      SETB     ISD_CE
162:                                ;MOV      A,REC_NUM
163:                                ;ADD      A,#28
164:                                ;MOV      REC_NUM,A
165:      00BF      74 01      MOV      A,#DISPCLR
166:      00C1      12 07 9C      CALL     LCDINS
167:      00C4      12 0B 6D      CALL     DELAY1D
168:                                ;RET
169:      00C7      80 DB      JMP      RING_AGAIN
170:      00C9      BUZZERZ:
171:      00C9      C2 B6      CLR      BUZZ
172:      00CB      12 0B 6D      CALL     DELAY1D
173:      00CE      D2 B6      SETB     BUZZ
174:      00D0      C2 B6      CLR      BUZZ
175:      00D2      12 0B 6D      CALL     DELAY1D
176:      00D5      D2 B6      SETB     BUZZ
177:      00D7      12 0B 6D      CALL     DELAY1D
178:      00DA      C2 B6      CLR      BUZZ
179:      00DC      12 0B 6D      CALL     DELAY1D
180:      00DF      D2 B6      SETB     BUZZ
181:      00E1      22      RET
182:
183:      00E2      OUT_SUARA:
184:      00E2      F5 A0      MOV      P1SD,A
185:      00E4      12 B1      SETB     ISD_PR
186:      00E6      C2 93      CLR      ISD_PD
187:      00E8      C2 92      CLR      ISD_CE
188:      00EA      20 94 FD      JB      ISD_EOM,$
189:      00ED      D2 94      SETB     ISD_EOM
190:      00EF      D2 93      SETB     ISD_PD
191:      00F1      D2 92      SETB     ISD_CE
192:      00F3      74 01      MOV      A,#DISPCLR
193:      00F5      12 07 9C      CALL     LCDINS
194:      00F8      22      RET
195:
196:      ;=====
197:      00F9      CHK_ALARM:
198:      ;=====
199:      00F9      JAM_KE_1:      ; 07.00
200:      00F9      E5 35      MOV      A, HOUR
201:      00FB      B4 07 19      CJNE     A,#7,JAM_KE_2
202:      00FE      E5 34      MOV      A, MIN
203:      0100      B4 00 14      CJNE     A,#0,JAM_KE_2
204:      0103      E5 33      MOV      A, SEC
205:      0105      B4 00 0F      CJNE     A,#0,JAM_KE_2
206:      0108      74 01      MOV      A,#DISPCLR
207:      010A      12 07 9C      CALL     LCDINS
208:      010D      12 0A 2C      CALL     DSP_1
209:      0110      11 C9      CALL     BUZZERZ

```

```

210:      0112      74 00      MOV      A, #0
211:      0114      11 E2      CALL     OUT_SUARA

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 5

Line	i	Addr	Code	Source
212:		0116	22	RET
213:				
214:		0117		JAM_KE_2: ; 07.45
215:		0117	E5 35	MOV A, HOUR
216:		0119	B4 07 19	CJNE A, #7, JAM_KE_3
217:		011C	E5 34	MOV A, MIN
218:		011E	B4 2D 14	CJNE A, #45, JAM_KE_3
219:		0121	E5 33	MOV A, SEC
220:		0123	B4 00 0F	CJNE A, #0, JAM_KE_3
221:		0126	74 01	MOV A, #DISPCLR
222:		0128	12 07 9C	CALL LCDINS
223:		012B	12 0A 43	CALL DSP_2
224:		012E	11 C9	CALL BUZZERZ
225:		0130	74 1C	MOV A, #28
226:		0132	11 E2	CALL OUT_SUARA
227:		0134	22	RET
228:		0135		JAM_KE_3: ; 08.30
229:		0135	E5 35	MOV A, HOUR
230:		0137	B4 08 19	CJNE A, #8, JAM_KE_4
231:		013A	E5 34	MOV A, MIN
232:		013C	B4 1E 14	CJNE A, #30, JAM_KE_4
233:		013F	E5 33	MOV A, SEC
234:		0141	B4 00 0F	CJNE A, #0, JAM_KE_4
235:		0144	74 01	MOV A, #DISPCLR
236:		0146	12 07 9C	CALL LCDINS
237:		0149	12 0A 5A	CALL DSP_3
238:		014C	11 C9	CALL BUZZERZ
239:		014E	74 38	MOV A, #56
240:		0150	11 E2	CALL OUT_SUARA
241:		0152	22	RET
242:				
243:		0153		JAM_KE_4: ; 09.15
244:		0153	E5 35	MOV A, HOUR
245:		0155	B4 09 19	CJNE A, #9, JAM_KE_5
246:		0158	E5 34	MOV A, MIN
247:		015A	B4 0F 14	CJNE A, #15, JAM_KE_5
248:		015D	E5 33	MOV A, SEC
249:		015F	B4 00 0F	CJNE A, #0, JAM_KE_5
250:		0162	74 01	MOV A, #DISPCLR

251:	0164	12 07 9C	CALL	LCDINS
252:	0167	12 0A 71	CALL	DSP_4
253:	016A	11 C9	CALL	BUZZER2
254:	016C	74 54	MOV	A, #84
255:	016E	11 E2	CALL	OUT_SUARA
256:	0170	22	RET	
257:				
258:	0171		JAM_KE_5:	; 10.00
259:	0171	E5 35	MOV	A, HOUR
260:	0173	34 0A 19	CJNE	A, #10, JAM_ISTIRAHAT
261:	0176	E5 34	MOV	A, MIN
262:	0178	34 00 14	CJNE	A, #0, JAM_ISTIRAHAT
263:	017B	E5 33	MOV	A, SEC
264:	017D	34 00 0F	CJNE	A, #0, JAM_ISTIRAHAT
265:	0180	74 01	MOV	A, #DISPCLR
266:	0182	12 07 9C	CALL	LCDINS
267:	0185	12 0A 88	CALL	DSP_5

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 6

Line	I	Addr	Code	Source
268:		0188	11 C9	CALL BUZZER2
269:		018A	74 70	MOV A, #112
270:		018C	11 E2	CALL OUT_SUARA
271:		018E	22	RET
272:				
273:		018F		JAM_ISTIRAHAT: ; 10.45
274:		018F	E5 35	MOV A, HOUR
275:		0191	34 0A 19	CJNE A, #10, JAM_KE_6
276:		0194	E5 34	MOV A, MIN
277:		0196	34 2D 14	CJNE A, #45, JAM_KE_6
278:		0199	E5 33	MOV A, SEC
279:		019B	34 00 0F	CJNE A, #0, JAM_KE_6
280:		019E	74 01	MOV A, #DISPCLR
281:		01A0	12 07 9C	CALL LCDINS
282:		01A3	12 0A 9F	CALL DSP_ISTIRAHAT
283:		01A6	11 C9	CALL BUZZER2
284:		01A8	74 8C	MOV A, #140
285:		01AA	11 E2	CALL OUT_SUARA
286:		01AC	22	RET
287:				
288:		01AD		JAM_KE_6: ; 11.00
289:		01AD	E5 35	MOV A, HOUR
290:		01AF	B4 0B 19	CJNE A, #11, JAM_KE_7
291:		01B2	E5 34	MOV A, MIN
292:		01B4	B4 00 14	CJNE A, #0, JAM_KE_7

293:	01B7	E5 33	MOV A, SEC
294:	01B9	B4 C0 0F	CJNE A, #0, JAM_KE_7
295:	01BC	74 01	MOV A, #DISPCLR
296:	01BE	12 07 9C	CALL LCDINS
297:	01C1	12 0A 86	CALL DSP_6
298:	01C4	11 C9	CALL BUZZERZ
299:	01C6	74 A8	MOV A, #168
300:	01C8	11 E2	CALL OUT_SUARA
301:	01CA	22	RET
302:	01CB		JAM_KE_7: ; 11.45
303:	01CB	E5 35	MOV A, HOUR
304:	01CD	B4 0B 19	CJNE A, #11, JAM_PULANG
305:	01DE	E5 34	MOV A, MIN
306:	01D2	B4 2D 14	CJNE A, #45, JAM_PULANG
307:	01D5	E5 33	MOV A, SEC
308:	01D7	B4 00 0F	CJNE A, #0, JAM_PULANG
309:	01DA	74 01	MOV A, #DISPCLR
310:	01DC	12 07 9C	CALL LCDINS
311:	01DF	12 0A CD	CALL DSP_7
312:	01E2	11 C9	CALL BUZZERZ
313:	01E4	74 C4	MOV A, #196
314:	01E6	11 E2	CALL OUT_SUARA
315:	01E8	22	RET
316:			
317:	01E9		JAM_PULANG: ; 12.30
318:	01E9	E5 35	MOV A, HOUR
319:	01EB	B4 0C 19	CJNE A, #12, OUT_C
320:	01EE	E5 34	MOV A, MIN
321:	01F0	B4 1E 14	CJNE A, #30, OUT_C
322:	01F3	E5 33	MOV A, SEC
323:	01F5	B4 C0 0F	CJNE A, #0, OUT_C

Line	Addr	Code	Source
324:	01F8	74 01	MOV A,#DISPCLR
325:	01FA	12 07 9C	CALL LCDINS
326:	01FD	12 0A E4	CALL ESP_PULANG
327:	0200	11 C9	CALL BUZZERZ
328:	0202	74 E0	MOV A,#224
329:	0204	11 E2	CALL OUT_SUARA
330:	0206	22	RET
331:	0207		OUT_C:
332:	0207	22	RET
333:			=====
334:	0208		DIS_BACK:
335:			=====
336:	0208	74 81	MOV A,#081H
337:	020A	12 07 9C	CALL LCDINS
338:	020D	74 54	MOV A,#'T'
339:	020F	12 07 A2	CALL LCDDAT
340:	0212	74 47	MOV A,#'G'
341:	0214	12 07 A2	CALL LCDDAT
342:	0217	74 4C	MOV A,#'L'
343:	0219	12 07 A2	CALL LCDDAT
344:	021C	74 C1	MOV A,#CC1H
345:	021E	12 07 9C	CALL LCDINS
346:	0221	74 4A	MOV A,#'J'
347:	0223	12 07 A2	CALL LCDDAT
348:	0226	74 41	MOV A,#'A'
349:	0228	12 07 A2	CALL LCDDAT
350:	022B	74 4D	MOV A,#'M'
351:	022D	12 07 A2	CALL LCDDAT
352:	0230		DIS_R:
353:	0230	74 89	MOV A,#89H
354:	0232	12 07 9C	CALL LCDINS
355:	0235	74 2D	MOV A,#'-'
356:	0237	12 07 A2	CALL LCDDAT
357:	023A	74 8D	MOV A,#8DH
358:	023C	12 07 9C	CALL LCDINS
359:	023F	74 2D	MOV A,#'-'
360:	0241	12 07 A2	CALL LCDDAT
361:	0244	74 CA	MOV A,#0CAH
362:	0246	12 07 9C	CALL LCDINS
363:	0249	74 3A	MOV A,#':'
364:	024B	12 07 A2	CALL LCDDAT
365:	024E	74 CD	MOV A,#3CDH
366:	0250	12 07 9C	CALL LCDINS
367:	0253	74 3A	MOV A,#':'
368:	0255	12 07 A2	CALL LCDDAT

