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

SKRIPSI

PERENCANAAN DAN PEMBUATAN REMOTE KONTROL PENYETING TIMER UNTUK MENGUBAH CHANEL SECARA OTOMATIS PADA TELEVISI DENGAN MENGGUNAKAN MIKROKONTROLER AT89S8252

Disusun Oleh :
NUR SURYANINGRAT
Nim 01.17.155

MARET 2006

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

SKRIPSI

[illegible]

Disusun oleh :
[illegible]
NIM [illegible]

[illegible]

LEMBAR PERSETUJUAN

PERENCANAAN DAN PEMBUATAN REMOTE KONTROL PENYETING TIMER UNTUK MENGUBAH CHANEL SECARA OTOMATIS PADA TELEVISI DENGAN MENGGUNAKAN MIKROKONTROLER AT89S8252

SKRIPSI

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

Disusun oleh :

NUR SURYANINGRAT

Nim : 01.17.155

Mengetahui,
Ketua Jurusan T.Elektro S-1

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

Diperiksa dan Disetujui
Dosen Pembimbing

Joseph Dedy Irawan ST, MT.
NIP. 132315178

KONSENTRASI T.ELEKTRONIKA
JURUSAN T.ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2006

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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Nur Suryaningrat.

Nim : 01.17.155.

Jurusan : Teknik Elektro S-1.

Konsentrasi : Teknik Elektronika.

Judul Skripsi : **Perencanaan Dan Pembuatan Remote Kontrol Penyeting Timer Untuk Mengubah Chanel Secara Otomatis Pada Televisi Dengan Menggunakan Mikrokontroler AT89S8252.**

Dipertahankan Dihadapan Tim Penguji Skripsi Jenjang Strata Satu (S-1) Pada :

Hari : Sabtu.

Tanggal : 18 Maret 2006.

Dengan Nilai : 80.6 (A).

Panitia Ujian Skripsi

Ketua	Sekretaris
(Ir. Mochtar Asroni, MSME)	(Ir. F. Yudi Limpraptono, MT)
NIP.Y. 1018100036	NIP.Y. 1039500274

Anggota Penguji

Penguji I	Penguji II
(Ir. Sidik Noertjahjono, MT)	(Sotyohadi, ST)
NIP. 1028700163	

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

LEMBAR BIMBINGAN SKRIPSI

1. Nama	: Nur Suryaningrat
2. Nim	: 01.17.155
3. Jurusan	: Teknik Elektro S-1
4. Konsentrasi	: Teknik Elektronika
5. Masa Bimbingan	: 5 Desember 2005 s/d 5 Mei 2006
6. Judul Skripsi	: Perencanaan Dan Pembuatan Remote Kontrol Penyeting Timer Untuk Mengubah Chanel Secara Otomatis Pada Televisi Dengan Menggunakan Mikrokontroler AT89S8252
7. Tanggal Pengajuan Skripsi	: 8 Oktober 2005
8. Selesai Pengajuan Skripsi	: 18 Maret 2006
9. Pembimbing	: Joseph Dedy Irawan ST, MT.
10. Telah Dievaluasi Dengan Nilai	: 83.5

Diperiksa dan Disetujui Dosen Pembimbing	Mengetahui, Ketua Jurusan T. Elektro S-1
Joseph Dedy Irawan ST, MT NIP. 132315178	Ir. F.Yudi Limpraptono, MT NIP.Y.1039500274

بسم الله الرحمن الرحيم

Alhamdulillahhirobbilalamin saya panjatkan puji syukur kehadirat_Mu Ya ALLAH, yang menciptakan bumi dan langit, juga shalawat serta salam saya panjatkan kepada junjungan besar Nabi Muhammad Saw beserta para Rasulnya, hanya dengan izin_Mu,dan ridho'Mu saya mampu menyelesaiakan penulisan skripsi ini.

Kupersembahkan karya kecilku ini.............

Special untuk's kedua orang tuaku, bapak n mama tercinta (Drs, H, Sibik - Pramujito,Bc.Ap & Hj, Sri Kamarumi) yang telah merawat dan mendidik aku tanpa lelah keluh kesah sampai menjadi seorang Sarjana dengan segala pengorbanan material spiritual dan semoga Beliau selalu mulia dihadapan-Mu Ya Allah. Amin

Untuk Sodara-Sodaraku tercinta:

Mbakku Sri Suryawati, Sp.(moga makin cantik aja, n makasih semangatnya buat adek ya, adek da jadi ST beneran ni mbak,he2x..), mas ipunk,SE.(pinjaman dasinya ok2x, ntar abis wisuda adek balikin ya..), mas Mono, mas Joko, masToro, Aa' Yono dan mas Dinto. makasih atas perhatian yang tlah engkau curahkan ke adek n thank's banget juga untuk kucuran2x dana segarnya,he..he..,

Tanpa kalian semua aku sadar betapa tidak ada artinya aku hidup di dunia ini, hanya dalam dekapan kasih sayang Beliaulah detik demi detik terlewati begitu berharga dan memberikan warna-warna kehidupan yang tak pernah dan tak akan pernah terbayangkan sebelumnya.

Allah akan meninggikan orang-orang yang beriman dan
berilmu pengetahuan itu beberapa derajat,
dan Allah Maha Mengetahui apa-apa yang kamu kerjakan.
(QS. Al-Mujaadalah : 11)

MATOR SAKALANGKONG

My self, dengan ihtiar tanpa kenal lelah dan tawakkal kepada-Nya sehingga bisa menyelesaikan skripsi ini tepat pada waktunya, dan berhak menyandang gelar ST, juga Motor Astrea Grand M 3706 TH yang selalu setia mengantarkan tukang insinyur ini kemana-mana tanpa lelah..,

Bpk Joseph Dedy Irawan ST, MT, dengan kesabaran dan kebaikannya dalam membimbing saya sampai saya menjadi seorang Sarjana Teknik. Thank's banget...

Andik(seksi hardware), mas Heri(seksi software), Ismianto alias Hupp al abbas(seksi compt), Agung Jakarta(seksi-pertelevisian).

Teman-teman satu kos: ungguh P Amd, caca, nanang (sori guss...,diyena ST kaadhek ya), Hupp, Rori bin horton, agus, dani "kita lewati kebersamaan dengan canda tawa duka dan maafkan ya pabila aku egois dan ada kesalahan yang disengaja atau ngga', kalian semua adalah sahabat sejatiku."

Pak Sodik, Bu Jannah dengan penjara *Alqatras*-nya jl Sumbersari IV/62f, tempat aku bernaung selama menimba ilmu.

A.Kadir & Beni ;kapan ni maen PS lagi.??,kalian adalah teman sejatiku.
Teman-teman Elka4'01 :rosi, nana, didik, suprio, helmi, medi', yudha dll.
teman seperjuangan dari seminar proposal ampe'kompre : fadil, sigit, sulis, dan sa'jurusan elektronika S-1 angktn'2001.

Pak Walikota Malang beserta aparat desa atas izin tinggal sementaranya selama aku kuliah di malang, dan para cewek-cewek malang yang telah banyak memberi konspirasi khayalan tingkat tinggi pada aku,he..hee.

Dan semua pihak yang tidak tercantum/disebutkan yang telah banyak membantu, saya ucapkan terimakasih, tanpa kalian semua aku adalah sebutir pasir diantara jutaan pasir di padang pasir. Kalian semua akan selalu dalam hatiku sampai waktu memisahkan kita.

ABSTRAKSI

PERENCANAAN DAN PEMBUATAN REMOTE KONTROL PENYETING TIMER UNTUK MENGUBAH CHANEL SECARA OTOMATIS PADA TELEVISI DENGAN MENGGUNAKAN MIKROKONTROLER AT89S8252

(Nur Suryaningrat, 01.17.155, Jurusan Teknik ElektroS-1/Elektronika)

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

Kata Kunci : Mikrokontroller AT89S8252, *Keypad*, RTC DS1307, *Optocoupler* 4N35, LCD M1632.

Dewasa ini perkembangan teknologi sangatlah pesat terutama dalam bidang elektronika, berbagai aplikasi yang diterapkan dalam teknologi elektronika salah satu contoh yang telah familiar adalah teknologi yang disebut control dan pengaturan. Dengan adanya pengontrolan dan pengaturan secara elektronik ini maka dapat meningkatkan efektifitas dari suatu alat elektronik ini yang hanya memiliki fungsi terbatas menjadi lebih komplek fungsinya. Sistem tersebut memanfaatkan chip mikrokontroler yang menjadikan sistem yang dirancang semakin fleksibel.

Alat penyeting timer untuk mengubah chanel secara otomatis ini dikontrol oleh mikrokontroler AT89S8252, pada dasarnya prinsip kerja alat ini adalah membandingkan data pewaktuan yang telah disimpan pada EEPROM mikrokontroler dengan pewaktuan pada RTC. Apabila data pada EEPROM sama dengan pewaktuan RTC, maka mikrokontroler akan memerintahkan *driver relay* untuk mengaktifkan remote. Dalam sehari data yang dapat diinputkan maksimal 6 data. Alat yang dibuat meliputi perancangan perangkat keras (*hardware)* dan perangkat lunak *(software)*, perancangan perangkat keras meliputi: rangkaian mikrokontroler AT89S8252, rangkaian RTC DS1307, rangkaian *driver relay*, rangkaian *keypad*, serta rangkaian LCD sebagai *display*, sedangkan untuk perancangan perangkat lunak adalah *Flowchart* cara kerja sistem.

Dari hasil pengujian didapatkan *error* antara perhitungan dan pengukuran pada rangkaian driver relay untuk Ic = 9% sedangkan *error* untuk adalah Ib = 2%.

KATA PENGANTAR

Atas Berkat Rahmat Allah SWT, sehingga penulis dapat menyelesaikan laporan Skripsi dengan judul:

" Perencanaan Dan Pembuatan Remote Kontrol Penyeting Timer Untuk Mengubah Chanel Secara Otomatis Pada Televisi Dengan Menggunakan Mikrokontroler AT89S8252"

Pembuatan Skripsi ini disusun guna memenuhi syarat akhir kelulusan pendidikan jenjang Strata-1 di Institut Teknologi Nasional Malang. Laporan Skripsi ini merupakan tanggung jawab tertulis atas ilmu pengetahuan yang didapat selama penyusun mengikuti kuliah.

Atas terselesaikannya Skripsi ini, penulis mengucapkan terima kasih kepada :

1. Kedua Orang Tua dan Saudaraku yang telah banyak memberi dukungan baik material maupun spiritual
2. Bapak Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
3. Bapak Ir. Mochtar Asroni, MSME selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
4. Bapak Ir. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S1 / Elektronika.
5. Bapak Joseph Dedy Irawan ST, MT selaku Dosen Pembimbing yang telah memberikan bimbingan, pengarahan, serta ilmu-ilmu yang sangat berharga sehingga Skripsi ini dapat terselesaikan.

6 Teman – teman ku dan semua pihak yang telah banyak membantu dalam proses penyelesaian Skripsi ini.

Penulis menyadari bahwa laporan ini masih banyak yang perlu disempurnakan. Oleh sebab itu kritik dan saran yang membangun sangat diharapkan.

Akhir kata, penulis mohon maaf kepada semua pihak bilamana selama penyusunan Skripsi ini penyusun membuat kesalahan secara tidak sengaja dan semoga Skripsi ini dapat bermanfaat bagi kita semua.

Malang, Maret 2006

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i

LEMBAR PERSETUJUAN ii

ABSTRAK iii

KATA PENGANTAR iv

DAFTAR ISI vi

DAFTAR GAMBAR ix

DAFTAR TABEL xi

BAB I. PENDAHULUAN 1

1.1. Latar Belakang 1

1.2. Tujuan 2

1.3. Rumusan Masalah 2

1.4. Batasan Masalah 2

1.5. Metodologi Pembahasan 3

1.6. Sistematika Penulisan 3

BAB II. LANDASAN TEORI 5

2.1. Mikrokontroller AT89S8252 5

2.1.1. Fitur Mikrokontroler Atmel AT89S8252 5

2.1.2. SFR Tambahan pada Atmel AT89S8252 10

2.1.3.Data Memori (EEPROM) dan RAM 11

2.1.4. Programable Watchdog Timer (WDT) 12

2.1.5. Timer 2 13

2.2. Serial RTC DS1307 14

2.3. *Keypad* 20

2.4. *Relay* 20

2.5. *Optocoupler* 23

2.6. LCD (*Liquid Crystal Display*) 24

2.7. *Remote Kontrol* 25

BAB III. PERENCANAAN DAN PEMBUATAN ALAT 28

3.1. Perencanaan Perangkat Keras (Hardware) 28

3.1.1. Diagram Blok 28

3.1.2. Prinsip Kerja Alat 29

3.1.3 Perancangan dan Pembuatan Rangkaian Mikrokontroler AT89S8252 30

3.1.3.1. Perancangan Pengunaan Port-Port Pada Mikrokontroler AT89S8252 30

3.1.4. LCD (Liquid Crystal Display) 33

3.1.5. *Keypad* 34

3.1.6. Rangkaian *Driver* Relay 35

3.1.7. Rangkaian RTC DS1307 37

3.2. Perancangan Perangkat Lunak (Software) 39

3.2.1. *Flowchart* 40

BAB IV. PENGUJIAN ALAT 42

4.1. Umum 42

4.2. Pengujian Perangkat Keras 42
4.2.1. Pengujian Sistem Mikrokontroler AT89S8252 42
4.2.2. Pengujian RTC DS1307 45
4.2.3. Pengujian Keypad 48
4.2.4. Pengujian LCD 49
4.2.5. Pengujian Driver Relay 51
4.3. Pengujian Perangkat Lunak Keseluruhan 52

BAB V. PENUTUP 56
5.1. Kesimpulan 56
5.2. Saran 56

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar **Halaman**

2.1 Mikrokontroller AT89S8252 6

2.2 Pin-Out Serial RTC DS1307 15

2.3 Memori Map pada DS1307 16

2.4 Map Register Pada DS1307 17

2.5 Proses Penulisan DS1307 18

2.6 Pembacaan Pada DS1307 19

2.7 Rangkaian Keypad 20

2.8 Cara Kerja Relay 21

2.9 Relay SPST 21

2.10 Relay SPDT 22

2.11 Relay DPST 22

2.12 Relay DPDT 22

2.13 Konfigurasi Pin 4N35 23

2.14 Bentuk fisik dari LCD 25

2.15 Pengiriman kode Dengan Tipe *Pulse-Coded Signal* 26

2.16 Pengiriman kode Dengan Tipe *Space-Coded Signal* 26

2.17 Pengiriman kode Dengan tipe *Shift-Coded Signal* 27

3.1 Blok Diagram Keseluruhan Sistem 28

3.2 Pemakaian Port-Port Mikrokontroler AT89S8252 31

3.3 Rangkaian LCD M1632 34

3.4 Rangkaian *Keypad* 35

3.5 Rangkaian Driver Relay 37
3.6 Rangkaian Serial RTC DS1307 38
3.7 Flowchart Keseluruhan Sistem 41
4.1 Diagram Blok Pengujian Mikrokontroler AT89S8252 43
4.2 Diagram Blok Pengujian RTC 46
4.3 Tampilan Hasil RTC 47
4.4 Diagram Blok Pengujian Keypad 48
4.5 Diagram Blok Pengujian LCD 50
4.6 Hasil Pengujian LCD 51
4.7 Rangkain Pengujian Driver Relay 51
4.8 Tampilan Awal LCD 53
4.9 Tampilan Menu Utama 53
4.10 Tampilan LCD Saat Pemrograman 54
4.11 Proses Membandingkan Data Dan Tampilan Standby 54
4.12 Gambar Alat Keseluruhan 55

DAFTAR TABEL

Tabel **Halaman**

2.1 Konfigurasi Port 3 Atmel AT89S8252 8

2.2 Fungsi Khusus Pada Port 1 Atmel AT89S8252 10

2.3 Pemilihan Perioda Data WDT 12

2.4 *Mode Operasi Timer 2* 14

2.5 Tabel Fungsi Bit Control RS1 dan RS0 18

4.1 Hasil Pengujian Sistem Mikrokontroler AT89S8252 44

4.2 Hasil Pengujian Rangkaian Keypad 49

4.3 Hasil Pengujian Rangkaian Relay 51

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini perkembangan teknologi sangatlah pesat terutama dalam bidang elektronika, berbagai aplikasi yang diterapkan dalam teknologi elektronika salah satu contoh yang telah familiar adalah teknologi yang disebut kontrol dan pengaturan yaitu sebuah pengontrolan dan pengaturan yang dilakukan pada sebuah proses kerja sebelumnya.

Dengan adanya pengontrolan dan pengaturan secara elektronik ini maka dapat meningkatkan efektifitas dari suatu alat elektronik ini yang hanya memiliki fungsi terbatas menjadi lebih komplek fungsinya. Sistem tersebut memanfaatkan chip mikrokontroler yang sekarang makin banyak digunakan. Penggunaan sistem mikrokontroler ini menjadikan sistem yang dirancang semakin fleksibel.

Sesuai dengan beberapa faktor yang telah diuraikan diatas maka kami membuat suatu perangkat yang menerapkan suatu teknologi pengontrolan dan pengaturan dengan memanfaatkan mikrokontroler. Alat yang kami buat ini dinamakan **'Perencanaan dan Pembuatan Remote Kontrol Penyeting Timer untuk Mengubah Chanel Secara Otomatis pada Televisi dengan Menggunakan Mikrokontroler AT89S8252'**, dengan alat ini diharapkan dapat menambah fungsi dari remote kontrol biasa pada televisi menjadi lebih komplek yaitu dapat mengubah chanel secara otomatis sesuai dengan kita inputkan atau program sebelumnya.

1.2. Tujuan

Tujuan dari penulisan skripsi ini adalah Perencanaan dan pembuatan *remote kontrol untuk menyeting timer secara otomatis pada televisi dengan* menggunakan mikrokontroler AT89S8252.

1.3. Rumusan Masalah

Mengacu pada permasalahan yang ada, maka dalam perencanaan dan pembuatan alat ini diutamakan pada hal-hal sebagai berikut :

1. Bagaimana merencanakan dan membuat program untuk mengontrol alat.
2. Bagaimana merencanakan dan membuat perangkat keras (hardware) serta rangkaian pendukung untuk mengoptimalkan kerja sistem.
3. Bagaimana membuat rangkaian keypad, LCD dan RTC.

1.4. Batasan Masalah

Dalam laporan akhir "Perencanaan dan Pembuatan Remote Kontrol Penyeting Timer untuk Mengubah Chanel Secara Otomatis pada Televisi dengan Menggunakan Mikrokontroler AT89S8252", penulis akan memberikan batasan-batasan masalah agar tidak terjadi penyimpangan maksud dan tujuan utama penyusunan laporan akhir ini.

1. Membahas pembuatan sistem kontrol menggunakan AT89S8252 beserta komponen pendukungnya.
2. Dalam rangkaian ini catu daya dianggap konstan dan tidak akan dibahas.

3. Perancangan menggunakan pada satu merk televisi saja, yaitu: *Samsung*.

1.5. Metodelogi Pembahasan

Metodelogi penelitian yang dipakai dalam pembuatan Skripsi ini:

1. Studi Literatur.
2. Perancangan dan pembuatan alat.
3. Pelaksanaan uji coba alat.
4. Penyusunan Laporan Skripsi.

1.6. Sistematika Penulisan

Pada penulisan laporan skripsi ini ditulis sedemikian rupa sehingga diperoleh hubungan yang jelas antara bagaian yang satu dengan yang lainnya. Sistematika penulisan dari laporan ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Berisi latar belakang, tujuan, permasalahan, batasan masalah, metodelogi, dan sistematika penulisan.

BAB II. TEORI

Membahas teori – teori dasar penunjang, perancangan dan pembuatan alat.

BAB III. PERANCANGAN DAN PEMBUATAN ALAT

Membahas tentang perancangan alat baik perangkat keras maupun perangkat lunak, serta cara kerja blok diagram.

BAB IV. PENGUJIAN ALAT

Mencakup pembahasan tentang proses pengujian alat yang terdiri dari peralatan yang digunakan, langkah kerja dan analisa hasil pengujian.

BAB V. PENUTUP

Berisi kesimpulan dan saran.

BAB II

TINJAUAN PUSTAKA

2.1. Mikrokontroler AT89S8252

Mikrokontroler *Atmel* AT89S8252 merupakan pengembangan dari mikrokontroler standart MCS-51. Hal-hal yang terdapat pada penjelasan mikrokontroler MCS-51 juga berlaku untuk mikrokontroler *Atmel* AT89S8252.

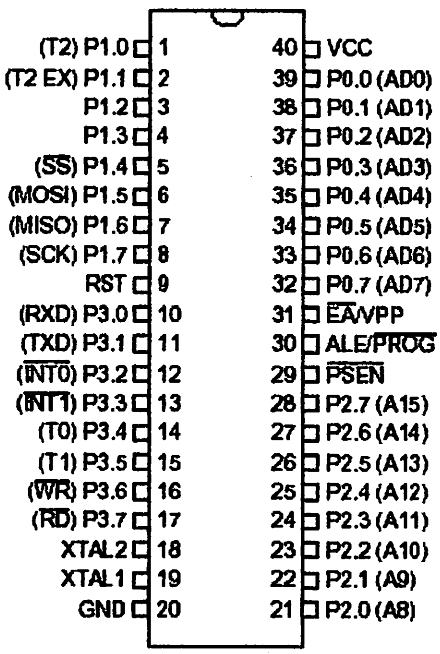
2.1.1. Fitur Mikrokontroler Atmel AT89S8252

Mikrokontroler AT 89S8252 merupakan pengembangan dari mikrokontroler standard MCS – 51, dengan banyak kelebihan yang ditawarkan antara lain :

- Kompatibel dengan Mikrokontroler MCS – 51
- 8K byte Downloadable Flash memori
- 2K byte EEPROM
- 3 level program memori lock
- 256 byte RAM internal
- 32 I/O yang dapat dipakai semua
- 3 buah timer / counter 16 bit
- Programmable watchdog timer
- Dual data pointer
- Frekuensi kerja 0 sampai 24 MHZ
- Tegangan operasi 2,7 – 6 volt

Dipakainya downloadable flash memori memungkinkan mikrokontroler ini bekerja sendiri tanpa diperlukan tambahan chip lainnya. Dan flash memori dapat diprogram hingga seribu kali. Hal lain yang menguntungkan adalah sistem pemrograman jauh lebih sederhana dan tidak memerlukan rangkaian yang rumit seperti rangkaian untuk memprogram AT89C51.

Timer / counter juga bertambah satu dari standar 2 buah pada MCS – 51. Selain itu frekuensi kerja yang lebar dan rancangan statik sangat membantu untuk proses *debugging*. Dengan adanya beberapa fitur tambahan itu, maka akan mengakibatka bertambahnya SFR (*Special Function Register*). Gambar berikut adalah gambar mikrokontroler AT 89S8252. Tata letak pin – pin ini masih mengacu pada mikrokontroler MCS – 51 sehingga AT 89S8252 dapat menggantikan mikrokontoler MCS – 51.



Gambar 2.1. Mikrokontroler AT89S8252

Sumber: Data Sheet ATMEL AT89S8252

Keterangan pin :

a. Pin 40 (VCC)

   Merupakan pin catu daya dengan tegangan sebesar +5 V (DC)

b. Pin 20 (GND)

   Merupakan pin GROUND yang nanti terhubung dengan grounding rangkaian.

c. Pin 32 – 39 (Port 0)

   PORT 0 mempunyai fungsi sebagai port alamat dan data , maka jika mikrokontroler sedang mengakses alamat, P0 akan aktif sebagai pembawa alamat 8 bit yang bawah (A0 – A8). Ketika mengakses data (bisa input atau output) port ini sebagai jalur data (D7 – D0).

d. Pin 21 – 28 (Port 2)

   Port 2 berfungsi sebagai pembawa alamat 8 bit atas (A8 – A15). Berbeda dengan port 0, port ini tidak bersifat sebagai jalur data hanya sebagai pembawa alamat. Dengan demikian jelas bahwa untuk alamat AT89S8252 menyediakan 16 bit sedangkan untuk jalur data hanya 8 bit.

e. Pin 10 – 17 (port 3)

   Port 3 ini mempunyai fungsi yang berlainan dari setiap pin-pinnya, seperti yang di tunjukkan di bawah ini:

   - P 3.7: Kaki *read* yang aktif manakala sedang melakukan eksekusi yang sifatnya membaca data.

- P 3.6 : Kaki *write* yang aktif saat melakukan eksekusi yang sifatnya menulis data ke suatu alamat.
- P 3.5 : Merupakan pin yang berhubungan dengan timer register timer 1 (T1).
- P 3.4 : Merupakan pin yang berhubungan dengan timer register 0 (T0).
- P 3.3 : Berhubungan dengan control interupt (INT1).
- P 3.2 : Berhubungan dengan control interupt (INT0).
- P 3.1 : Berhubungan dengan port serial (TXD).
- P 3.0 : Berhubungan dengan port serial (RXD).

Untuk lebih jelasnya lihat daftar tabel dibawah ini :

Tabel 2.1. Konfigurasi Port 3 Atmel 89S8252

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

Sumber: Data Sheet ATMEL AT89S8252

f. Pin 9 (RST (*Reset*))

Pin reset ini aktif tinggi (1), jika pin ini aktif tinggi minimal dua kali siklus mesin bekerja maka akan mereset peralatan.

g. Pin 30 (ALE/PROG (*Address Latch Enable/Program*))

Pin ALE ini aktif tinggi dengan mengeluarkan pulsa output untuk melatch (mengunci/menahan) 1 byte alamat rendah selama mengakses ke alamat memori eksternal. ALE dapat mengendalikan 8 beban TTL dan juga merupakan input program yang aktif rendah selama pemrograman *Flash Eprom.* Pada operasi normal, ALE dikeluarkan pada suatu kecepatan yang konstan yaitu 1/6 dari frekuensi osilator, dan juga dapat dipergunakan untuk pewaktu atau timing eksternal atau untuk pemberian clock.

h. Pin 29 (PSEN (*Program Strobe Enable*))

Pin ini aktif rendah, yang merupakan pulsa pengaktif untuk pembacaan ke program memori eksternal.

i. Pin 19 (XTAL - !)

Sebagai pin input ke penguat osilator pembalik dan input rangkaian clock internal untuk operasi system.

j. Pin 18 ((XTAL – 2)

Pin output dari penguat osilator.

k. Pin 31 (EA/VPP (*Externall Access/Program Supplay Voltage*))

Pin ini harus di tahan dalam kondisi rendah secara eksternal atau dihubungkan ke-*ground* agar AT 89S8252 dapat mengakses kode mesin dari memori eksternal. Jika menggunakan internal program memori maka pin ini harus diberi logika tinggi (1).

Semua pin pada mikrokontroler AT 89S8252 adalah sama dengan mikrokontroler MCS – 51. Namun pada port 1 mikrokontroler AT 89S8252 terdapat beberapa fungsi khusus yang tidak terdapat pada mikrokontoler MCS – 51. Fungsi khusus tersebut dijelaskan pada tabel berikut :

Tabel 2.2. Fungsi khusus pada Port 1 Atmel 89S8252

Port PIN	**Fungsi Khusus**
P 1.0	T2 (masukan luar untuk timer / counter 2)
P 1.1	T2 EX (timer / counter capture / reload trigger dan control arah)
P 1.2	-
P 1.3	-
P 1.4	SS (*slave port select input*)
P 1.5	MOSI (master data ouput, slave dan input untuk kanal SPI)
P 1.6	MISO (master data input, slave data output untuk kanal SPI)
P 1.7	SCK (master clock output, slave clock input untuk kanal SPI)

Sumber: Data Sheet ATMEL AT89S8252

2.1.2. SFR tambahan pada Atmel AT89S8252

Selain memiliki SFR(*Special Function Register*) seperti halnya pada MCS-51, mikrokontroler Atmel AT89S8252 memiliki tambahan SFR. Hal ini tak lain adalah karena adanya fitur tambahan pada mikrokontroler Atmel AT89S8252.

SFR tambahan ini meliputi: T2CON(Timer 2 Register dangan alamat 0C8H), T2MOD(*Timer 2 Mode* dengan alamaat 0C9H), WMCON(*Watchdog and Memory Control Register* dengan alamat 96H), SPCR(*SPI Control Register* dengan alamat D5H), SPSR(*SPI Status Refister* dengan alamat AAH), SPDR(SPI Data Register dengan alamat 86H).

2.1.3. Data Memory (EEPROM) dan RAM

Berbeda dengan mikrokontroler standart MCS-51, mikrokontroler Atmel AT89S8252 juga dilengkapi dengan data memori yang berupa EEPROM (*Electrically Erasable Programmable Read Only Memory*). EEPROM yang ditanamkan ini besarnya 2 kilo byte (2K) dan dipakai untuk penyimpanan data.

EEPROM ini diakses dengan mengeset bit EEMEN pada registerWMCON pada alamat 96H. Alamat EEPROM ini adalah 000H sampai 7FFH. Instruksi movx digunakan untuk mengakses EEPROM internal ini. Namun jika ingin mengakses data memori luar (diluar mikrokontroler Atmel AT89S8252) dengan menggunakan instruksi movx ini maka bit EEMEN harus dibuat"0".

Bit EEMWE pada register WMCON harus diset ke 1 sebelum sembarang lokasi pada EEPROM dapat ditulisi. Program pengguna harus mereset bit EEMWE ke "0" jika proses penulisan ke EEPROM tidak diperlukan lagi. Proses penulisan pada EEPROM dapat dilihat dengan membaca bt RDY/BSY pada SFR WMCON. Jika bit ini berlogika rendah maka penulisan EEPROM sedang berlangsung, jika bit ini berlogika tinggi maka penulisan sudah selesai dan penulisan lain dapat dimulai lagi. Sedangkan RAM yang ada pada mikrokontroler

Atmel AT89S8252 adalah berkapasitas 256 byte. Penjelasan mengenai RAM ini adalah sama dengan RAM yang ada pada mikrokontroler standart MCS-51.

2.1.4. Programable Watchdog Timer (WDT)

Pada mikrokontroler Atmel AT89S8252 juga dilkengkapi oleh *watchdog Timer*. *Watchdog Timer* ini menggunakan detak tersendiri. Untuk mengatur rentang waktu(perioda) pada WDT ini maka terdapat bit prescaler yang dapat mengatur rentang waktru yang dibutuhkan.

Bit prascaler ini adalah bit PS0, PS1dan PS2 pada register WMCON. Perioda waktu pada WDT ini berkisar dari 16 mili detik sampai 2048 mili detik. Karena bit prescalernya ada 3, maka akan ada 8 buah kemungkinan yaitu:

Tabel 2.3. Pemilihan Perioda Waktu WDT

PS2	PS1	PS0	Perioda
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Sumber: Data Sheet ATMEL AT89S8252

WDT dilumpuhkan oleh Power on Reset (POR) dan selama Power Down. WDT diaktifkan dengan menseting bit WDTEN pada SFR WMCON (alamat

96H). Jika perhitungan waktu WDT telah selesai tanpa ada reset atau dilumpuhkan, maka suatu pulsa reset internal akan dihasilkan untuk mereset CPU.

2.1.5. Timer 2

Pada mikrokontroler Atmel AT89S8252 terdapat tambahan *Timer* 2. *Timer* yang lain adalah *timer* 0 dan *timer* 1. Hal yang perlu diperhatikan adalah *Timer/Counter* dapat digunakan sebagai generator *baudrate* untuk serial *port.* Pada standart MS-51 biasanya yang digunakan adalah *timer* 1 sebagai penghasil *baudrate*. Pada mikrokontroler Atmel AT89S8252 selain menggunakan *timer* 1 sebagai *baudrate* (untuk menjaga kompabilitas dengan MCS-51) juga dapat menggunakan *Timer* 2 sebagai penghasil *baudrate* untuk serial *port. Timer* 2 ini merupakan Timer/Counter yang berukuran 16 bit yang dapat beroperasi sebagai timer atau dapat beroperasi sebagai penghitung kejadian dengan detak dari luar. Untuk mengatur fungsi ini dilakukan dengan mengatur bit C/T2 pada SFR T2CON. Terlihat bahwa jika bit ini tinggi maka akan terpilih fungsai *counter*, tapi jika bit ini rendah maka akan terpilih fungsi *Timer* 2.

Timer 2 ini memiliki 3 mode operasi yaitu: *capture, auto reload (up dan down counting)* dan *baud rate* generator. Untuk memilih mode ini dilakukan dengan mengatur bit pada SFR T2CON.

Timer 2 ini terdiri dari 2 buah *timer* 8 bit *register* yaitu TH2 dan TL2 dinaikkan tiap siklus mesin. Karena siklus mesin terdiri dari 1 *periode* osilasi, maka *count rate* menjadi 1/12 dari frekuensi osilator.

Pada fungsi *counter, register* dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada pena yang bersesuaian (dalam hal ini pin T2 atau

Pi.0). pada fungsi ini, masukan luar akan disampling selama S5P2 dari tiap siklus mesin. Jika hasil sampling menunjukkan logika tinggi pada selama satu siklus dan logika rendah pada siklus selanjutnya maka akan terdeteksi transisi tinggi ke rendah dan akibatnya perhitungan akan dinaikkan. Nilai perhitungan yang baru akan muncul pada register selama S3P1 dari siklus setelah transisi tinggi ke rendah terdeteksi.

Tabel 2.4. Mode Operasi Timer 2

RCLK+TCLK	CP/RL2	TR2	MODE
0	0	1	16 bit auto reload
0	1	1	16 bit capture
1	X	1	Baud rate generator
X	X	0	Off

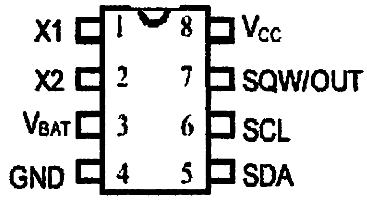
Sumber: Data Sheet ATMEL AT89S8252

2.2. Serial RTC DS1307

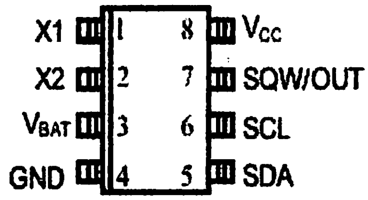
Pada system ini RTC difungsikan sebagai referansi waktu (detik, menit, jam, tanggal, bulan, hari, tahun dan hari). RTC (Real Time Clock) DS 1307 adalah serial RTC produksi MAXIM. Serial RTC DS1307 memiliki beberapa kelebihan jika dibandingkan dengan jenis RTC lainnya, yaitu :

- Akses datanya secara serial, sehingga menghemat pin-out untuk perangkat pengaksesnya.
- Dilengkapi pin untuk baterai back-up, sehingga tidak repot menambah rangkaian untuk baterai back-up.
- Y2K compatible, sehingga tidak perlu lagi ada pembenahan penanggalan.

