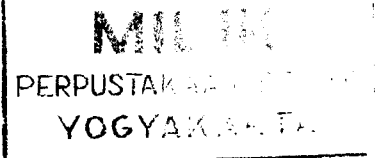


MESIN PRESENSI DIGITAL BERBASIS MIKROKONTROLER AT89S8252

TUGAS AKHIR



Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik
Jurusan Teknik Elektro



No. Insk.	78/A/ST/IT.EL-VII/05
Tanggal	5 Jan 05
Asal	FAK. TEKNOLOGI INDUSTRI - VII.
Harga	Rp. 2.000,-
PERPUSTAKAAN FAK. TEKNOLOGI INDUSTRI UNIVERSITAS ISLAM INDONESIA YOGYAKARTA	

Oleh :

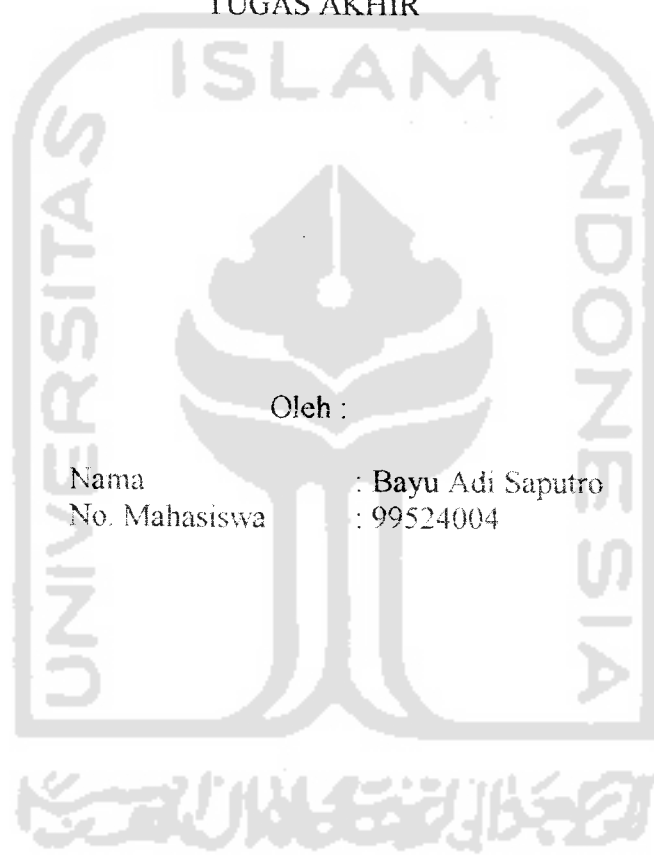
Nama : Bayu Adi Saputro
No. Mahasiswa : 99524004

JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2004

LEMBAR PENGESAHAN PEMBIMBING

**MESIN PRESENSI DIGITAL
BERBASIS MIKROKONTROLER AT89S8252**

TUGAS AKHIR



Oleh :

Nama : Bayu Adi Saputro
No. Mahasiswa : 99524004

Yogyakarta, Desember 2004

Pembimbing I

Ir. H. Suyanto

Pembimbing II

Sisdarmanto Adinandra, ST

LEMBAR PENGESAHAN PENGUJI

MESIN PRESENSI DIGITAL BERBASIS MIKROKONTROLER AT89S8252

TUGAS AKHIR

Oleh :

Nama : Bayu Adi Saputro
No. Mahasiswa : 99524004

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 13 Desember 2004

Tim Penguji

Ir. H. Suyamto
Ketua

Sisdarmanto Adinandra, ST
Anggota I

Yusuf Aziz Amrullah, ST
Anggota II



Mengetahui :

Dekan Fakultas Teknologi Industri
Universitas Islam Indonesia

Ir. H. Bachrun Sutrisno, Msc

KATA PENGANTAR

Assalaamu'alaikum Wr.Wb

Alhamdullilaahhirabbil'aalamien, puji syukur kehadiran Allah SWT, atas limpahan hidayah, taufiq, serta 'inayah-Nya, sehingga Laporan Tugas Akhir dengan judul "MESIN PRESENSI DIGITAL BERBASIS MIKROKONTROLER AT89S8252" ini dapat terselesaikan dengan baik tanpa halangan yang berarti.