369:	0258	74 85	MOV	A, #085H
370:	025A	12 07 9C	CALL	LCDINS
371:	025D	74 3A	MOV	A, #' :
372:	025F	12 07 A2	CALL	LCDDAT
373:	0262	74 C5	MOV	A, #0C5H
374:	0264	12 07 9C	CALL	LCDINS
375:	0267	74 3A	MOV	A, #' :
376:	0269	12 07 A2	CALL	LCDDAT
377:				
378:	026C	75 32 04	MOV	RTC_ADR, #04
379:	026F	12 03 23	CALL	GET_RTC

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 3

Line	I	Addr	Code	Source
380:		0272	12 06 38	CALL PINDAH_DATA_RTC
381:		0275	74 87	MOV A, #087H
382:		0277	12 07 9C	CALL LCDINS
383:		027A	E5 36	MOV A, DATE
384:		027C	75 F0 0A	MOV B, #010
385:		027F	84	DIV AB
386:		0280	24 30	ADD A, #030H
387:		0282	12 07 A2	CALL LCDDAT
388:		0285	E5 F0	MOV A, B
389:		0287	24 30	ADD A, #030H
390:		0289	12 07 A2	CALL LCDDAT
391:		028C	75 32 05	MOV RTC_ADR, #05
392:		028F	12 03 23	CALL GET_RTC
393:		0292	12 06 38	CALL PINDAH_DATA_RTC
394:		0295	74 8A	MOV A, #08AH
395:		0297	12 07 9C	CALL LCDINS
396:		029A	E5 37	MOV A, MONTH
397:		029C	12 03 E3	CALL CARI_BULAN
398:		029F	12 07 80	CALL LCDSTRING
399:		02A2	75 32 06	MOV RTC_ADR, #06
400:		02A5	12 03 23	CALL GET_RTC
401:		02A8	12 06 38	CALL PINDAH_DATA_RTC
402:		02AB	74 8E	MOV A, #08EH
403:		02AD	12 07 9C	CALL LCDINS
404:		02B0	E5 38	MOV A, YEAR
405:		02B2	75 F0 0A	MOV B, #010
406:		02B5	84	DIV A3
407:		02B6	24 30	ADD A, #030H
408:		02B8	12 07 A2	CALL LCDDAT
409:		02BB	E5 F0	MOV A, B

410:	02B0	24 30	ADD	A,#030H
411:	02B7	12 07 A2	CALL	LCDDAT
412:	02C2	75 32 02	MOV	RTC_ADR,#02
413:	02C5	12 03 23	CALL	GET_RTC
414:	02C8	12 06 38	CALL	PINDAH_DATA_RTC
415:	02CB	74 C8	MOV	A,#0C8H
416:	02CD	12 07 9C	CALL	LCDINS
417:	02D0	35 35	MOV	A, HOUR
418:	02D2	75 F0 0A	MOV	B,#010
419:	02D5	84	DIV	AB
420:	02D6	24 30	ADD	A,#030H
421:	02D8	12 07 A2	CALL	LCDDAT
422:	02DB	E5 F0	MOV	A,B
423:	02DD	24 30	ADD	A,#030H
424:	02DF	12 07 A2	CALL	LCDDAT
425:	02E2	75 32 01	MOV	RTC_ADR,#01
426:	02E5	12 03 23	CALL	GET_RTC
427:	02E8	12 06 38	CALL	PINDAH_DATA_RTC
428:	02EB	74 CB	MOV	A,#0CBH
429:	02ED	12 07 9C	CALL	LCDINS
430:	02F0	E5 34	MOV	A, MIN
431:	02F2	75 F0 0A	MOV	B,#010
432:	02F5	84	DIV	AB
433:	02F6	24 30	ADD	A,#030H
434:	02F8	12 07 A2	CALL	LCDDAT
435:	02FB	E5 F0	MOV	A,B

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 9

Line	I	Addr	Code	Source
436:		02FD	24 30	ADD A,#030H
437:		02FF	12 07 A2	CALL LCDDAT
438:		0302	75 32 00	MOV RTC_ADR,#0
439:		0305	12 03 23	CALL GET_RTC
440:		0308	12 06 38	CALL PINDAH_DATA_RTC
441:		030B	74 CE	MOV A,#0CEH
442:		030D	12 07 9C	CALL LCDINS
443:		0310	E5 33	MOV A, SEC
444:		0312	75 F0 0A	MOV B,#010
445:		0315	84	DIV AB
446:		0316	24 30	ADD A,#030H
447:		0318	12 07 A2	CALL LCDDAT
448:		031B	E5 F0	MOV A,B
449:		031D	24 30	ADD A,#030H
450:		031F	12 07 A2	CALL LCDDAT

```

451:      0322      22      RET
452:
453:
454:
455:
456:
457:      0323      GET_RTC:
458:
459:      0323      C2 AF      CLR      EA
460:      0325      71 BD      ACALL     SIGNAL_START
461:      0327      4C 1F      JC        ERROR_GET
462:      0329      74 D0      MOV      A, #11010000B ;==> ADRESS
RTC DAN PROSES WRITE
463:      032B      71 7F      ACALL OUT8BIT
464:      032D      4C 19      JC        ERROR_GET
465:      032F      E5 32      MOV      A, RTC_ADR
466:      0331      71 7F      ACALL OUT8BIT
467:      0333      4C 13      JC        ERROR_GET
468:      0335      71 BD      ACALL     SIGNAL_START
469:      0337      74 D1      MOV      A, #11010001B ;==> ADRESS
RTC DAN PROSES READ
470:      0339      71 7F      ACALL OUT8BIT
471:      033B      4C 0B      JC        ERROR_GET
472:      033D      71 A4      ACALL IN8BIT
473:      033F      F5 31      MOV      RTC_DAT, A
474:      0341      40 05      JC        ERROR_GET
475:      0343      71 D5      ACALL     SIGNAL_STOP
476:      0345      D2 AF      SETB     EA
477:      0347      22      RET
478:
479:      0348      ERROR_GET:
480:      0348      74 80      MOV      A, #080H
481:      034A      12 07 9C    CALL     LCDINS
482:      034D      90 07 5A    MOV      DPTR, #TXT_RTC_ERROR_GET
483:      0350      12 07 90    CALL     LCDSTRING
484:      0353      22      RET
485:
486:
487:      0354      SET_RTC:
488:
489:      0354      C2 AF      CLR      EA
490:      0356      12 03 BD    LCALL     SIGNAL_START
491:      0359      40 18      JC        ERROR_SET

```

Line	I	Addr	Code	Source
492:		0358	74 D0	MOV A, #11010000B ;==> ADDRESS
RTC IAN PROSES WRITE				
493:		035D	12 03 7F	CALL OUT8BIT
494:		0360	40 11	JC ERROR_SET
495:		0362	E5 32	MOV A, RTC_ADR
496:		0364	71 7F	ACALL OUT8BIT
497:		0366	40 0B	JC ERROR_SET
498:		0368	E5 31	MOV A, RTC_DAT
499:		036A	71 7F	ACALL OUT8BIT
500:		036C	40 05	JC ERROR_SET
501:		036E	71 D5	ACALL SIGNAL_STOP
502:		0370	D2 AF	SETB EA
503:		0372	22	RET
504:				
505:		0373		ERROR_SET:
506:		0373	74 80	MOV A, #080H
507:		0375	12 07 9C	CALL LCDINS
508:		0378	90 07 6B	MOV DPTR, #TEXT_RTC_ERROR_SET
509:		037B	12 07 80	CALL LCDSTRING
510:		037E	22	REI
511:				
512:				=====
513:		037F		CUT8BIT:
514:				=====
515:		037F	CC F0	PUSH B
516:		0381	75 F0 08	MOV B, #8 ; akan digeser 8
kali (bit)				
517:		0384		CUTLOOP:
518:		0384	33	RLC A ; bit A.7 digeser
ke C di PSW				
519:		0385	92 B4	MOV SDA, C ; nilai C di
SDA				
520:		0387	0C	NOP ; tunggu
sementar sebelum.. ..				
521:		0388	D2 B5	SETB SCL ; SCL dibuat
= "1"				
522:		038A	0C	NOP ; tunggu lagi
523:		038B	0C	NOP
524:		038C	0C	NOP
525:		038D	0C	NOP
526:		038E	C2 B5	CLR SCL ; SCL dibuat
= "0", BYTE diambil "slave"				
527:		0390	D5 F0 F1	DJNZ B, CUTLOCP ; ulangi
terus sampai 8 kali				
528:		0393	D2 B4	SETB SDA ; SDA dibuat
"1"				
529:		0395	0C	NOP ; agar 'slave'
bisa mengirim ACK				
530:		0396	0C	NOP

```

531:      0397      D2 B5      SETB   SCL      ; clock ke 9
untuk menerima ACK
532:      0399      C0          NOP          ; tunggu data
533:      039A      C0          NOP
534:      039B      C0          NOP
535:      039C      C0          NOP
536:      039D      A2 B4      MOV     C,SDA    ; ambil ACK
yang dikirim "slave"
537:      039F      C2 B5      CLR     SCL      ; SCK kembali
ke "0"
538:      03A1      D0 F0      POP     B
539:      03A3      22          RET
540:
541:
542:      03A4          ;=====
                        IN8BIT:
543:          ;=====
544:      03A4      C0 F0      PUSH    B
545:      03A6      D2 B4      SETB    SDA      ; SDA="1"
agar "slave" bisa kirim BYTE
546:      03A8      75 F0 08    MOV     B,#8    ; akan digeser 8
kali (bit)
547:      03AB          INLOOP:

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 11

Line	I	Addr	Code	Source
548:		03AB	C0	NOP ; "slave"
boleh mengubah BYTE selama SCK="0"				
549:		03AC	C0	NOP
550:		03AD	C0	NOP
551:		03AE	D2 B5	SETB SCL ; SCK="1"
552:		03B0	C0	NOP ; tunggu
sementara				
553:		03B1	C0	NOP
554:		03B2	A2 B4	MOV C,SDA ; ambil
kiriman bit dari "slave"				
555:		03B4	E3	RLC A ; ditampung di A
556:		03B5	C2 B5	CLR SCL ; "slave"
boleh merubah BYTE selama SCK="0"				
557:		03B7	E5 F0 F1	DJNZ B,INLOOP
558:		03BA	D0 F0	POP B
559:		03BC	22	RET
560:				