- Konsumsi daya yang relative kecil, dengan hanya memakai 3V baterai CR2032 bisa untuk masa pakai 10 tahun



DS1307 8-Pin DIP (300-mil)



DS1307 8-Pin SOIC (150-mil)

Gambar 2.2. Pin-Out Serial RTC DS1307

Sumber: Data Sheet RTC DS1307

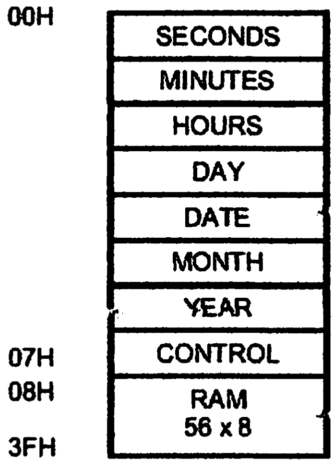
Dari gambar 2.2, terlihat bahwa untuk lalu-lintas data hanya menggunakan 2 pin-out, yaitu SCL (Serial Clock, pin 6) dan SDA (Serial Data, pin 5).

Fungsi masing-masing pin-out pada DS1307 adalah sebagai berikut :

- VCC-GND, kedua pin-out ini untuk supply tegangan.
- Vbat, pin untuk baterai back-up. Baterai back-up digunakan baterai dengan tegangan 3V.
- SCL (Serial Clock Input), merupakan pin input untuk clock yang dikirim oleh peripheral master (peripheral yang mengakses DS1307).
- SDA (Serial Data), merupakan pin bidirectional, dapat berfungsi sebagai input atau output, tergantung proses yang sedang berlangsung. Saat proses read (baca) pin SDA berfungsi sebagai pin output,

sebaliknya saat proses write (tulis/isi data) pin SDA berfungsi sebagai pin output.

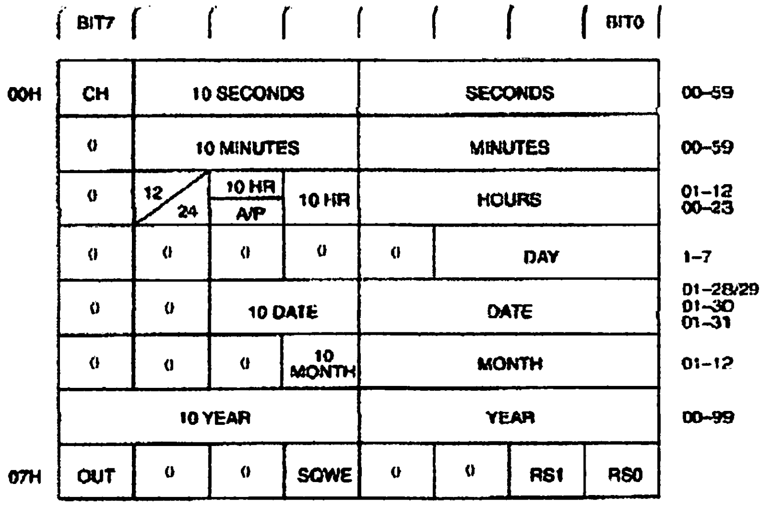
- X1,X2, merupakan pin-out untuk kristal. Kristal yang dipakai adalah standart quartz kristal yang bernilai 32,768KHz.
- SQW (Square Wave Output), adalah pin out yang mengeluarkan sinyal kotak dengan nilai yang dapat diset. Tetapi fungsi ini bisa dimatikan kalau memang tidak dibutuhkan.



Gambar 2.3. Memory Map pada DS1307

Sumber: Data sheet RTC DS1307

Peta pengalamatan pada RTC DS1307 dapat dilihat pada gambar 2.3, dari gambar tersebut terlihat bahwa untuk lokasi memory 08H sampai dengan lokasi 3FH berupa RAM yang dapat digunakan untuk menyimpan data untuk fungsi-fungsi tertentu.



Gambar 2.4. Map Register Pada DS1307

Sumber: Data Sheet RTC DS1307

Dari gambar 2.4, dapat dilihat bahwa untuk alamat 00H pada bit ke-7 terdapat bit CH (clock halt) yang bila berlogic 0, maka clock akan berjalan (run) tetapi bila CH berlogic 1, clock akan berhenti (halt). Kondisi ini perlu diperhatikan saat inisialisasi RTC DS1307.

Untuk control register yang terletak pada alamat 07H, ada 4 bit yang berfungsi sebagai bit control, yaitu :

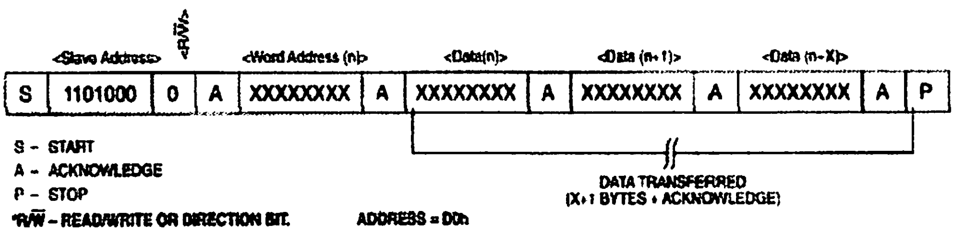
- Bit OUT (bit ke-7), adalah output control untuk level dari SQW
- Bit SQWE (Square Wave Enable, bit ke-4), merupakan bit control untuk enable/disable SQW. Bila SQWE berlogic 1 maka SQW enable, kebalikannya bila SWQE berlogic 0 maka SQW disable.
- Bit RS1 dan RS0, merupakan bit control untuk memilih frekuensi dari sinyal kotak yang dikeluarkan pada pin SQW bila SQW di-enable-kan. Tabel untuk fungsi dari RS1 dan RS0 dapat dilihat pada table 2.5

Tabel 2.5. Tabel Fungsi Bit Control RS1 dan RS0

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

Sumber: Data Sheet RTC DS 1307

Pada DS1307 semua data diakses secara serial, begitu pula untuk pengisian data ke DS1307, dilakukan secara serial.



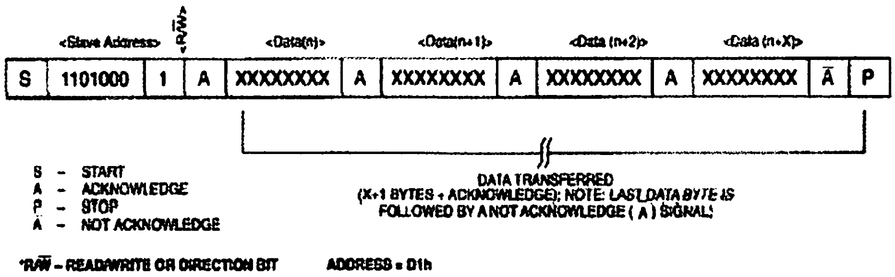
Gambar 2.5. Proses Penulisan DS1307

Sumber: Data Sheet RTC DS1307

Pengisian data pada DS1307, langkah-langkahnya dapat dilihat pada gambar. Dari gambar ini terlihat bahwa langkah pengisian adalah sebagai berikut :

- Diawali dengan bit start
- Diikuti slave address selebar 7 bit, dan nilai dari dari slave address adalah 1101000b. Dan bit ke-8 dari slave address merupakan bit penentu proses write atau read, bila proses write nilai bit ini harus 0. Dan diteruskan oleh bit acknowledgement.
- Diikuti oleh word address, yang merupakan lokasi register/lokasi memory yang akan ditulis/diisi datanya. Dan diteruskan dengan bit acknowledgement.

- Baru diikuti oleh byte data yang akan diisikan dan diteruskan oleh bit acknowledgment .
- Bila masih ada data yang akan diisikan pada lokasi memory selanjutnya, bisa langsung diisikan datanya dan diteruskan dengan bit acknowledgement.
- Tetapi apabila tidak akan mengisikan data pada lokasi memory selanjutnya setelah bit acknowledgement langsung diberi bit stop, sebagai tanda proses berakhir.



Gambar 2.6. Pembacaan pada DS1307

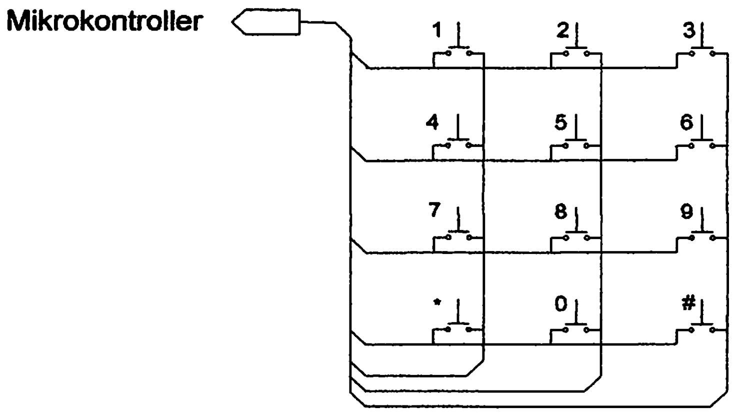
Sumber: Data Sheet RTC DS1307

Untuk proses pembacaan data dari DS1307 prosesnya dapat dilihat pada gambar 2.6. Dari gambar ini terlihat bahwa proses pembacaan adalah mirip dengan proses penulisan data ke DS1307. Yang membedakan yaitu :

- Pada bit ke-8 pada slave address, bernilai 1 untuk mode pembacaan data. Sedangkan untuk mode penulisan data bernilai 0.
- Pada proses pembacaan, apabila sudah pada akhir pembacaan data, bit acknowledgement tidak perlu diberikan.

2.3. Keypad

Keypad digunakan sebagai sarana untuk memasukkan data ke komputer atau mikrokontroler. Untuk rangkaian keypad menggunakan 3x4 yaitu 12 buah saklar tekan (push button) terdiri dari angka 0 sampai 9 serta tanda # dan * yang dirangkai dalam bentuk matrik. Gambar rangkaian keypad ditunjukkan pada gambar berikut

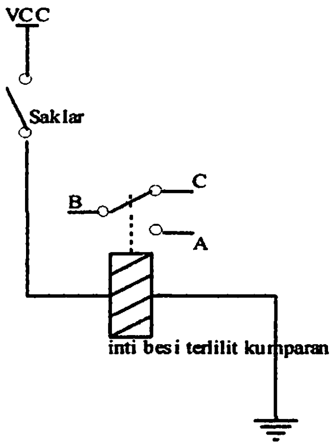


Gambar 2.7. Rangkaian Keypad.

Sumber: Data Sheet LCD M1632.

2.4. Relay

Relay adalah komponen elektronika yang terdiri dari sebuah lilitan kawat (kumparan/koil) yang terlilit pada sebuah besi lunak. Jika kumparan dialiri arus listrik maka inti besi akan menjadi magnet dan menarik pegas sehingga kotak AB terhubung dan BC terputus.



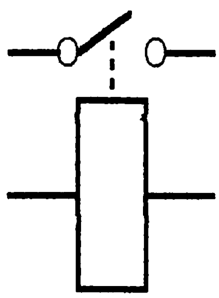
Gambar 2.8. Cara Kerja Relay

Sumber: Fourth Edition Electronic Devices And Circuit Theory.

Relay merupakan suatu alat untuk menghubungkan atau memerlukan kontak antara komponen yang satu dengan yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang ditimbulkan dari adanya medan magnet listrik yang dihasilkan oleh kumparan yang melilit pada besi lunak.

Ada beberapa macam relay, antara lain:

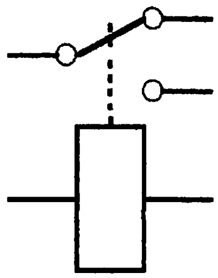
- SPST *(Single Pin Single Terminal)*



Gambar 2.9. Relay SPST

Sumber: Fourth Edition Electronic Devices And Circuit Theory.

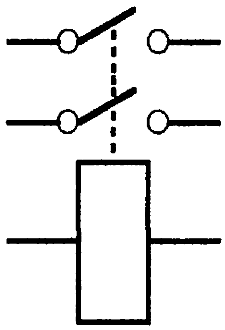
• SPDT *(Single Pin Dual Terminal)*



Gambar 2.10. Relay SPDT

Sumber: Fourth Edition Electronic Devices And Circuit Theory.

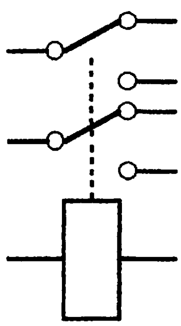
• DPST *(Dual Pin Single Terminal)*



Gambar 2.11. Relay DPST

Sumber: Fourth Edition Electronic Devices And Circuit Theory.

• DPDT *(Dual Pin Dual Terminal)*



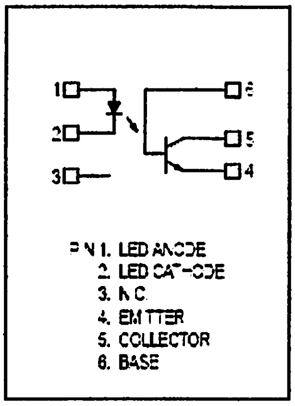
Gambar 2.12. Relay DPDT

Sumber: Fourth Edition Electronic Devices And Circuit Theory.

2.5. Optocoupler (4N35)

Optocoupler disebut juga optoisolator merupakan gabungan dari LED (pada sisi input) dan Photodioda (pada sisi output) dalam satu kemasan. Fungsi dari optocupler adalah mengisolasi antara satu bagian rangkaian dengan bagian rangkaian yang lain. Tujuan dari pengisolasian ini adalah untuk mencegah agar tidak terjadi kerusakan komponen pada suatu bagian sebagai akibat dari munculnya tegangan tinggi yang tidak diinginkan pada bagian lainnya.

Keuntungan pokok dari optocoupler adalah terjadinya isolasi elektrik antara satu rangkaian input dan output. Dengan optocoupler, hanya terdapat kontak input dan output dalam bentuk pancaran sinar. Oleh karena itu, dimungkinkan untuk mengisolasi resistansi antara dua rangkaian dalam orde ribuan megaohm. Isolasi yang seperti itu berguna dalam aplikasi tegangan tinggi dimana beda potensial dua rangkaian sampai dengan ribuan volt. [Malvino, 2003:171]. Dalam tugas akhir ini digunakan optocoupler 4N35 yang dapat mengisolasi tegangan sampai dengan 7500 Vac (pk) volt pada sisi photodioda.



Gambar 2.13.Konfigurasi Pin 4N35

Sumber: Data Sheet 4N35REV2

2.6. LCD (Liquid Crystal Display)

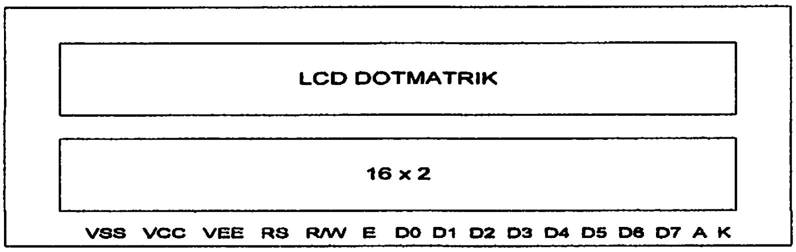
LCD (Liquid Crystal Display) adalah komponen display yang tidak memancar (nonemissive), sehingga tidak menghasilkan sumber cahaya seperti CRT (Cathode Ray Tube), dan berdaya sangat rendah (lebih rendah dari LED) yaitu dalam hitungan mikrowatt (LED dalam hitungan miliwatt). LCD menahan atau membiarkan cahaya yang dipantulkan dari sunber cahaya luar dan cahaya yang berasal dari belakang atau samping yang melewatinya. LCD dikontrol oleh ROM/RAM generator karakter dan RAM data display. Semua fungsi display dikontrol dengan instruksi dan LCD dapat dengan mudah diinterfacekan dengan MPU (Mikroprosessor Unit).

Karakteristik dari LCD dot-matriks adalah sebagai berikut:

- 16X2 karakter dengan 5X7 dot matriks+kursor
- ROM generator karakter dengan 8 tipe karakter (untuk program write)
- 80X8 bit RAM data display
- Dapat diinterfacekan dengan 4 atau 8 bit MPU
- RAM data dan RAM generator karakter dapat dibaca dari MPU
- +5V single power supply
- Power-on reset
- Range temperature operasi 0-60°C
- Beberapa fungsi instruksi:

  Display clear, Cursor home, Display ON/OFF, Cursor ON/OFF, Display charackter blink, Cursor Shift dan Display shift.

LCD disini dapat menampilkan karakter yang ada pada ROM generator karakter, yang sudah berisi 192 jenis karakter, dengan cara memberikan kode karakter untuk tiap-tiap karakter yang diinginkan pada pada bus data dengan menggunakan sinyal kontrol.



Gambar 2.14. Bentuk fisik dari LCD.

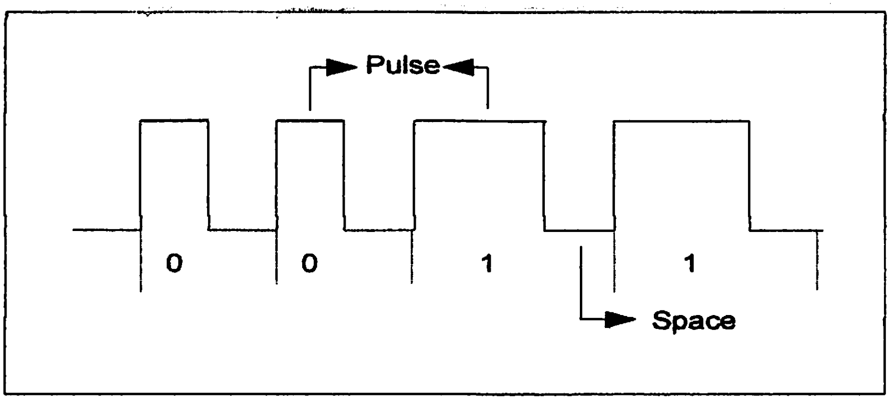
Sumber: Data sheet M1632

2.7.Remote Kontrol

Remote kontrol inframerah menggunakan cahaya inframerah sebagai media dalam mengirimkan data ke penerima. Data yang dikirimkan berupa pulsa-pulsa cahaya dengan modulasi frekuensi 40 kHz. Sinyal yang dikirimkan berupa data-data biner. Untuk membentuk data-data biner tersebut, ada tiga metode yang digunakan yaitu pengubahan lebar pulsa, lebar jeda, dan gabungan keduanya.

- *Pulse – Coded Signals*

  Dalam mengirimkan kode, lebar jeda tetap yaitu t sedangkan lebar pulsa adalah 2t. Jika lebar pulsa dan lebar jeda adalah sama yaitu t, berarti yang dikirim adalah bit 0, jika lebar pulsa adalah 2t dan lebar jeda adalah t, berarti yang dikirim adalah 1.

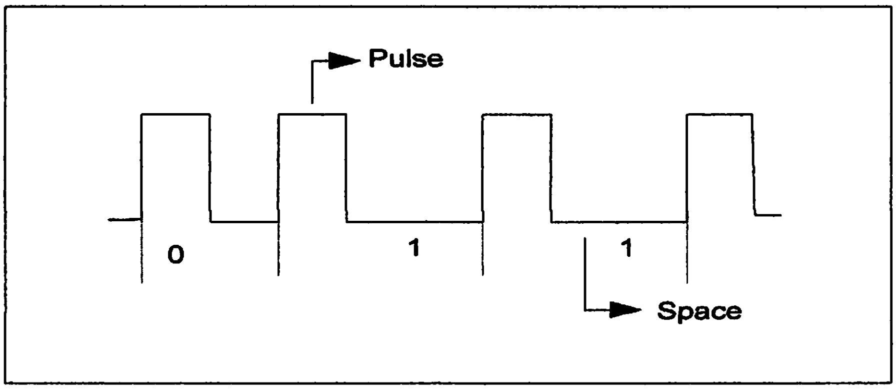


Gambar 2.15. Pengiriman Kode dengan Tipe *Pulse – Coded Signal.*

Sumber: Dave Negro, 1999:5.

- *Space – Coded Signal*

Dalam mengirimkan kode remote control dilakukan dengan cara mengubah lebar jeda, sedangkan lebar pulsa tetap. Jika lebar jeda dan lebar pulsa adalah sama yaitu t, berarti yang dikirim adalah 0. jika lebar jeda 3t, berarti data yang dikirim adalah 1.

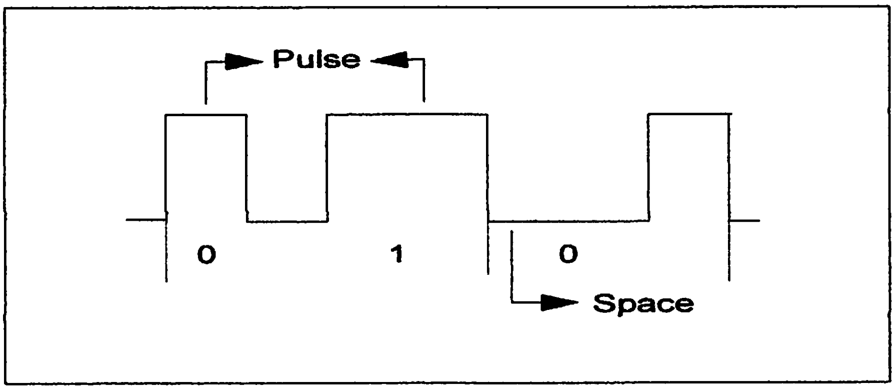


Gambar 2.16. Pengiriman kode dengan Tipe *Space – Coded Signal.*

Sumber: Dave Negro, 1999.

- *Shift – Coded Signal*

  Tipe ini merupakan gabungan dari tipe *pulse* dan *space,* yaitu dalam mengirimkan kode remote control, dengan cara mengubah lebar pulsa dan lebar jeda. Jika lebar jeda adalah t dan lebar pulsa adalah 2t, maka ini diartikan sebagai data 1. Jika lebar adalah 2t dan lebar pulsa adalah t, maka ini diartikan sebagai data 0 (low).



Gambar 2.17. Pengiriman kode dengan Tipe *Shift – Coded signal.*

Sumber: Dave Negro, 1999.

Sebelum kode dikirim, terlebih dahulu mengirimkan sinyal awal yang disebut sebagai *header*. *Header* adalah sinyal yang dikirimkan sebelum kode sebenarnya, dan juga merupakan sinyal untuk mengaktifkan penerima. *Header* selalu dikirimkan dengan lebar pulsa yang jauh lebih panjang daripada kode. Setelah *header* dikirimkan, baru kemudian kode remote kontrol. Kode remote kontrol dibagi menjadi dua fungsi, yaitu fungsi pertama digunakan sebagai penunjuk alamat peralatan yang akan diaktifkan, fungsi kedua adalah sebagai *command* atau perintah untuk melaksanakan instruksi dari remote kontrol.

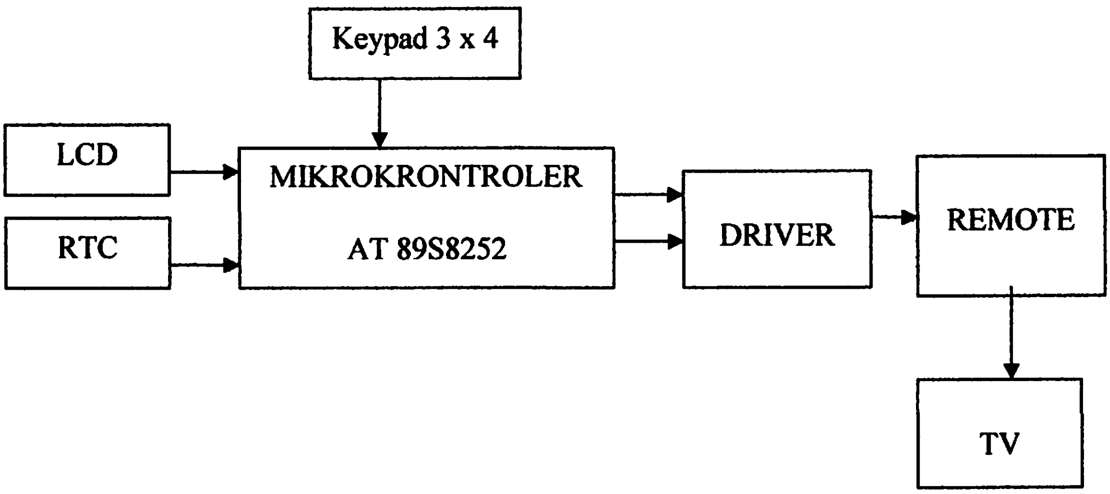
BAB III

PERENCAANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai peralatan yang direncanakan dan akan direalisasikan sebagaimana fungsinya. Adapun perencanaan dan pembuatan alat meliputi perencanaan dan pembuatan perangkat keras serta perencanaan dan pembuatan perangkat lunak secara garis besarnya.

3.1. Perencanaan Perangkat Keras (Hardware)

3.1.1. Diagram Blok



Gambar 3.1

Blok Diagram Keseluruhan Sistem

Fungsi dari tiap blok adalah sebagai berikut :

1. Mikrokontroler AT89S8252 merupakan komponen pengendali utama kerja keseluruhan sistem yang dibuat. Instruksi yang terdapat di dalam memori akan dikerjakan oleh mikrokontroler.
2. *LCD (Liquid Crystal Display)* berfungsi untuk menampilkan inputan berupa hari, tanggal, jam, menit, detik, dan saluran tv yang akan diprogram untuk meng-on atau meng-off kan. Sedangkan pada waktu stand by, berfungsi menampilkan hari, jam,menit.
3. *Keypad* berfungsi sebagai media masukan untuk mengeset data yang akan digunakan dalam pemberian informasi.
4. Driver relay berfungsi untuk mengaktifkan remote televisi, setelah mendapat instruksi dari mikrokontroler.
5. *RTC (Real Time Clock)* berfungsi untuk menghitung waktu sekarang dengan waktu yang ada di mikrokontroler pada saat sistem bekerja.
6. Remote berfungsi hanya sebagai mengirim data ke tv.

3.1.2. Prinsip Kerja Alat

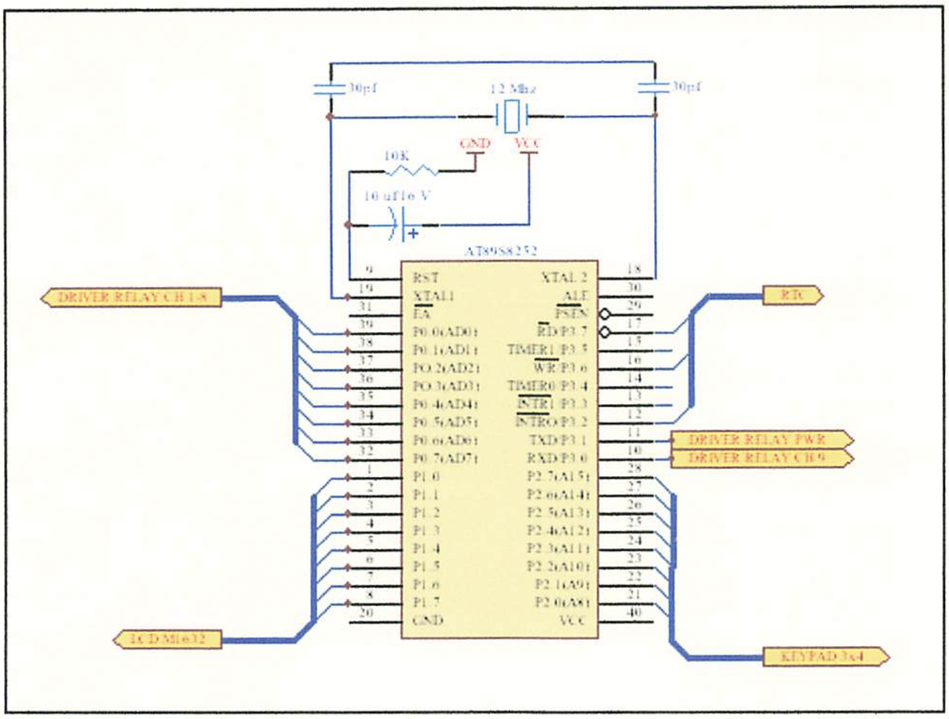
Pada prinsipnya kerja alat ini adalah membandingkan pewaktuan pada mikrokontroler dengan *RTC (Real Time Clock)* DS1307. Pengisian jadwal dapat dilakukan dengan memilih menu-menu pada listing program yang ditampilkan oleh LCD. Tombol # pada keypad digunakan untuk masuk ke menu utama. Diawali dengan pengisian hari dalam satu minggu, kemudian pengisian agenda yang dibatasi 6 jadwal untuk satu hari. Jadwal terdiri dari dua pengisian, yaitu pengisian waktu dan pengisian channel televisi. Pengisian waktu terdiri dari jam, dan menit. Setelah kita melakukan semua pengesetan, langkah berikutnya adalah menjalankan menu run, sehingga data ini akan disimpan pada mikrokontroler AT89S8252 sebagai pusat pengatur dan pengolah data. Kemudian pada proses ini mikrokontroler akan membandingkan terus pewaktuan yang telah disimpan dengan RTC, ketika waktu pada mikrokontroler sama dengan RTC, maka mikrokontroler akan memberi outputan ke driver relay untuk mengubah chanel atau meng-on kan atau meng-off kan yang kemudian dikirimkan ke televisi melalui inframerah sebagai transmiter. Untuk penghapusan semua data yang sudah diset, dapat dipilih menu hapus. Data untuk satu hari bisa dihapus hanya dalam sekali proses penghapusan saja. Pada menu rubah jam dan rubah tanggal, ini dilakukan apabila alat mengalami *error* atau penggantian baterai pada RTC, sehingga dibutuhkan lagi pengesetan dari awal terhadap pewaktuan.

3.1.3. Perancangan dan Pembuatan Rangkaian Mikrokontroler AT89S8252

3.1.3.1. Perancangan Penggunaan Port-Port Pada Mikrokontoller AT89S8252

Pada skripsi ini IC mikrokontroller AT89S8252 digunakan sebagai pusat pengendali kerja dari alat yang dibuat karena pada IC inilah akan disimpan program-program *(software)* perintah serta alamat yang akan dituju program. Untuk melaksanakan fungsi tersebut diatas maka perlu dirancang port-port I/O serta sinyal-sinyal yang akan digunakan dengan seksama.

Gambar 3.2 menunjukkan rancangan port-port I/O serta sinyal-sinyal pada IC mikrokontroller AT89S8252 yang dimanfaatkan pada skripsi ini.



Gambar 3.2. Pemakaian Port-Port Mikrokontroller AT89S8252.

- Port 0

Port 0 merupakan port dua fungsi yang berada pada pin 32-39 dari IC AT89S8252. Dalam perancangan, P0.0-P0.7 digunakan sebagai port keluaran ke relay 1-8.

- Port 1

Port 1 disediakan sebagai port output untuk menampilkan data ke LCD dan menempati pin 1-8, dalam perancangan port ini yang digunakan P1.0-P1.7.

- Port 2

Port 2 yang berada pada pin 21-28 merupakan port dua fungsi yaitu sebagai I/O serbaguna, atau sebagai bus alamat byte tinggi untuk rancangan yang melibatkan memori eksternal. Karena dalam perancangan tidak digunakan memory eksternal maka port ini digunakan untuk keypad yakni P2.0-P2.7.

- Port 3

Port 3 dalam perancangan ini P3.2, P3.6, P3.7 dengan pin 12, pin 16, dan pin17 digunakan sebagai inputan dari RTC (Real Time Clock), sedangkan P3.0-P3.1 dengan pin 10-11 digunakan relay channel 9 dan relay power.

- $\overline{PSEN}$ *(Program Strore Enable)*

$\overline{PSEN}$ adalah suatu sinyal keluaran yang terdapat pada pin 29. Fungsinya adalah sebagai sinyal kontrol untuk memungkinkan mikrokontroller membaca program (code) dari memori eksternal. Jika eksekusi program dari ROM internal (8051/8052) atau dari flash memori AT89S8252 , maka $\overline{PSEN}$ berada pada kondisi tidak aktif (high).

- ALE *(Address Latch Enable)*

Sinyal output ALE yang berada pada pin 30 fungsinya untuk demultipleks bus alamat dan bus data. Sinyal ALE membangkitkan pulsa sebesar 1/6 frekwensi oscilator dan dapat dipakai sebagai clock yang dipergunakan secara umum.

- $\overline{EA}$ *(External Access)*

Masukan sinyal $\overline{EA}$ terdapat pada pin 31 yang dapat diberikan logika rendah (pin terhubung ground) atau logika tinggi (pin terhubung Vcc). Jika $\overline{EA}$ diberikan logika tinggi, maka mikrokontroller akan mengakses program dari ROM internal (EPROM/flash memory). Jika $\overline{EA}$ diberikan logika rendah, maka mikrokontroller akan mengakses program dari memori eksternal. Pada skripsi ini $\overline{EA}$ dihubungkan ke Vcc

- RST *(Reset)*

Input reset pada pin 9 adalah reset master untuk AT89S8252.

- Oscilator

Oscilator yang disediakan pada chip dikemudikan dengan XTAL yang dihubungkan pada pin 18 dan pin 19. Besar nilai XTAL yang digunakan sebesar 11,0592 MHz untuk keluarga MCS-51, dan diperlukan dua buah kapasitor penstabil sebesar 30pF.