```

561:                                     ;-----P-----
562:      03BD                                SIGNAL_START:
563:                                     ;-----
564:      03BD      D2 B4                      SETB   SDA
565:      03BF      D2 B5                      SETB   SCL      ; Memastikan
I2C Bus bisa dipakai
566:
567:      03C1      30 B4 0F                    JNB    SDA,BUSBUSY ; kalau SDA=1
I2C Bus tidak siap pakai
568:      03C4      30 B5 0C                    JNB    SCL,BUSBUSY ; kalau SCL=1
I2C Bus tidak siap pakai
569:
570:                                     ; Membuat sinyal START
571:      03C7      00                          NOP      ; tunggu
sementara
572:      03C8      C2 B4                      CLR     SDA
573:      03CA      00                          NOP      ; tunggu agar
BYTE benar-benar stabil
574:      03CB      00                          NOP
575:      03CC      00                          NOP
576:      03CD      00                          NOP
577:      03CE      00                          NOP
578:      03CF      C2 B5                      CLR     SCL
579:      03D1      C3                          CLR     C      ; C=0 berarti
berhasil membuat START
580:      03D2      22                          RET
581:      03D3                                BUSBUSY:
582:      03D3      D3                          SETB   C      ; C=1 berarti
gagal membuat START
583:      03D4      22                          RET
584:
585:                                     ;-----
586:      03D5                                SIGNAL_STOP:
587:                                     ;-----
588:      03D5      C2 B4                      CLR     SDA      ; SDA=0 low
beberapa saat
589:      03D7      00                          NOP
590:      03D8      00                          NOP
591:      03D9      D2 B5                      SETB   SCL      ; SCL=1
592:      03DB      00                          NOP      ; tunggu
sementara
593:      03DC      00                          NOP
594:      03DD      00                          NOP
595:      03DE      00                          NOP
596:      03DF      00                          NOP
597:      03E0      D2 B4                      SETB   SDA      ; SDA=1
598:      03E2      22                          RET
599:
600:                                     ;-----
601:      03E3                                CAR1_BULAN:
602:                                     ;-----
603:      03E3      B4 01 05                    CJNE    A,#01,CB2

```

688:	048A	80 BB		JMP	RSJC
689:	048C		RSJ2_1:		
690:	048C	15 36		DEC	DATE
691:	048E	80 B7		JMP	RSJC
692:	0490		RSJ3:		
693:	0490	20 95 D5		JB	TBMENU, RSJ1
694:	0493	30 95 FD		JNB	TBMENU, \$
695:	0496	E5 36		MOV	A, DATE
696:	0498	75 F0 0A		MOV	B, #10
697:	049B	84		DIV	AB
698:	049C	C4		SWAP	A
699:	049D	25 F0		ADD	A, B
700:	049F	F5 36		MOV	DATE, A
701:	04A1	75 32 04		MOV	RTC_ADR, #04H
702:	04A4	85 36 31		MOV	RTC_DAT, DATE
703:	04A7	71 54		CALL	SET_RTC
704:	04A9	02 05 0F		JMP	RSJ8
705:					
706:					; <-- SET BULAN
707:	04AC		RSJ4:		
708:	04AC	74 8A		MOV	A, #08AH
709:	04AE	12 07 9C		CALL	LCDINS
710:	04B1	E5 37		MOV	A, MONTH
711:	04B3	71 E3		CALL	CARI_BULAN
712:	04B5	12 07 8C		CALL	LCDSTRING
713:	04B8	74 8C		MOV	A, #08CH
714:	04BA	12 07 9C		CALL	LCDINS
715:	04BD	12 07 D3		CALL	DELAY

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 14

Line	I	Addr	Code	Source
716:		04C0	12 07 D3	CALL DELAY
717:		04C3		RSJ5:
718:		04C3	20 96 1A	JB TBUP, RSJ6
719:		04C6	30 96 FD	JNB TBUP, \$
720:		04C9	12 07 D3	CALL DELAY
721:		04CC	12 07 D3	CALL DELAY
722:		04CF	12 07 D3	CALL DELAY
723:		04D2	E5 37	MOV A, MONTH
724:		04D4	84 0C 05	CJNE A, #012, RSJ5_1
725:		04D7	75 37 01	MOV MONTH, #01
726:		04DA	80 D0	JMP RSJ4
727:		04DC		RSJ5_1:

857:	05DA	84		DIV	AB
858:	05DB	24 30		ADD	A, #030H
859:	05DD	12 07 A2		CALL	LCDDAT
860:	05E0	E5 F0		MOV	A, B
861:	* 05E2	24 30		ADD	A, #030H
862:	05E4	12 07 A2		CALL	LCDDAT
863:	05E7	74 CC		MOV	A, #0CCH
864:	05E9	12 07 9C		CALL	LCDINS
865:	05EC		RSJ17:		
866:	05EC	20 96 11		JB	TBUP, RSJ18
867:	05EF	30 96 FD		JNB	TBUP, \$
868:	05F2	E5 34		MOV	A, MIN
869:	05F4	B4 3B 05		CJNE	A, #059, RSJ17_1
870:	05F7	75 34 00		MOV	MIN, #0
871:	05FA	80 D4		JMP	RSJ16
872:	05FC		RSJ17_1:		
873:	05FC	05 34		INC	MIN
874:	05FE	80 D0		JMP	RSJ16
875:	0600		RSJ18:		
876:	0600	20 97 10		JB	TBDOWN, RSJ19
877:	0603	30 97 FD		JNB	TBDOWN, \$
878:	0606	E5 34		MOV	A, MIN
879:	0608	70 05		JNZ	RSJ18_1
880:	060A	75 34 3B		MOV	MIN, #059
881:	060D	80 C1		JMP	RSJ16
882:	060F		RSJ18_1:		
883:	060F	15 34		DEC	MIN

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 17

Line	I	Addr	Code	Source
884:		0611	80 BD	JMP RSJ16
885:		0613		RSJ19:
886:		0613	20 95 D6	JB TBMENU, RSJ17
887:		0616	30 95 FD	JNB TBMENU, \$
888:		0619	E5 34	MOV A, MIN
889:		061B	75 F0 0A	MOV B, #10
890:		061E	84	DIV AB
891:		061F	C4	SWAP A
892:		0620	25 F0	ADD A, B
893:		0622	F5 34	MOV MIN, A
894:		0624	75 32 01	MOV RTC_ADR, #01H
895:		0627	85 34 31	MOV RTC_DAT, MIN
896:		062A	71 54	CALL SET_RTC

```

897:      062C      12 07 D3      CALL      DELAY
898:      062F      74 0C      MOV      A, #DISPON
899:      0631      12 07 9C      CALL      LCDINS
900:      0634      12 07 D3      CALL      DELAY
901:      0637      22      RET
902:
903:
904:
905:      0638      ;=====
                      PINDAH_DATA_RTC:
906:      ;=====
907:      0638      E5 32      MOV      A, RTC_ADR
908:      063A      70 15      JNZ      PDR1
909:      063C      E5 31      MOV      A, RTC_DAT
910:      063E      54 0F      ANL      A, #CFH
911:      0640      F5 3A      MOV      TEMP, A
912:      0642      E5 31      MOV      A, RTC_DAT
913:      0644      C4      SWAP      A
914:      0645      54 0F      ANL      A, #CFH
915:      0647      75 F0 0A      MOV      B, #010
916:      064A      A4      MUL      AB
917:      064B      25 3A      ADD      A, TEMP
918:      064D      F5 33      MOV      SEC, A
919:      064F      80 76      SJMP     PDRX
920:
921:      0651      PDR1:      CJNE     A, #01, PDR2
922:      0651      B4 01 15      MOV      A, RTC_DAT
923:      0654      E5 31      ANL      A, #CFH
924:      0656      54 0F      MOV      TEMP, A
925:      0658      F5 3A      MOV      A, RTC_DAT
926:      065A      E5 31      MOV      A, RTC_DAT
927:      065C      C4      SWAP      A
928:      065D      54 0F      ANL      A, #CFH
929:      065F      75 F0 0A      MOV      B, #010
930:      0662      A4      MUL      AB
931:      0663      25 3A      ADD      A, TEMP
932:      0665      F5 34      MOV      MIN, A
933:      0667      80 5E      SJMP     PDRX
934:
935:      0669      PDR2:      CJNE     A, #02, PDR3
936:      0669      B4 02 15      MOV      A, RTC_DAT
937:      066C      E5 31      ANL      A, #CFH
938:      066E      54 0F      MOV      TEMP, A
939:      0670      F5 3A      MOV      A, RTC_DAT
940:      0672      E5 31      MOV      A, RTC_DAT
941:      0674      C4      SWAP      A

```

Line	I	Addr	Code	Source
940:		0675	54 0F	ANL A, #0FH
941:		0677	75 F0 0A	MOV B, #010
942:		067A	A4	MUL AB
943:		067B	25 3A	ADD A, TEMP
944:		067D	F5 35	MOV HOUR, A
945:		067F	80 46	SJMP PDRX
946:		0681		PDR3:
947:		0681	B4 04 15	CJNE A, #04, PDR4
948:		0684	E5 31	MOV A, RTC_DAT
949:		0686	54 0F	ANL A, #0FH
950:		0688	F5 3A	MOV TEMP, A
951:		068A	E5 31	MOV A, RTC_DAT
952:		068C	C4	SWAP A
953:		068D	54 0F	ANL A, #0FH
954:		068F	75 F0 0A	MOV B, #010
955:		0692	A4	MUL AB
956:		0693	25 3A	ADD A, TEMP
957:		0695	F5 36	MOV DATE, A
958:		0697	80 2E	SJMP PDRX
959:		0699		PDR4:
960:		0699	B4 05 15	CJNE A, #05, PDR5
961:		069C	E5 31	MOV A, RTC_DAT
962:		069E	54 0F	ANL A, #0FH
963:		06A0	F5 3A	MOV TEMP, A
964:		06A2	E5 31	MOV A, RTC_DAT
965:		06A4	C4	SWAP A
966:		06A5	54 0F	ANL A, #0FH
967:		06A7	75 F0 0A	MOV B, #010
968:		06AA	A4	MUL AB
969:		06AB	25 3A	ADD A, TEMP
970:		06AD	F5 37	MOV MONTH, A
971:		06AF	80 16	SJMP PDRX
972:		06B1		PDR5:
973:		06B1	B4 06 13	CJNE A, #06, PDRX
974:		06B4	E5 31	MOV A, RTC_DAT
975:		06B6	54 0F	ANL A, #0FH
976:		06B8	F5 3A	MOV TEMP, A
977:		06BA	E5 31	MOV A, RTC_DAT
978:		06BC	C4	SWAP A
979:		06BD	54 0F	ANL A, #0FH
980:		06BF	75 F0 0A	MOV B, #010
981:		06C2	A4	MUL AB
982:		06C3	25 3A	ADD A, TEMP
983:		06C5	F5 38	MOV YEAR, A
984:		06C7		PDRX:
985:		06C7	75 3A 00	MOV TEMP, #0
986:		06CA	22	RET
987:				
988:				
989:		06CB		C_JML_HR:

```

1031:      071B      B4 0B 05      CJNE      A,#011,CJH12
1032:      071E      75 39 1E      MOV       JMLHR,#03C
1033:      0721      80 06      SJMP      CJHX
1034:      0723      CJH12:
1035:      0723      B4 0C 03      CJNE      A,#012,CJHX
1036:      0726      75 39 1F      MOV       JMLHR,#031
1037:      0729      CJHX:
1038:      0729      22      RET
1039:      ;=====
1040:      072A      4A 41 4E 00      BLN_1      : DB 'JAN',0
1041:      072E      46 45 42 00      BLN_2      : DB 'FEB',0
1042:      0732      4D 41 52 00      BLN_3      : DB 'MAR',0
1043:      0736      41 50 52 00      BLN_4      : DB 'APR',0
1044:      073A      4D 45 49 00      BLN_5      : DB 'MAY',0
1045:      073E      4A 55 4E 00      BLN_6      : DB 'JUN',0
1046:      0742      4A 55 4C 00      BLN_7      : DB 'JUL',0
1047:      0746      41 47 54 00      BLN_8      : DB 'AUG',0
1048:      074A      53 45 50 00      BLN_9      : DB 'SEP',0
1049:      074E      4F 4B 54 00      BLN_10     : DB 'OCT',0
1050:      0752      4E 4F 56 00      BLN_11     : DB 'NOV',0
1051:      0756      44 45 53 00      BLN_12     : DB 'DEC',0

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 20

```

Line  I  Addr  Code      Source
-----
1052:
1053:      075A      20 45 52 52      TXT_RTC_ERROR_GET      :      DB
' ERROR GET RTC ',0
      075E      4F 52 20 47
      0762      45 54 20 52
      0766      54 43 20 20
      076A      00
1054:      076B      20 45 52 52      TXT_RTC_ERROR_SET      :      DB
' ERROR SET RTC ',0
      076F      4F 52 20 53
      0773      45 54 20 52
      0777      54 43 20 20
      077B      00
1055:
1056:      ;=====
1057:      077C      PRINTSTRINGLOOP:
1058:      ;=====
1059:      077C      12 07 A2      CALL      LCDDAT
1060:      077F      A3      INC      DPTR

```

```

1153:      07FA      22      RET
1154:      ;=====
1155:      07FB      DSP_PLAY:

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 22

Line	I	Addr	Code	Source
1156:		07FB	74 80	MOV A, #080H
1157:		07FD	F1 9C	CALL LCDINS
1158:		07FF	90 08 3F	MOV DPTR, #IDLE_1
1159:		0802	12 07 80	CALL LCDSTRING
1160:		0805	74 C0	MOV A, #0C0H
1161:		0807	12 07 9C	CALL LCDINS
1162:		080A	90 08 50	MOV DPTR, #IDLE_2
1163:		080D	12 07 80	CALL LCDSTRING
1164:		0810	22	RET
1165:		0811		DSP_RECORD:
1166:		0811	74 80	MOV A, #080H
1167:		0813	12 07 9C	CALL LCDINS
1168:		0816	90 08 61	MOV DPTR, #IDLE_3
1169:		0819	12 07 80	CALL LCDSTRING
1170:		081C	74 C0	MOV A, #0C0H
1171:		081E	12 07 9C	CALL LCDINS
1172:		0821	90 08 72	MOV DPTR, #IDLE_4
1173:		0824	12 07 80	CALL LCDSTRING
1174:		0827	22	RET
1175:				
1176:		0828		DSP_ISI_REC:
1177:		0828	74 80	MOV A, #080H
1178:		082A	12 07 9C	CALL LCDINS
1179:		082D	90 08 83	MOV DPTR, #IDLE_5
1180:		0830	12 07 80	CALL LCDSTRING
1181:		0833	74 C0	MOV A, #0C0H
1182:		0835	12 07 9C	CALL LCDINS
1183:		0838	90 08 94	MOV DPTR, #IDLE_6
1184:		083B	12 07 80	CALL LCDSTRING
1185:		083E	22	RET
1186:				
1187:				
1188:				;=====
1189:				;TEXT SECTION
1190:				;=====
1191:		083F	50 4C 41 59	IDLE_1 : DB 'PLAY SUARA', 0
	0843	20 53 55 41		
	0847	52 41 20 20		
	084B	20 20 20 20		

```

08CB      4F 50 20 52
08CF      45 43 4F 52
08D3      44 20 20 20
08D7      00
1204:    08D8      20 54 65 6B      IDLE_A : DB ' Tekan PB Menu ',0
08DC      61 6E 20 50
08E0      42 20 4D 65
08E4      6E 75 20 20
08E8      00
1205:
1206:
1207:    08E9      20 20 20 20      ABOUT_1 : DB '      BEL/ALARM',0
08ED      20 20 20 42
08F1      45 4C 2F 41
08F5      4C 41 52 4D
08F9      00
1208:    08FA      20 20 20 20      ABOUT_2 : DB '      SEKOLAH',0
08FE      20 20 20 20
0902      20 53 45 4B
0906      4F 4C 41 48
090A      00
1209:    090B      20 20 20 20      ABOUT_3 : DB '      BERBASIS',0
090F      20 20 20 20
0913      42 45 52 42
0917      41 53 49 53
091B      00
1210:    091C      20 20 20 20      ABOUT_4 : DB '      MCU AT89S51',0
0920      20 4D 43 55
0924      20 41 54 38
0928      39 53 35 31
092C      00
1211:    092D      20 20 54 52      ABOUT_7 : DB '      TRY ANTIRC A M',0

```

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 24

Line	I	Addr	Code	Source
		0931	59 20 41 4E	
		0935	54 4F 52 4F	
		0939	20 41 20 4D	
		093D	00	
1212:		093E	20 20 20 4E	ABOUT_8 : DB ' NIM:97.17.026',0
		0942	49 4D 3A 39	
		0946	37 2E 31 37	
		094A	2E 30 32 36	
		094E	00	

1213:	094F	44 4F 53 45	ABOUT_9 : DB 'DOSEN PEMBIMBING',0
	0953	4E 20 50 45	
	0957	4D 42 49 4D	
	095B	42 49 4E 47	
	095F	00	
1214:	0960	49 52 2E 20	ABOUT_A : DB 'IR. F.YUDI.L, MT',0
	0964	46 2E 59 55	
	0968	44 49 2E 4C	
	096C	2C 20 4D 54	
	0970	00	
1215:			
1216:	0971	20 20 20 4A	DS1 : DB ' JAM PELAJARAN',0
	0975	41 4D 20 50	
	0979	45 4C 41 4A	
	097D	41 52 41 4E	
	0981	00	
1217:	0982	20 20 20 20	DS2 : DB ' PERTAMA',0
	0986	20 20 20 20	
	098A	20 50 45 52	
	098E	54 41 4D 41	
	0992	00	
1218:	0993	20 20 20 20	DS3 : DB ' KEDUA',0
	0997	20 20 20 20	
	099B	20 20 20 4B	
	099F	45 44 55 41	
	09A3	00	
1219:	09A4	20 20 20 20	DS4 : DB ' KETIGA',0
	09A8	20 20 20 20	
	09AC	20 20 4B 45	
	09B0	54 49 47 41	
	09B4	00	
1220:	09B5	20 20 20 20	DS5 : DB ' KEEMPAT',0
	09B9	20 20 20 20	
	09BD	20 4B 45 45	
	09C1	4D 50 41 54	
	09C5	00	
1221:	09C6	20 20 20 20	DS6 : DB ' KELIMA',0
	09CA	20 20 20 20	
	09CE	20 20 4B 45	
	09D2	4C 49 4D 41	
	09D6	00	
1222:	09D7	20 20 20 20	DS7 : DB ' JAM',0
	09DB	20 20 20 20	
	09DF	20 20 20 20	
	09E3	20 4A 41 4D	
	09E7	00	
1223:	09E8	20 20 20 20	DS8 : DB ' ISTIRAHAT',0

Line	I	Addr	Code	Source
		09EC	20 20 20 49	
		09F0	53 54 49 52	
		09F4	41 48 41 54	
		09F8	00	
1224:		09F9	20 20 20 20	DS9 : DB ' KEENAM',0
		09FD	20 20 20 20	
		0A01	20 20 4B 45	
		0A05	45 4E 41 4D	
		0A09	00	
1225:		0A0A	20 20 20 20	DSA : DB ' KETUJJH',0
		0A0E	20 20 20 20	
		0A12	20 4B 45 54	
		0A16	55 4A 55 48	
		0A1A	00	
1226:		0A1B	20 20 20 20	DSB : DB ' PULANG',0
		0A1F	20 20 20 20	
		0A23	20 20 50 55	
		0A27	4C 41 4E 47	
		0A2B	00	
1227:				
1228:		0A2C		DSP_1:
1229:		0A2C	74 80	MOV A,#080H
1230:		0A2E	12 07 9C	CALL LCDINS
1231:		0A31	90 09 71	MOV DPTR,#DS1
1232:		0A34	12 07 80	CALL LCDSTRING
1233:		0A37	74 C0	MOV A,#0C0H
1234:		0A39	12 07 9C	CALL LCDINS
1235:		0A3C	90 09 82	MOV DPTR,#DS2
1236:		0A3F	12 07 80	CALL LCDSTRING
1237:		0A42	22	RET
1238:		0A43		DSP_2:
1239:		0A43	74 80	MOV A,#080H
1240:		0A45	12 07 9C	CALL LCDINS
1241:		0A48	90 09 71	MOV DPTR,#DS1
1242:		0A4B	12 07 80	CALL LCDSTRING
1243:		0A4E	74 C0	MOV A,#0C0H
1244:		0A50	12 07 9C	CALL LCDINS
1245:		0A53	90 09 93	MOV DPTR,#DS3
1246:		0A56	12 07 80	CALL LCDSTRING
1247:		0A59	22	RET
1248:				
1249:		0A5A		DSP_3:

1250:	0A5A	74 80	MOV	A, #080H
1251:	0A5C	12 07 9C	CALL	LCDINS
1252:	0A5F	9C 09 71	MOV	DPTR, #DS1
1253:	0A62	12 07 80	CALL	LCDSTRING
1254:	0A65	74 C0	MOV	A, #0C0H
1255:	0A67	12 07 9C	CALL	LCDINS
1256:	0A6A	9C 09 A4	MOV	DPTR, #DS4
1257:	0A6D	12 07 80	CALL	LCDSTRING
1258:	0A70	22	RET	
1259:	0A71		DSP_4:	
1260:	0A71	74 80	MOV	A, #080H
1261:	0A73	12 07 9C	CALL	LCDINS
1262:	0A76	9C 09 71	MOV	DPTR, #DS1
1263:	0A79	12 07 80	CALL	LCDSTRING

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 26

Line	I	Addr	Code	Source
1264:		0A7C	74 C0	MOV A, #0C0H
1265:		0A7E	12 07 9C	CALL LCDINS
1266:		0A81	9C 09 B5	MOV DPTR, #DS5
1267:		0A84	12 07 80	CALL LCDSTRING
1268:		0A87	22	RET
1269:		0A88		DSP_5:
1270:		0A88	74 80	MOV A, #080H
1271:		0A8A	12 07 9C	CALL LCDINS
1272:		0A8D	9C 09 71	MOV DPTR, #DS1
1273:		0A90	12 07 80	CALL LCDSTRING
1274:		0A93	74 C0	MOV A, #0C0H
1275:		0A95	12 07 9C	CALL LCDINS
1276:		0A98	9C 09 C6	MOV DPTR, #DS6
1277:		0A9B	12 07 80	CALL LCDSTRING
1278:		0A9E	22	RET
1279:		0A9F		DSP_ISTIRAHAT:
1280:		0A9F	74 80	MOV A, #080H
1281:		0AA1	12 07 9C	CALL LCDINS
1282:		0AA4	9C 09 D7	MOV DPTR, #DS7
1283:		0AA7	12 07 80	CALL LCDSTRING
1284:		0AAA	74 C0	MOV A, #0C0H
1285:		0AAC	12 07 9C	CALL LCDINS
1286:		0AAF	9C 09 E8	MOV DPTR, #DS8
1287:		0AB2	12 07 80	CALL LCDSTRING
1288:		0AB5	22	RET
1289:		0AB6		DSP_6:
1290:		0AB6	74 80	MOV A, #080H