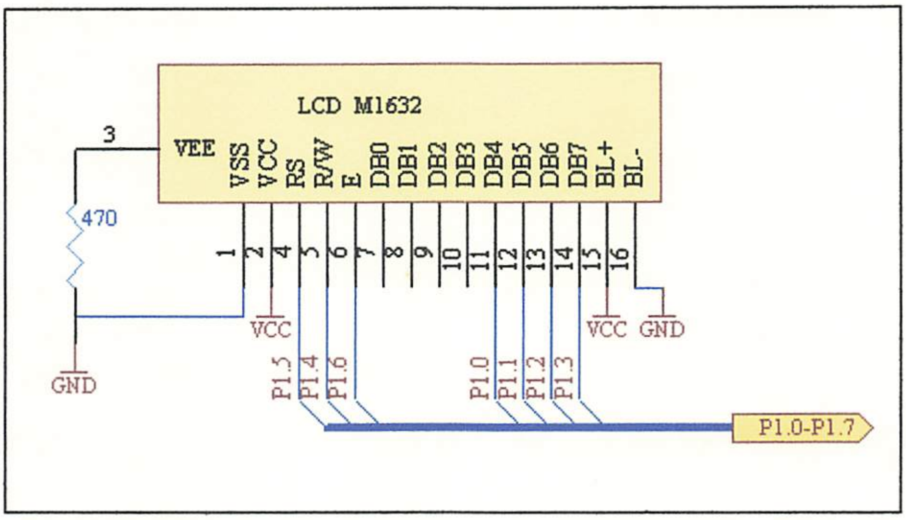
- Vcc

AT89S8252 dioperasikan dengan tegangan supply +5V. Pin Vcc berada pada pin nomor 40 yang dipararel dengan EA/VPP pada pin 31, sedangkan Vss (ground) pada AT89S8252 berada pada pin 20.

3.1.4. LCD (Liquid Crystal Display)

Sebagai penampil (display) hari, jam, menit, tanggal, bulan dan tahun pada keadaan standby, serta pesan untuk memasukkan nomor channel dan penyetingan waktu pada alat ini digunakan LCD jenis TM162 yang membutuhkan arus maksimal 3 mA dan tegangan 5V. LCD jenis TM162 dapat menampilkan 16 karakter dalam 2 baris dihubungkan dengan data bus mikrokontroler..

Dengan bantuan perangkat lunak yang dibuat, dapat ditampilkan karakter yang bisa dilihat pada layar tampilan, yaitu dengan dengan mengendalikan pada pena E, R/W dan RS. Untuk lebih jelasnya perancangan LCD dan pin-pin yang terhubung pada mikrokontroler dapat dilihat pada gambar 3.3.

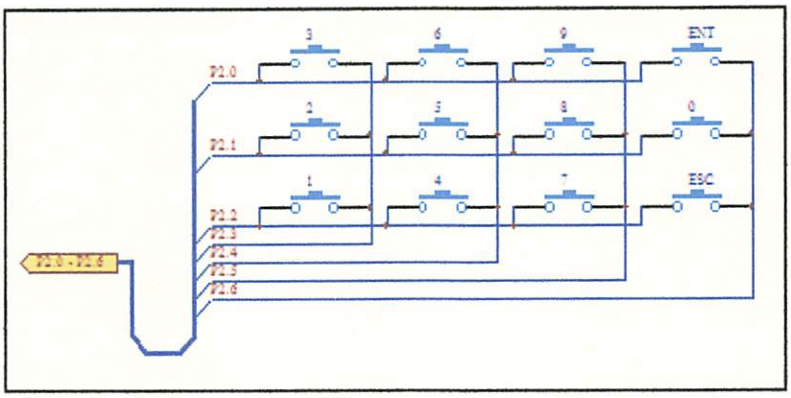


Gambar 3.3. Rangkaian LCD M1632.

Dalam perencanaan alat ini menggunakan mode 4 bit untuk menghemat port pada mikrokontoler AT89S8252, penggunaan pin-pin pada LCD yang terhubung dengan AT89S8252 dapat dilihat dari gambar diatas, bahwa pada pin 4 (FS) akan terhubung dengan P1.5, pin 5 (R/W) terhubung dengan P1.4, pin 6 (E) terhubung dengan P1.6, pin 11 (DB4) terhubung dengan P1.0, pin 12 (DB5) terhubung pada P1.1, pin 13 (DB6) terhubung dengan P1.2, dan pin terakhir yang terhubung dengan AT89S8252 pin 14 (DB7) terhubung dengan P1.4. Sedangkan antara VEE dan VSS dihubungkan resistor dengan nilai 470 ohm kemudian digroundkan. BL+ terhubung dengan Vcc dan BL- terhubung dengan ground.

3.1.5. Keypad

Pada prinsipnya keypad adalah switch yang dipasang secara matrik, yang dipakai dalam hal ini adalah matrik 3 × 4, keypad ini berfungsi untuk memasukkan angka-angka yang nantinya digunakan untuk pengisian waktu disesuaikan pada menu yang ditampilkan pada LCD. Proses penjadwalan perencanaan waktu yang diinginkan dapat dimasukkan sesuai dengan penekanan pada keypad ini. Penekanan keypad ini akan dideteksi dan akan mengeluarkan data yang diterjemahkan dalam bentuk biner. Data biner ini akan langsung masuk ke mikrokontroller AT89S8252. Dalam perencanaan ini memakai port 2 sebagai inputan ke mikrokontroler.



Gambar 3.4. Rangkaian Keypad.

3.1.6. Driver Relay

Rangkaian driver berfungsi untuk mengaktifkan remote yang nantinya diteruskan ke televisi. Pada perancangan ini terdiri dari sebuah *optocoupler* 4N35. Pemasangan *optocoupler* berfungsi sebagai penahan *bouncing* yang muncul akibat kelebihan arus yang ditimbulkan oleh beban, sehingga dapat menyebabkan mikrokontroler menjadi rusak baik secara fisik maupun non fisik (program di dalam mikrokontroler itu sendiri). Walaupun sebenarnya *bouncing* itu sendiri telah diantisipasi dengan memasang sebuah diode yang terpasang paralel dengan *relay*-nya sebelum masuk ke beban. *Relay* yang digunakan adalah *relay* dengan tegangan 5 volt dan hambatan 100 Ω, maka arus yang mengalir dapat dihitung dengan rumus:

$$Irelay = \frac{\text{Vrelay}}{Rrelay}$$

$$Irelay = \frac{5volt}{100\Omega} = 0.05A$$

Transistor yang dapat digunakan untuk arus tersebut adalah tipe C9012 karena memiliki Ic maksimum 1.5 A(data sheet). Oleh karena relay ini terhubung pada bagian kolektor dari transistor ,maka dapat dianggap:

Ic = I*relay* = 0.05 A.

Jika transistor ini memiliki β = 150.

Ic = 0.05 A.

Maka :

$$Ib = \frac{Ic}{\beta}$$

$$Ib = \frac{0.05}{150} = 0.33mA.$$

Sehingga ,

$$Rb = \frac{Vcc - Vbe}{Ib}$$

$$Rb = \frac{5volt - 0.7}{0.33mA}$$

= 13030 Ω. = 13 kΩ.

Karena harga resistor 13030 Ω tidak ada dipasaran, maka digunakan nilai resistor sebesar 12 kΩ.

Untuk LED pada *optocoupler* besarnya V*led=3*volt dan I*led*=60mA(berdasarkan data sheet),.

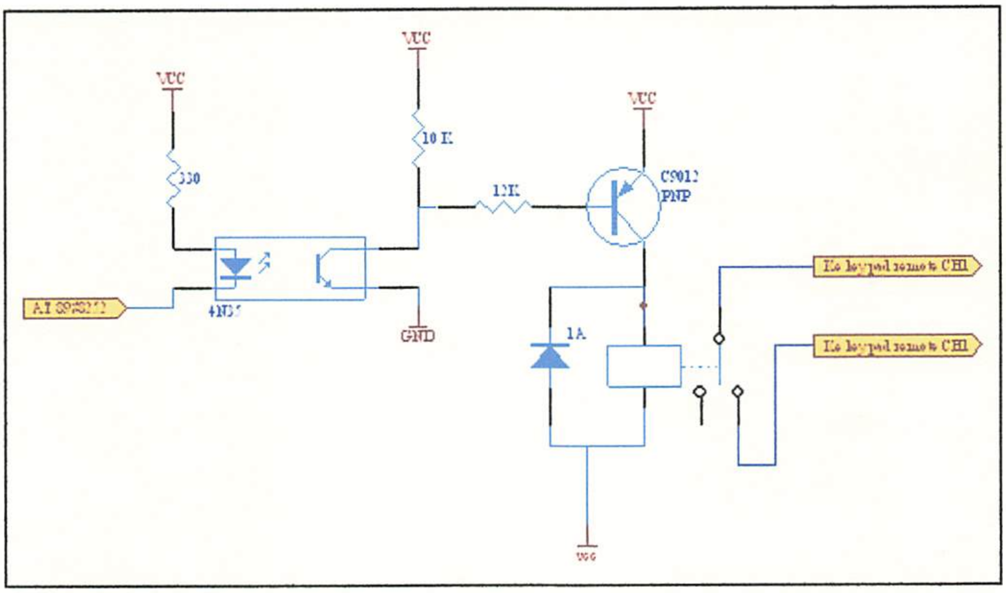
Sehingga digunakan resistor yang besarnya adalah:

$$R1 = \frac{Vcc - V_{LED}}{I_{LED}}$$

$$= \frac{5Volt - 3Volt}{60mA}$$

R1 = 333 Ω ≈ 330 Ω.

Karena di pasaran tidak ada nilai 333 Ω, sehingga dipilih R1 sebesar 330 Ω.



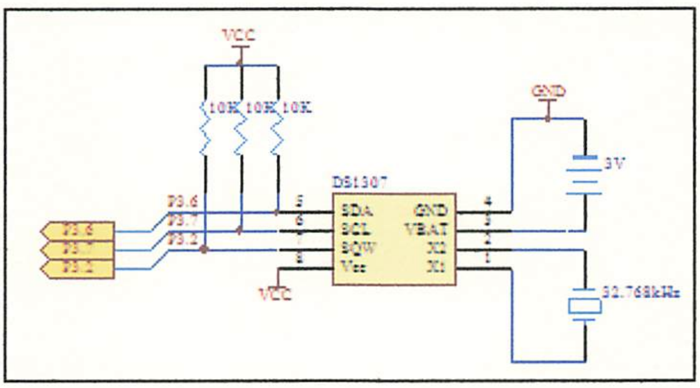
Gambar 3.5. Rangkaian *Driver Relay.*

3.1.7. Rangkaian RTC DS1307

Dalam sistem yang dirancang ini RTC difungsikan sebagai pewaktu, yaitu peripheral yang menyediakan data detik, menit, jam, hari, tanggal, bulan dan tahun biasa ataupun kabisat. Data waktu ini (RTC) nantinya akan diolah oleh mikrokontroler dan ditampilkan pada LCD serta akan dijadikan sebagai inputan data untuk mengatur setting waktu yang dibutuhkan saat proses pembuatan jadwal televisi.

RTC yang dipilih adalah serial RTC produksi MAXIM yaitu DS1307, pemilihan serial RTC dari MAXIM karena beberapa hal dibawah ini antara lain:

- Harga yang relative lebih murah jika dibandingkan dengan RTC lainnya.
- Karena akses data dilakukan secara serial, maka hanya butuh 2 pin saja, ini akan menghemat port pada mikrokontroller.
- Y2K kompatibel, jadi tidak perlu lagi ada perbaikan penanggalan untuk penggunaan ditahun 2000 ke atas.
- Sudah tersedia pin untuk baterai back-up, sehingga tidak perlu lagi dibuatkan lagi rangkaian untuk baterai back-up, apabila kehilangan supply tegangan .
- Komponen pendukungnya mudah diperoleh dipasaran.



Gambar 3.6. Rangkaian Serial RTC DS1307

Dari gambar 3.6 dapat dilihat bahwa pin SDA dan SCL terhubung dengan mikrokontroler tepatnya pada P3.6 dan 3.7. Pin SQW/OUT juga terhubung pada pin 3.2 (INTO) pada mikrokontoler AT89S8252, dengan maksud agar pulsa yang dikeluarkan dari pin SQW/OUT akan dijadikan interrupt bagi mikrokontroler. Seperti kita ketahui bahwa SQW/OUT dapat mengeluarkan pulsa yang dapat diset nilainya. Fungsi dari pulsa ini adalah sebagai interrupt bagi mikrokontroler untuk membaca nilai waktu dari RTC (Real Time Clock).

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 dihubungkan dengan port 2.7 dan port 2.6 pada mikrokontroller.

3.2. Perancangan Perangkat Lunak (Software)

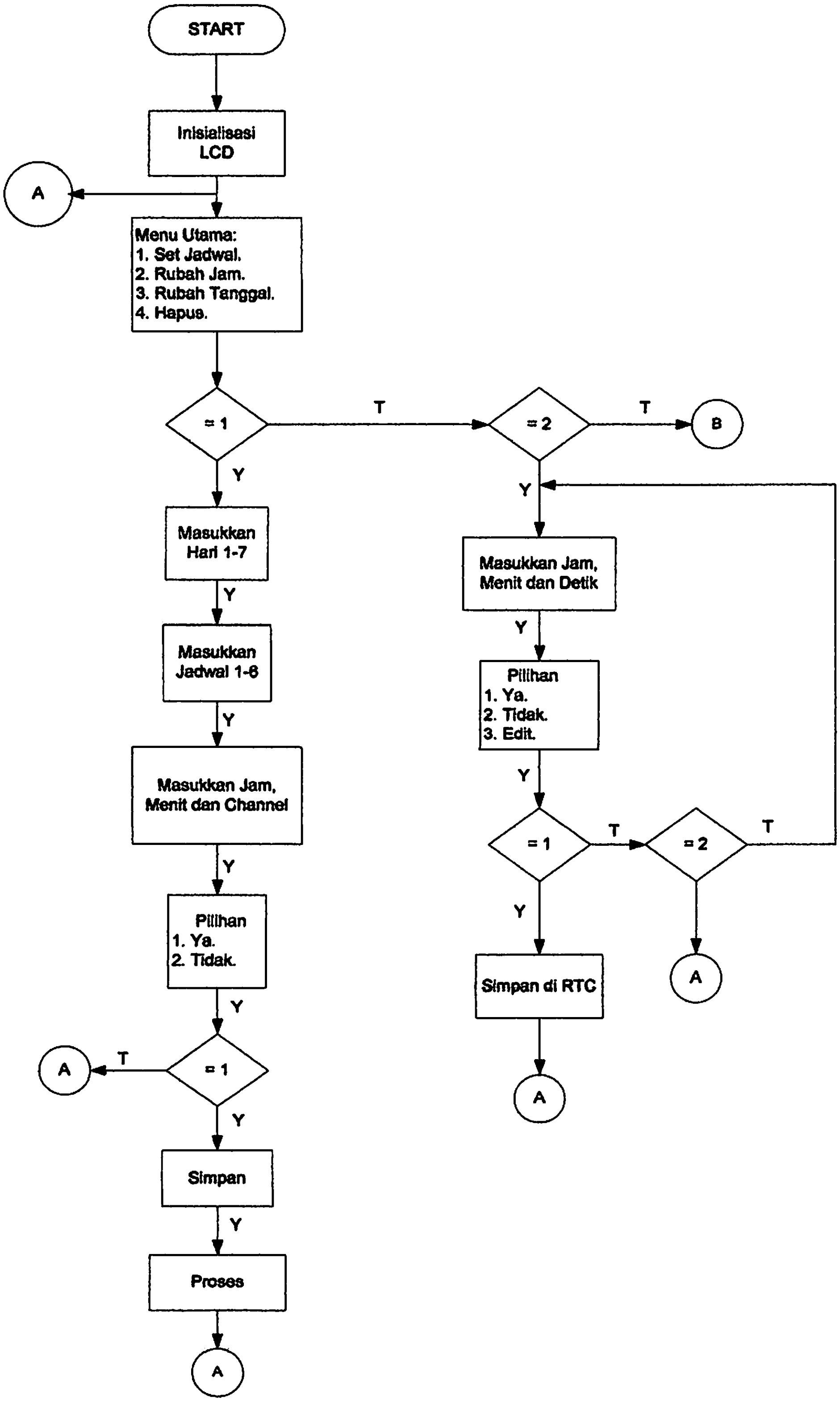
Untuk pemakaian mikrokontroler didalam suatu sistem, perlu direncanakan perangkat lunak mikrokontroler yang dapat mengatur system tersebut. Perangkat lunak disini adalah perintah-perintah (program) didalam memori yang harus dilaksanakan oleh mikrokontroller.

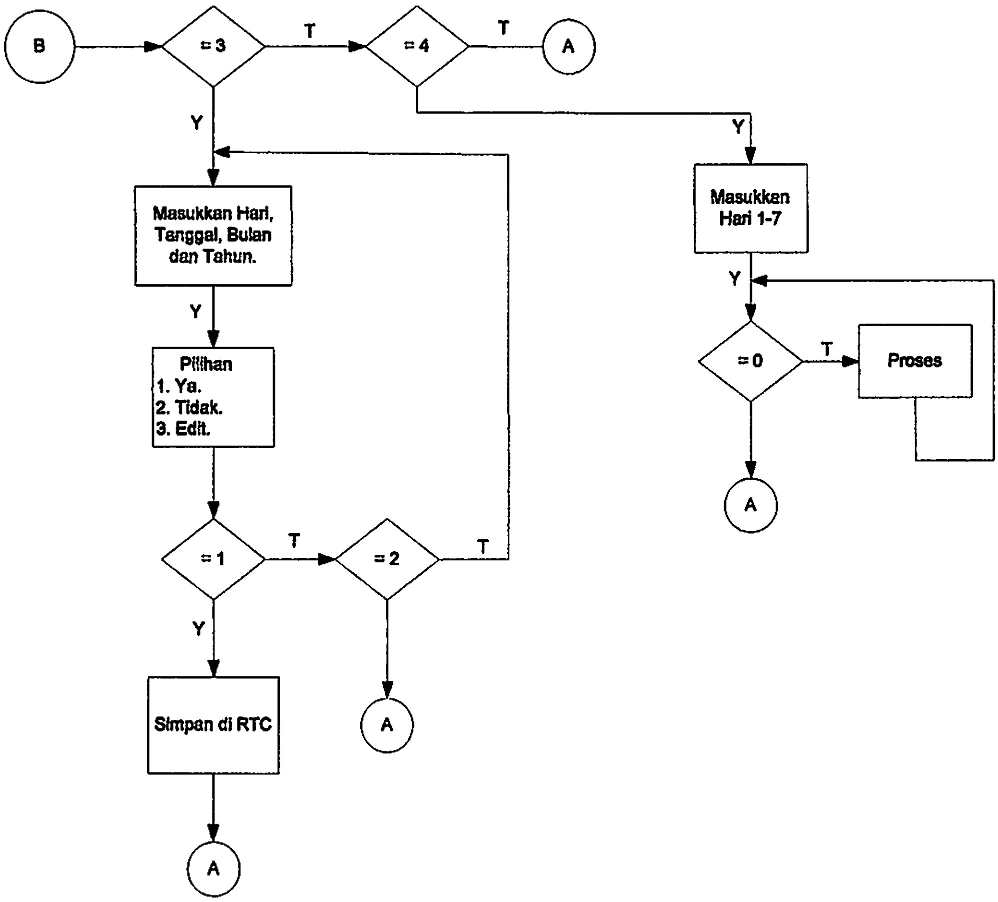
Didalam suatu mikrokontroler memori merupakan fasilitas utama karena disinilah disimpan perintah-perintah yang harus dikerjakan. Memori disini dapat dibedakan menurut fungsinya menjadi memori program dan memori data.

Perencanaan perangkat lunak (software) didasarkan perencanaan perangkat keras yang telah dibuat sebelumnya.

3.2.1. Flow Chart

Cara kerja dari perangkat lunak (*software*) secara umum sebagai berikut:





Gambar 3.7. Flowchart Keseluruhan Sistem.

BAB IV

PENGUJIAN ALAT

4.1. Umum

Pengujian alat ini dilakukan untuk mengetahui kinerja dari keseluruhan sistem rangkaian. Jadi pada tahap ini akan diketahui nilai-nilai serta parameter-parameter dari setiap bagian yang menyusun sistem secara keseluruhan. Secara umum, pengujian ini bertujuan untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pengujian yang dilakukan antara lain rangkaian Sistem Mikrokontroler, rangkaian RTC DS1307, rangkaian Keypad, rangkaian LCD, dan rangkaian Driver Relay.

4.2. Pengujian Perangkat Keras

4.2.1. Pengujian Sistem Mikrokontroller

1. Tujuan.

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

2. Peralatan Yang Dibutuhkan.

   - Komputer.
   - Sistem mikrokontroller.

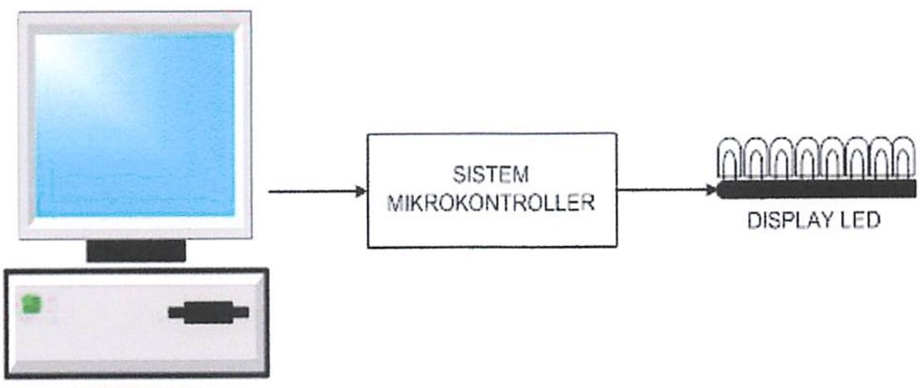
3. Prosedur Pengujian.

   - Membuat program yang digunakan dalam pengujian mikrokontroller.

- Program yang digunakan dalam pengujian mikrokontroller ini merupakan program sederhana dengan memasukkan bilangan 0FH dan F0H ke *port* 2.
- *Port* 2 AT89S8252. Program yang dibuat adalah sebagai berikut :

```
           ORG     0H
MULAI:     MOV     P2,#0FH          ;kondisi satu
           CALL    DELAY
           MOV     P2,# F0H         :kondisi dua
           CALL    DELAY
           JMP     MULAI
DELAY:     MOV     R0,#0
DELAY1:    MOV     R5,#50H
           DJNZ    R5,$
           DJNZ    R0,DELAY1
           END
```

- Membuat rangkaian seperti Gambar 4.1:



Gambar 4.1. Diagram Blok Pengujian Mikrokontroller

- Memasang catu rangkaian sebesar 5V.

- Men*download* program diatas.
- Mengamati keluaran pada LED d*isplay*.
- Hasil pengujian.

Hasil pengujian mikrokontroller adalah sebagai berikut :

Tabel 4.1

Hasil Pengujian Sistem Mikrokontroller

KONDISI	**KELUARAN PADA LED DISPLAY**							
	Bit 0	**Bit 1**	**Bit 2**	**Bit 3**	**Bit 4**	**Bit 5**	**Bit 6**	**Bit 7**
Satu	0	0	0	0	1	1	1	1
Dua	1	1	1	1	0	0	0	0

Apabila sistem mikrokontroler dapat bekerja dengan baik maka hasil keluaran tampilan LED akan sesuai dengan data masukan yang diberikan lewat program. Dalam tabel tersebut, nilai 1 menunjukkan bahwa LED dalam keadaan menyala, sedangkan nilai 0 menunjukkan LED dalam keadaan mati.

Pada saat *port* 2 diberi bilangan 0FH, LED pada P2.0 sampai dengan P2.3 padam, dan LED pada P2.4 sampai P2.7 nyala. Ketika *port* 1 diberi bilangan F0H, P2.0 sampai P2.3 nyala dan P2.4 sampai P2.7 padam. Kemudian program terus mengulang sehingga LED akan menyala dan padam secara bergantian.

4.2.2. Pengujian RTC DS1307

1. Tujuan.

Tujuan dari pengujian Serial RTC DS1307 dan EEPROM AT89S8252 adalah untuk mengetahui apakah Serial RTC DS1307 dapat berfungsi dengan baik dan data dapat tetap tersimpan meskipun tegangan pada rangkaian terputus (mati), sehinga sistem rangkaian dapat bekerja seperti yang telah kita rencanakan, yaitu :

- Dapat melakukan perhitungan waktu meskipun supply tegangan telah diputuskan.
- Benar-benar Y2K kompatibel, artinya bahwa perhitungan penanggalan telah sesuai untuk penggunaan diatas tahun 2000.

2. Peralatan yang digunakan.

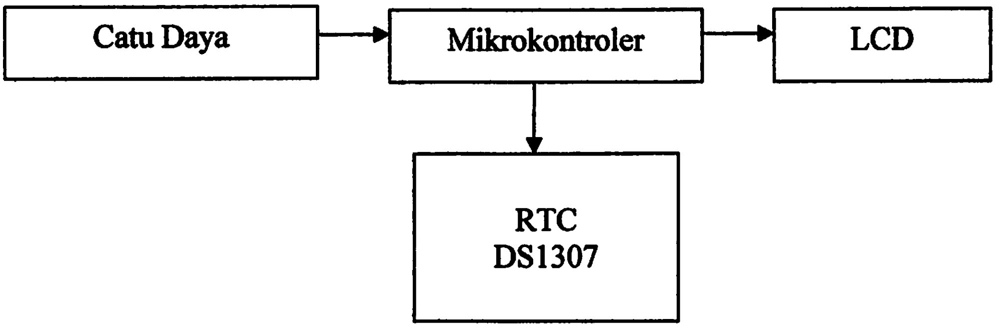
Rangkaian pengujian Serial RTC DS1307 dapat dilihat pada gambar 4.2. Dari gambar tersebut dapat dilihat bahwa komponen yang digunakan dalam pengujian adalah :

- Mikrokontroler AT89S8252, sebagai pengendali dari seluruh rangakaian penguji Serial RTC DS1307.
- Serial RTC DS1307.
- Baterai litium CR2032 sebagai baterai back-up untuk Serial RTC DS1307.
- X'Tal standart quartz 32,768KHz sebagai osilator referensi bagi Serial RTC DS1307.
- LCD 16x2 sebagai media penampil waktu.
- Beberapa komponen pasif pendukung.

3. Prosedur Pengujian.

Langkah-langkah pengujian rangkaian Serial RTC DS1307 adalah sebagai berikut:

- Membuat rangkaian seperti terlihat pada gambar 4.2.



Gambar 4.2 Diagram Blok Pengujian RTC.

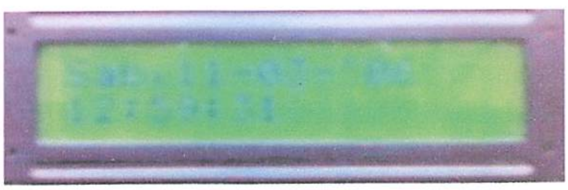
- Mengadakan pengamatan pada tampilan LCD (yang menampilkan hari, tanggal, bulan, tahun, jam dan menit).
- Membuat program untuk pengujian RTC seperti berikut ini:

```
Subroutine RTC_ShowTime:
rtc_sacajam1:
     Mov   DPTR,#rtc_bufjam
     Call  RAM_Read
     Mov   06h,A
     Anl   A,#0F0h
     Add   A,#30h
     Call  LCD_Tulis_Data
     Mov   A,06h
     Anl   A,#0Fh
     Call  LCD_Tulis_Data
     Mov   A,#':'
     Call  LCD_Tulis_Data
rtc_sacamnt1:
     Mov   DPTR,#rtc_bufmin
     Call  RAM_Read
     Mov   06h,A
     Anl   A,#0F0h
```

```
    Add   A,#30h
    Call  LCD_Tulis_Data
    Mov   A,06h
    Anl   A,#0Fh
    Call  lcd_tulis_data
    Mov   A,#':'
    Call  LCD_Tulis_Data
rtc_sacadet1:
    Mov   DPTR,#rtc_bufdet
    Call  RAM_Read
    Mov   06h,A
    Anl   A,#0F0h
    Add   A,#30h
    Call  LCD_Tulis_Data
    Mov   A,06h
    Anl   A,#0Fh
    Call  LCD_Tulis_Data
    Ret
EndSub
```

4. Dari hasil pengujian dapat dianalisa sebagai berikut:

- Bahwa RTC mampu bekerja meskipun power supply telah dimatikan (RTC bekerja dengan menggunakan baterai back-up).
- Bahwa Serial RTC DS1307 benar-benar kompatibel untuk penggunaan diatas tahun 2000, ini terlihat bahwa mampu mendeteksi adanya tahun kabisat



Gambar 4.3. Tampilan Hasil Uji RTC.

Dengan dua analisa tersebut, dapat kita simpulkan bahwa Serial RTC DS1307 bisa berfungsi dengan baik sesuai dengan rancangan yang telah kita rencanakan sebelumnya.

4.2.3. Pengujian Keypad.

1. Tujuan.

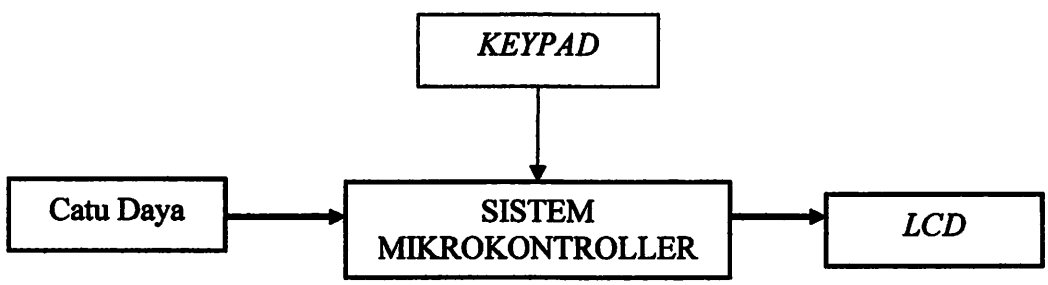
Untuk mengetahui keluaran dari unit papan tombol saat tombol ditekan apakah sesuai dengan tampilan yang ada di *LCD.*

2. Peralatan yang dibutuhkan.

- Catu daya 5V DC.
- Rangkaian *Keypad.*
- *MCU.*
- Rangkaian *LCD*

3. Prosedur pengujian :

Alat-alat dirangkai seperti ditunjukkan pada gambar diagram blok dibawah ini:



Gambar 4.4. Diagram Blok Pengujian *Keypad.*

4. Hasil Pengujian.

Dari hasil pengujian dapat diketahui bahwa rangkaian *Keypad* dapat berfungsi dengan baik hal ini dapat diketahui dari tombol yang ditekan sesuai dengan tampilan yang ada di *LCD,* berikut ini tabel perbandingan tombol yang ditekan dengan tampilan di *LCD* :

Tabel 4.2. Hasil Pengujian Rangkaian *Keypad*

SWITCH	Tampilan *LCD*
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0
*	*
#	#

4.2.4. Pengujian LCD

1. Tujuan.

Mengetahui apakah rangkaian LCD dapat menampilkan data karakter yang sesuai dengan data yang dikirimkan.

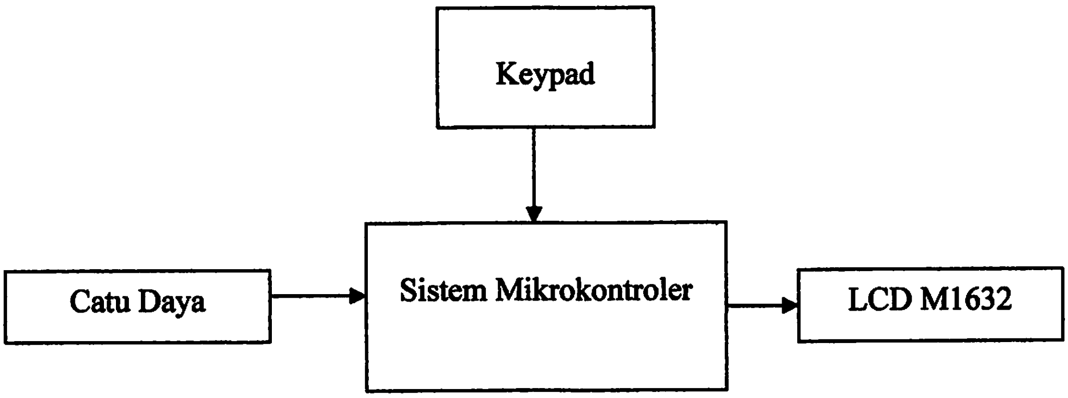
2. Peralatan Yang Dibutuhkan.

- LCD (Liquid Crystal Display).
- Keypad.
- Sistem Mikrokontroler AT89S8252.

- Catu Daya 5V.

3. Prosedur Pengujian.

- Merangkai peralatan seperti dalam Gambar 4.5.

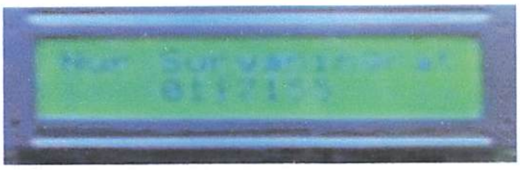


Gambar 4.5. Diagram Blok Pengujian *LCD.*

- Membuat perangkat lunak pengujian rangkaian LCD. Program ini berisi inisialisasi Mikrokontroller, dan LCD.
- Mengaktifkan catu daya
- Mengoperasikan program, hasil keluaran akan ditunjukkan pada layar penampil kristal cair.

4. Hasil Pengujian.

Dari hasil pengujian didapatkan bahwa rangkaian *LCD* dapat menampilkan karakter-karakter, sesuai dengan data yang dikirimkan. Tampilan terdiri atas 2 baris yang masing-masing mempunyai 11 karakter.



Gambar 4.6. Hasil Pengujian *LCD.*

4.2.5. Pengujian Driver Relay

1. Tujuan.

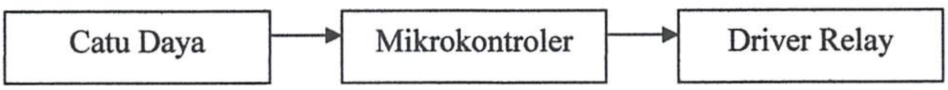
   Mengetahui apakah rangkaian *driver* relay dapat bekerja dengan baik.

2. Peralatan yang dibutuhkan:

   - *Power supply* 5V.
   - Rangkaian Driver Relay.

3. Prosedur pengujian.

   - Alat-alat dirangkai seperti ditunjukkan pada gambar 4.7.
   - Memberi tegangan 5V pada rangkaian driver.



Gambar 4.7. Rangkaian Pengujian *Driver* Relay.