```

1291:    0AB8    12 07 9C    CALL    LCDINS
1292:    0ABB    90 09 71    MOV     DPTR,#DS1
1293:    0ABE    12 07 80    CALL    LCDSTRING
1294:    0AC1    74 C0      MOV     A,#0C0H
1295:    0AC3    12 07 9C    CALL    LCDINS
1296:    0AC6    90 09 F9    MOV     DPTR,#DS9
1297:    0AC9    12 07 80    CALL    LCDSTRING
1298:    0ACC    22          RET
1299:    0ACD                    DSP_7:
1300:    0ACD    74 80      MOV     A,#080H
1301:    0ACF    12 07 9C    CALL    LCDINS
1302:    0AD2    90 09 71    MOV     DPTR,#DS1
1303:    0AD5    12 07 80    CALL    LCDSTRING
1304:    0AD8    74 C0      MOV     A,#0C0H
1305:    0ADA    12 07 9C    CALL    LCDINS
1306:    0ADD    90 0A 0A    MOV     DPTR,#DSA
1307:    0AE0    12 07 80    CALL    LCDSTRING
1308:    0AE3    22          RET
1309:    0AE4                    DSP_PULANG:
1310:    0AE4    74 80      MOV     A,#080H
1311:    0AE6    12 07 9C    CALL    LCDINS
1312:    0AE9    90 09 D7    MOV     DPTR,#DS7
1313:    0AEC    12 07 80    CALL    LCDSTRING
1314:    0AEF    74 C0      MOV     A,#0C0H
1315:    0AF1    12 07 9C    CALL    LCDINS
1316:    0AF4    90 0A 1B    MOV     DPTR,#DSB
1317:    0AF7    12 07 80    CALL    LCDSTRING
1318:    0AFA    22          RET
1319:

```

Line	I	Addr	Code	Source
1320:				=====
1321:				; DAFTAR IKLAN
1322:				=====
1323:		0AFB		IKLAN:
1324:		0AFB	74 80	MOV A,#080H
1325:		0AFD	12 07 9C	CALL LCDINS
1326:		0B00	90 08 E9	MOV DPTR,#ABOUT_1
1327:		0B03	12 07 80	CALL LCDSTRING
1328:		0B06	74 C0	MOV A,#0C0H
1329:		0B08	12 07 9C	CALL LCDINS
1330:		0B0B	90 08 FA	MOV DPTR,#ABOUT_2
1331:		0B0E	12 07 80	CALL LCDSTRING

1332:	0B11	74 03	MOV	A, #03
1333:	0B13	12 07 BE	CALL	DELAY3
1334:	0B16	74 80	MOV	A, #080H
1335:	0B18	12 07 9C	CALL	LCDINS
1336:	0B1B	90 09 0B	MOV	DPTR, #ABOUT_3
1337:	0B1E	12 07 80	CALL	LCDSTRING
1338:	0B21	74 C0	MOV	A, #0C0H
1339:	0B23	12 07 9C	CALL	LCDINS
1340:	0B26	90 09 1C	MOV	DPTR, #ABOUT_4
1341:	0B29	12 07 80	CALL	LCDSTRING
1342:	0B2C	74 03	MOV	A, #03
1343:	0B2E	12 07 BE	CALL	DELAY3
1344:				
1345:	0B31	74 80	MOV	A, #080H
1346:	0B33	12 07 9C	CALL	LCDINS
1347:	0B36	90 09 2D	MOV	DPTR, #ABOUT_7
1348:	0B39	12 07 80	CALL	LCDSTRING
1349:	0B3C	74 C0	MOV	A, #0C0H
1350:	0B3E	12 07 9C	CALL	LCDINS
1351:	0B41	90 09 3E	MOV	DPTR, #ABOUT_8
1352:	0B44	12 07 80	CALL	LCDSTRING
1353:	0B47	74 C3	MOV	A, #03
1354:	0B49	12 07 BE	CALL	DELAY3
1355:	0B4C	74 80	MOV	A, #080H
1356:	0B4E	12 07 9C	CALL	LCDINS
1357:	0B51	90 09 4F	MOV	DPTR, #ABOUT_9
1358:	0B54	12 07 80	CALL	LCDSTRING
1359:	0B57	74 C0	MOV	A, #0C0H
1360:	0B59	12 07 9C	CALL	LCDINS
1361:	0B5C	90 09 60	MOV	DPTR, #ABOUT_A
1362:	0B5F	12 07 80	CALL	LCDSTRING
1363:	0B62	74 03	MOV	A, #03
1364:	0B64	12 07 BE	CALL	DELAY3
1365:	0B67	74 01	MOV	A, #DISPCLR
1366:	0B69	12 07 9C	CALL	LCDINS
1367:	0B6C	22	RET	
1368:				
1369:	0B6D		DELAY1D:	
1370:	0B6D	75 22 C8	MOV	C_5mS, #0200
1371:	0B70		WAIT_1D:	
1372:	0B70	12 0B 77	CALL	DELAY_5MS
1373:	0B73	D5 22 FA	DJNZ	C_5mS, WAIT_1D
1374:	0B76	22	RET	
1375:				

Line	I	Addr	Code	Source
1376:		0B77		DELAY_5MS:
1377:		0B77	C0 89	PUSH TMOD
1378:		0B79	75 89 21	MOV TMOD,#21H
1379:		0B7C	75 8C ED	MOV TH0,#0EDH
1380:		0B7F	75 8A FF	MOV TL0,#0FFH
1381:		0B82	D2 8C	SETB TR0
1382:		0B84		WAIT_5MS:
1383:		0B84	10 8D 02	JBC TF0,DAH_5MS
1384:		0B87	80 FB	JMP WAIT_5MS
1385:		0B89		DAH_5MS:
1386:		0B89	C2 8C	CLR TR0
1387:		0B8B	D0 89	POP TMOD
1388:		0B8D	22	RET
1389:				END

register banks used: ---

no errors

LIST OF SYMBOLS

SYMBOL	TYPE	VALUE	LINE
??ASEM_51	NUMBER	8051	
??VERSION	NUMBER	0130	
ABOUT_1	CODE	08E9	1207
ABOUT_2	CODE	08FA	1208
ABOUT_3	CODE	090B	1209
ABOUT_4	CODE	091C	1210
ABOUT_7	CODE	092D	1211
ABOUT_8	CODE	093E	1212
ABOUT_9	CODE	094F	1213
ABOUT_A	CODE	0960	1214

AC	BIT	D6		
ACC	DATA		E0	
B	DATA		F0	
BLN_1	CODE		072A	1040
BLN_10	CODE		074E	1049
BLN_11	CODE		0752	1050
BLN_12	CODE		0756	1051
BLN_2	CODE		072E	1041
BLN_3	CODE		0732	1042
BLN_4	CODE		0736	1043
BLN_5	CODE		073A	1044
BLN_6	CODE		073E	1045
BLN_7	CODE		0742	1046
BLN_8	CODE		0746	1047
BLN_9	CODE		074A	1048
BUSBUSY	CODE		03D3	581
BUZZ	BIT	B6	18	
BUZZERZ	CODE		00C9	170
CARI_BULAN	CODE		03E3	601
CB10	CODE		042B	638
CB11	CODE		0433	642
CB12	CODE		043B	646
CB2	CODE		03EE	606
CB3	CODE		03F3	610
CB4	CODE		03FB	614
CB5	CODE		0403	618
CB6	CODE		040B	622
CB7	CODE		0413	626
CB8	CODE		041B	630
CB9	CODE		0423	634
CBX	CODE		043E	648
CEK_ALARM	CODE		00F9	197
CJH10	CODE		0713	1026
CJH11	CODE		071B	1030
CJH12	CODE		0723	1034
CJH2	CODE		06D3	994
CJH3	CODE		06DB	998
CJH4	CODE		06E3	1002
CJH5	CODE		06EB	1006
CJH6	CODE		06F3	1010

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 30

SYMBOL	TYPE	VALUE	LINE
CJH7	CODE	06FB	1014

CJH8	CODE	0703	1018
CJH9	CODE	070B	1022
CJHX	CODE	0729	1037
CY	BIT	D7	
C_5MS	NUMBER	0022	64
C_JML_HR	CODE	06CB	989
DAH_5MS	CODE	0B89	1385
DATE	NUMBER	0036	51
DATE_P	NUMBER	003D	58
DELAY	CODE	07D3	1125
DELAY1D	CODE	0B6D	1369
DELAY2	CODE	07C7	1117
DELAY3	CODE	07B2	1111
DELAY4	CODE	07B3	1106
DELAY4_1	CODE	07BB	1108
DELAY_5MS	CODE	0B77	1376
DISPCLR	NUMBER	0001	8
DISPLON	CODE	0783	1072
DISPON	NUMBER	000C	11
DIS_BACK	CODE	0208	334
DIS_R	CODE	0230	352
DPH	DATA	83	
DPL	DATA	82	
DS1	CODE	0971	1216
DS2	CODE	0982	1217
DS3	CODE	0993	1218
DS4	CODE	09A4	1219
DS5	CODE	09B5	1220
DS6	CODE	09C6	1221
DS7	CODE	09D7	1222
DS8	CODE	09E8	1223
DS9	CODE	09F9	1224
DSA	CODE	0A0A	1225
DSB	CODE	0A1B	1226
DSP_1	CODE	0A2C	1228
DSP_2	CODE	0A43	1238
DSP_3	CODE	0A5A	1249
DSP_4	CODE	0A71	1259
DSP_5	CODE	0A88	1269
DSP_6	CODE	0AB6	1289
DSP_7	CODE	0ACD	1299
DSP_ISI_REC	CODE	0828	1176
DSP_ISTIRAHAT	CODE	0A9F	1279
DSP_ON	CODE	07B1	1099
DSP_PLAY	CODE	07FB	1155
DSP_PULANG	CODE	0AE4	1309
DSP_RECORD	CODE	0811	1165
EA	BIT	AF	
ENTRMOD	NUMBER	0006	13
ERROR_GET	CODE	0348	479
ERROR_SET	CODE	0373	505
ES	BIT	AC	

ETO	BIT	A9
ETI	BIT	AB
EX0	BIT	A8

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 31

SYMBOL	TYPE	VALUE	LINE
EXI	BIT	AA	
EXTI0	CODE	C003	
EXTI1	CODE	C013	
FO	BIT	D5	
FUNCSET	NUMBER	0038	9
GET_RTC	CODE	0323	457
HOUR	NUMBER	0035	50
HOUR_P	NUMBER	003C	57
IDLE_1	CODE	083F	1191
IDLE_2	CODE	0850	1192
IDLE_3	CODE	0861	1194
IDLE_4	CODE	0872	1195
IDLE_5	CODE	0883	1197
IDLE_6	CODE	0894	1198
IDLE_7	CODE	08A5	1200
IDLE_8	CODE	08B6	1201
IDLE_9	CODE	08C7	1203
IDLE_A	CODE	08D8	1204
IE	DATA	A8	
IE0	BIT	89	
IE1	BIT	8B	
IKLAN	CODE	0AFB	1323
IN8BIT	CODE	03A4	542
INITLCD	CODE	07DC	1136
INLOOP	CODE	03AB	547
INPUT_SUARA	CODE	0077	125
INT0	BIT	B2	
INT1	BIT	B3	
IP	DATA	B8	
IS1	CODE	0077	126
IS2	CODE	0089	135
ISD_CE	BIT	92	29
ISD_EOM	BIT	94	30
ISD_PD	BIT	93	28
ISD_PR	BIT	B1	31
ISI_RECNUM	CODE	0053	109
IT0	BIT	88	
IT1	BIT	8A	

JAM_ISTIRAHAT	CODE	018F	273
JAM_KE_1	CCDE	00F9	199
JAM_KE_2	CCDE	0117	214
JAM_KE_3	CCDE	0135	228
JAM_KE_4	CCDE	0153	243
JAM_KE_5	CCDE	0171	258
JAM_KE_6	CCDE	01AD	288
JAM_KE_7	CCDE	01CB	302
JAM_PULANG	CODE	01E9	317
JMLHR	NUMBER	0039	54
KURSOR	CODE	0785	1067
LCDDAT	CODE	07A2	1090
LCDE	BIT	90	15
LCDINS	CCDE	079C	1035
LCDOUT	CCDE	07A6	1093
LCDRS	BIT	91	16
LCDSTRING	CODE	0780	1061
M	NUMBER	0040	61

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 32

SYMBOL	TYPE	VALUE	LINE
MIN	NUMBER	0034	49
MIN_P	NUMBER	003B	56
MONTH	NUMBER	0037	52
MONTH_P	NUMBER	003E	59
MUTER	CODE	C7D5	1127
MUTERX	CODE	07C9	1119
MUTERZ	CODE	07C0	1113
CUT8BIT	CODE	037F	513
CUTLOOP	CODE	0384	517
OUT_C	CODE	0207	331
OUT_SUARA	CODE	00E2	183
OV	BIT	D2	
P	BIT	D0	
PC	DATA	80	
P1	DATA	90	
P2	DATA	A0	
P3	DATA	B0	
PCON	DATA	87	
PDR1	CODE	0651	920
PDR2	CODE	0669	933
PDR3	CODE	0681	946
PDR4	CODE	0699	959

PDR5	CODE	06B1	972	
PDRX	CODE	06C7	984	
PINDAH_DATA_RTC	CODE		0638	905
PISD	NUMBER	00A0	32	
PLCD	NUMBER	0080	17	
PRINTSTRINGLOOP	CODE		077C	1057
PS	BIT	BC		
PSW	DATA		D0	
PT0	BIT	B9		
PT1	BIT	BB		
PUTR	NUMBER	0030	45	
PX0	BIT	B8		
PX1	BIT	BA		
R	NUMBER	0041	62	
RB8	BIT	9A		
RD	BIT	B7		
REC_NUM	NUMBER		0042	63
REN	BIT	9C		
RESET	CODE		0000	
R	BIT	98		
RING_AGAIN	CODE		00A4	149
RS0	BIT	D3		
RS1	BIT	D4		
RSJ0	CODE		0447	657
RSJ1	CODE		0468	672
RSJ10	CODE		053F	778
RSJ10_1	CODE		054E	785
RSJ11	CODE		0552	788
RSJ12	CODE		056B	802
RSJ13	CODE		0587	815
RSJ13_1	CODE		059A	823
RSJ14	CODE		059E	826
RSJ14_1	CODE		05B0	834
RSJ15	CODE		05B4	837

ASEM-51 V1.3
W.W. Heinz

Copyright (c) 2002 by
PAGE 33

SYMBOL	TYPE	VALUE	LINE
RSJ16	CODE	05D0	852
RSJ17	CODE	05EC	865
RSJ17_1	CODE	05FC	872
RSJ18	CODE	0600	875
RSJ18_1	CODE	060F	882
RSJ19	CODE	0613	885

RSJ1_1	CODE	0478	679
RSJ2	CODE	047C	682
RSJ2_1	CODE	048C	689
RSJ3	CODE	049C	692
RSJ4	CODE	04AC	707
RSJ5	CODE	04C3	717
RSJ5_1	CODE	04DC	727
RSJ6	CODE	04EC	730
RSJ6_1	CODE	04F0	737
RSJ7	CODE	04F4	740
RSJ8	CODE	050F	755
RSJ9	CODE	052B	768
RSJ9_1	CODE	053B	775
RTC_ADR	NUMBER	0032	47
RTC_DAT	NUMBER	0031	46
RXD	BIT	B0	
3BUF	DATA	99	
SCL	BIT	B5	22
3CON	DATA	98	
3DA	BIT	B4	23
SEC	NUMBER	0033	48
SET_RTC	CODE	0354	487
SET_WAKTU	CODE	043F	651
SIGNAL_START	CODE	03BD	562
SIGNAL_STOP	CODE	03D5	586
SINT	CODE	0023	
SM0	BIT	9F	
SM1	BIT	9E	
SM2	BIT	9D	
3P	DATA	81	
3START	CODE	0003	69
3START_A	CODE	0019	81
T0	BIT	B4	
T1	BIT	B5	
TAMPIL	CODE	0796	1079
TB8	BIT	9B	
TBDOWN	BIT	97	37
TBMENU	BIT	95	39
TBUP	BIT	96	38
TCON	DATA	88	
TEMP	NUMBER	003A	55
TEST_ISD	CODE	0037	95
TEST_ISD_OUT	CODE	0045	102
TF0	BIT	8D	
TF1	BIT	8F	
TH0	DATA	8C	
TH1	DATA	8D	
TI	BIT	99	
TIMER0	CODE	000B	
TIMER1	CODE	001B	

SYMBOL	TYPE	VALUE	LINE
TL0	DATA	8A	
TL1	DATA	8B	
TMOD	DATA	89	
TR0	BIT	8C	
TR1	BIT	8E	
TXD	BIT	B1	
TXT_RTC_ERROR_SET	CODE	075A	1053
TXT_RTC_ERROR_SET	CODE	076B	1054
WAIT_1D	CODE	0B70	1371
WAIT_5MS	CODE	0B84	1382
WR	BIT	B6	
YEAR	NUMBER	0038	53
YEAR_F	NUMBER	003F	60