4. Hasil Pengujian.

   Hasil pengujian rangkaian *driver* relay adalah sebagai berikut :

Tabel 4.3. Hasil Pengujian Rangkaian *Relay.*

Arus Yang Diukur	Hasil Pengukuran (mA)	Hasil Perhitungan (mA)
Ic	49	50
Ib	0.30	0.33

$$\text{Eror} = \frac{|perhitungan - pengukuran|}{perhitungan} \times 100\%$$

$$\text{Erorr Ib} = \frac{|50 - 49|}{50} \times 100\%$$

Erorr Ib = 2%.

$$\text{Erorr Ic} = \frac{|0.33 - 0.30|}{0.33} \times 100\%$$

Erorr Ic = 9 %.

4.3. Pengujian Perangkat Keras Secara Keseluruhan

1. Tujuan.

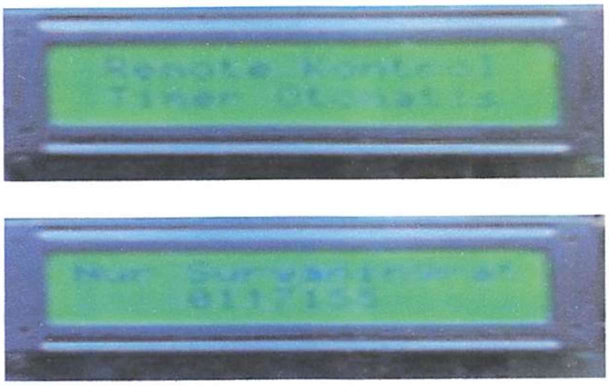
Tujuan dari pengujian perangkat keras secara keseluruhan ini adalah untuk mengetahui kinerja perangkat keras secara keseluruhan apabila dijalankan sesuai perintah yang kita jalankan.

2. Peralatan Yang Dibutuhkan.

- Catu daya 5V.

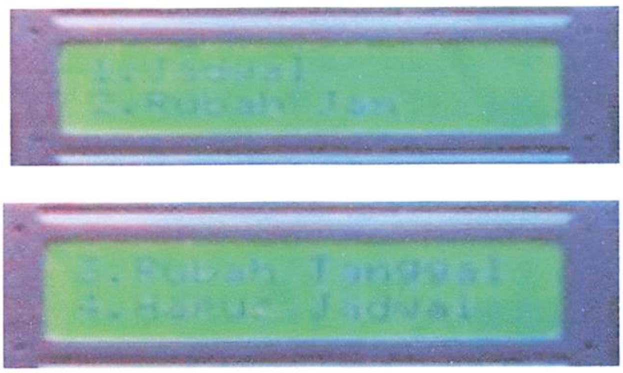
3. Prosedur Pengujian.

- Merangkai semua perangkat keras menjadi satu sesuai dengan blok diagram.
- Mengaktifkan catu daya.



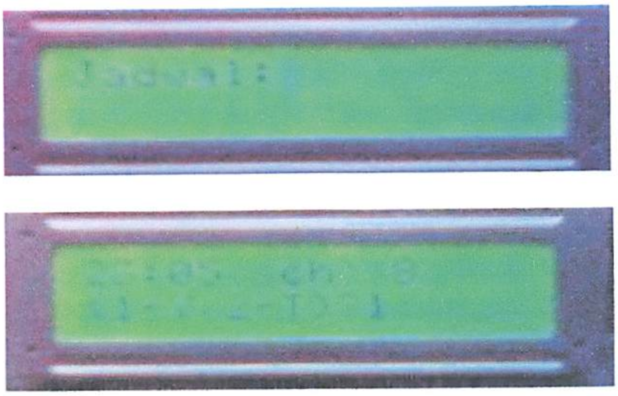
Gambar 4.8. Tampilan Awal LCD.

- Menekan tombol # pada *keypad* untuk masuk ke menu utama.



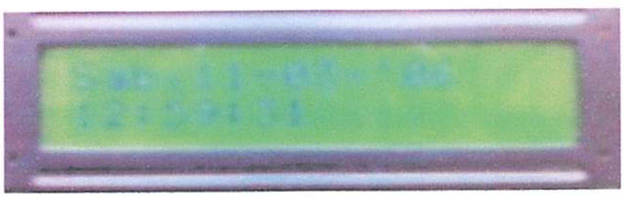
Gambar 4.9. Tampilan Menu Utama Pada LCD.

- Pilih menu, selanjutnya pilih hari, dan jadwal ke berapa yang akan diisi, ini dibatasi maksimal 6 dalam sehari. Dan untuk channel yang bisa diprogram hanya 9 channel. Kemudian memasukkan jam, menit, dan nomor channel yang akan diprogram seperti tampak pada gambar 4.10.

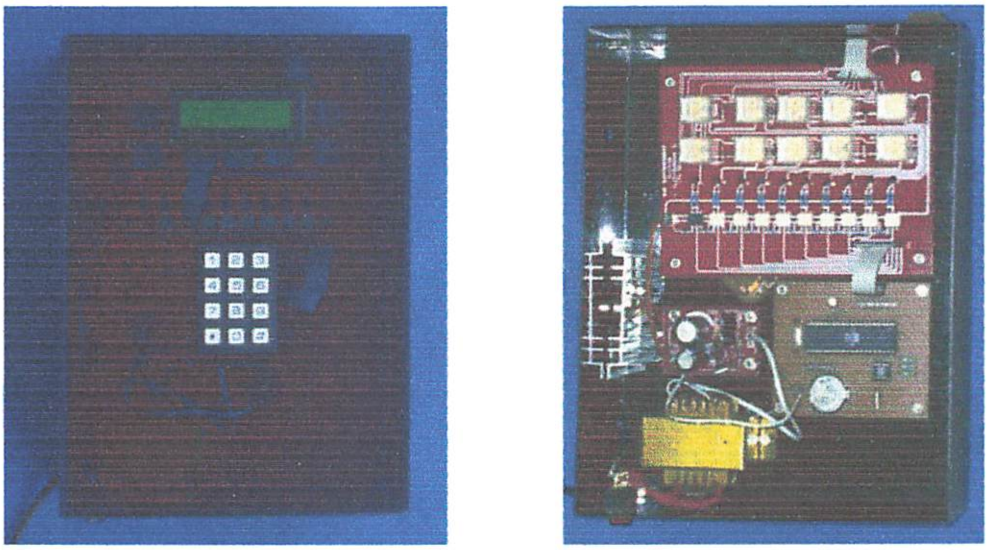


Gambar 4.10. Tampilan LCD Saat Proses Pemprograman.

- Proses membandingkan data yang tersimpan pada Mikrokontroler dengan RTC.



Gambar 4.11.Proses Membandingkan Data Dan Tampilan Standby.



4.12. Gambar Alat Keseluruhan.

BAB V

PENUTUP

5.1. Kesimpulan

Dari perencanaan, pembuatan, dan pengujian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Penggunaan mikrokontroler pada alat ini sangat membantu dalam system pengendalian alat.
2. Driver relay dalam alat ini dapat mengaktifkan tombol remote, dan dapat berjalan dengan baik,
3. Pada pengujian rangkaian driver relay terdapat adanya *error,* yaitu untuk Ic = 9% dan Ib = 2%.
4. RTC mampu menyediakan data pewaktu dengan benar.
5. Data yang sudah disimpan pada EEPROM tidak berubah walaupun catu daya mati.
6. Pada pengujian *Liquid Crystal Display,* dapat menampilkan informasi dengan baik sesuai dengan program.

5.2. Saran

Untuk pengembangan lebih lanjut, alat ini dapat ditambahkan dengan fitur – fitur lainnya agar lebih kompleks, antara lain: alarm, kalkulator dll.

DAFTAR PUSTAKA

[1] Ibnu M, "*Belajar MIKROKONTORLER ATMEL AT89S8252*" Edisi Pertama, Penerbit Gava Media, Jogkarta,2003.

[2] *http://www.atmel.com*

[3] *http://www.wcs.com*

[4] *http://www.datasheet.com*

LAMPIRAN

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting) Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang ,6 Desember 2005

Nomor : ITN-918/7/TA.GNP/2005
Lampiran :
Perihal : ***Bimbingan Skripsi***

Kepada : Yth. Sdr. **Joseph Dedy Irawan,ST,MT**
Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat.
Sesuai dengan permohonan dan persetujuan dalam proposal skipsi untuk mahasiswa:

Nama	: Nur Suryaningrat
Nim	: 0117155
Semester	: IX
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu 6 (enam) bulan, terhitung mulai tanggal:

5 Desember 2005 s/d 5 Mei 2006

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana.
Demikian agar maklum, atas perhatian dan bantuannya kami ucapkan banyak terima kasih.

Ketua Jurusan
Teknik Elektro S-1

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

FORMULIR BIMBINGAN SKRIPSI

Nama : NUR SURYANINGRAT
NIM : 01.17.155
Masa Bimbingan : 05 Desember 2005 s/d 05 Mei 2006
Judul Skripsi : PERENCANAAN DAN PEMBUATAN REMOTE KONTROL PENYETING TIMER UNTUK MENGUBAH CHANEL SECARA OTOMATIS PADA TELEVISI DENGAN MENGGUNAKAN MIKROKONTROLLER AT89S8252.

NO	Tanggal	Uraian	Paraf Pembimbing
1	02-01-06	Penulisan Bab I kurang benar	[paraf]
2	06-01-06	Bab II tidak ada sumber data pada gambar	[paraf]
3	09-01-06	Revisi Driver Relay Bab III -	[paraf]
4	26-01-06	Pemilihan komponen Transistor pada relay ?	[paraf]
5	03-02-06	Bab IV pengujian alat kurang foto	[paraf]
6	04-03-06	Flowchart tidak benar / salah alur.	[paraf]
7	13-03-06	Penulisan harap di cek ulang.	[paraf]
8	15-03-2006	Acc KOMPRE	[paraf]
9			
10			

Malang, 15 - 03 - 2006
Dosen pembimbing

Joseph Dedy Irawan, ST, MT
NIP.P 103.9800.324

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Nilai Ujian Skripsi

Nama Mahasiswa : NUR SURYANINGRAT / Nim : 0117155
Fakultas / Jurusan : Teknologi Industri / Teknik Elektro
Konsentrasi : T. ~~Energi Listrik~~ / T. Elektronika S-1 *)

No	Indikator Penilaian	Nilai
1.	**Penguasaan materi skripsi :** • Ketajaman perumusan masalah dan tujuan penelitian • Ketepatan metode yang digunakan • Ketepatan penarikan kesimpulan dengan tujuan dan hasil penelitian	
2.	**Penguasaan materi penunjang Skripsi / tinjauan pustaka :** • Relevansi, kemuktakhiran dan penyusunan daftar pustaka	
3.	**Konstribusi hasil penelitian** • Manfaat hasil penelitian bagi pengembangan IPTEKS, pembangunan dan atau pengembangan kelembagaan	
	Nilai rata-rata	75

Catatan :

1. Nilai diberikan dalam bentuk angka dengan kisaran : 0 sampai dengan 100
2. Nilai komulatip dari penguji kurang dari 56 peserta ujian dinyatakan tidak lulus

Malang, ____________

Moderator / Dosen Pembimbing

Joseph

Dosen Penguji

18/3 06

Bambang N.

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA :
N I M :
Perbaikan meliputi :

Malang, 18/7 06

( )

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Nilai Ujian Skripsi

Nama Mahasiswa : NUR SURYANINGRAT / Nim : 01.17.155
Fakultas / Jurusan : Teknologi Industri / Teknik Elektro
Konsentrasi : ~~T. Energi Listrik~~ / T. Elektronika S-1 *)

No	Indikator Penilaian	Nilai
1.	**Penguasaan materi skripsi :** • Ketajaman perumusan masalah dan tujuan penelitian • Ketepatan metode yang digunakan • Ketepatan penarikan kesimpulan dengan tujuan dan hasil penelitian	
2.	**Penguasaan materi penunjang Skripsi / tinjauan pustaka :** • Relevansi, kemuktakhiran dan penyusunan daftar pustaka	
3.	**Konstribusi hasil penelitian** • Manfaat hasil penelitian bagi pengembangan IPTEKS, pembangunan dan atau pengembangan kelembagaan	
	Nilai rata-rata	71

Catatan :

1. Nilai diberikan dalam bentuk angka dengan kisaran : 0 sampai dengan 100
2. Nilai komulatip dari penguji kurang dari 56 peserta ujian dinyatakan tidak lulus

Malang, 18.03.2006

Moderator / ~~Dosen~~ Pembimbing

Dosen Penguji

INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

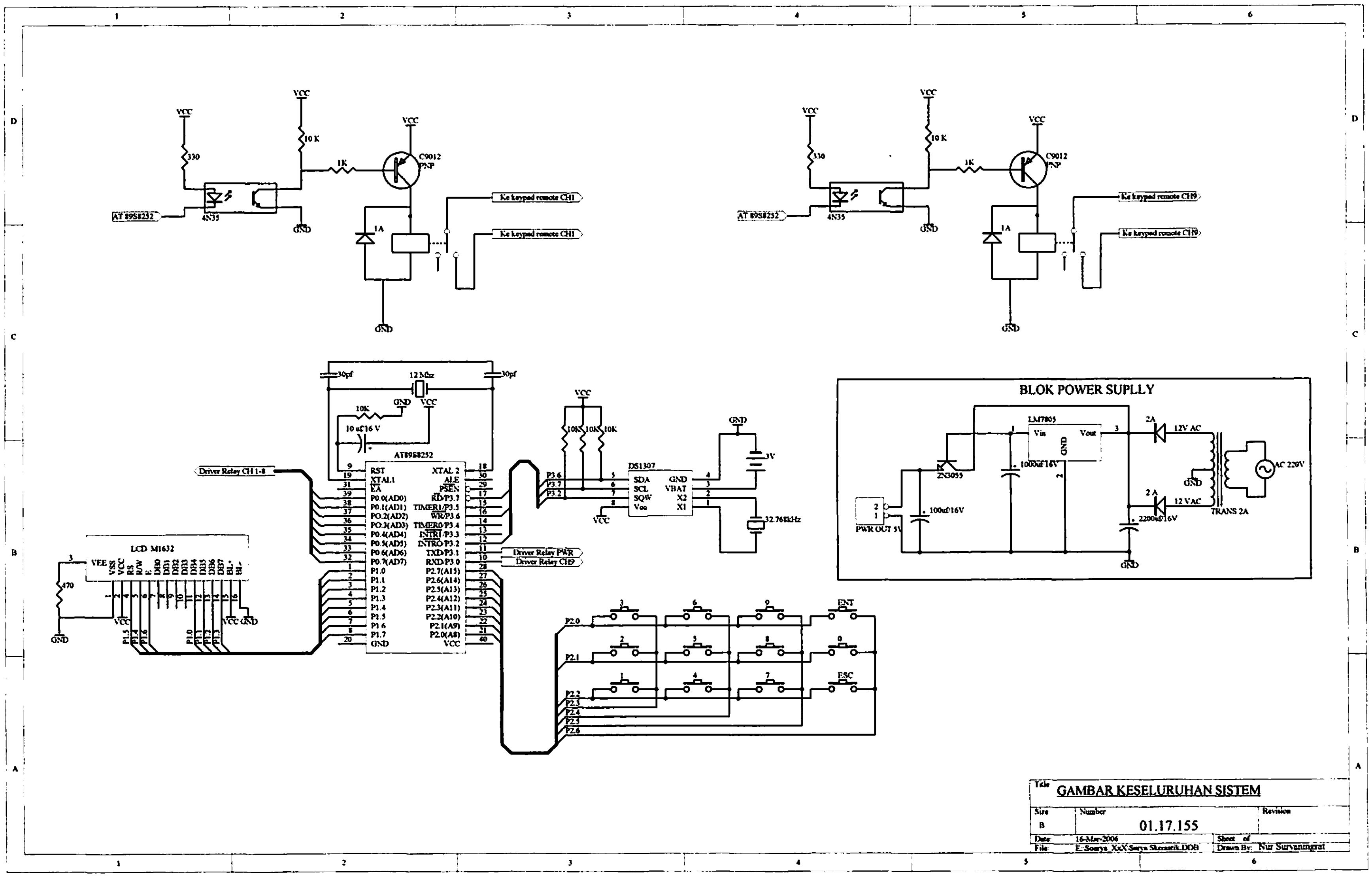
Formulir Perbaikan Ujian Skripsi

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

NAMA :
N I M :
Perbaikan meliputi :

Malang,

(______________________)



VCC
330
10 K
1K
C9012 PNP
4N35
AT 89S8252
GND
1A
Ke keypad remote CH1
Ke keypad remote CH9
30pf
12 Mhz
10K
10 uf/16 V
AT89S8252
Driver Relay CH 1-8
Driver Relay PWR
Driver Relay CH9
LCD M1632
470
DS1307
SDA
SCL
SQW
Vcc
VBAT
X2
X1
3V
32.768kHz
BLOK POWER SUPLLY
LM7805
Vin
Vout
2N3055
1000uf/16V
100uf/16V
2200uf/16V
PWR OUT 5V
2A
12V AC
12 VAC
AC 220V
TRANS 2A
ENT
ESC
P2.0
P2.1
P2.2
P2.3
P2.4
P2.5
P2.6
GAMBAR KESELURUHAN SISTEM
Size
B
Number
01.17.155
Revision
Date 16-Mar-2006
Sheet of
File E:\Soerya_XxX\Serya Skematik.DDB
Drawn By: Nur Suryaningrat

```
kolom1          Bit     P2.4    ; kiri (1,4,7,redial)
kolom2          Bit     P2.5    ;      (2,5,8,0)
kolom3          Bit     P2.6    ;      (3,3,9,#
kolom4          Bit     P2.7

baris1          Bit     P2.0    ; atas (1,2,3)
baris2          Bit     P2.1    ;      (4,5,6)
baris3          Bit     P2.2    ;      (7,8,9)
baris4          Bit     P2.3    ;      (*,0,#)
;
keyport         Data    P2
keydata         Data    15h
FlagKey         Data    16h
FlagKey_matrix  Data    17h

Data_LCD        Equ     P1
lcdrs           Bit     P1.4        ;LCD rs pin connected to P6
lcdrw           Bit     P1.5        ;LCD r/w pin connected to PA5
lcde            Bit     P1.6        ;LCD e pin connected to PA4

;=============== MODULE RTC ============
detrtc   Equ    0000h     ;alamat detik rtc
mntrtc   Equ    0001h
jamrtc   Equ    0002h
harirtc  Equ    0003h
tglrtc   Equ    0004h
blnrtc   Equ    0005h
thnrtc   Equ    0006h

Slave_AddrRTC   Data    20h
DataRTC         Data    20h
SDA             Bit     P3.7    ; pin 5
SCL             Bit     P3.6    ; pin 6
WMCON           Data    96h


;=============== MODULE RTC ============
;--------------------------------------
EE_Senin1       Equ     0000h  ; jam,menit,channel
EE_Senin2       Equ     EE_Senin1+4
EE_Senin3       Equ     EE_Senin2+4
EE_Senin4       Equ     EE_Senin3+4
EE_Senin5       Equ     EE_Senin4+4
EE_Senin6       Equ     EE_Senin5+4
EE_Senin7       Equ     EE_Senin6+4
EE_Senin8       Equ     EE_Senin7+4
EE_Senin9       Equ     EE_Senin8+4

EE_Selasa1      Equ     0040h  ; jam,menit,channel
EE_Selasa2      Equ     EE_Selasa1+4
EE_Selasa3      Equ     EE_Selasa2+4
EE_Selasa4      Equ     EE_Selasa3+4
EE_Selasa5      Equ     EE_Selasa4+4
EE_Selasa6      Equ     EE_Selasa5+4
EE_Selasa7      Equ     EE_Selasa6+4
EE_Selasa8      Equ     EE_Selasa7+4
EE_Selasa9      Equ     EE_Selasa8+4

EE_Rabu1        Equ     0080h  ; jam,menit,channel
EE_Rabu2        Equ     EE_Rabu1+4
EE_Rabu3        Equ     EE_Rabu2+4
EE_Rabu4        Equ     EE_Rabu3+4
EE_Rabu5        Equ     EE_Rabu4+4
EE_Rabu6        Equ     EE_Rabu5+4
EE_Rabu7        Equ     EE_Rabu6+4
EE_Rabu8        Equ     EE_Rabu7+4
```

```
EE_Rabu9            Equ      EE_Rabu8+4

EE_Kamis1           Equ      0120h   ; jam,menit,channel
EE_Kamis2           Equ      EE_Kamis1+4
EE_Kamis3           Equ      EE_Kamis2+4
EE_Kamis4           Equ      EE_Kamis3+4
EE_Kamis5           Equ      EE_Kamis4+4
EE_Kamis6           Equ      EE_Kamis5+4
EE_Kamis7           Equ      EE_Kamis6+4
EE_Kamis8           Equ      EE_Kamis7+4
EE_Kamis9           Equ      EE_Kamis8+4

EE_Jumat1           Equ      0160h   ; jam,menit,channel
EE_Jumat2           Equ      EE_Jumat1+4
EE_Jumat3           Equ      EE_Jumat2+4
EE_Jumat4           Equ      EE_Jumat3+4
EE_Jumat5           Equ      EE_Jumat4+4
EE_Jumat6           Equ      EE_Jumat5+4
EE_Jumat7           Equ      EE_Jumat6+4
EE_Jumat8           Equ      EE_Jumat7+4
EE_Jumat9           Equ      EE_Jumat8+4

EE_Sabtu1           Equ      0200h   ; jam,menit,channel
EE_Sabtu2           Equ      EE_Sabtu1+4
EE_Sabtu3           Equ      EE_Sabtu2+4
EE_Sabtu4           Equ      EE_Sabtu3+4
EE_Sabtu5           Equ      EE_Sabtu4+4
EE_Sabtu6           Equ      EE_Sabtu5+4
EE_Sabtu7           Equ      EE_Sabtu6+4
EE_Sabtu8           Equ      EE_Sabtu7+4
EE_Sabtu9           Equ      EE_Sabtu8+4

EE_Minggu1          Equ      0240h   ; jam,menit,channel
EE_Minggu2          Equ      EE_Minggu1+4
EE_Minggu3          Equ      EE_Minggu2+4
EE_Minggu4          Equ      EE_Minggu3+4
EE_Minggu5          Equ      EE_Minggu4+4
EE_Minggu6          Equ      EE_Minggu5+4
EE_Minggu7          Equ      EE_Minggu6+4
EE_Minggu8          Equ      EE_Minggu7+4
EE_Minggu9          Equ      EE_Minggu8+4
                            ;0280h
;-----------------------------------------

RelayOff            Bit      P3.0
;Relay1              Bit      P3.0
Relay2              Bit      P3.1
Relay3              Bit      P0.7
Relay4              Bit      P0.6
Relay5              Bit      P0.5
Relay6              Bit      P0.4
Relay7              Bit      P0.3
Relay8              Bit      P0.2
Relay9              Bit      P0.1
Relay10             Bit      P0.0

Dphh                Data     40h
Dpll                Data     41h
StatusRelay         Data     42h
NoRelay             Data     43h

TmpTgl              Data     44h
TmpBln              Data     45h
TmpThn              Data     46h
```

```
Tmp1            Data    4Ah
Tmp2            Data    4Bh
Tmp3            Data    4Ch
Tmp4            Data    4Dh

TmpJam          Data    4Eh
TmpMenit        Data    4Fh
TmpDetik        Data    50h

tmpHari         Data    51h
tmpJadwal       Data    52h

Booting:
        Mov     SP,#80h
        Mov     WMCON,#11101010b
        Mov     keyport,#0FFh

        Clr     EA
        Call    LCD_Inisialisasi

        Call    LCD_Blink_off
        Call    LCD_Clear
        Call    lcd_line_1
        Mov     DPTR,#Txt_info_new
        Call    LCD_String
        Call    LCD_Line_2
        Mov     DPTR,#Txt_Info_new1
        Call    LCD_String
        Call    Delay_Fix_1s


        Call    LCD_Clear
        Call    lcd_line_1
        Mov     DPTR,#Txt_info_new2
        Call    LCD_String
        Call    LCD_Line_2
        Mov     DPTR,#Txt_info_new3
        Call    LCD_String
        Call    Delay_Fix_1s

        Call    LCD_Clear
        Call    lcd_line_1
        Mov     DPTR,#Txt_info_new6
        Call    LCD_String
        Call    Delay_Fix_1s

        Call    LCD_Clear
        Call    lcd_line_1
        Mov     DPTR,#Txt_info_new4
        Call    LCD_String
        Call    LCD_Line_2
        Mov     DPTR,#Txt_Info_new5
        Call    LCD_String
        Call    Delay_Fix_1s


;==========================================
; Menu Utama
;==========================================
Menu_Awal:
Menu_Utama:
;==========================================
; Menu Jadwal Waktu
;==========================================
Menu_Jadwal_Waktu:
```

```
        Call     LCD_Blink_Off
        Call     LCD_Clear
        Call     lcd_line_1
        Mov      DPTR,#Txt_Info
        Call     LCD_String
        Call     LCD_Line_2
        Mov      DPTR,#Txt_Info1
        Call     LCD_String
        Call     Delay_Fix_1s

        Call     LCD_Clear
        Call     lcd_line_1

Get_Menu_RTC:
        Clr      kolom3
        Jb       P2.3,Cekjadwalx
        Jmp      lanjut

Cekjadwalx:
        Jmp      CekJadwal

Lanjut:
        Call     LCD_Blink_Off
        Call     LCD_Clear
        Call     lcd_line_1
        Mov      DPTR,#Txt_RTC1
        Call     LCD_String
        Call     LCD_Line_2
        Mov      DPTR,#Txt_RTC3
        Call     LCD_String
        Call     Delay_Fix_1s
        ;Call     Delay_Fix_1s

        Call     LCD_Clear
        Call     lcd_line_1
        Mov      DPTR,#Txt_RTC4
        Call     LCD_String
        Call     LCD_Line_2
        Mov      DPTR,#Txt_RTC5
        Call     LCD_String
        Call     Delay_Fix_1s
        ;Call     Delay_Fix_1s

        Call     LCD_Clear
        Call     lcd_line_1
        Mov      DPTR,#Txt_Pilih
        Call     LCD_String
        Call     LCD_Line_2
        Mov      DPTR,#Txt_Pilih1
        Call     LCD_String
        ;Call     Delay_Fix_1s
;        Call     Delay_Fix_1s

        Mov      FlagKey,#1
        Call     Keypad_KeyIn

        Cjne     A,#'1',Get_Menu_RTC2
        Call     Delay_Fix_100ms

        Jmp      Menu_RTC_SetJadwal
Get_Menu_RTC2:

Get_Menu_RTC3:
        Cjne     A,#'2',Get_Menu_RTC4
        Call     Delay_Fix_100ms
```

```
        Jmp     Menu_RTC_UbahJam            ;ok
Get_Menu_RTC4:
        Cjne    A,#'3',Get_Menu_RTC5
        Call    Delay_Fix_100ms

        Jmp     Menu_RTC_UbahTanggal        ;ok
Get_Menu_RTC5:
        Cjne    A,#'4',Get_Menu_RTC6
        Call    Delay_Fix_100ms

        Jmp     Menu_RTC_HapusJadwal
Get_Menu_RTC6:
Get_Menu_RTC7:
        Cjne    A,#'0',Get_Menu_RTCxx
        Jmp     Menu_Utama
Get_Menu_RTCxx:
        Jmp     Get_Menu_RTC
;-------------------------------------
CekJadwal:
        Call    RTC_GetTime
        Call    RTC_GetDate
        Call    LCD_Line_1
        Call    RTC_ShowDate
        Call    LCD_Line_2
        Call    RTC_ShowTime
        Mov     A,#' '
        Call    LCD_Tulis_Data

        Mov     A,TmpDetik
        Cjne    A,#0,Get_Menu_RTCxx

NextJadwal1:
        Mov     DPTR,#rtc_bufhar        ; pindahkan ke buffer
sementara
        Call    EEPROM_Read
        Cjne    A,#1,NextJadwal2
        Mov     DPTR,#EE_Minggu1
        Jmp     CekHari1
NextJadwal2:
        Cjne    A,#2,NextJadwal3
        Mov     DPTR,#EE_Senin1
        Jmp     CekHari1
NextJadwal3:
        Cjne    A,#3,NextJadwal4
        Mov     DPTR,#EE_Selasa1
        Jmp     CekHari1
NextJadwal4:
        Cjne    A,#4,NextJadwal5
        Mov     DPTR,#EE_Rabu1
        Jmp     CekHari1
NextJadwal5:
        Cjne    A,#5,NextJadwal6
        Mov     DPTR,#EE_Kamis1
        Jmp     CekHari1
NextJadwal6:
        Cjne    A,#6,NextJadwal7
        Mov     DPTR,#EE_Jumat1
        Jmp     CekHari1
NextJadwal7:
        Mov     DPTR,#EE_Sabtu1
        Jmp     CekHari1
;----------------
CekHari1:
        Mov     70h,#0
CekHari2:
        Mov     A,70h
```

```
        Call    Get_DPTR
        Call    EEPROM_Read
        Cjne    A,#0FFh,CekHari3
        Call    EEPROM_Read
        Cjne    A,TmpJam,CekHari3
        Call    EEPROM_Read
        Cjne    A,TmpMenit,CekHari3
        Call    EEPROM_Read
        Call    ProsesRelay
        Jmp     Get_Menu_RTC

CekHari3:
        Inc     70h
        Mov     A,70h
        Cjne    A,#7,CekHari2
        Jmp     Get_Menu_RTC

Subroutine ProsesRelay:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#Txt_Found1
        Call    LCD_String
        Mov     A,NoRelay
        Cjne    A,#1,CekRelay2
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_1
        Call    LCD_String

        Clr     Relay2
        Jmp     RelayExit
CekRelay2:
        Cjne    A,#2,CekRelay3
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_2
        Call    LCD_String

        Clr     Relay3
        Jmp     RelayExit
CekRelay3:
        Cjne    A,#3,CekRelay4
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_3
        Call    LCD_String

        Clr     Relay4
        Jmp     RelayExit
CekRelay4:
        Cjne    A,#4,CekRelay5
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_4
        Call    LCD_String

        Clr     Relay5
        Jmp     RelayExit
CekRelay5:
        Cjne    A,#5,CekRelay6
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_5
        Call    LCD_String

        Clr     Relay6
        Jmp     RelayExit
CekRelay6:
        Cjne    A,#6,CekRelay7
        Call    LCD_Line_2
        Mov     DPTR,#txt_Chanel_6
```

```
        Call      LCD_String

        Clr       Relay7
        Jmp       RelayExit
CekRelay7:
        Cjne      A,#7,CekRelay8
        Call      LCD_Line_2
        Mov       DPTR,#txt_Chanel_7
        Call      LCD_String

        Clr       Relay8
        Jmp       RelayExit
CekRelay8:
        Cjne      A,#8,CekRelay9
        Call      LCD_Line_2
        Mov       DPTR,#txt_Chanel_8
        Call      LCD_String

        Clr       Relay9
        Jmp       RelayExit
CekRelay9:
        Cjne      A,#9,CekRelay10
        Call      LCD_Line_2
        Mov       DPTR,#txt_Chanel_9
        Call      LCD_String

        Clr       Relay10
        Jmp       RelayExit
CekRelay10:
        Cjne      A,#0,CekRelay11

        ;Call      LCD_Line_2
        Mov       DPTR,#Txt_Off
        Call      LCD_String
        Clr       RelayOff
        Jmp       RelayExit

RelayExit:
        Call      Delay_Fix_1s
        Mov       P0,#0FFh
        Setb      Relay2
        Setb      RelayOff

CekRelay11:
        Call      Delay_Fix_1s
        Call      Delay_Fix_1s
        Call      LCD_Clear

EndSub

;-------------------------------------------
Menu_RTC_SetJadwal:
        Call      LCD_Clear
        Call      LCD_Line_1
        Mov       DPTR,#txt_hari1
        Call      LCD_String
        Call      LCD_Line_2
        Mov       DPTR,#txt_hari2
        Call      LCD_String

Menu_RTC_SetJadwal_Key1:
        Mov       FlagKey,#1
        Call      Keypad_KeyIn
        Cjne      A,#'1',Menu_RTC_SetJadwal_Key2
        Mov       tmpHari,A
        Jmp       Proses_Menu_RTC_SetJadwal
```

```
Menu_RTC_SetJadwal_Key2:
        Cjne    A,#'2',Menu_RTC_SetJadwal_Key3
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Key3:
        Cjne    A,#'3',Menu_RTC_SetJadwal_Key4
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Key4:
        Cjne    A,#'4',Menu_RTC_SetJadwal_Key5
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Key5:
        Cjne    A,#'5',Menu_RTC_SetJadwal_Key6
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Key6:
        Cjne    A,#'6',Menu_RTC_SetJadwal_Key7
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Key7:
        Cjne    A,#'7',Menu_RTC_SetJadwal_Keyxx
        Mov     tmpHari,A
        Jmp     Proses_Menu_RTC_SetJadwal
Menu_RTC_SetJadwal_Keyxx:
        Cjne    A,#'0',Menu_RTC_SetJadwal_Keyxx
        Mov     tmpHari,A
        Jmp     Menu_Jadwal_Waktu
;----------------
Proses_Menu_RTC_SetJadwal:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_jadwal
        Call    LCD_String
        Call    LCD_Blink_On
Menu_RTC_SetJadwal_Status:
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Cjne    A,#'1',SetJadwalKe2
        Jmp     SetJadwalNow
SetJadwalKe2:
        Cjne    A,#'2',SetJadwalKe3
        Jmp     SetJadwalNow
SetJadwalKe3:
        Cjne    A,#'3',SetJadwalKe4
        Jmp     SetJadwalNow
SetJadwalKe4:
        Cjne    A,#'4',SetJadwalKe5
        Jmp     SetJadwalNow
SetJadwalKe5:
        Cjne    A,#'5',SetJadwalKe6
        Jmp     SetJadwalNow
SetJadwalKe6:
        Cjne    A,#'6',SetJadwalKe7
        Jmp     SetJadwalNow
SetJadwalKe7:

        Cjne    A,#'7',SetJadwalKe8
        Jmp     SetJadwalNow
SetJadwalKe8:
        Cjne    A,#'8',SetJadwalKe9
        Jmp     SetJadwalNow
SetJadwalKe9:
        Cjne    A,#'9',SetJadwalKe10
        Jmp     SetJadwalNow
SetJadwalKe10:
```

```
        Jmp     Proses_Menu_RTC_SetJadwal
;--------------------------
SetJadwalNow:
        Call    Delay_Fix_100ms
        Call    LCD_Blink_Off
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#Txt_SetJadwal3a
        Call    LCD_String

        Call    LCD_Blink_On

        Mov     A,#10h
        Call    LCD_Cursor_Position

        Mov     A,#80h
        Call    LCD_Command

Show_Date_SetJadwal:
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     70h,A

        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     70h,A

        Mov     A,#':'
        Call    LCD_Tulis_Data

        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     71h,A

        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     71h,A

        Mov     A,#8Bh
        Call    LCD_Command

        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     72h,A

        Call    LCD_Blink_Off

        Call    LCD_Line_2
        Mov     DPTR,#Txt_YN
        Call    LCD_String
SetJadwalYakin:
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Cjne    A,#'1',JadwalYakinNo
        Jmp     ProsesJadwalYakin
JadwalYakinNo:
        Cjne    A,#'2',SetJadwalYakin
        Jmp     Menu_Jadwal_Waktu
ProsesJadwalYakin:
        Mov     A,tmpHari
```

```
        Cjne    A,#'1',SimpanNoRelay2
        Mov     DPTR,#EE_minggu1
        Mov     A,tmpJadwal
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay2:
        Cjne    A,#'2',SimpanNoRelay3
        Mov     DPTR,#EE_Senin1
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay3:
        Cjne    A,#'3',SimpanNoRelay4
        Mov     DPTR,#EE_Selasa1
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay4:
        Cjne    A,#'4',SimpanNoRelay5
        Mov     DPTR,#EE_Rabu1
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay5:
        Cjne    A,#'5',SimpanNoRelay6
        Mov     DPTR,#EE_Kamis1
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay6:
        Cjne    A,#'6',SimpanNoRelay7
        Mov     DPTR,#EE_Jumat1
        Jmp     SimpanJadwalNow
;---------------
SimpanNoRelay7:
        Mov     DPTR,#EE_Sabtu1
        Jmp     SimpanJadwalNow
;----------------
SimpanJadwalNow:
        Call    Get_DPTR
        Mov     A,#0FFh             ; id
        Call    EEPROM_Write

        Mov     A,70h                 ; jam
        Call    EEPROM_Write
                   ; men
        Mov     A,71h
        Call    EEPROM_Write

        Mov     A,72h
        Call    EEPROM_Write

        Call    Delay_Fix_1s
        Ljmp     Menu_Jadwal_Waktu


Menu_RTC_UbahJam:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_jam1
        Call    LCD_String
        Call    LCD_Blink_On

        Mov     R0,#10h
        Call    LCD_Cursor_Position

        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     70h,A
```

```
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     R0,#13h
        Call    LCD_Cursor_Position
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     71h,A
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     R0,#16h
        Call    LCD_Cursor_Position
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     72h,A
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Call    Delay_Fix_1s
set_jam_yakin:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_jam2
        Call    LCD_String
        Call    LCD_Line_2
        Mov     DPTR,#txt_jam3
        Call    LCD_String
        Call    LCD_Blink_off
set_jam_yakin1:
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Cjne    A,#'1',set_jam_yakin2
        Jmp     proses_set_jam
set_jam_yakin2:
        Cjne    A,#'2',set_jam_yakin3
        Jmp     Menu_Jadwal_Waktu
set_jam_yakin3:
        Cjne    A,#'3',set_jam_yakin1
        Jmp     Menu_RTC_UbahJam
proses_set_jam:
        Mov     R0,70h
        Mov     R1,71h
        Mov     R2,72h
        Call    RTC_SetTime
        Call    Delay_Fix_100ms
        Call    LCD_Blink_Off
        Jmp     Menu_Jadwal_Waktu
;----------------------------------------
Menu_RTC_UbahTanggal:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_jam4
        Call    LCD_String
        Call    LCD_Blink_On

        Mov     R0,#10h
        Call    LCD_Cursor_Position
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     70h,A

        Mov     R0,#12h
```

```
        Call    LCD_Cursor_Position
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     71h,A
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh

        Mov     R0,#15h
        Call    LCD_Cursor_Position
        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     72h,A
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh

        Mov     R0,#18h
        Call    LCD_Cursor_Position

        Mov     flagkey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Mov     73h,A
        Mov     FlagKey,#0
        Call    Keypad_Keyin
        Anl     A,#0Fh
        Call    Delay_Fix_1s
        Call    LCD_Blink_Off
set_tgl_yakin:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_jam2
        Call    LCD_String
        Call    LCD_Line_2
        Mov     DPTR,#txt_jam3
        Call    LCD_String
set_tgl_yakin1:
        Mov     FlagKey,#1
        Call    Keypad_Keyin
        Cjne    A,#'1',set_tgl_yakin2
        Jmp     proses_set_tgl
set_tgl_yakin2:
        Cjne    A,#'2',set_tgl_yakin3
        Jmp     Menu_Jadwal_Waktu
set_tgl_yakin3:
        Cjne    A,#'3',set_tgl_yakin1
        Jmp     Menu_RTC_UbahTanggal
proses_set_tgl:
        Mov     R0,70h
        Mov     R1,71h
        Mov     R2,72h
        Mov     R3,73h
        Call    RTC_SetDate
        Call    Delay_Fix_100ms
        Call    LCD_Blink_Off
        Jmp     Menu_Jadwal_Waktu
;------------------------------------------
Menu_RTC_HapusJadwal:
        Call    LCD_Clear
        Call    LCD_Line_1
        Mov     DPTR,#txt_hari1
        Call    LCD_String
        Call    LCD_Line_2
```

```
        Mov     DPTR,#txt_hari2
        Call    LCD_String
        Mov     R0,#50
        Call    Delay_Var_10ms
RTC_HapusJadwalKey1:
        Mov     FlagKey,#1
        Call    Keypad_KeyIn
        Cjne    A,#'1',RTC_HapusJadwalKey2
        Mov     70h,#0
HapusMinggu1:
        Mov     A,70h
        Mov     DPTR,#EE_Minggu1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusMinggu1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey2:
        Cjne    A,#'2',RTC_HapusJadwalKey3
HapusSenin1:
        Mov     A,70h
        Mov     DPTR,#EE_Senin1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusSenin1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey3:
        Cjne    A,#'3',RTC_HapusJadwalKey4
HapusSelasa1:
        Mov     A,70h
        Mov     DPTR,#EE_Selasa1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusSelasa1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey4:
        Cjne    A,#'4',RTC_HapusJadwalKey5
HapusRabu1:
        Mov     A,70h
        Mov     DPTR,#EE_Rabu1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusRabu1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey5:
        Cjne    A,#'5',RTC_HapusJadwalKey6
HapusKamis1:
        Mov     A,70h
        Mov     DPTR,#EE_Kamis1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusKamis1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey6:
        Cjne    A,#'6',RTC_HapusJadwalKey7
HapusJumat1:
        Mov     A,70h
```

```
        Mov     DPTR,#EE_Jumat1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusJumat1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey7:
        Cjne    A,#'7',RTC_HapusJadwalKey8
HapusSabtu1:
        Mov     A,70h
        Mov     DPTR,#EE_Sabtu1
        Call    Get_DPTR
        Mov     A,#0
        Call    EEPROM_Write
        Mov     A,70h
        Cjne    A,#7,HapusSabtu1
        Jmp     HapusJadwalOK
RTC_HapusJadwalKey8:
        Cjne    A,#'0',RTC_HapusJadwalKey1ss
        Jmp     Menu_Jadwal_Waktu
RTC_HapusJadwalKey1ss:
        Jmp     RTC_HapusJadwalKey1
HapusJadwalOK:
        Jmp     RTC_HapusJadwalKeyEnd
RTC_HapusJadwalKeyEnd:
        Call    LCD_Clear
        Call    lcd_line_1
        Mov     DPTR,#txt_hapus4
        Call    LCD_String
        Mov     R0,#2
        Call    Delay_Var_1s
        Jmp     Menu_RTC_HapusJadwal
;----------------------------------------
;=======================================
; Menu Sensor
;=======================================

;===========================
Subroutine RTC_Save:

        Mov     DataRTC,A
        Call    TulisDPTRRTC16b
        Mov     A,DataRTC

EndSub
Subroutine RTC_Read:

        Mov     R7,#0D0h
        Mov     B,DPH
        Mov     R6,DPL
        Call    Baca_RTC16b

EndSub
Subroutine Tulis_RTC16b:
        Call    Siapkan16bAlamatRTC
        Jc      Wrong_Write
        Mov     A,DataRTC
        Call    KirimDataRTC
        Jc      Wrong_Write
        Call    Buat_StopBit
        Ret
Wrong_Write:
        Call    Buat_StopBit
        Jmp     Tulis_RTC16b
EndSub
```

```
Subroutine Tulis_RTC8b:
        Call    Siapkan8bAlamatRTC
        Jc      Wrong_Write8b
        Mov     A,DataRTC
        Call    KirimDataRTC
        Jc      Wrong_Write8b
        Call    Buat_StopBit
        Ret
Wrong_Write8b:
        Call    Buat_StopBit
        Jmp     Tulis_RTC8b
EndSub
Subroutine Baca_RTC16b:
        Call    Siapkan16bAlamatRTC
        Jc      Wrong_Read
        Call    Buat_StartBit           ;Kirim Device Address dengan
        Call    ModeBacaRTC
        Jc      Wrong_Read
        Call    BacaDataRTC
        Ret
Wrong_Read:
        Call    Buat_StopBit
        Ajmp    Baca_RTC16b
EndSub
Subroutine Baca_RTC8b:
        Call    Siapkan8bAlamatRTC
        Jc      Wrong_Read8b
        Call    Buat_StartBit           ;Kirim Device Address dengan
        Call    ModeBacaRTC
        Jc      Wrong_Read8b
        Call    BacaDataRTC
        Ret
Wrong_Read8b:
        Call    Buat_StopBit
        Jmp     Baca_RTC8b
EndSub
Subroutine Ambil_Ack:
        Setb    SDA
        Setb    SCL
        Mov     C,SDA
        Clr     SCL
EndSub
Subroutine KirimDeviceAddress:
        Call    Buat_StartBit
        Mov     A,R7                ;Device Address
        Call    KirimDataRTC            ;
EndSub
Subroutine Kirim1WordAddress:
        Mov     A,B                ;First Word Address
        Call    KirimDataRTC            ;
EndSub
Subroutine Kirim2WordAddress:
        Mov     A,R6                ;Second Address
        Call    KirimDataRTC            ;
EndSub
Subroutine BacaDataRTC:
        Mov     B,#08h
        Clr     A
LoopBacaRTC16b:
        Rl      A
        Setb    SDA
        Setb    SCL
        Mov     C,SDA
        Mov     ACC.0,C
        Clr     SCL
        Djnz    B,LoopBacaRTC16b
```

```
        Call    Ambil_Ack
        Call    Ambil_Ack
        Call    Buat_StopBit
EndSub
Subroutine KirimDataRTC:
        Mov     B,#8
Send8_bitloop:
        Rlc     A
        Mov     SDA,C
        Call    Pulse_RTC
        Djnz    B,Send8_bitloop
        Call    Ambil_Ack
EndSub
Subroutine Pulse_RTC:
        Setb    SCL
        Clr     SCL
EndSub
Subroutine Buat_StartBit:
        Setb    SDA
        Setb    SCL
        Clr     SDA
        Clr     SCL
EndSub
Subroutine Buat_StopBit:
        Clr     SDA
        Setb    SCL
        Setb    SDA
EndSub
Subroutine PageRTC8bWrite:
        Call    Siapkan8bAlamatRTC
        Jmp     PageRTCWrite
PageRTC16bWrite:
Loop2PageRTC16bWrite:
        Call    Siapkan16bAlamatRTC
PageRTCWrite:
        Mov     A,DataRTC
        Call    KirimDataRTC
        Jc      WrongPWrite
        Mov     R5,#031
LoopPageRTC16bWrite:
        Mov     A,DataRTC
        Call    KirimDataRTC
        Jc      Loop2PageRTC16bWrite
        Djnz    R5,LoopPageRTC16bWrite
        Call    Buat_StopBit
        Ret
WrongPWrite:
        Call    Buat_StopBit
        Jmp     PageRTC8bWrite
EndSub
Subroutine Siapkan16bAlamatRTC:
        Call    KirimDeviceAddress
        Call    Kirim2WordAddress
SalahTulisAlamat:
        Ret
EndSub
Subroutine ModeBacaRTC:
        Mov     A,R7
        Setb    ACC.0
        Mov     Slave_AddrRTC,A
        Mov     A,Slave_AddrRTC
        Call    KirimDataRTC
EndSub
Subroutine DPTRRTC16bit:
        Mov     R7,#0A0h
        Mov     B,DPH
```

```
        Mov     R6,DPL
        Call    Baca_RTC16b
EndSub
Subroutine TulisDPTRRTC16b:
        Mov     R7,#0D0h
        Mov     B,DPH
        Mov     R6,DPL
        Call    Tulis_RTC16b
EndSub

Subroutine EEPROM_Write:
RAM_Write:
        Mov     01,A

        Mov     A,01
save_03:
        Movx    @DPTR,A
save_01:
        Movx    A,@DPTR
        Cjne    A,01h,save_01

        Mov     A,WMCON
        Clr     ACC.4
        Mov     WMCON,A
EndSub

Subroutine EEPROM_Read:
RAM_Read:
        Call    Set_EEProm
        Movx    A,@DPTR
EndSub

Subroutine Set_EEProm:
        Mov     A,WMCON
        Clr     ACC.0
        Setb    ACC.3
        Setb    ACC.4
        Mov     WMCON,A

EndSub

;=================================================================
=======
;       Clear entire LCD and delay for a bit
;=================================================================
=======

Subroutine LCD_Clear:
        Mov     A,#01h          ;Clear LCD command
        Call    LCD_Command
EndSub

Subroutine LCD_Blink_Off:
        Push    ACC
        Mov     A,#00001100b    ;
        Lcall   LCD_Command
        Pop     ACC
EndSub
Subroutine LCD_Blink_On:
        Push    ACC
        Mov     A,#00001101b    ;
        Lcall   LCD_Command
        Pop     ACC
EndSub
Subroutine lcd_line_1:
        Mov     A,#80h
```

```
        Call      LCD_Command
EndSub
Subroutine LCD_Line_2:
        Mov       A,#0C0h
        Call      LCD_Command
EndSub
Subroutine LCD_Cursor_Position:
        Push      ACC
        Mov       A,R0      ;15       26
        Cjne      A,#10h,lcd_cursor_position1
        Mov       A,R0      ;15
        Call      LCD_Command
        Jmp       lcd_cursor_position_end
lcd_cursor_position1:
        Cjne      A,#20h,lcd_cursor_position_end
        Mov       A,R0                ; 26
        Call      LCD_Command
lcd_cursor_position_end:
        Pop       ACC
EndSub

;==================================================================
=======
;         Initialize LCD module
;==================================================================
=======
Subroutine LCD_Inisialisasi:
        Mov       Data_LCD,#0
        Call      Delay_Fix_100ms
        Mov       Data_LCD,#3

        Call      Delay_Fix_10ms
        Mov       Data_LCD,#3

        Call      Delay_Fix_1ms

        Mov       Data_LCD,#3

        Mov       Data_LCD,#2

        Mov       A,#00101000b     ;Function set, 4 wire, 2 line, 5x7
font
        Call      LCD_Command

        Mov       A,#00001100b     ;Display on,  cursor off, blink off
        Call      LCD_Command

        Mov       A,#00000110b     ;Address increment, no scrolling
        Call      LCD_Command

EndSub

Subroutine LCD_Command:
        Push      B
        Mov       B,A

        Clr       lcdrs
        Clr       lcdrw

        Mov       A,B
        Anl       A,#0Fh
        Mov       Data_LCD,A

        Mov       A,B
        Anl       A,#0Fh     ;Strip off upper bits
        Mov       Data_LCD,A     ;Put on port
```

```
        Mov     A,B     ;Recall character
        Anl     A,#0Fh     ;Strip off upper bits
        Mov     Data_LCD,A     ;Put on port

        Setb    lcdrs     ;Register select set for data

        Pop     B

EndSub

Subroutine LCD_String:
TampilKata1:
        Clr     A
        Movc    A,@A+DPTR
        Cjne    A,#0,TampilKata2                    ;intinya
        Jmp     out
TampilKata2:
        Inc     DPTR
        Call    LCD_Tulis_Data
        Jmp     TampilKata1
out:
EndSub
Subroutine LCD_Hexa:
        Mov     7,A
        Anl     A,#0F0h
        Lcall   tes_huruf_
        Call    LCD_Tulis_Data
        Mov     A,7
        Anl     A,#0Fh
        Lcall   tes_huruf_
        Lcall   LCD_Tulis_Data

        Ret

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

Subroutine delayL:

        Push    0
        Push    1
        Push    2
```