```

        JNB     TBDOWN,$
        MOV     REC_NUM,#0
        MOV     A,#DISPCLR
        CALL    LCDINS
        CALL    DELAY1D
        JMP     TEST_ISD

;=====PROSEDUR MENU SUNTING INPUT SUARA=====
INPUT_SUARA:
IS1:
        MOV     A,#DISPCLR
        CALL    LCDINS
        CALL    DSP_RECORD
        MOV     A,REC_NUM
        MOV     P1SD,A
        CLR     ISD_PR
        CLR     ISD_PD
        CLR     ISD_CE

IS2:
        JB      TMENU,IS2
        JNB     TMENU,$
        SETB     ISD_PR
        SETB     ISD_PD
        SETB     ISD_CE
        MOV     A,REC_NUM
        ADD     A,#28
        MOV     REC_NUM,A
        MOV     A,#DISPCLR
        CALL    LCDINS
        CALL    DELAY1D
        RET

RING_AGAIN:
        MOV     A,#DISPCLR
        CALL    LCDINS
        CALL    DSP_PLAY
        MOV     A,#28
        MOV     P1SD,A
        SETB     ISD_PR
        CLR     ISD_PD
        CLR     ISD_CE
        JB      ISD_EOM,$
        SETB     ISD_EOM
        SETB     ISD_PD
        SETB     ISD_CE
        ;MOV     A,REC_NUM
        ;ADD     A,#28
        ;MOV     REC_NUM,A
        MOV     A,#DISPCLR
        CALL    LCDINS
        CALL    DELAY1D
        ;RET
        JMP     RING_AGAIN

BUZZER2:
        CLR     BUZZ
        CALL    DELAY1D
        SETB     BUZZ
        CLR     BUZZ
        CALL    DELAY1D

```

```

SETB BUZZ
CALL DELAY1D
CLR BUZZ
CALL DELAY1D
SETB BUZZ
RET

```

OUT_SUARA:

```

MOV P1SD,A
SETB ISD_PR
CLR ISD_PD
CLR ISD_CE
JB ISD_EOM,$
SETB ISD_EOM
SETB ISD_PD
SETB ISD_CE
MOV A,#DISPCLR
CALL LCDINS
RET

```

CEK_ALARM:

```

=====
JAM_KE_1: ; 07.00
MOV A, HOUR
CJNE A, #7, JAM_KE_2
MOV A, MIN
CJNE A, #0, JAM_KE_2
MOV A, SEC
CJNE A, #0, JAM_KE_2
MOV A, #DISPCLR
CALL LCDINS
CALL DSP_1
CALL BUZZERZ
MOV A, #0
CALL OUT_SUARA
RET

```

```

JAM_KE_2: ; 07.45
MOV A, HOUR
CJNE A, #7, JAM_KE_3
MOV A, MIN
CJNE A, #45, JAM_KE_3
MOV A, SEC
CJNE A, #0, JAM_KE_3
MOV A, #DISPCLR
CALL LCDINS
CALL DSP_2
CALL BUZZERZ
MOV A, #28
CALL OUT_SUARA
RET

```

```

JAM_KE_3: ; 08.30
MOV A, HOUR
CJNE A, #8, JAM_KE_4
MOV A, MIN
CJNE A, #30, JAM_KE_4
MOV A, SEC
CJNE A, #0, JAM_KE_4

```

```

MOV    A, #DISPCLR
CALL   LCDINS
CALL   DSP_3
CALL   BUZZERZ
MOV    A, #56
CALL   OUT_SUARA
RET

```

JAM_KE_4: ; 09.15

```

MOV    A, HOUR
CJNE   A, #9, JAM_KE_5
MOV    A, MIN
CJNE   A, #15, JAM_KE_5
MOV    A, SEC
CJNE   A, #0, JAM_KE_5
MOV    A, #DISPCLR
CALL   LCDINS
CALL   DSP_4
CALL   BUZZERZ
MOV    A, #94
CALL   OUT_SUARA
RET

```

JAM_KE_5: ; 10.00

```

MOV    A, HOUR
CJNE   A, #10, JAM_ISTIRAHAT
MOV    A, MIN
CJNE   A, #0, JAM_ISTIRAHAT
MOV    A, SEC
CJNE   A, #0, JAM_ISTIRAHAT
MOV    A, #DISPCLR
CALL   LCDINS
CALL   DSP_5
CALL   BUZZERZ
MOV    A, #112
CALL   OUT_SUARA
RET

```

JAM_ISTIRAHAT: ; 10.45

```

MOV    A, HOUR
CJNE   A, #10, JAM_KE_6
MOV    A, MIN
CJNE   A, #45, JAM_KE_6
MOV    A, SEC
CJNE   A, #0, JAM_KE_6
MOV    A, #DISPCLR
CALL   LCDINS
CALL   DSP_ISTIRAHAT
CALL   BUZZERZ
MOV    A, #140
CALL   OUT_SUARA
RET

```

JAM_KE_6: ; 11.00

```

MOV    A, HOUR
CJNE   A, #11, JAM_KE_7
MOV    A, MIN
CJNE   A, #0, JAM_KE_7
MOV    A, SEC

```

```

MOV A, #89H
CALL LCDINS
MOV A, #'-'
CALL LCDDAT
MOV A, #8DH
CALL LCDINS
MOV A, #'-'
CALL LCDDAT
MOV A, #0CAH
CALL LCDINS
MOV A, #':'
CALL LCDDAT
MOV A, #0CDH
CALL LCDINS
MOV A, #':'
CALL LCDDAT
MOV A, #085H
CALL LCDINS
MOV A, #':'
CALL LCDDAT
MOV A, #0C5H
CALL LCDINS
MOV A, #':'
CALL LCDDAT

```

```

MOV RTC_ADR, #04
CALL GET_RTC
CALL PINDAH_DATA_RTC
MOV A, #087H
CALL LCDINS
MOV A, DATE
MOV B, #010
DIV AB
ADD A, #030H
CALL LCDDAT
MOV A, B
ADD A, #030H
CALL LCDDAT
MOV RTC_ADR, #05
CALL GET_RTC
CALL PINDAH_DATA_RTC
MOV A, #08AH
CALL LCDINS
MOV A, MONTH
CALL CARI_BULAN
CALL LCDSTRING
MOV RTC_ADR, #06
CALL GET_RTC
CALL PINDAH_DATA_RTC
MOV A, #08EH
CALL LCDINS
MOV A, YEAR
MOV B, #010
DIV AB
ADD A, #030H
CALL LCDDAT
MOV A, B
ADD A, #030H
CALL LCDDAT

```

```

MOV     RTC_ADR,#02
CALL    GET_RTC
CALL    PINDAH_DATA_RTC
MOV     A,#0CBH
CALL    LCDINS
MOV     A,HOUR
MOV     B,#010
DIV     AB
ADD     A,#030H
CALL    LCDDAT
MOV     A,B
ADD     A,#030H
CALL    LCDDAT
MOV     RTC_ADR,#01
CALL    GET_RTC
CALL    PINDAH_DATA_RTC
MOV     A,#0CBH
CALL    LCDINS
MOV     A,MIN
MOV     B,#010
DIV     AB
ADD     A,#030H
CALL    LCDDAT
MOV     A,B
ADD     A,#030H
CALL    LCDDAT
MOV     RTC_ADR,#0
CALL    GET_RTC
CALL    PINDAH_DATA_RTC
MOV     A,#0CEH
CALL    LCDINS
MOV     A,SEC
MOV     B,#010
DIV     AB
ADD     A,#030H
CALL    LCDDAT
MOV     A,B
ADD     A,#030H
CALL    LCDDAT
RET
;=====

```

```

;=====
;=====
GET_RTC:
;=====
        CLR     EA
        ACALL    SIGNAL_START
        JC       ERROR_GET
        MOV     A,#11010000B    ;==> ADDRESS RTC LAN
PROSES WRITE
        ACALL    OUT8BIT
        JC       ERROR_GET
        MOV     A,RTC_ADR
        ACALL    OUT8BIT
        JC       ERROR_GET
        ACALL    SIGNAL_START

```

```

        MOV    A,#11010001B    ;==> ADDRESS RTC DAN
PROSES_READ
        ACALL  OUT8BIT
        JC     ERROR_GET
        ACALL  IN8BIT
        MOV    RTC_DAT,A
        JC     ERROR_GET
        ACALL  SIGNAL_STOP
        SETB   EA
        RET

```

```

ERROR_GET:
        MOV    A,#080H
        CALL   LCDINS
        MOV    DPTR,#TXT_RTC_ERROR_GET
        CALL   LCDSTRING
        RET

```

```

;=====
SET_RTC:
;=====
        CLR    EA
        LCALL  SIGNAL_START
        JC     ERROR_SET
        MOV    A,#11010000B    ;==> ADDRESS RTC DAN
PROSES_WRITE
        CALL   OUT8BIT
        JC     ERROR_SET
        MOV    A,RTC_ADR
        ACALL  OUT8BIT
        JC     ERROR_SET
        MOV    A,RTC_DAT
        ACALL  OUT8BIT
        JC     ERROR_SET
        ACALL  SIGNAL_STOP
        SETB   EA
        RET

```

```

ERROR_SET:
        MOV    A,#080H
        CALL   LCDINS
        MOV    DPTR,#TXT_RTC_ERROR_SET
        CALL   LCDSTRING
        RET

```

```

;=====
OUT8BIT:
;=====
        PUSH   B
        MOV    B,#8            ; akan digeser 8 kali (bit)
OUTLOOP:
        RLC    A                ; bit A.7 digeser ke C di
PSW
        MOV    SDA,C            ; nilai C di SDA
        NOP                     ; tunggu sebentar sebelum..
        SETB   SCL              ; SCL dibuat = "1"
        NOP                     ; tunggu lagi
        NOP

```

```

        NOP
        NOP
        CLR     SCL           ; SCL dibuat ="0", BYTE
diambil "slave"
        DJNZ    B,OUTLOOP    ; ulangi terus sampai 8
kali
        SETB    SDA           ; SDA dibuat "1"
        NOP      ; agar 'slave' bisa mengirim
ACK
        NOP
        SETB    SCL           ; clock ke 9 untuk menerima
ACK
        NOP      ; tunggu dulu
        NOP
        NOP
        NOP
        MOV     C,SDA         ; ambil ACK yang dikirim
"slave"
        CLR     SCL           ; SCK kembali ke "0"
        POP     B
        RET

;=====
IN8BIT:
;=====
        PUSH    B
        SETB    SDA           ; SDA="1" agar "slave" bisa
kirim BYTE
        MOV     B,#8          ; akan digeser 8 kali (bit)
INLOOP:
        NOP      ; "slave" boleh mengubah
BYTE selama SCK="0"
        NOP
        NOP
        SETB    SCL           ; SCK="1"
        NOP      ; tunggu sebentar
        NOP
        MOV     C,SDA         ; ambil kiriman bit dari
"slave"
        RLC     A             ; ditampung di A
        CLR     SCL           ; "slave" boleh merubah
BYTE selama SCK="0"
        DJNZ    B,INLOOP
        POP     B
        RET

;=====P=====
SIGNAL_START:
;=====
        SETB    SDA
        SETB    SCL           ; Memastikan I2C Bus bisa
dipakai
        JNB     SDA,BUSBUSY    ; kalau SDA=0 I2C Bus tidak
siap pakai
        JNB     SCL,BUSBUSY    ; kalau SCL=0 I2C Bus tidak
siap pakai
        ; Membuat sinyal START

```

```

        NOP                ; tunggu sebentar
        CLR    SDA
        NOP                ; tunggu agar BYTE benar-
benar stabil
        NOP
        NOP
        NOP
        NOP
        CLR    SCL
        CLR    C            ; C=0 berarti berhasil
membuat START
        RET
BUSBUSY:
        SETB   C            ; C=1 berarti gagal membuat
START
        RET

```

```

=====
SIGNAL_STOP:
=====

```

```

        CLR    SDA        ; SDA=0 low beberapa saat
        NOP
        NOP
        SETB   SCL        ; SCL=1
        NOP                ; tunggu sebentar
        NOP
        NOP
        NOP
        SETB   SDA        ; SDA=1
        RET

```

```

=====
CARI_BULAN:
=====

```

```

        CJNE   A, #01, CB2
        MOV    DPTR, #BLN_1
        SJMP   CBX
CB2:
        CJNE   A, #02, CB3
        MOV    DPTR, #BLN_2
        SJMP   CBX
CB3:
        CJNE   A, #03, CB4
        MOV    DPTR, #BLN_3
        SJMP   CBX
CB4:
        CJNE   A, #04, CB5
        MOV    DPTR, #BLN_4
        SJMP   CBX
CB5:
        CJNE   A, #05, CB6
        MOV    DPTR, #BLN_5
        SJMP   CBX
CB6:
        CJNE   A, #06, CB7
        MOV    DPTR, #BLN_6
        SJMP   CBX
CB7:

```