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

        Pop     2
        Pop     1
        Pop     0
EndSub

;=================================
Subroutine Keypad_KeyIn:
        Push    B                       ; amankan register B
KeyInGet1:
        Call    Keypad3x4              ;scan keypad
        Mov     A,keydata               ;isi keydata = data di
rutin keypad
        Cjne    A,#0FFh,KeyInGet0       ;jk isi a # 0ffh ==>lompat
        Jmp     KeyInGet1
KeyInGet0:
        Mov     B,A                     ;simpan isi a to b
KeyInGet:
        Call    Keypad3x4
        Mov     A,keydata
        Cjne    A,B,KeyInOut            ;jk isi A # B lompat
        Jmp     KeyInGet
KeyInOut:
        Mov     A,FlagKey
        Cjne    A,#0,KeyInOut1a
        Mov     A,B
        Call    LCD_Tulis_Data
        Jmp     KeyInOut1
Keyinout1a:

        Cjne    A,#1,KeyInOut1

KeyInOut1:
        Mov     A,B
        Pop     B                       ;idem
EndSub
;===================================
; routine u/ baca keypad 3x4
; output pd keydata(0-9,E=redial,F=#)
;===================================
Subroutine Keypad3x4:

        Mov     keyport,#0FFh
        Clr     kolom1                  ;
;------------
ul1:
        Jb      baris1,key1
        Mov     keydata,#'1'
        Ret
key1:

        Jb      baris2,key2
        Mov     keydata,#'4'
        Ret
key2:
        Jb      baris3,key3
        Mov     keydata,#'7'
```

```
        Ret
key3:
        Jb      baris4,key4
        Mov     keydata,#'*'
        Ret
;--------------
key4:
        Setb    kolom1
        Clr     kolom2
        ;-------------
        Jb      baris1,key5
        Mov     keydata,#'2'
        Ret
key5:
        Jb      baris2,key6
        Mov     keydata,#'5'
        Ret
key6:
        Jb      baris3,key7
        Mov     keydata,#'8'
        Ret
key7:
        Jb      baris4,key8
        Mov     keydata,#'0'
        Ret
;--------------
key8:
        Setb    kolom2
        Clr     kolom3
        ;-----------------
        Jb      baris1,key9
        Mov     keydata,#'3'
        Ret
key9:
        Jb      baris2,key10
        Mov     keydata,#'6'
        Ret
key10:
        Jb      baris3,key11
        Mov     keydata,#'9'
        Ret
key11:
        Jb      baris4,key12
        Mov     keydata,#'#'
        Ret
key12:
        Mov     keydata,#0FFh

EndSub


rtc_buftgl              Equ     0410h
rtc_bufbln              Equ     0411h
rtc_bufthn              Equ     0412h
rtc_bufdet              Equ     0413h
rtc_bufmin              Equ     0414h
rtc_bufjam              Equ     0415h
rtc_bufhar              Equ     0416h
;***************************************
; subrutin cek uip rtc
;***************************************

;******************************************************************
********
; rutin baca rtc (mengambil data dari rtc & menyimpannya di
accumulator)
;******************************************************************
```

```
********
Subroutine RTC_GetDate:

bacahar:
;          Call      RTC_Cek_UIP
;          Call      RTC_Delay0
         Mov       DPTR,#harirtc          ;baca hari rtc
         Call      RTC_Read
         ;Movx     A,@DPTR
bacatgl:
         ;Lcall    RTC_Cek_UIP
         ;Lcall    RTC_Delay0
         Mov       DPTR,#tglrtc
         Call      RTC_Read
         ;Movx     A,@DPTR
         Mov       TmpTgl,A
bacabln:
         ;Lcall    RTC_Cek_UIP
         ;Lcall    RTC_Delay0
         Mov       DPTR,#blnrtc
         Call      RTC_Read
         ;Movx     A,@DPTR
         Mov       TmpBln,A

bacathn:
;          Lcall     RTC_Cek_UIP
;          Lcall     RTC_Delay0
         Mov       DPTR,#thnrtc
         Call      RTC_Read
         ;Movx     A,@DPTR
         Mov       TmpThn,A
EndSub
Subroutine RTC_Delay0:
         Push      03
         Push      02
         Push      01

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

bacartc:
bacadet:
;          Lcall     RTC_Cek_UIP
;          Lcall     RTC_Delay0
         Mov       DPTR,#detrtc          ;baca detik rtc
         Call      RTC_Read
;          Movx      A,@DPTR
bacamnt:
         ;Lcall    RTC_Cek_UIP
         ;Lcall    RTC_Delay0
         Mov       DPTR,#mntrtc
```

```
        Call      RTC_Read
        ;Movx     A,@DPTR
bacajam:
        ;Lcall    RTC_Cek_UIP
        ;Lcall    RTC_Delay0
        Mov       DPTR,#jamrtc
        Call      RTC_Read
        ;Movx     A,@DPTR
        Ret
EndSub
;****************************************
; subrutin pindah new setting tanggal
;****************************************
Subroutine RTC_SetDate:

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

        Mov       DPTR,#harirtc       ; day
        Call      RTC_Save
        ;Movx      @DPTR,A

        Mov       DPTR,#tglrtc        ; date
        Call      RTC_Save
        ;Movx      @DPTR,A

        Mov       DPTR,#blnrtc        ; month
        Call       RTC_Save
        ;movx      @DPTR,A

        Mov       DPTR,#thnrtc        ; year
        Call      RTC_Save
;         Movx     @DPTR,A

;         Movx     @DPTR,A

        Ret
EndSub
;****************************************
; subrutin pindah new setting jam
;****************************************
Subroutine RTC_SetTime:

        Mov       DPTR,#detrtc
        Call      RTC_Save
;         Movx     @DPTR,A

        Mov       DPTR,#mntrtc
        Call      RTC_Save
;         Movx     @DPTR,A

        Mov       DPTR,#jamrtc
        Call      RTC_Save
;         Movx     @DPTR,A


        Ret
EndSub
;****************************************
; subrutin tampil jam:menit:detik lcd1
;****************************************
Subroutine RTC_ShowTime:
rtc_sacajam1:
        Mov       DPTR,#rtc_bufjam
        Call      RAM_Read
        Mov       06h,A
        Anl       A,#0F0h
```

```
        Add     A,#30h
        Call    LCD_Tulis_Data
        Mov     A,06h
        Anl     A,#0Fh
        Call    LCD_Tulis_Data
        Mov     A,#':'
        Call    LCD_Tulis_Data
rtc_sacamnt1:
        Mov     DPTR,#rtc_bufmin
        Call    RAM_Read
        Mov     06h,A
        Anl     A,#0F0h
        Add     A,#30h
        Call    LCD_Tulis_Data
        Mov     A,06h
        Anl     A,#0Fh
        Call    lcd_tulis_data
        Mov     A,#':'
        Call    LCD_Tulis_Data
rtc_sacadet1:
        Mov     DPTR,#rtc_bufdet
        Call    RAM_Read
        Mov     06h,A
        Anl     A,#0F0h
        Add     A,#30h
        Call    LCD_Tulis_Data
        Mov     A,06h
        Anl     A,#0Fh
        Call    LCD_Tulis_Data
        Ret
EndSub
;****************************************
; subrutin tampil tgl-bln-thn hari lcd1
;****************************************
Subroutine RTC_ShowDate:
sacaday1:
        Mov     DPTR,#rtc_bufhar
        Call    RAM_Read
        Cjne    A,#01h,opsen1
        Mov     DPTR,#txt_minggu
        Call    LCD_String
        Ljmp    sacatgl1
opsen1:
        Cjne    A,#02h,opsel1
        Mov     DPTR,#txt_senin
        Call    LCD_String
        Ljmp    sacatgl1
opsel1:
        Cjne    A,#03h,oprab1
        Mov     DPTR,#txt_selasa
        Call    LCD_String
        Ljmp    sacatgl1
oprab1:
        Cjne    A,#04h,opkam1
        Mov     DPTR,#txt_rabu
        Call    LCD_String
        Ljmp    sacatgl1
opkam1:
        Cjne    A,#05h,opjum1
        Mov     DPTR,#txt_kamis
        Call    LCD_String
        Ljmp    sacatgl1
opjum1:
        Cjne    A,#06h,opsab1
        Mov     DPTR,#txt_jumat
        Call    LCD_String
        Ljmp    sacatgl1
```

```
opsabl:
        Cjne    A,#07h,opsabx1
        Mov     DPTR,#txt_sabtu
        Call    LCD_String
        Ljmp    sacatgl1
opsabx1:

sacatgl1:
        Mov     A,#','
        Call    lcd_tulis_data

        Mov     DPTR,#rtc_buftgl
        Call    RAM_Read
        Mov     06h,A
        Anl     A,#0F0h
        Add     A,#30h
        Call    lcd_tulis_data
        Mov     A,06h
        Anl     A,#0Fh
        Call    LCD_Tulis_Data
        Mov     A,#'-'
        Call    lcd_tulis_data

        Mov     DPTR,#rtc_bufbln
        Call    RAM_Read
        Mov     06h,A
        Anl     A,#0F0h
        Add     A,#30h
        Call    lcd_tulis_data
        Mov     A,06h
        Anl     A,#0Fh
        Call    LCD_Tulis_Data
        Mov     A,#'-'
        Call    lcd_tulis_data
        Mov     A,#"'"
        Call    lcd_tulis_data

sacathnl:
        Mov     DPTR,#rtc_bufthn
        Call    RAM_Read
        Mov     06h,A
        Anl     A,#0F0h
        Add     A,#30h
        Call    LCD_Tulis_Data
        Mov     A,06h
        Anl     A,#0Fh
        Call    LCD_Tulis_Data

        Ret

EndSub

;****************************************
; hari-hari
;****************************************
txt_minggu:     Db      'Min',0
txt_senin:      Db      'Sen',0
txt_selasa:     Db      'Sel',0
txt_rabu:       Db      'Rab',0
txt_kamis:      Db      'Kam',0
txt_jumat:      Db      'Jum',0
txt_sabtu:      Db      'Sab',0

;================ DELAY ==============================
Subroutine Delay_Var_1ms:
```

```
        Call     Delay_Fix_1ms
        Djnz     R0,Delay_Var_1ms
EndSub
Subroutine Delay_Var_10ms:
        Call     Delay_Fix_10ms
        Djnz     R0,Delay_Var_10ms
EndSub
Subroutine Delay_Var_100ms:
        Call     Delay_Fix_1ms
        Djnz     R0,Delay_Var_100ms
EndSub
Subroutine Delay_Var_1s:
        Call     Delay_Fix_1s
        Djnz     R0,Delay_Var_1s
EndSub
Subroutine Delay_Var_10s:
        Call     Delay_Fix_10s
        Djnz     R0,Delay_Var_10s
EndSub
Subroutine Delay_Var_10us:
        Call     Delay_Fix_10us
        Djnz     R0,Delay_Var_10us
EndSub
Subroutine Delay_Fix_10us:
        Push     1
        Mov      1,#20
        Djnz     1,$
        Pop      1
EndSub
Subroutine Delay_Fix_10s:
        Push     1
        Mov      1,#100
delay_fix_10s_1:
        Call     Delay_Fix_100ms
        Djnz     1,delay_fix_10s_1
        Pop      1
EndSub
Subroutine Delay_Fix_1s:
        Push     1
        Mov      1,#100
delay_fix_1000ms_1:
        Call     Delay_Fix_10ms
        Djnz     1,delay_fix_1000ms_1
        Pop      1
EndSub
Subroutine Delay_Fix_100ms:
        Push     1
        Mov      1,#10
delay_fix_100ms_1:
        Call     Delay_Fix_10ms
        Djnz     1,delay_fix_100ms_1
        Pop      1
EndSub
Subroutine Delay_Fix_10ms:
        Mov      TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
        Mov      TL0,#3Dh    ; siapkan waktu tunda 50 mili-detik
        Mov      TH0,#0B0h
        Clr      TF0                ; me-nol-kan bit limpahan
        Setb     TR0                ; timer mulai bekerja
        Jnb      TF0,$              ; tunggu di sini sampai melimpah
        Clr      TR0                ; timer berhenti kerja
        Ret
EndSub
Subroutine Delay_Fix_1ms:
        Mov      TMOD,#00000001b    ; Timer 1 bekerja pada mode 1
        Mov      TL0,#0EDh   ; siapkan waktu tunda 50 mili-detik
        Mov      TH0,#78h
```

```
        Clr     TF0                 ; me-nol-kan bit limpahan
        Setb    TR0                 ; timer mulai bekerja
        Jnb     TF0,$               ; tunggu di sini sampai melimpah
        Clr     TR0                 ; timer berhenti kerja
        Ret
EndSub
Subroutine Get_DPTR:
        Push    B
       ; Mov     DPTR,#suhu0
        Mov     B,#4               ; jumlah antar lokasi nama
variabel = 12
        Mul     AB                  ; kalikan a dengan b
        Add     A,DPL               ; jumlahkan a dengan dpl
        Mov     DPL,A               ; simpan ke dpl
        Mov     A,B                 ; ambil b
        Addc    A,DPH
        Mov     DPH,A
        Pop     B
        ;Ret
EndSub
Txt_RTC1:           Db      '1.Jadwal',0
Txt_RTC3:           Db      '2.Rubah Jam',0
Txt_RTC4:           Db      '3.Rubah Tanggal',0
Txt_RTC5:           Db      '4.Hapus Jadwal',0
Txt_RTC6:           Db      '6.On/Off Relay',0
txt_RTC7:           Db      'Pilih menu:',0
Txt_RTC8:           Db      '1-2-3-4-5-6',0
Txt_Pilih:          Db      'Pilih',0
Txt_Pilih1:         Db      '1-2-3-4-0',0
Txt_Info:           Db      'Back to Menu',0
Txt_Info1:          Db      'Pres Tombol #',0

;                            0123456789abcdef
txt_hari1:          Db      '1-M 2-S 3-S 4-R',0
txt_hari2:          Db      '   5-K 6-J 7-S',0
txt_jadwal:         Db      'Jadwal:',0

Txt_SetJadwal1a: Db          ' Hapus Channel',0
Txt_SetJadwal1b: Db          '   1-2-3-4-5-6',0
Txt_SetJadwal2a: Db          '1-On/2-Off',0
Txt_SetJadwal2b: Db          'Status:',0
Txt_SetJadwal3:  Db          'Tg-Bl-Th',0
Txt_SetJadwal3a: Db          'jj:mm  ch>',0
Txt_SetJadwal3b: Db          'Channel:',0

;                            0123456789abcdef
Txt_SetJadwal4: Db          '      Simpan?',0
Txt_SetJadwal5: Db          '   1-Y/2-T/3-E',0
Txt_SetJadwal6: Db          ' Data disimpan!',0

txt_jam1:           Db      'jj:mm:dd',0
txt_jam2:           Db      '      Yakin?',0
txt_jam3:           Db      '   1-Y/2-T/3-E',0
txt_jam4:           Db      'h tg-bb-th',0

;                            0123456789abcdef
txt_hapus1:         Db      'Masukkan Data:',0
txt_hapus2:         Db      'Tg-Bl-Th',0
txt_hapus3:         Db      'Data Tidak Ada!',0
txt_hapus4:         Db      ' Data dihapus!',0
;----------------------------------------------------
;                            0123456789abcdef
Txt_On:             Db      'On',0
Txt_Off:            Db      ' Off',0
Txt_YN:             Db      '(1-Y/2-T)?',0
Txt_NoJadwal:       Db      'Jadwal tidak ada',0
```

```
Txt_Found1:        Db      'Channel:',0
;                           0123456789abcdef
Txt_info_new:      Db      ' Remote Kontrol',0
Txt_info_new1:     Db      ' Timer Otomatis',0
Txt_info_new2:     Db      ' Pada Televisi',0
Txt_info_new3:     Db      ' ATMEL 89S8252',0
Txt_info_new4:     Db      'Nur Suryaningrat',0
Txt_info_new5:     Db      '     0117155',0
Txt_info_new6:     Db      '       By',0

txt_Chanel_1:      Db      '     RCTI',0
txt_Chanel_2:      Db      '     SCTV',0
txt_Chanel_3:      Db      '    INDOSIAR',0
txt_Chanel_4:      Db      '     TPI',0
txt_Chanel_5:      Db      '    METROTV',0
txt_Chanel_6:      Db      '     ANTV',0
txt_Chanel_7:      Db      '     TV7',0
txt_Chanel_8:      Db      '    LATIV',0
txt_Chanel_9:      Db      '    TRANSTV',0
```

ıtures

- mpatible with MCS®51 Products
- Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- to 6V Operating Range
- ly Static Operation: 0 Hz to 24 MHz
- ee-level Program Memory Lock
- ; x 8-bit Internal RAM
- Programmable I/O Lines
- ee 16-bit Timer/Counters
- e Interrupt Sources
- grammable UART Serial Channel
- I Serial Interface
- v-power Idle and Power-down Modes
- errupt Recovery from Power-down
- grammable Watchdog Timer
- al Data Pointer
- wer-off Flag

8-bit Microcontroller with 8K Bytes Flash

AT89S8252

scription

AT89S8252 is a low-power, high-performance CMOS 8-bit microcontroller with 8K
; of downloadable Flash programmable and erasable read-only memory and 2K
; of EEPROM. The device is manufactured using Atmel's high-density nonvolatile
ıory technology and is compatible with the industry-standard 80C51 instruction
ınd pinout. The on-chip downloadable Flash allows the program memory to be
ıgrammed In-System through an SPI serial interface or by a conventional nonvol-
memory programmer. By combining a versatile 8-bit CPU with downloadable
ı on a monolithic chip, the Atmel AT89S8252 is a powerful microcontroller, which
ides a highly-flexible and cost-effective solution to many embedded control
cations.

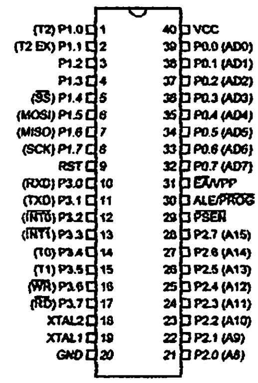
AT89S8252 provides the following standard features: 8K bytes of downloadable
ı, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog
r, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt
itecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition,
AT89S8252 is designed with static logic for operation down to zero frequency and
orts two software selectable power saving modes. The Idle Mode stops the CPU
allowing the RAM, timer/counters, serial port, and interrupt system to continue
ioning. The Power-down mode saves the RAM contents but freezes the oscillator,
oling all other chip functions until the next external interrupt or hardware reset.

downloadable Flash can be changed a single byte at a time and is accessible
ıgh the SPI serial interface. Holding RESET active forces the SPI bus into a serial
ramming interface and allows the program memory to be written to or read from
ss lock bits have been activated.

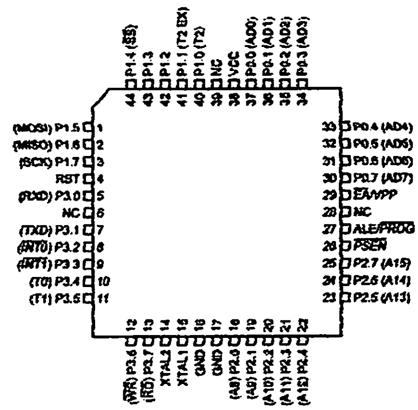
0401F–MICRO–11/03

Configurations

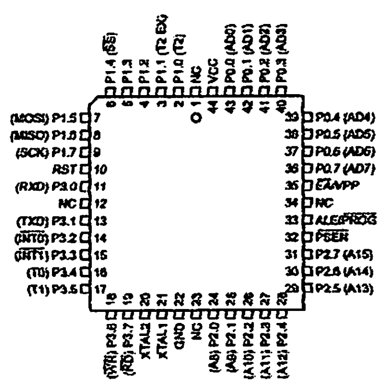
PDIP



TQFP



PLCC



Description

)

Supply voltage.

)

Ground.

0

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

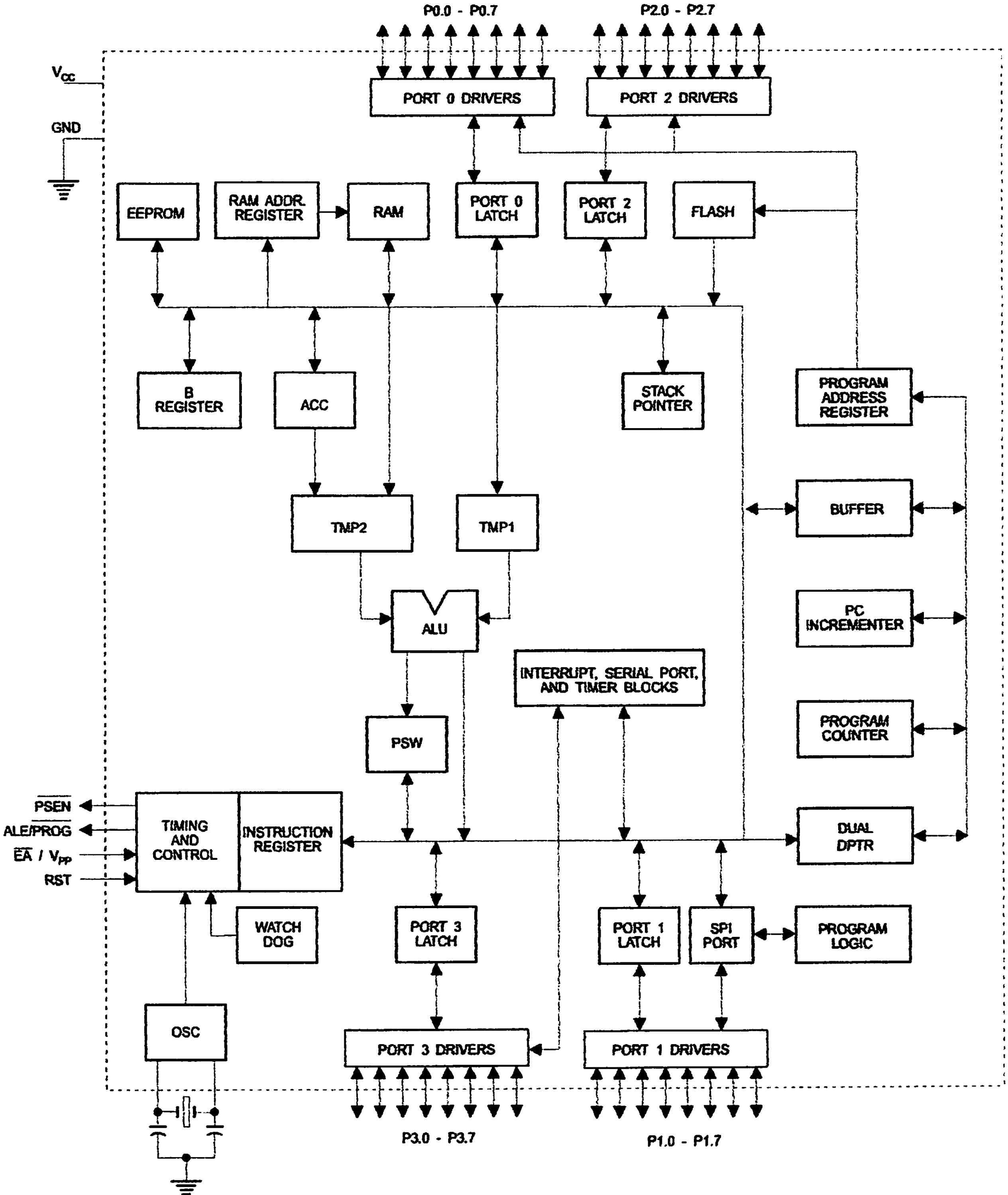
Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

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

t 1

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

ck Diagram



P0.0 - P0.7
P2.0 - P2.7
VCC
GND
PORT 0 DRIVERS
PORT 2 DRIVERS
EEPROM
RAM ADDR. REGISTER
RAM
PORT 0 LATCH
PORT 2 LATCH
FLASH
B REGISTER
ACC
STACK POINTER
PROGRAM ADDRESS REGISTER
TMP2
TMP1
BUFFER
ALU
PC INCREMENTER
INTERRUPT, SERIAL PORT, AND TIMER BLOCKS
PSW
PROGRAM COUNTER
PSEN
ALE/PROG
EA / VPP
RST
TIMING AND CONTROL
INSTRUCTION REGISTER
DUAL DPTR
WATCH DOG
PORT 3 LATCH
PORT 1 LATCH
SPI PORT
PROGRAM LOGIC
OSC
PORT 3 DRIVERS
PORT 1 DRIVERS
P3.0 - P3.7
P1.0 - P1.7

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

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

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

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

t 2

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

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

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

t 3

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

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

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

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

.

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

:/$\overline{PROG}$

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

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

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

:N

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

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

VPP

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

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

L1

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

L2

Output from the inverting oscillator amplifier.

cial Function isters

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

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

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

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

1. AT89S8252 SFR Map and Reset Values

H									0FFH
H	B 00000000								0F7H
H									0EFH
H	ACC 00000000								0E7H
H									0DF H
H	PSW 00000000					SPCR 000001XX			0D7H
H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CF H
H									0C7H
H	IP XX000000								0BFH
H	P3 11111111								0B7H
H	IE 0X000000		SPSR 00XXXXXX						0AFH
H	P2 11111111								0A7H
H	SCON 00000000	SBUF XXXXXXXX							9FH
H	P1 11111111						WMCON 00000010		97H
H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H

ı 2. T2CON – Timer/Counter 2 Control Register

ON Address = 0C8H							Reset Value = 0000 0000B
ıddressable							
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
7	6	5	4	3	2	1	0

ıbol	Function
	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
.K	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
K	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
:N2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
:	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
$\overline{2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

:hdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in › 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The bit selects one of two DPTR registers available.

› 3. WMCON—Watchdog and Memory Control Register

CON Address = 96H Reset Value = 0000 0010B

PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
7	6	5	4	3	2	1	0

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

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

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

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

Power Off Flag The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

ı 4. SPCR – SPI Control Register

:R Address = D5H Reset Value = 0000 01XXB

SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
7	6	5	4	3	2	1	0

ıbol	Function
Ξ	SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
:	SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
ᴙD	Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
ΓR	Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode.
ƆL	Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control.
ɨA	Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control.
ʀ0 ʀ1	SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{OSC.}$, is as follows: SPR1 SPR0 SCK = $F_{OSC.}$ divided by 0 0 4 0 1 16 1 0 64 1 1 128

5. SPSR – SPI Status Register

R Address = AAH Reset Value = 00XX XXXXB

SPIF	WCOL	–	–	–	–	–	–
7	6	5	4	3	2	1	0

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

6. SPDR – SPI Data Register

R Address = 86H Reset Value = unchanged

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

a Memory – PROM and RAM

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

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

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

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

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

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

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

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

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

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

grammable chdog Timer

The programmable Watchdog Timer (WDT) operates from an independent internal oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at V_{CC} = 5V) are within ±30% of the nominal.

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

Table 7. Watchdog Timer Period Selection

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

er 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the Atmel web site (http://www.atmel.com). From the home page, select "Products", then "Microcontrollers, then "8051-Architecture". Click on "Documentation", then on "Other Documents". Open the document "AT89 Series Hardware Description".

er 2

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

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

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

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

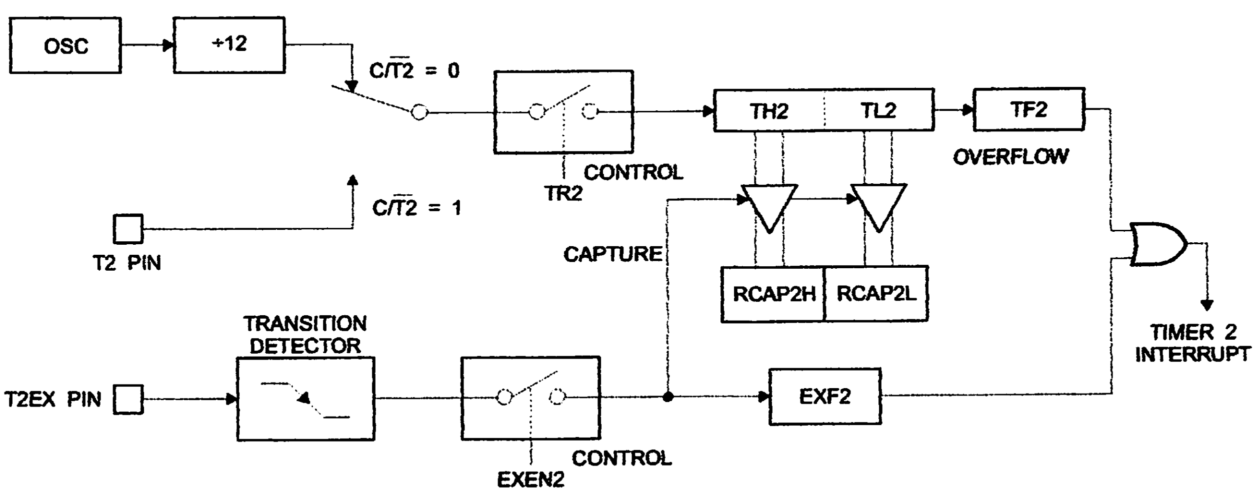
Table 8. Timer 2 Operating Modes

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

ture Mode

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

re 1. Timer 2 in Capture Mode



›-reload (Up or Down nter)

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

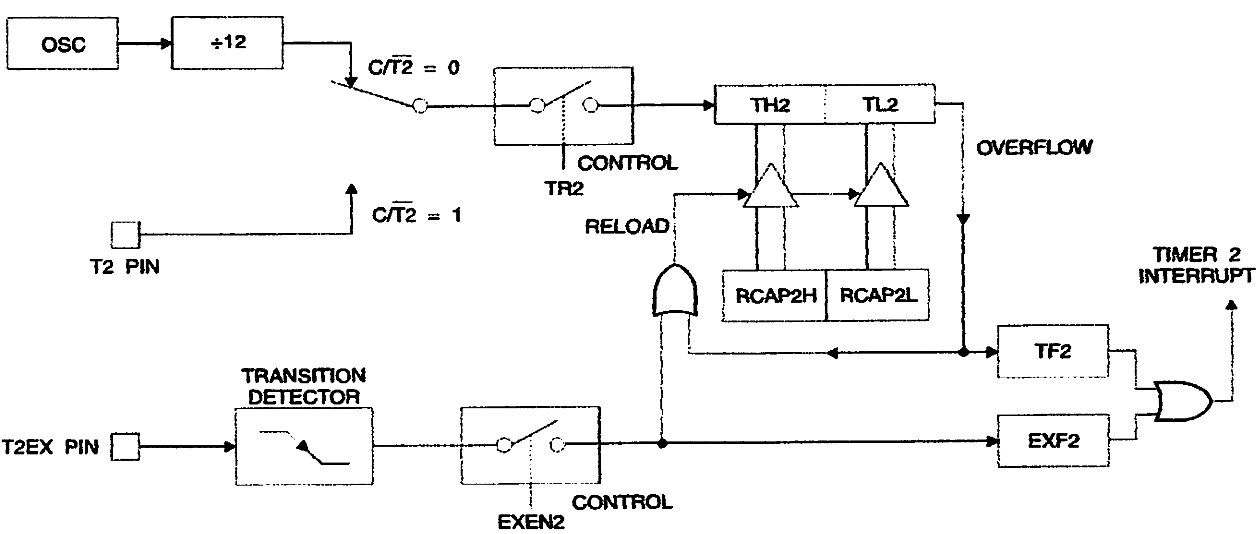
Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

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

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

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

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



9. T2MOD – Timer 2 Mode Control Register

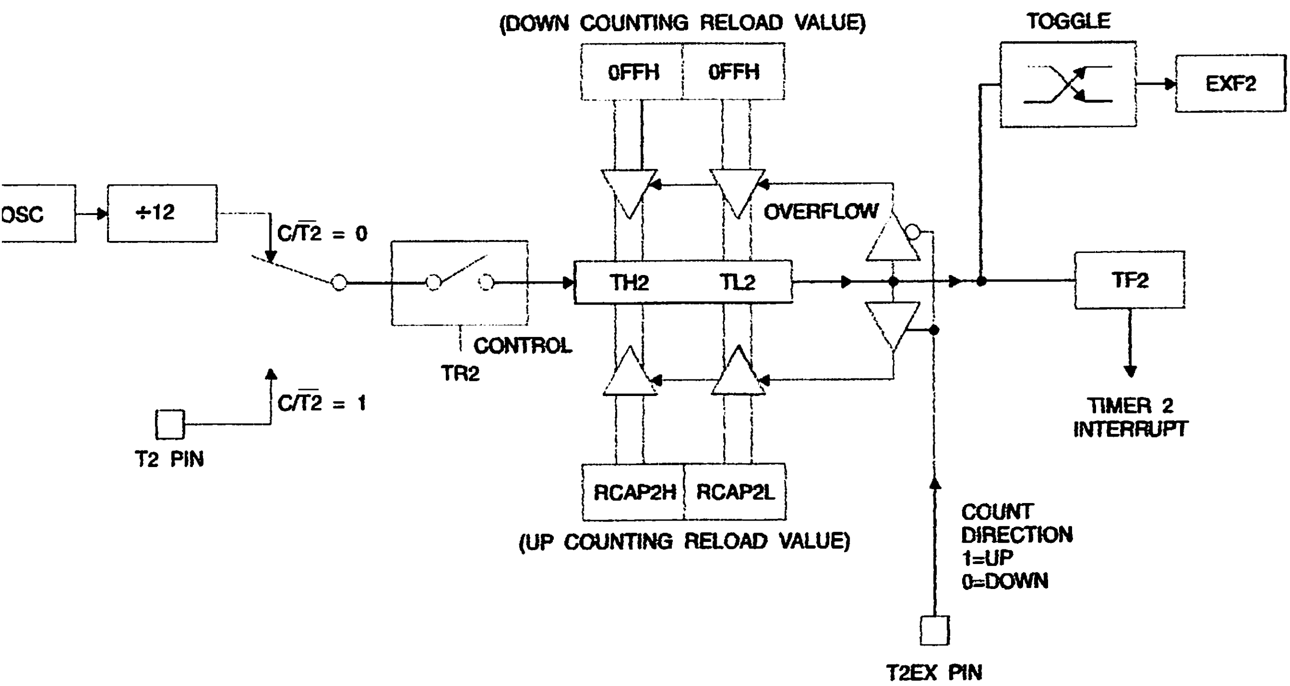
OD Address = 0C9H Reset Value = XXXX XX00B

Bit Addressable

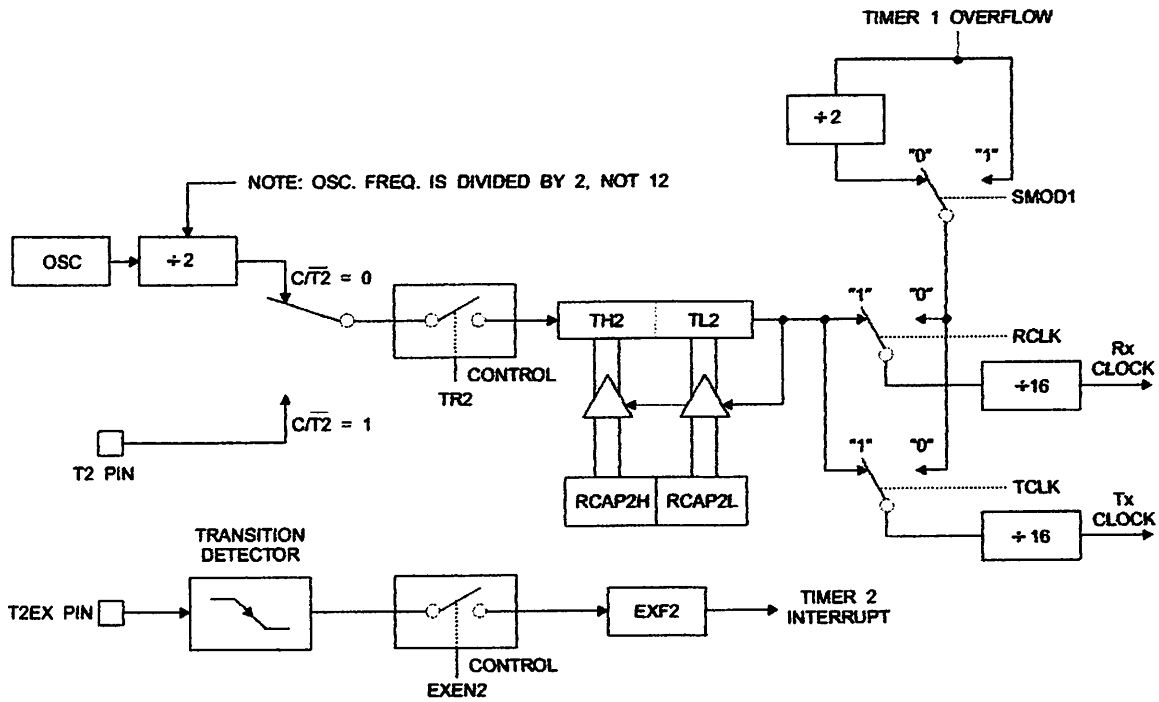
–	–	–	–	–	–	T2OE	DCEN
7	6	5	4	3	2	1	0

bol	Function
	Not implemented, reserved for future use.
E	Timer 2 Output Enable bit.
N	When set, this bit allows Timer 2 to be configured as an up/down counter.

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



e 4. Timer 2 in Baud Rate Generator Mode



ıd Rate Generator

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

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

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

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

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

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

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

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

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

grammable ck Out

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

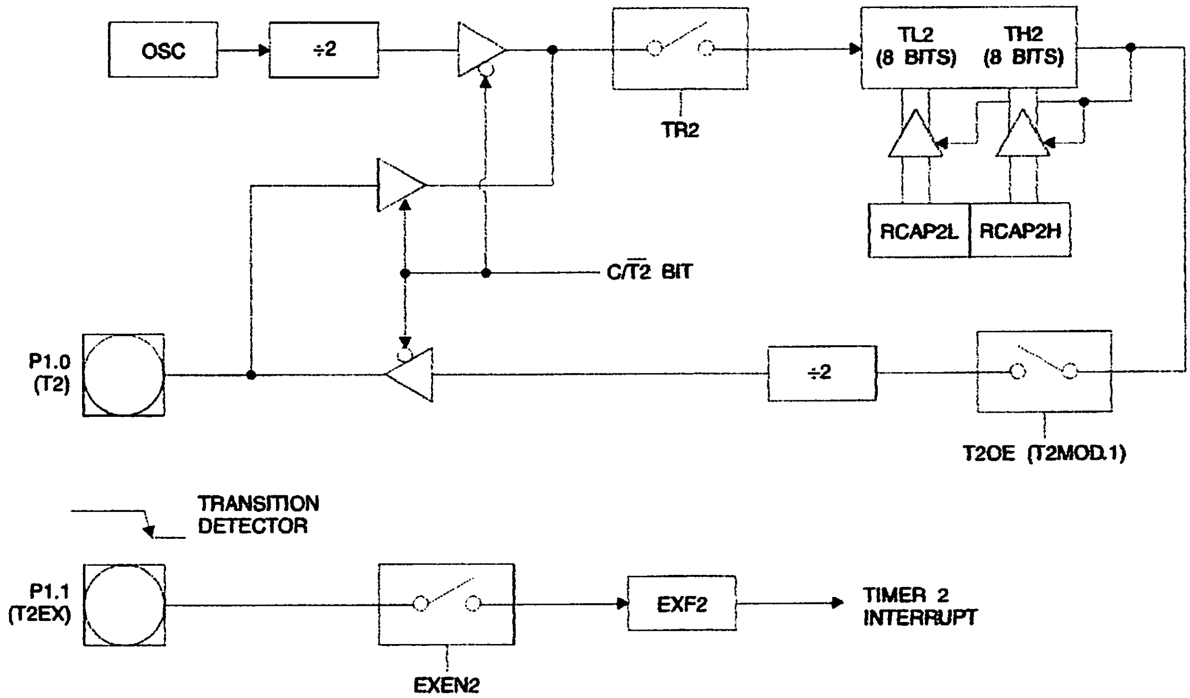
To configure the Timer/Counter 2 as a clock generator, bit $C/\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

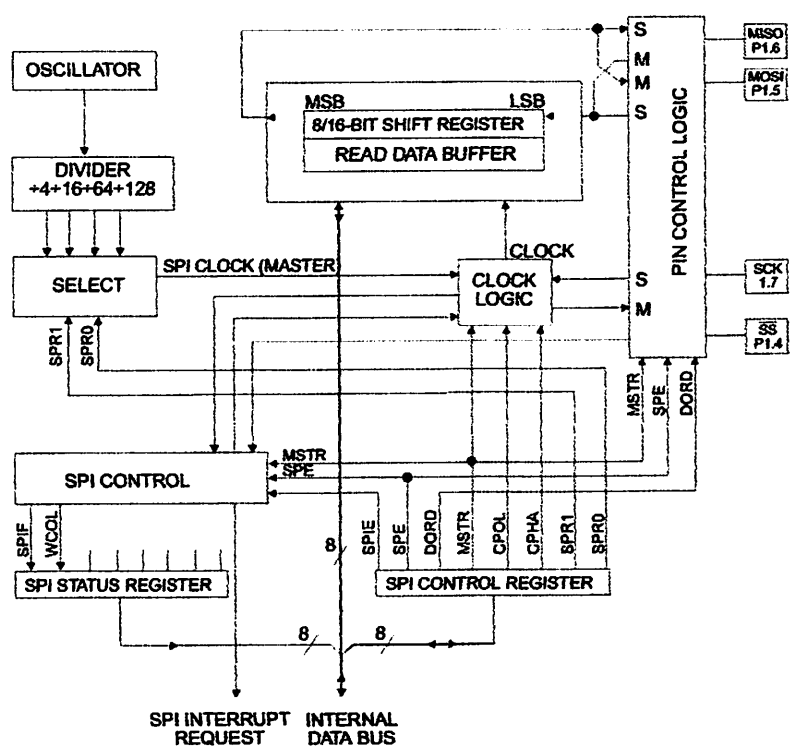
$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

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

'e 5. Timer 2 in Clock-out Mode



re 6. SPI Block Diagram



RT

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

ial Peripheral rface

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

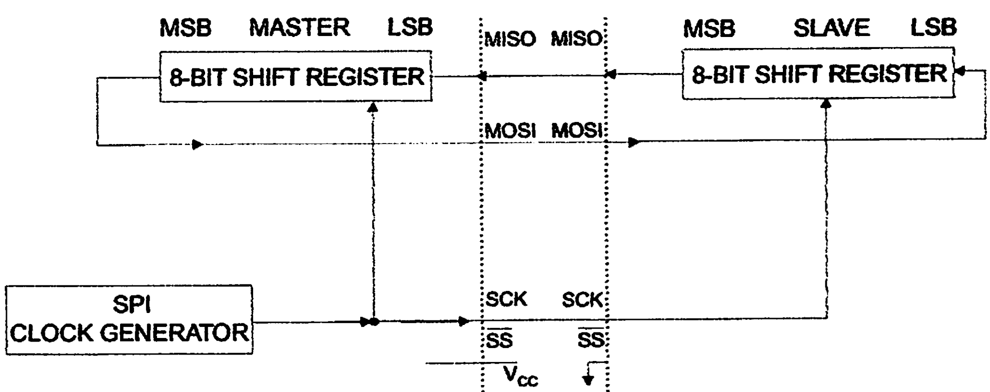
- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

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

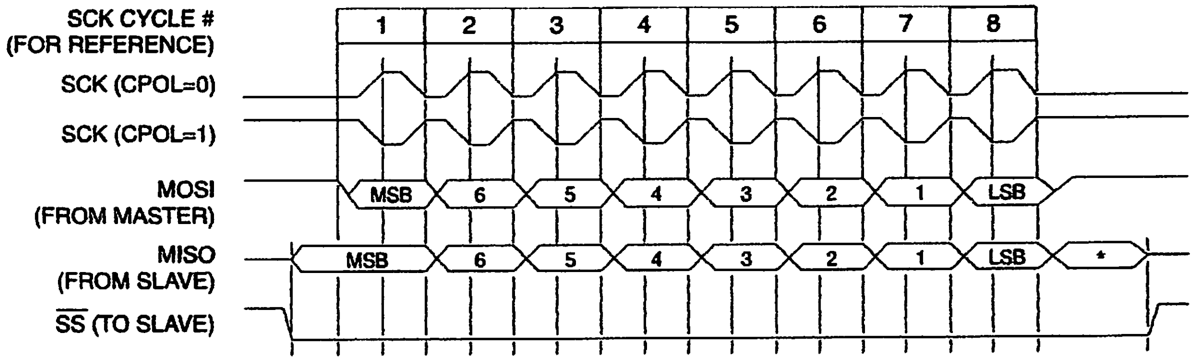
The Slave Select input, $\overline{SS}$/P1.4, is set low to select an individual SPI device as a slave. When $\overline{SS}$/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

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

re 7. SPI Master-slave Interconnection

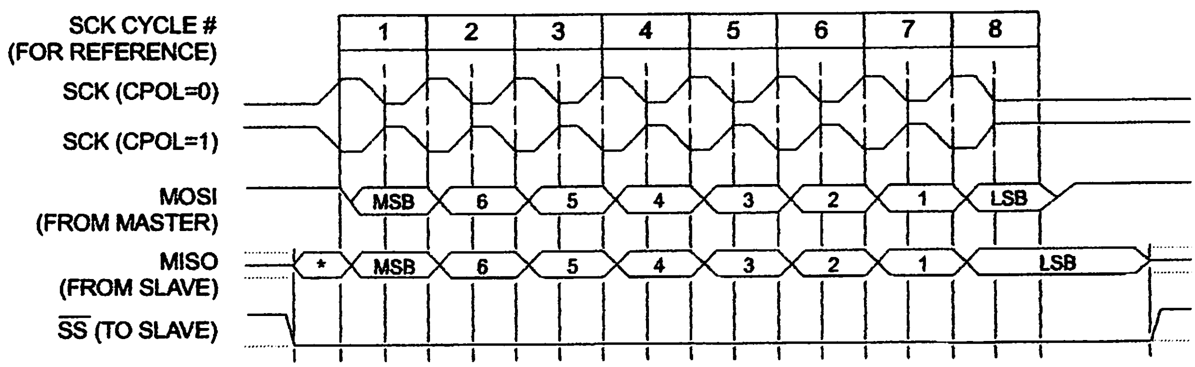


e 8. SPI transfer Format with CPHA = 0



*Not defined but normally MSB of character just received

e 9. SPI Transfer Format with CPHA = 1



*Not defined but normally LSB of previously transmitted character.

rrupts

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

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

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

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

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

10. Interrupt Enable (IE) Register

3)(LSB)

EA	–	ET2	ES	ET1	EX1	ET0	EX0

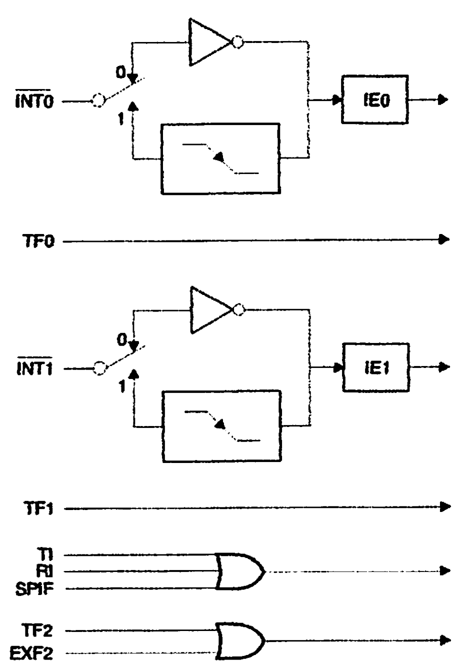
Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
–	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

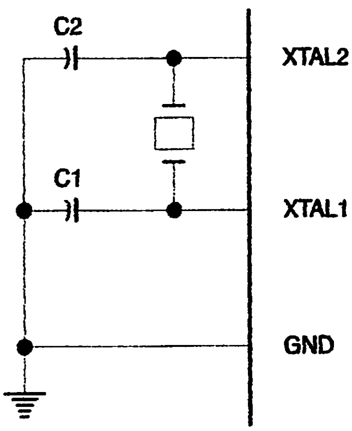
e 10. Interrupt Sources



illator racteristics

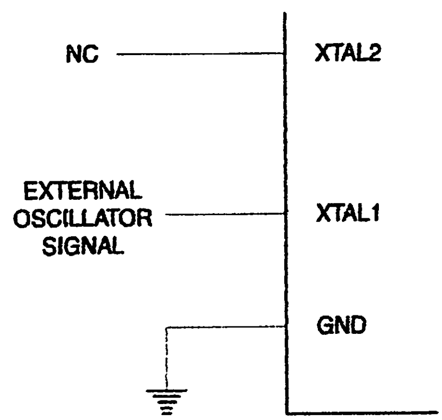
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 11. Oscillator Connections



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

Figure 12. External Clock Drive Configuration



Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Status of External Pins During Idle and Power-down Modes

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

ver-down Mode

In the power-down mode, the oscillator is stopped and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. Exit from power-down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

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

gram Memory
k Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

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

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

k Bit Protection Modes(1)(2)

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

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

gramming the ;h and EEPROM

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The AT89S8252 is shipped with the Serial Programming Mode enabled.

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

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

gramming rface

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

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

ial Downloading

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

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

The Code and Data memory arrays have separate address spaces:

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

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

al Programming ›rithm

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

1. Power-up sequence:

   Apply power between VCC and GND pins.

   Set RST pin to "H".

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

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

   Set RST to "L".

   Turn V_{CC} power off.

ıl Programming uction

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

ıction Set

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

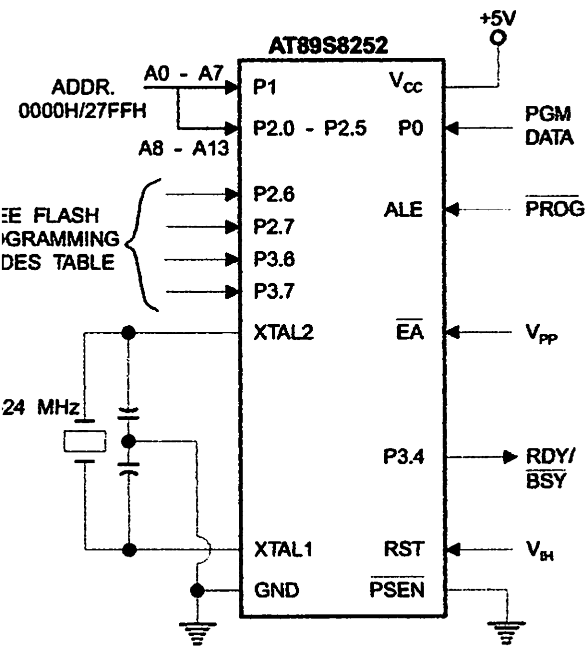
1. DATA polling is used to indicate the end of a byte write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.

;h and EEPROM Parallel Programming Modes

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

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

e 13. Programming the Flash/EEPROM Memory



e 14. Verifying the Flash/EEPROM Memory

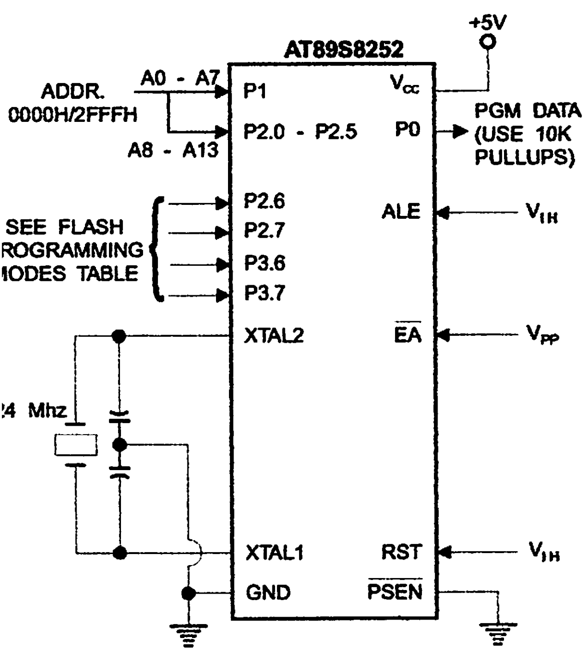
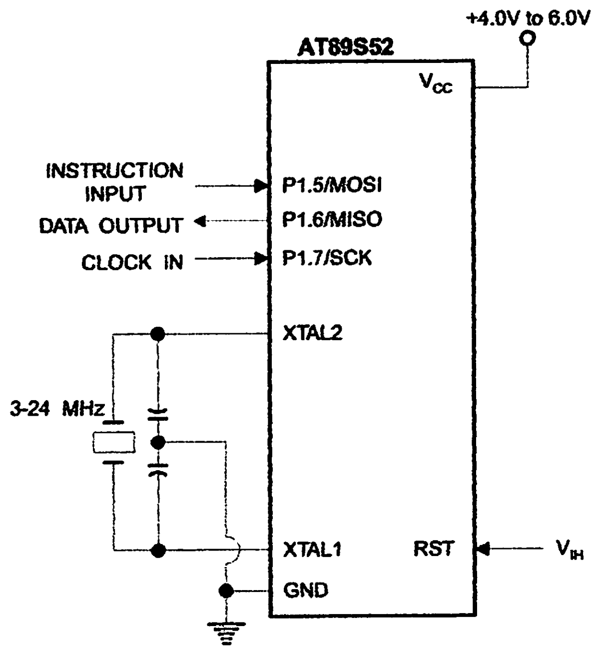


Figure 15. Flash/EEPROM Serial Downloading

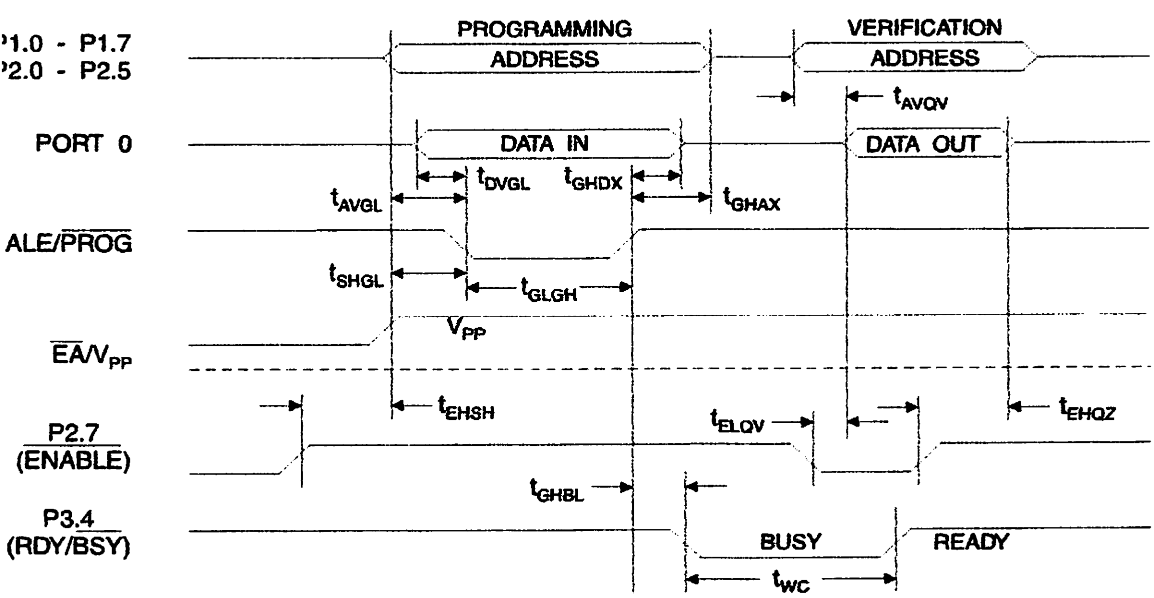


sh Programming and Verification Characteristics – Parallel Mode

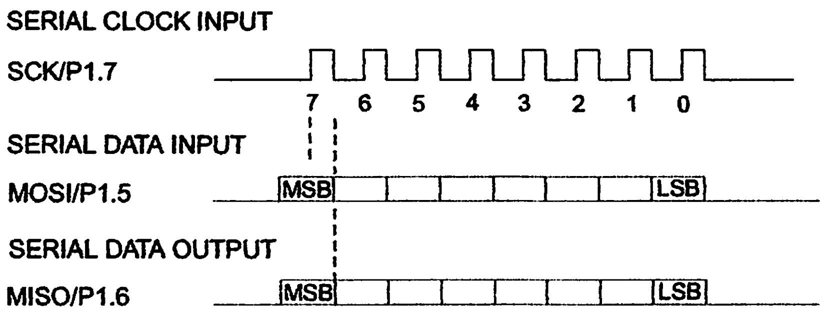
)°C to 70°C, V_{CC} = 5.0V ± 10%

ıbol	Parameter	Min	Max	Units
	Programming Enable Voltage	11.5	12.5	V
	Programming Enable Current		1.0	mA
CL	Oscillator Frequency	3	24	MHz
-	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
X	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
-	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
X	Data Hold after $\overline{PROG}$	$48t_{CLCL}$		
H	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
L	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
H	$\overline{PROG}$ Width	1	110	µs
/	Address to Data Valid		$48t_{CLCL}$	
/	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
Z	Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
L	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
	Byte Write Cycle Time		2.0	ms

sh/EEPROM Programming and Verification Waveforms – Parallel Mode

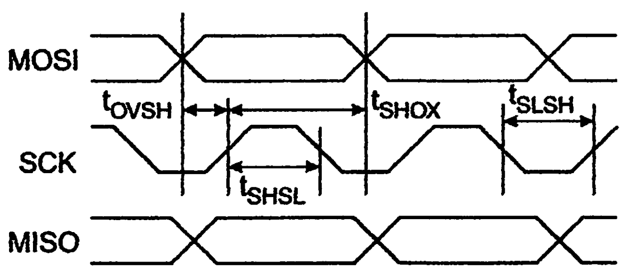


ial Downloading Waveforms



al Programming Characteristics

re 16. Serial Programming Timing



; 11. Serial Programming Characteristics, T_A = -40°C to 85°C, V_{CC} = 4.0 - 6.0V (Unless Otherwise Noted)

ıbol	Parameter	Min	Typ	Max	Units
LCL	Oscillator Frequency	0		24	MHz
L	Oscillator Period	41.6			ns
L	SCK Pulse Width High	24 t_{CLCL}			ns
H	SCK Pulse Width Low	24 t_{CLCL}			ns
H	MOSI Setup to SCK High	t_{CLCL}			ns
X	MOSI Hold after SCK High	2 t_{CLCL}			ns

solute Maximum Ratings*

rating Temperature	-55°C to +125°C
age Temperature	-65°C to +150°C
age on Any Pin Respect to Ground	-1.0V to +7.0V
imum Operating Voltage	6.6V
Output Current	15.0 mA

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

Characteristics

alues shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0V ± 20%, unless otherwise noted.

bol	Parameter	Condition	Min	Max	Units
	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
	Output Low-voltage [1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.5	V
	Output Low-voltage [1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.5	V
	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
T	Reset Pull-down Resistor		50	300	KΩ
	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode [2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

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

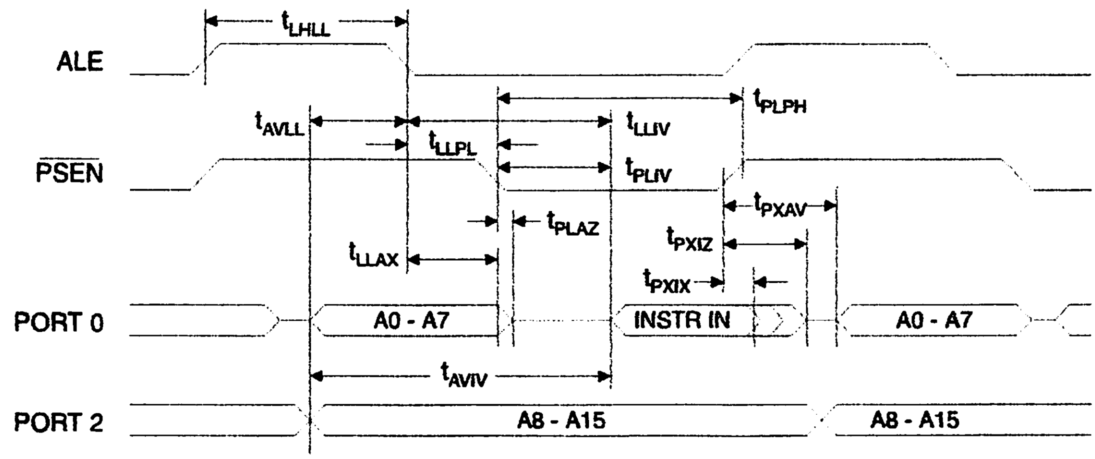
Characteristics

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

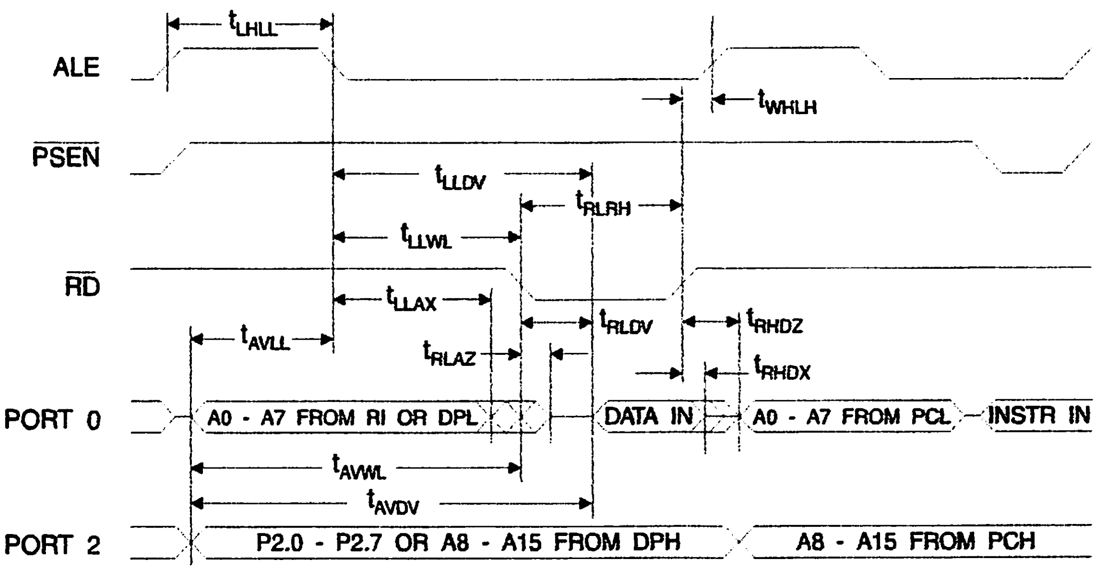
ırnal Program and Data Memory Characteristics

ıbol	Parameter	Variable Oscillator		Units
		Min	Max	
.CL	Oscillator Frequency	0	24	MHz
.	ALE Pulse Width	$2t_{CLCL}$ - 40		ns
.	Address Valid to ALE Low	t_{CLCL} - 13		ns
(	Address Hold after ALE Low	t_{CLCL} - 20		ns
	ALE Low to Valid Instruction In		$4t_{CLCL}$ - 65	ns
.	ALE Low to $\overline{PSEN}$ Low	t_{CLCL} - 13		ns
H	$\overline{PSEN}$ Pulse Width	$3t_{CLCL}$ - 20		ns
	$\overline{PSEN}$ Low to Valid Instruction In		$3t_{CLCL}$ - 45	ns
:	Input Instruction Hold after $\overline{PSEN}$	0		ns
:	Input Instruction Float after $\overline{PSEN}$		t_{CLCL} - 10	ns
/	$\overline{PSEN}$ to Address Valid	t_{CLCL} - 8		ns
	Address to Valid Instruction In		$5t_{CLCL}$ - 55	ns
Z	$\overline{PSEN}$ Low to Address Float		10	ns
H	$\overline{RD}$ Pulse Width	$6t_{CLCL}$ - 100		ns
H	$\overline{WR}$ Pulse Width	$6t_{CLCL}$ - 100		ns
/	$\overline{RD}$ Low to Valid Data In		$5t_{CLCL}$ - 90	ns
X	Data Hold after $\overline{RD}$	0		ns
Z	Data Float after $\overline{RD}$		$2t_{CLCL}$ - 28	ns
/	ALE Low to Valid Data In		$8t_{CLCL}$ - 150	ns
/	Address to Valid Data In		$9t_{CLCL}$ - 165	ns
L	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	$3t_{CLCL}$ - 50	$3t_{CLCL}$ + 50	ns
L	Address to $\overline{RD}$ or $\overline{WR}$ Low	$4t_{CLCL}$ - 75		ns
X	Data Valid to $\overline{WR}$ Transition	t_{CLCL} - 20		ns
H	Data Valid to $\overline{WR}$ High	$7t_{CLCL}$ - 120		ns
X	Data Hold after $\overline{WR}$	t_{CLCL} - 20		ns
Z	$\overline{RD}$ Low to Address Float		0	ns
H	$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{CLCL} - 20	t_{CLCL} + 25	ns

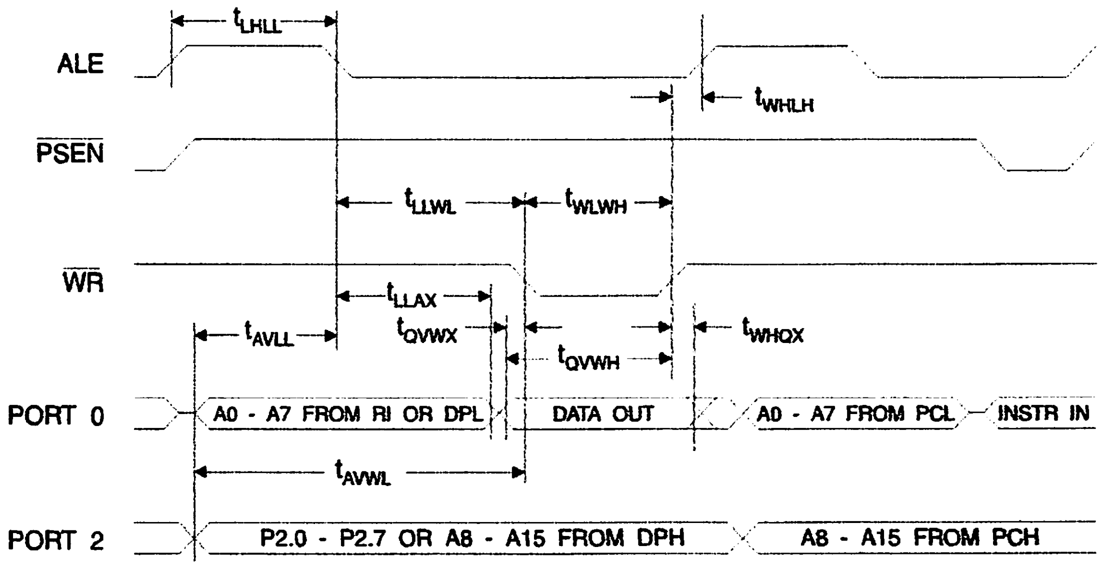
ernal Program Memory Read Cycle



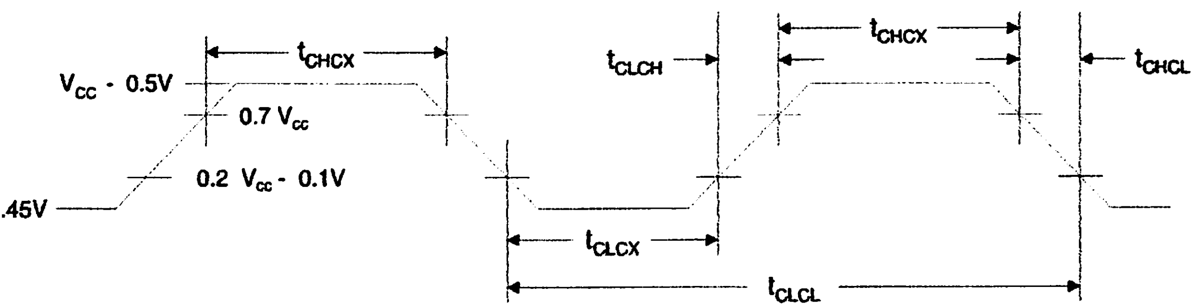
ernal Data Memory Read Cycle



ernal Data Memory Write Cycle



ernal Clock Drive Waveforms



ernal Clock Drive

bol	Parameter	V_{CC} = 4.0V to 6.0V		Units
		Min	Max	
.CL	Oscillator Frequency	0	24	MHz
-	Clock Period	41.6		ns
x	High Time	15		ns
‹	Low Time	15		ns
‑ı	Rise Time		20	ns
L	Fall Time		20	ns

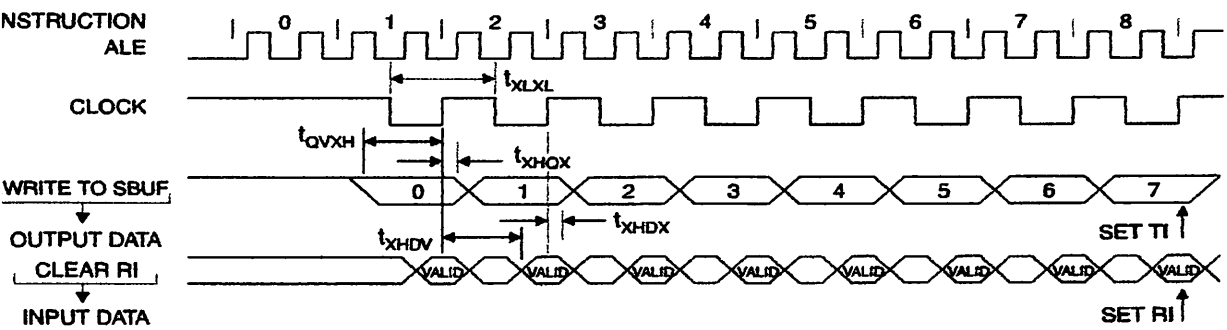
0401F–MICRO–11/03

ial Port Timing: Shift Register Mode Test Conditions

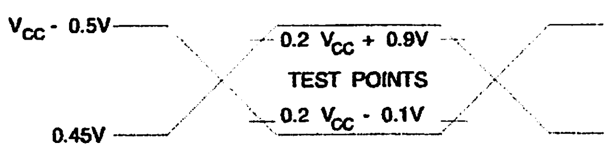
'alues in this table are valid for V_{CC} = 4.0V to 6V and Load Capacitance = 80 pF.

ıbol	Parameter	Variable Oscillator		Units
		Min	Max	
.	Serial Port Clock Cycle Time	$12t_{CLCL}$		µs
·l	Output Data Setup to Clock Rising Edge	$10t_{CLCL}$ - 133		ns
K	Output Data Hold after Clock Rising Edge	$2t_{CLCL}$ - 117		ns
(	Input Data Hold after Clock Rising Edge	0		ns
/	Clock Rising Edge to Input Data Valid		$10t_{CLCL}$ - 133	ns

ft Register Mode Timing Waveforms

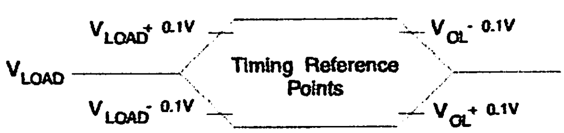


Testing Input/Output Waveforms(1)

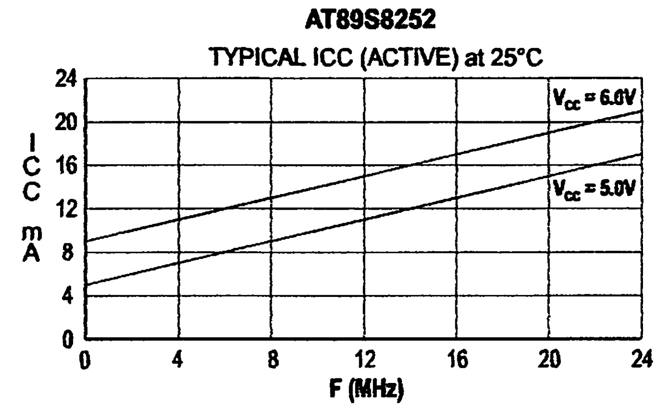


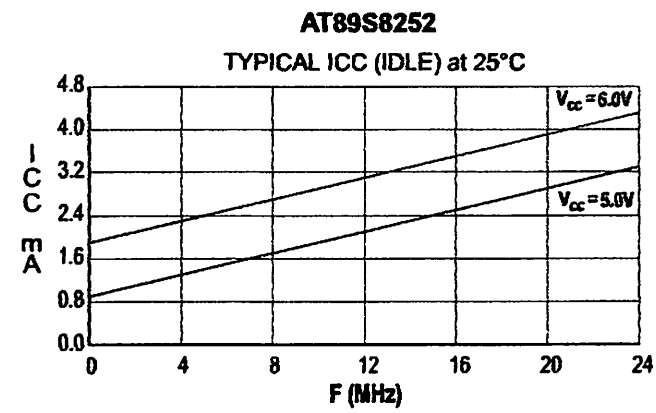
1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

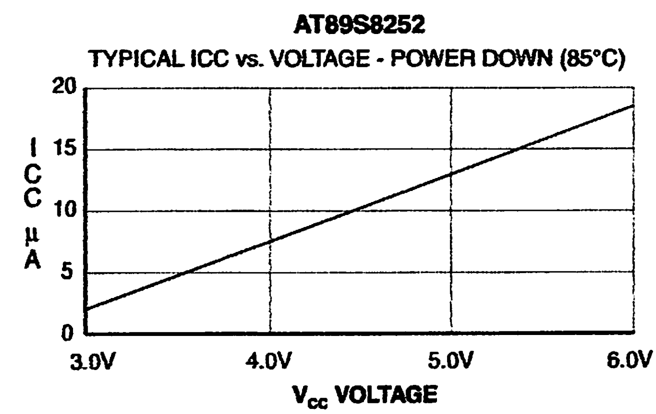
at Waveforms(1)



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







Notes: 1. XTAL1 tied to GND for Icc (power-down)
2. Lock bits programmed

0401F–MICRO–11/03

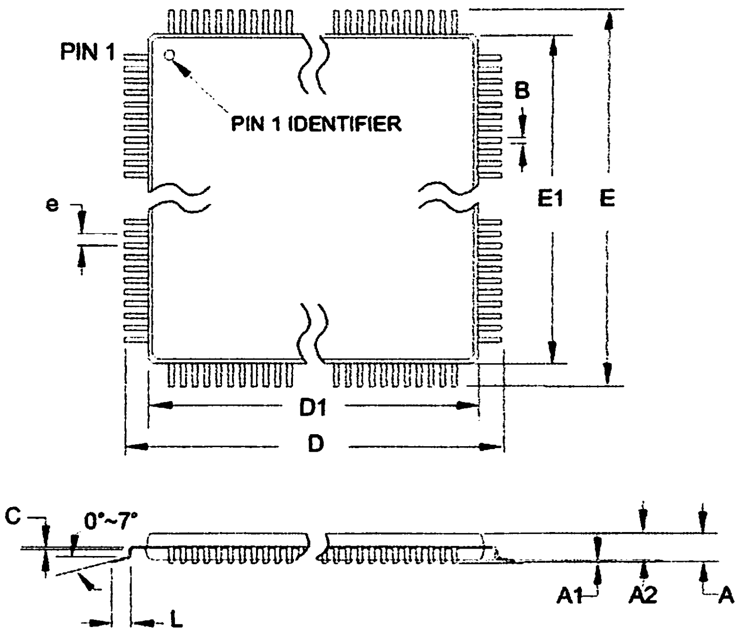
lering Information

›eed lHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 6.0V	AT89S8252-24AC AT89S8252-24JC AT89S8252-24PC	44A 44J 40P6	Commercial (0° C to 70° C)
	4.0V to 6.0V	AT89S8252-24AI AT89S8252-24JI AT89S8252-24PI	44A 44J 40P6	Industrial (-40° C to 85° C)

	Package Type
	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
	44-lead, Plastic J-leaded Chip Carrier (PLCC)
6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)

kaging Information

– TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

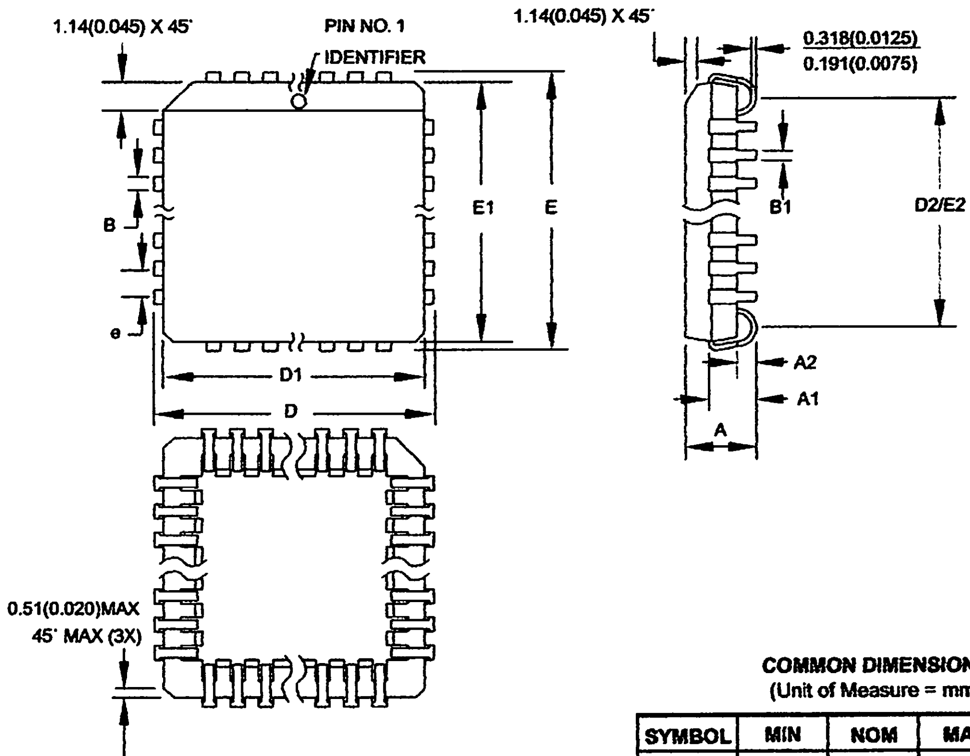
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10 mm maximum.

10/5/2001

2325 Orchard Parkway San Jose, CA 95131	**TITLE** 44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	**DRAWING NO.** 44A	**REV.** B

– PLCC



COMMON DIMENSIONS
(Unit of Measure = mm)

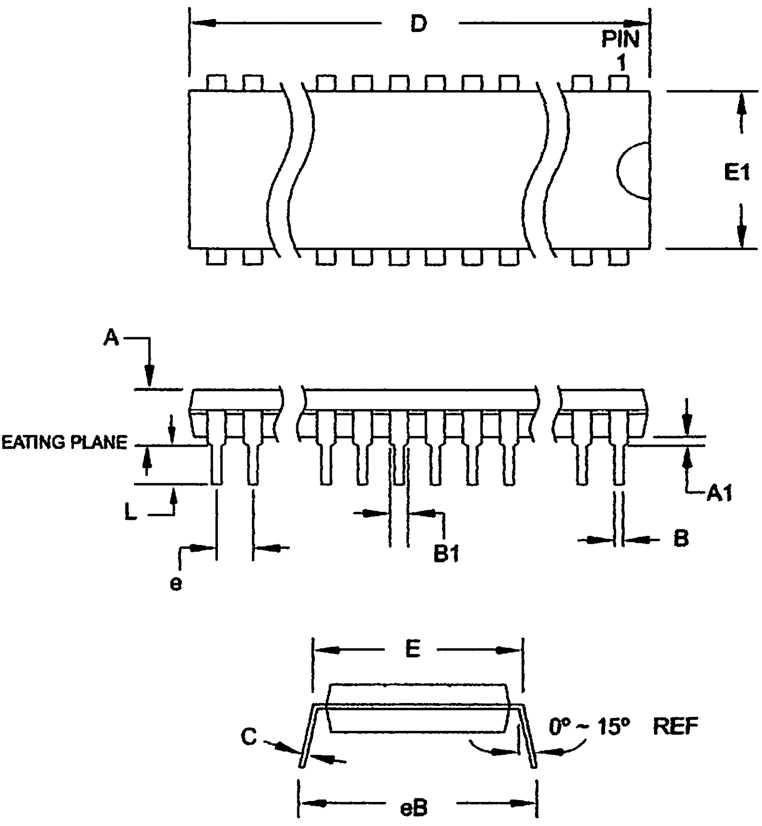
SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	–	4.572	
A1	2.286	–	3.048	
A2	0.508	–	–	
D	17.399	–	17.653	
D1	16.510	–	16.662	Note 2
E	17.399	–	17.653	
E1	16.510	–	16.662	Note 2
D2/E2	14.986	–	16.002	
B	0.660	–	0.813	
B1	0.330	–	0.533	
e	1.270 TYP			

Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010"(0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01

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

6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

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

Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

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

nel Corporation

:25 Orchard Parkway
ın Jose, CA 95131, USA
ɔl: 1(408) 441-0311
ıx: 1(408) 487-2600

ɟional Headquarters

ɔpe
mel Sarl
ɔute des Arsenaux 41
ɪse Postale 80
-1-1705 Fribourg
vitzerland
ɔl: (41) 26-426-5555
ıx: (41) 26-426-5500

ɪ
ɔom 1219
ıinachem Golden Plaza
' Mody Road Tsimshatsui
ɪst Kowloon
ɔng Kong
ɔl: (852) 2721-9778
ıx: (852) 2722-1369

ɪn
:, Tonetsu Shinkawa Bldg.
24-8 Shinkawa
ıuo-ku, Tokyo 104-0033
ıpan
ɔl: (81) 3-3523-3551
ıx: (81) 3-3523-7581

Atmel Operations

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

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

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

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

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

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

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

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

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

0401F–MICRO–11/03 xM

MOTOROLA
SEMICONDUCTOR TECHNICAL DATA

Order this document
by 4N35/D

GlobalOptoisolator™

VDE | UL | CSA | SETI | SEMKO | DEMKO | NEMKO | BABT

6-Pin DIP Optoisolators Transistor Output

4N35*
4N36
4N37
[CTR = 100% Min]

*Motorola Preferred Device

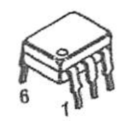
The 4N35, 4N36 and 4N37 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Current Transfer Ratio — 100% Minimum @ Specified Conditions
- Guaranteed Switching Speeds
- Meets or Exceeds all JEDEC Registered Specifications
- ***To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.***

Applications

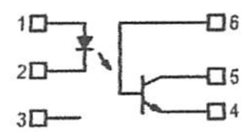
- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- Regulation Feedback Circuits
- Monitor & Detection Circuits
- Solid State Relays

STYLE 1 PLASTIC



STANDARD THRU HOLE
CASE 730A–04

SCHEMATIC



PIN 1. LED ANODE
2. LED CATHODE
3. N.C.
4. EMITTER
5. COLLECTOR
6. BASE

MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INPUT LED			
Reverse Voltage	V_R	6	Volts
Forward Current — Continuous	I_F	60	mA
LED Power Dissipation @ T_A = 25°C with Negligible Power in Output Detector Derate above 25°C	P_D	120 1.41	mW mW/°C
OUTPUT TRANSISTOR			
Collector–Emitter Voltage	V_{CEO}	30	Volts
Emitter–Base Voltage	V_{EBO}	7	Volts
Collector–Base Voltage	V_{CBO}	70	Volts
Collector Current — Continuous	I_C	150	mA
Detector Power Dissipation @ T_A = 25°C with Negligible Power in Input LED Derate above 25°C	P_D	150 1.76	mW mW/°C
TOTAL DEVICE			
Isolation Source Voltage(1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V_{ISO}	7500	Vac(pk)
Total Device Power Dissipation @ T_A = 25°C Derate above 25°C	P_D	250 2.94	mW mW/°C
Ambient Operating Temperature Range(2)	T_A	–55 to +100	°C
Storage Temperature Range(2)	T_{stg}	–55 to +150	°C
Soldering Temperature (10 sec, 1/16″ from case)	T_L	260	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating.
For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

Preferred devices are Motorola recommended choices for future use and best overall value.
GlobalOptoisolator is a trademark of Motorola, Inc.

REV 2

MOTOROLA

4N35 4N36 4N37

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)(1)

Characteristic		Symbol	Min	Typ(1)	Max	Unit
INPUT LED						
Forward Voltage (I_F = 10 mA)	T_A = 25°C T_A = –55°C T_A = 100°C	V_F	0.8 0.9 0.7	1.15 1.3 1.05	1.5 1.7 1.4	V
Reverse Leakage Current (V_R = 6 V)		I_R	—	—	10	µA
Capacitance (V = 0 V, f = 1 MHz)		C_J	—	18	—	pF
OUTPUT TRANSISTOR						
Collector–Emitter Dark Current (V_{CE} = 10 V, T_A = 25°C) (V_{CE} = 30 V, T_A = 100°C)		I_{CEO}	— —	1 —	50 500	nA µA
Collector–Base Dark Current (V_{CB} = 10 V)	T_A = 25°C T_A = 100°C	I_{CBO}	—	0.2 100	20 —	nA