```

        CJNE    A, #07, CB8
        MOV     DPTR, #BLN_7
        SJMP    CBX
CB8:
        CJNE    A, #08, CB9
        MOV     DPTR, #BLN_8
        SJMP    CBX
CB9:
        CJNE    A, #09, CB10
        MOV     DPTR, #BLN_9
        SJMP    CBX
CB10:
        CJNE    A, #010, CB11
        MOV     DPTR, #BLN_10
        SJMP    CBX
CB11:
        CJNE    A, #011, CB12
        MOV     DPTR, #BLN_11
        SJMP    CBX
CB12:
        MOV     DPTR, #BLN_12
CBX:
        RET
;=====
SET_WAKTU:
;=====
        MOV     A, #00001101B
        CALL    LCDINS
        JMP     RSJ4
; <-- SET TANGGAL
RSJ0:
        MOV     A, MONTH
        CALL    C_JML_HR
        MOV     A, #087H
        CALL    LCDINS
        MOV     A, DATE
        MOV     B, #010
        DIV     AB
        ADD     A, #030H
        CALL    LCDDAT
        MOV     A, B
        ADD     A, #030H
        CALL    LCDDAT
        MOV     A, #088H
        CALL    LCDINS
RSJ1:
        JB      TBUP, RSJ2
        JNB     TBUP, $
        MOV     A, DATE
        CJNE    A, JMLHR, RSJ1_1
        MOV     DATE, #01
        JMP     RSJ0
RSJ1_1:
        INC     DATE
        JMP     RSJ0
RSJ2:
        JB      TBDOWN, RSJ3
        JNB     TBDOWN, $
        MOV     A, DATE

```

```

        CJNE    A, #01, RSJ2_1
        MOV     DATE, JMLHR
        JMP     RSJ0
RSJ2_1:
        DEC     DATE
        JMP     RSJ0
RSJ3:
        JB      TBMENU, RSJ1
        JNB     TBMENU, $
        MOV     A, DATE
        MOV     B, #10
        DIV     AB
        SWAP    A
        ADD     A, B
        MOV     DATE, A
        MOV     RTC_ADR, #04H
        MOV     RTC_DAT, DATE
        CALL    SET_RTC
        JMP     RSJ8

                                ;<-- SET BULAN
RSJ4:
        MOV     A, #08AH
        CALL    LCDINS
        MOV     A, MONTH
        CALL    CARI_BULAN
        CALL    LCDSTRING
        MOV     A, #08CH
        CALL    LCDINS
        CALL    DELAY
        CALL    DELAY
RSJ5:
        JB      TBUP, RSJ6
        JNB     TBUP, $
        CALL    DELAY
        CALL    DELAY
        CALL    DELAY
        MOV     A, MONTH
        CJNE    A, #012, RSJ5_1
        MOV     MONTH, #01
        JMP     RSJ4
RSJ5_1:
        INC     MONTH
        JMP     RSJ4
RSJ6:
        JB      TBDOWN, RSJ7
        JNB     TBDOWN, $
        MOV     A, MONTH
        CJNE    A, #01, RSJ6_1
        MOV     MONTH, #012
        JMP     RSJ4
RSJ6_1:
        DEC     MONTH
        JMP     RSJ4
RSJ7:
        JB      TBMENU, RSJ5
        JNB     TBMENU, $
        MOV     A, MONTH
        MOV     B, #10

```

```

        DIV AB
        SWAP      A
        ADD A,B
        MOV MONTH,A
        MOV      RTC_ADR,#05H
        MOV      RTC_DAT,MONTH
        CALL     SET_RTC
        JMP      RSJ0

; <-- SET TAHUN
RSJ8:
        MOV      A,#08EH
        CALL     LCDINS
        MOV      A,YEAR
        MOV      B,#C10
        DIV      AB
        ADD      A,#030H
        CALL     LCDDAT
        MOV      A,B
        ADD      A,#030H
        CALL     LCDDAT
        MOV      A,#08FH
        CALL     LCDINS
RSJ9:
        JB       TBUP,RSJ10
        JNB      TBUP,$
        MOV      A,YEAR
        CJNE     A,#099,RSJ9_1
        MOV      YEAR,#0
        JMP      RSJ8
RSJ9_1:
        INC      YEAR
        JMP      RSJ8
RSJ10:
        JB       TBDOWN,RSJ11
        JNB      TBDOWN,$
        MOV      A,YEAR
        JNZ      RSJ10_1
        MOV      YEAR,#099
        JMP      RSJ8
RSJ10_1:
        DEC      YEAR
        JMP      RSJ8
RSJ11:
        JB       TBMENU,RSJ9
        JNB      TBMENU,$
        MOV      A,YEAR
        MOV      B,#10
        DIV      AB
        SWAP      A
        ADD      A,B
        MOV      YEAR,A
        MOV      RTC_ADR,#06H
        MOV      RTC_DAT,YEAR
        CALL     SET_RTC

; <-- SET JAM
RSJ12:
        MOV      A,#0C8H

```

```

        MOV     A, #0CCH
        CALL    LCDINS
RSJ17:
        JB      TBUP, RSJ18
        JNB     TBUP, $
        MOV     A, MIN
        CJNE    A, #059, RSJ17_1
        MOV     MIN, #0
        JMP     RSJ16
RSJ17_1:
        INC     MIN
        JMP     RSJ16
RSJ18:
        JB      TBCDOWN, RSJ19
        JNB     TBCDOWN, $
        MOV     A, MIN
        JNZ     RSJ18_1
        MOV     MIN, #059
        JMP     RSJ16
RSJ18_1:
        DEC     MIN
        JMP     RSJ16
RSJ19:
        JB      TBMENU, RSJ17
        JNB     TBMENU, $
        MOV     A, MIN
        MOV     B, #10
        DIV     AB
        SWAP    A
        ADD     A, B
        MOV     MIN, A
        MOV     RTC_ADR, #01H
        MOV     RTC_DAT, MIN
        CALL    SET_RTC
        CALL    DELAY
        MOV     A, #DISPON
        CALL    LCDINS
        CALL    DELAY
        RET

```

```

;=====
PINDAH_DATA_RTC:
;=====
        MOV     A, RTC_ADR
        JNZ     PDR1
        MOV     A, RTC_DAT
        ANL     A, #0FH
        MOV     TEMP, A
        MOV     A, RTC_DAT
        SWAP    A
        ANL     A, #0FH
        MOV     B, #010
        MUL     AB
        ADD     A, TEMP
        MOV     SEC, A
        SJMP    PDRX
PDR1:
        CJNE    A, #01, PDR2

```

```

        MUL      AB
        ADD      A,TEMP
        MOV      YEAR,A
PDRX:
        MOV      TEMP,#0
        RET

;=====
C_JML_HR:
;=====
        CJNE     A,#01,CJH2
        MOV      JMLHR,#031
        SJMP     CJHX
CJH2:
        CJNE     A,#02,CJH3
        MOV      JMLHR,#028
        SJMP     CJHX
CJH3:
        CJNE     A,#03,CJH4
        MOV      JMLHR,#031
        SJMP     CJHX
CJH4:
        CJNE     A,#04,CJH5
        MOV      JMLHR,#030
        SJMP     CJHX
CJH5:
        CJNE     A,#05,CJH6
        MOV      JMLHR,#031
        SJMP     CJHX
CJH6:
        CJNE     A,#06,CJH7
        MOV      JMLHR,#030
        SJMP     CJHX
CJH7:
        CJNE     A,#07,CJH8
        MOV      JMLHR,#031
        SJMP     CJHX
CJH8:
        CJNE     A,#08,CJH9
        MOV      JMLHR,#031
        SJMP     CJHX
CJH9:
        CJNE     A,#09,CJH10
        MOV      JMLHR,#030
        SJMP     CJHX
CJH10:
        CJNE     A,#010,CJH11
        MOV      JMLHR,#031
        SJMP     CJHX
CJH11:
        CJNE     A,#011,CJH12
        MOV      JMLHR,#030
        SJMP     CJHX
CJH12:
        CJNE     A,#012,CJHX
        MOV      JMLHR,#031
CJHX:
        RET
;=====

```

```

CALL DELAY
RET
DSP_ON:
MOV     A,#DISPON
CALL    LCDINS
CALL    DELAY
RET

;=====
DELAY4:
MOV     R7,#0250
DELAY4_1:
DJNZ    R7,DELAY4_1
RET
DELAY3:
MOV     PUTR,A
MUTERZ:
CALL    DELAY2
DJNZ    PUTR,MUTERZ
RET
DELAY2:
MOV     R5,#130
MUTERX:
MOV     R6,#250
CALL    DELAY
DJNZ    R6,$
DJNZ    R5,MUTERX
RET
DELAY:
MOV     R3,#08
MUTER:
MOV     R4,#0255
DJNZ    R4,$
DJNZ    R3,MUTER
RET

```

```

;=====
INITLCD:
;=====
MOV     A,#DISPCLR
CALL    LCDINS
CALL    DELAY
MOV     A,#FUNCSET
CALL    LCDINS
CALL    DELAY
MOV     A,#DISPON
CALL    LCDINS
CALL    DELAY
MOV     A,#ENTRMOD
CALL    LCDINS
CALL    DELAY
MOV     A,#DISPCLR
CALL    LCDINS
CALL    DELAY2
RET

```

```

;=====
DSP_PLAY:

```

```

CALL    DELAY3
MOV     A, #080H
CALL    LCDINS
MOV     DPTR, #ABOUT_3
CALL    LCDSTRING
MOV     A, #0C0H
CALL    LCDINS
MOV     DPTR, #ABOUT_4
CALL    LCDSTRING
MOV     A, #03
CALL    DELAY3

MOV     A, #080H
CALL    LCDINS
MOV     DPTR, #ABOUT_7
CALL    LCDSTRING
MOV     A, #0C0H
CALL    LCDINS
MOV     DPTR, #ABOUT_8
CALL    LCDSTRING
MOV     A, #03
CALL    DELAY3
MOV     A, #080H
CALL    LCDINS
MOV     DPTR, #ABOUT_9
CALL    LCDSTRING
MOV     A, #0C0H
CALL    LCDINS
MOV     DPTR, #ABOUT_A
CALL    LCDSTRING
MOV     A, #03
CALL    DELAY3
MOV     A, #DISPCLR
CALL    LCDINS
RET

```

```

DELAY1D:
MOV     C_5mS, #0200
WAIT_1D:
CALL    DELAY_5MS
DJNZ    C_5mS, WAIT_1D
RET

DELAY_5MS:
PUSH    TMOD
MOV     TMOD, #21H
MOV     TH0, #0EDH
MOV     TL0, #0FFH
SETB    TR0
WAIT_5MS:
JBC     TF0, DAH_5MS
JMP     WAIT_5MS
DAH_5MS:
CLR     TR0
POP     TMOD
RET

END

```

10

11

12

13

14

15