Collector–Emitter Breakdown Voltage (I_C = 1 mA)		$V_{(BR)CEO}$	30	45	—	V
Collector–Base Breakdown Voltage (I_C = 100 µA)		$V_{(BR)CBO}$	70	100	—	V
Emitter–Base Breakdown Voltage (I_E = 100 µA)		$V_{(BR)EBO}$	7	7.8	—	V
DC Current Gain (I_C = 2 mA, V_{CE} = 5 V)		h_{FE}	—	400	—	—
Collector–Emitter Capacitance (f = 1 MHz, V_{CE} = 0)		C_{CE}	—	7	—	pF
Collector–Base Capacitance (f = 1 MHz, V_{CB} = 0)		C_{CB}	—	19	—	pF
Emitter–Base Capacitance (f = 1 MHz, V_{EB} = 0)		C_{EB}	—	9	—	pF
COUPLED						
Output Collector Current (I_F = 10 mA, V_{CE} = 10 V)	T_A = 25°C T_A = –55°C T_A = 100°C	I_C (CTR)(2)	10 (100) 4 (40) 4 (40)	30 (300) — —	— — —	mA (%)
Collector–Emitter Saturation Voltage (I_C = 0.5 mA, I_F = 10 mA)		$V_{CE(sat)}$	—	0.14	0.3	V
Turn–On Time	(I_C = 2 mA, V_{CC} = 10 V, R_L = 100 Ω)(3)	t_{on}	—	7.5	10	µs
Turn–Off Time		t_{off}	—	5.7	10	
Rise Time		t_r	—	3.2	—	
Fall Time		t_f	—	4.7	—	
Isolation Voltage (f = 60 Hz, t = 1 sec)		V_{ISO}	7500	—	—	Vac(pk)
Isolation Current(4) (V_{I-O} = 3550 Vpk) (V_{I-O} = 2500 Vpk) (V_{I-O} = 1500 Vpk)	4N35 4N36 4N37	I_{ISO}	— — —	— — 8	100 100 100	µA
Isolation Resistance (V = 500 V)(4)		R_{ISO}	10^{11}	—	—	Ω
Isolation Capacitance (V = 0 V, f = 1 MHz)(4)		C_{ISO}	—	0.2	2	pF

1. Always design to the specified minimum/maximum electrical limits (where applicable).
2. Current Transfer Ratio (CTR) = I_C/I_F x 100%.
3. For test circuit setup and waveforms, refer to Figure 11.
4. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

TYPICAL CHARACTERISTICS

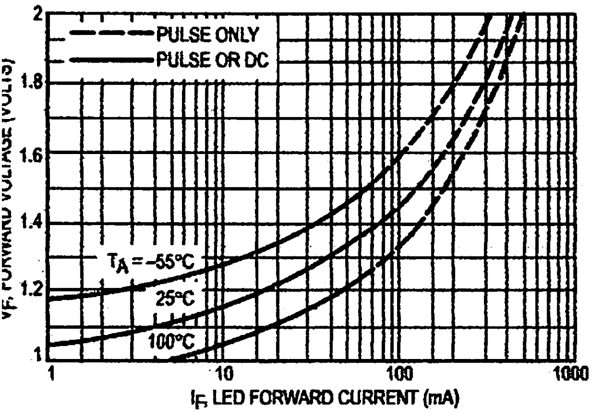


Figure 1. LED Forward Voltage versus Forward Current

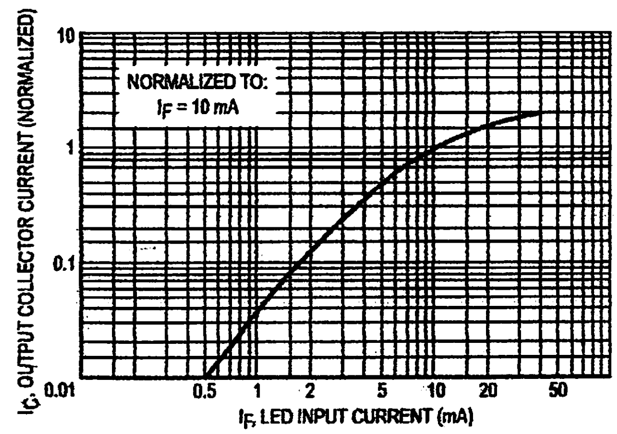


Figure 2. Output Current versus Input Current

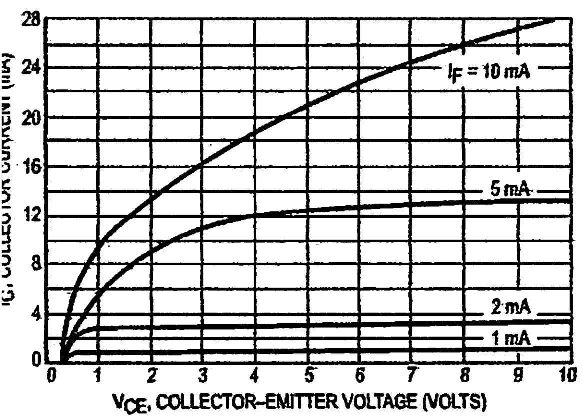


Figure 3. Collector Current versus Collector–Emitter Voltage

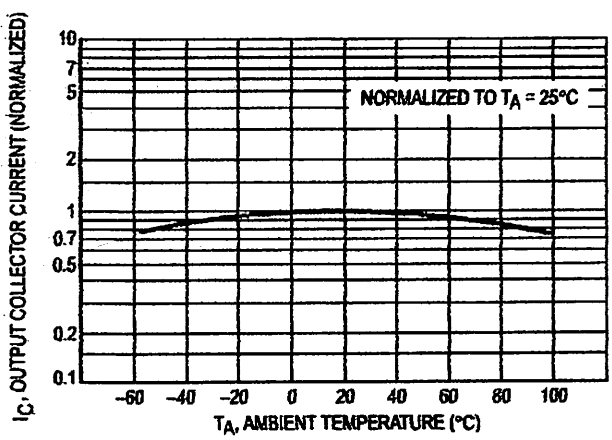


Figure 4. Output Current versus Ambient Temperature

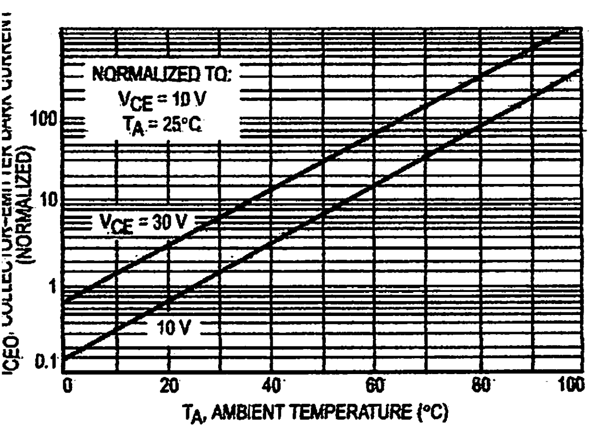


Figure 5. Dark Current versus Ambient Temperature

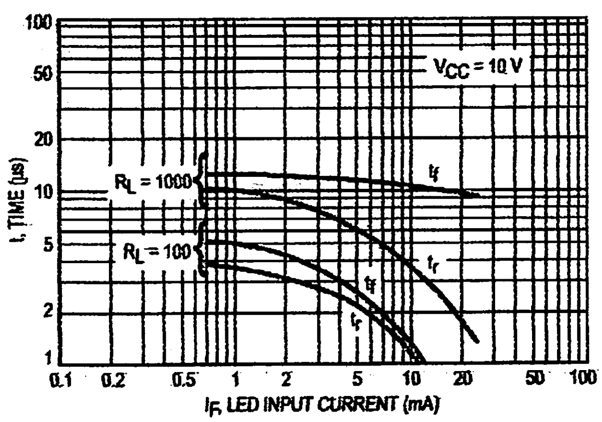


Figure 6. Rise and Fall Times (Typical Values)

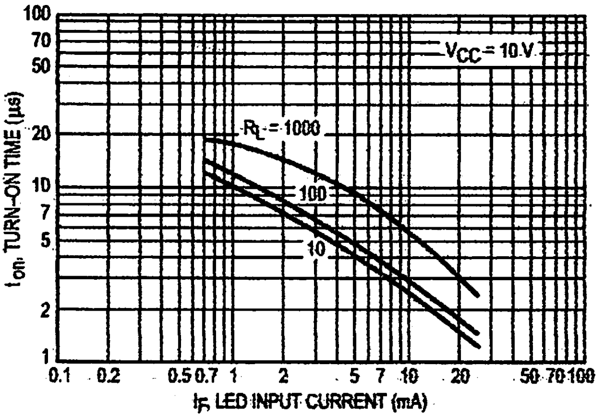


Figure 7. Turn–On Switching Times

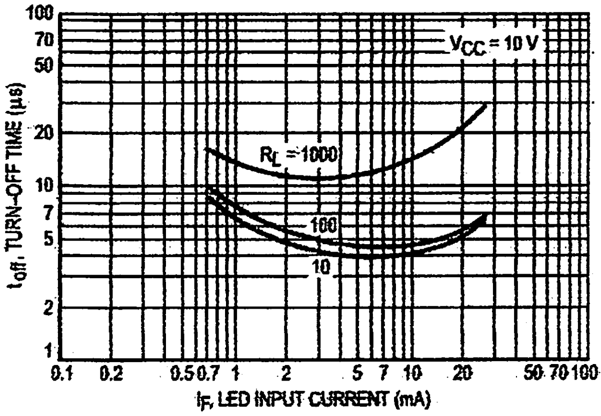


Figure 8. Turn–Off Switching Times

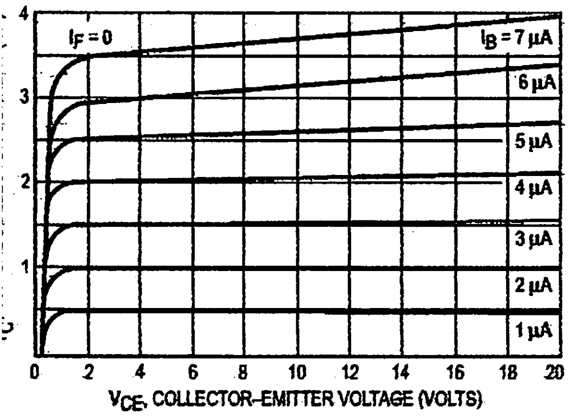


Figure 9. DC Current Gain (Detector Only)

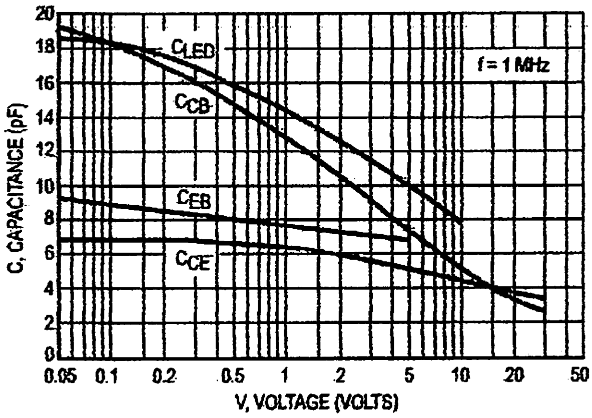


Figure 10. Capacitances versus Voltage

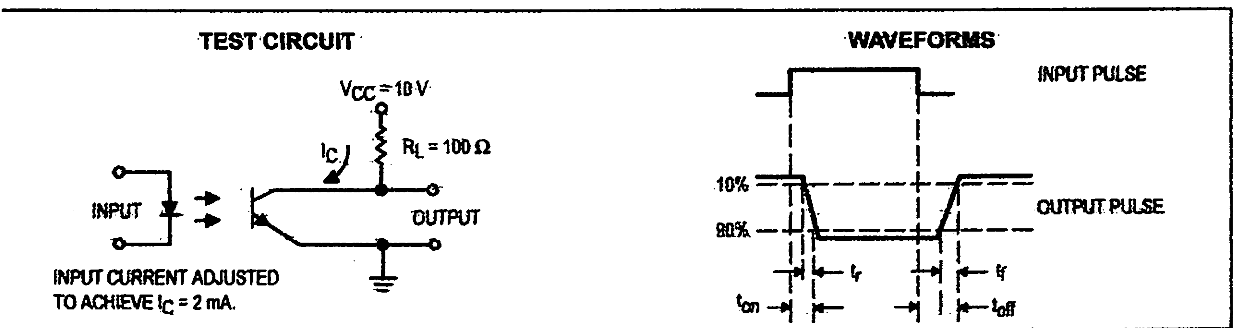
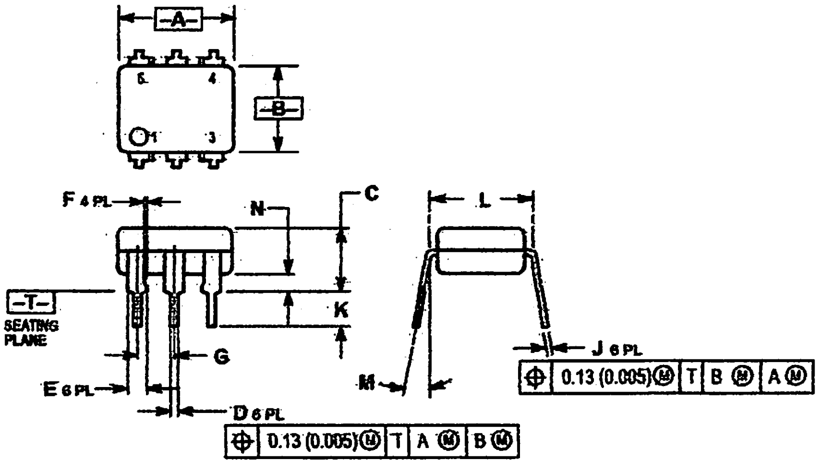


Figure 11. Switching Time Test Circuit and Waveforms

PACKAGE DIMENSIONS

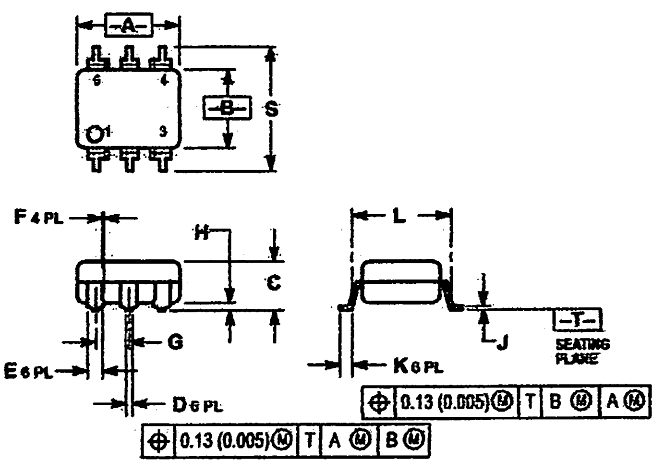


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.015	0.100	0.38	2.54

STYLE 1:
PIN 1. ANODE
2. CATHODE
3. NC
4. EMITTER
5. COLLECTOR
6. BASE

CASE 730A–04
ISSUE G

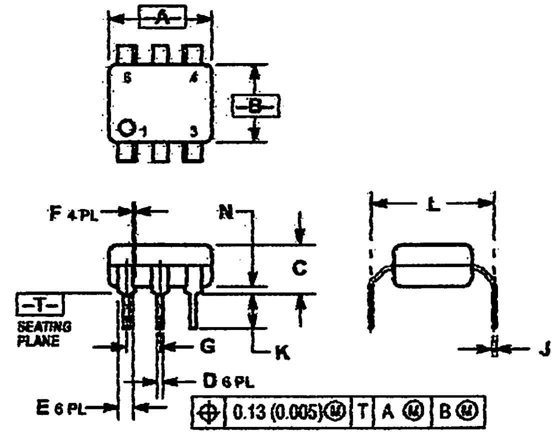


NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.

DIM	INCHES MIN	INCHES MAX	MILLIMETERS MIN	MILLIMETERS MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
H	0.020	0.025	0.51	0.63
J	0.008	0.012	0.20	0.30
K	0.006	0.035	0.16	0.88
L	0.320 BSC		8.13 BSC	
S	0.332	0.390	8.43	9.90

***Consult factory for leadform option availability**

CASE 730C–04
ISSUE D



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.400	0.425	10.16	10.80
N	0.015	0.040	0.38	1.02

***Consult factory for leadform option availability**

CASE 730D–05
ISSUE D

ow to reach us:
SA / EUROPE: Motorola Literature Distribution;
O. Box 20912; Phoenix, Arizona 85036. 1–800–441–2447

IFAX: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244–6609
ITERNET: http://Design–NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi–SPD–JLDC, Toshikatsu Otsuki,
6F Seibu–Butsuryu–Center, 3–14–2 Tatsumi Koto–Ku, Tokyo 135, Japan. 03–3521–8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

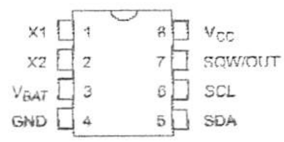
FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2–wire serial interface
- Programmable squarewave output signal
- Automatic power fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode at 25°C
- Optional industrial temperature range –40°C to +85°C (IND)
- Available in 8–pin DIP or SOIC

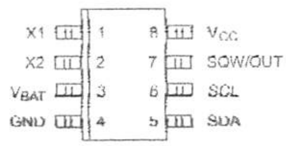
ORDERING INFORMATION

DS1307	Serial Timekeeping Chip; 8–pin DIP
DS1307Z	Serial Timekeeping Chip; 8–pin SOIC (150 mil)
DS1307N	8–pin DIP (IND)
DS1307ZN	8–pin SOIC (IND)

PIN ASSIGNMENT



DS1307 8 PIN DIP (300 MIL)



DS1307Z 8 PIN SOIC (150 MIL)

PIN DESCRIPTION

V_{CC}	–	Primary Power Supply
X1, X2	–	32.768 KHz Crystal Connection
V_{BAT}	–	+3 Volt Battery Input
GND	–	Ground
SDA	–	Serial Data
SCL	–	Serial Clock
SQW/OUT	–	Square wave/Output Driver

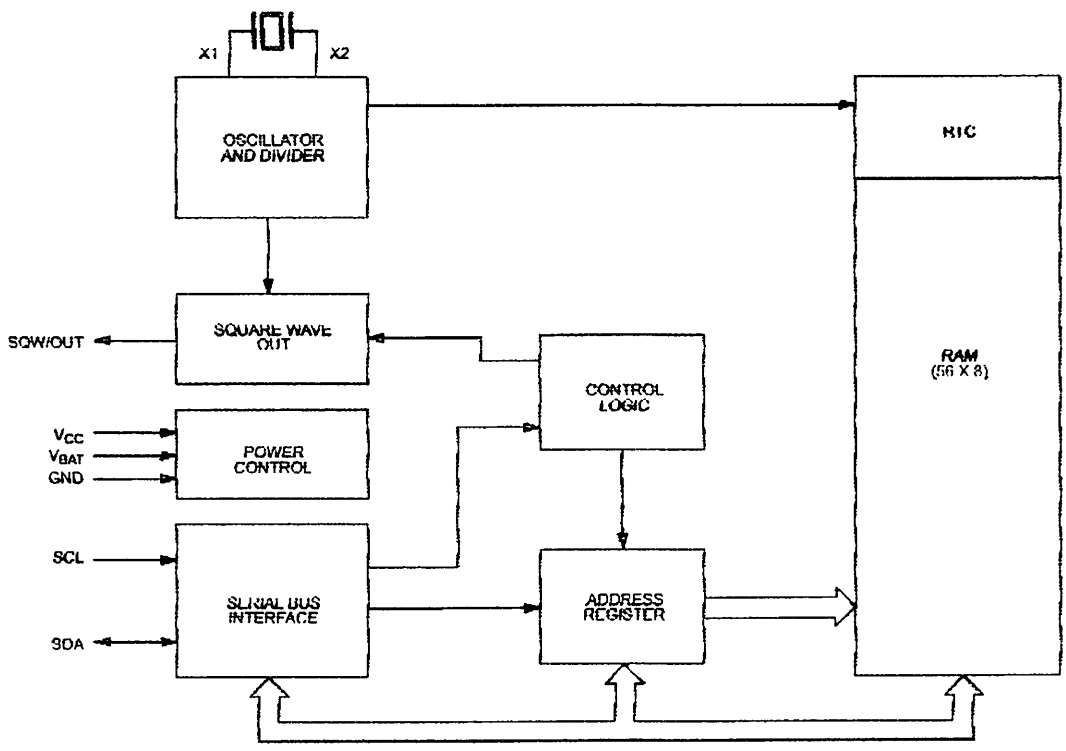
DESCRIPTION

The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24–hour or 12–hour format with AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT} the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low current battery backup mode. Upon power up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT}+0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock. The following paragraphs describe the function of each pin.

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5 volt input. When 5 volts are applied within normal limits, the device is fully accessible and data can be written and read. When a 3 volt battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external 3 volt battery.

V_{BAT} – Battery input for any standard 3 volt lithium cell or other energy source. Battery voltage must be held between 2.5 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A Lithium battery with 35 mAh or greater will back up the DS1307 for more than 10 years in the absence of power.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2–wire serial interface. The SDA pin is open drain which requires an external pull–up resistor.

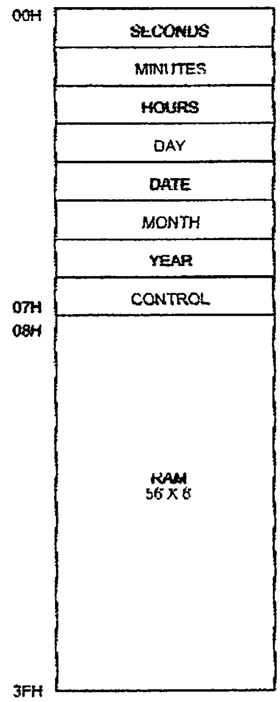
SQW/OUT (Square Wave/ Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 KHz, 8 KHz, 32 KHz). The SQW/OUT pin is open drain which requires an external pull–up resistor.

X1, X2 – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

RTC AND RAM ADDRESS MAP

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

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary–Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a one, the oscillator is disabled. When cleared to a zero, the oscillator is enabled.

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

DS1307 TIMEKEEPER REGISTERS Figure 3

	BIT7							BIT0	
00H	CH	10 SECONDS			SECONDS				00–59
	X	10 MINUTES			MINUTES				00–59
	X	12/24	10 HR A/P	10 HR	HOURS				01–12 00–23
	X	X	X	X	X	DAY			1–7
	X	X	10 DATE		DATE				01–28/29 01–30 01–31
	X	X	10 MONTH		MONTH				01–12
	10 YEAR				YEAR				00–99
07H	OUT	X	X	SQWE	X	X	RS1	RS0	

CONTROL REGISTER

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

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

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

SQWE (Square wave Enable): This bit when set to a logic 1 will enable the oscillator output. The frequency of the square wave output depends on the value of the RS0 and RS1 bits.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

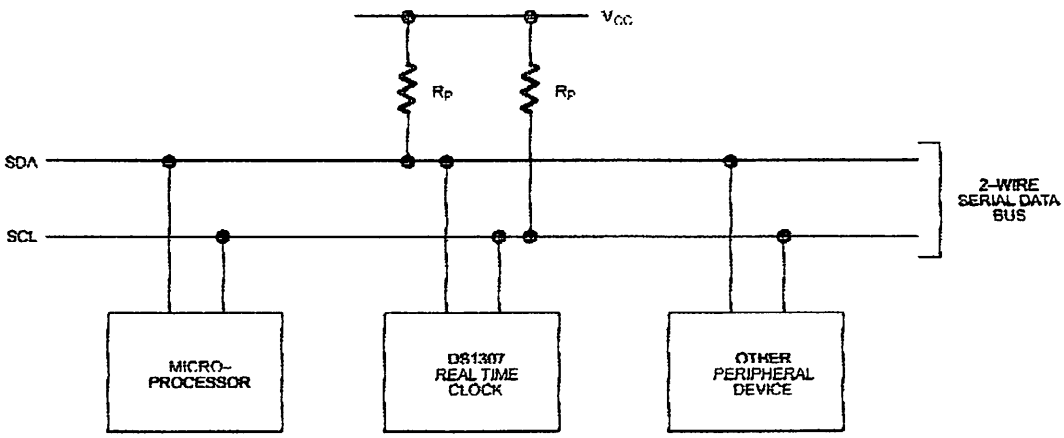
SQUAREWAVE OUTPUT FREQUENCY Table 1

RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4 KHz
1	0	8 KHz
1	1	32 KHz

2-WIRE SERIAL DATA BUS

The DS1307 supports a bi-directional 2-wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2-wire bus. A typical bus configuration using this 2-wire protocol is show in Figure 4.

TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



The following bus protocol has been defined (see Figure 5).

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

Accordingly, the following bus conditions have been defined:

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

Start data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

Stop data transfer: A change in the state of the data line from low to high, while the clock line is high defines the STOP condition.

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

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit.

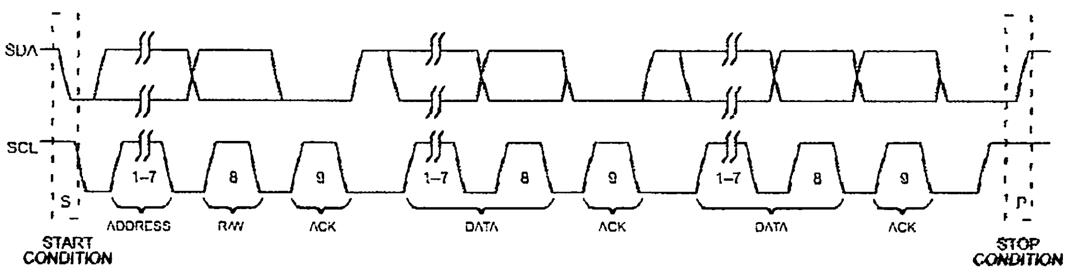
Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

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

DATA TRANSFER

Figures 5, 6, and 7 detail how data transfer is accomplished on the 2-wire bus. Depending on the state of the R/$\overline{W}$ bit in the transmission protocols as shown in Figures 6 and 7, two types of data transfer are possible:

DATA TRANSFER ON 2–WIRE SERIAL BUS Figure 5



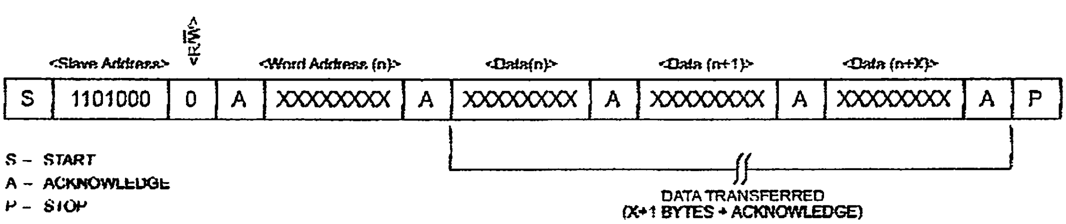
1. *Data transfer from a master transmitter to a slave receiver.* The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transfered with the most significant bit (MSB) first.
2. Data transfer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

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

The DS1307 may operate in the following two modes:

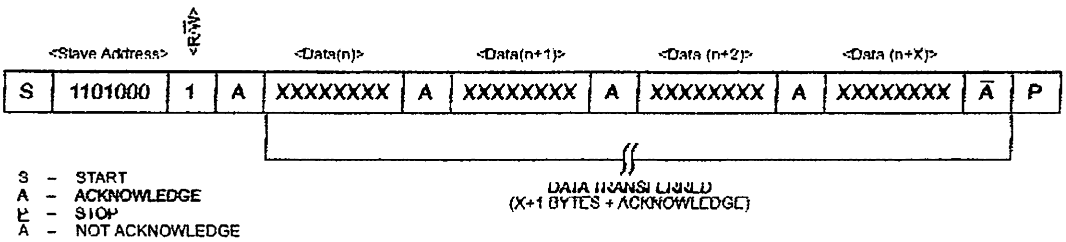
1. Slave receiver mode (DS1307 write mode): Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/W) which for a write is a 0. After receiving and decoding the address byte the DS1307 outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307 This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

DATA WRITE – SLAVE RECEIVER MODE Figure 6



2. Slave transmitter mode (DS1307 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/W\) which for a read is a 1. After receiving and decoding the address byte the DS1307 inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a Not Acknowledge to end a read.

DATA READ – SLAVE TRANSMITTER MODE Figure 7



ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	–0.5V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	–55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

The Dallas Semiconductor DS1307 is built to the highest quality standards and manufactured for long term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, standard versions of the DS1307 are not exposed to environmental stresses, such as burn–in, that some industrial applications require. Products which have successfully passed through this series of environmental stresses are marked IND or N, denoting their extended operating temperature and reliability rating. For specific reliability information on this product, please contact the factory at (972) 371–4448.

RECOMMENDED DC OPERATING CONDITIONS (0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	1
Logic 1	V_{IH}	2.2		V_{CC}+0.3	V	1
Logic 0	V_{IL}	–0.3		+0.8	V	1
V_{BAT} Battery Voltage	V_{BAT}	2.5		3.5	V	1

DC ELECTRICAL CHARACTERISTICS (0°C to 70°C; V_{CC}=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage	I_{LI}			1	µA	10
I/O Leakage	I_{LO}			1	µA	11
Logic 0 Output	V_{OL}			0.4	V	2
Active Supply Current	I_{CCA}			1.5	mA	9
Standby Current	I_{CCS}			200	µA	3
Battery Current (OSC ON); SQW/OUT OFF	I_{BAT1}		300	500	nA	4
Battery Current (OSC ON); SQW/OUT ON (32 KHz)	I_{BAT2}		480	800	nA	4

AC ELECTRICAL CHARACTERISTICS

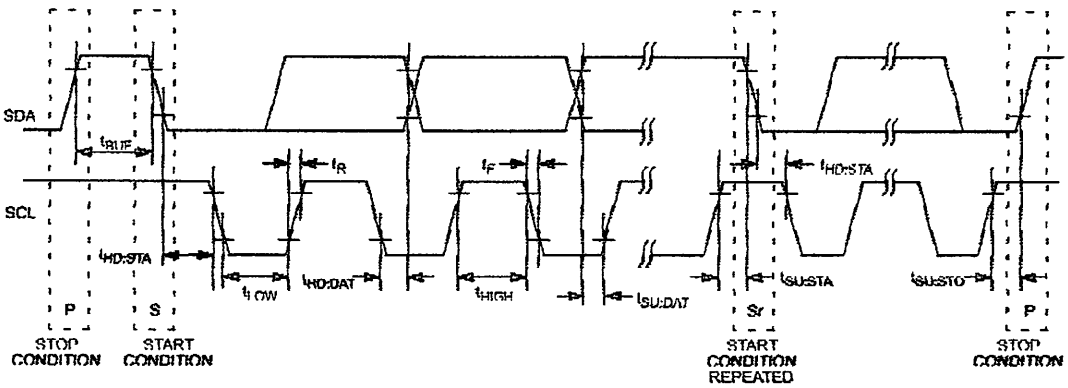
($0°C$ to $70°C$; V_{CC}=4.5V to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	KHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			µs	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			µs	5
LOW Period of SCL Clock	t_{LOW}	4.7			µs	
HIGH Period of SCL Clock	t_{HIGH}	4.0			µs	
Set–up Time for a Repeated START Condition	$t_{SU:STA}$	4.7			µs	
Data Hold Time	$t_{HD:DAT}$	0			µs	6, 7
Data Set–up Time	$t_{SU:DAT}$	250			ns	
Rise Time of Both SDA and SCL Signals	t_R			1000	ns	
Fall Time of Both SDA and SCL Signals	t_F			300	ns	
Set–up Time for STOP Condition	$t_{SU:STO}$	4.7			µs	
Capacitive Load for each Bus Line	C_B			400	pF	8
I/O Capacitance	$C_{I/O}$		10		pF	
Crystal Capacitance			12.5		pF	

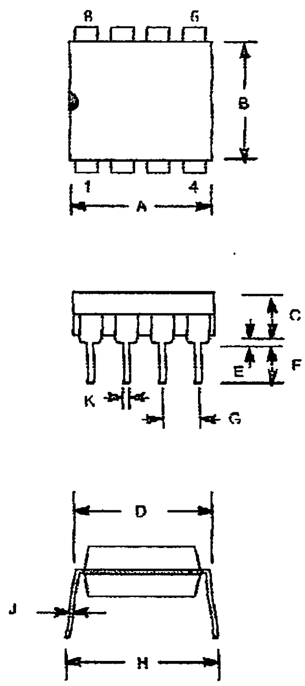
NOTES:

1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at V_{CC}=4.5V, V_{OL}=GND for capacitive loads.
3. I_{CCS} specified with V_{CC}=5.0V and SDA, SCL=5.0V.
4. V_{CC}=0V, V_{BAT}=3V.
5. After this period, the first clock pulse is generated.
6. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the V_{IHMIN} of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
7. The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.
8. C_B – total capacitance of one bus line in pF.
9. I_{CCA} – SCL clocking at max frequency = 100 KHz.
10. SCL only.
11. SDA and SQW/OUT

TIMING DIAGRAM

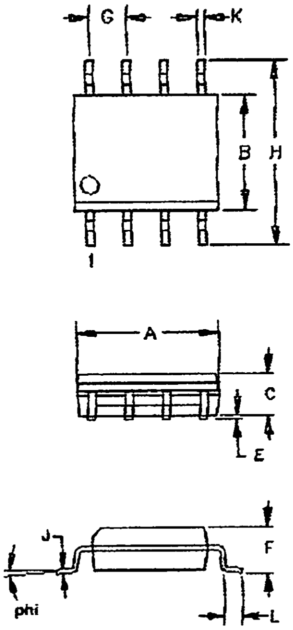


DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP



PKG	8–PIN	
DIM	MIN	MAX
A IN. MM	0.360 9.14	0.400 10.16
B IN. MM	0.240 6.10	0.260 6.60
C IN. MM	0.120 3.05	0.140 3.56
D IN. MM	0.300 7.62	0.325 8.26
L IN. MM	0.015 0.38	0.040 1.02
F IN. MM	0.120 3.04	0.140 3.56
G IN. MM	0.090 2.29	0.110 2.79
H IN. MM	0.320 8.13	0.370 9.40
J IN. MM	0.008 0.20	0.012 0.30
K IN. MM	0.015 0.38	0.021 0.53

DS1307Z 64 X 8 SERIAL REAL TIME CLOCK 8–PIN SOIC (150 MIL)



PKG	8–PIN (150 MIL)	
DIM	MIN	MAX
A IN. MM	0.188 4.78	0.196 4.98
B IN. MM	0.150 3.81	0.158 4.01
C IN. MM	0.046 1.22	0.062 1.57
E IN. MM	0.004 0.10	0.010 0.25
F IN. MM	0.053 1.35	0.069 1.75
G IN. MM	0.050 BSC 1.27 BSC	
H IN. MM	0.230 5.84	0.244 6.20
J IN. MM	0.007 0.18	0.011 0.28
K IN. MM	0.012 0.30	0.020 0.51
L IN. MM	0.016 0.41	0.050 1.27
phi	0°	8°

56–G2008–001