Sholawat serta salam semoga senantiasa tercurah atas junjungan kita Nabi Muhammad SAW, para kerabat, sahabat, pengikut setianya hingga hari kiamat nanti, Amien.

Sehubungan dengan terselesaikannya penyusunan laporan ini penyusun mengucapkan banyak terima kasih kepada:

1. Bapak Ir. H Bachrum Sutrisno, Msc, selaku Dekan Fakultas Teknologi Industri.
2. Ibu Ir. Hj. Budi Astuti, selaku Ketua Jurusan Teknik Elektro
3. Bapak Ir. H. Suyamto selaku Dosen pembimbing I yang telah memberikan bimbingan dan arahan dalam penelitian dan penyusunan skripsi ini.
4. Bapak Sisdarmanto Adnandra, ST, selaku Dosen pembimbing II yang telah memberikan bimbingan dan arahan dalam penelitian dan penyusunan skripsi ini.
5. Pada kedua orang tua yang telah memberikan dukungan moral dan spiritual dan,
6. Teman-teman yang turut membantu dan memberikan dukungan.

. Semoga skripsi ini bermanfaat untuk masyarakat dan semua pihak yang berkepentingan. Penulis menyadari bahwa penulisan skripsi ini banyak memerlukan arahan, bimbingan, dorongan, tenaga, pikiran, dan waktu dari berbagai pihak. Maka dari itu penulis mengucapkan banyak terima kasih atas saran dan masukan yang diberikan.

Yogyakarta, 10 September 2004

Penulis



ABSTRAKSI

Sekarang ini pada beberapa instansi dan perusahaan baik pemerintah maupun swasta masih dilakukan presensi karyawan secara manual. presensi secara manual ini selain tidak praktis juga sering terjadi kesalahan atau kerusakan data. Atas pertimbangan tersebut maka dibuat penelitian tentang perancangan mesin presensi secara digital yang dikendalikan dengan mikrokontroler AT89S8252. Alat ini diharapkan dapat mengurangi kelemahan-kelemahan pada sistem presensi manual, karena dilengkapi dengan sistem pengecekan nomor PIN (*Personal Identification Number*) dan penyimpanan data secara permanen selama 1 tahun serta penampil LCD 16 x 2 karakter yang menjadikan alat ini mudah digunakan.

Alat ini hanya bisa diakses oleh 10 orang yang sudah terdaftar dalam sistem, masing-masing orang memiliki nomor PIN. Pemasukan nomor PIN dilakukan melalui papan tombol (*keypad*) 3 x 4 dan penyimpanan data digunakan serial EEPROM AT24C256 yang berkapasitas 32 kbyte. Pemilihan mikrokontroler AT89S8252 sebagai pengendali pada alat ini selain karena faktor ekonomis juga karena mikrokontroler ini memiliki internal EEPROM sebesar 2 kbyte dan fasilitas SPI (*Serial Peripheral Interface*). Internal EEPROM digunakan untuk menyimpan data 10 karyawan dan SPI digunakan untuk pengisian program ke mikrokontroler secara langsung.

Mesin presensi digital pada perancangan ini dibatasi hanya dapat diakses oleh 10 karyawan saja. Sehingga jika diinginkan untuk menambah jumlah karyawan yang dapat mengakses alat ini, maka perlu penambahan eksternal *Flash memory* dan eksternal EEPROM. Atau dapat menggunakan mikrokontroler dengan kapasitas *Flash memory* yang lebih besar dan ditambah dengan serial EEPROM yang lebih besar.

Dari perancangan alat ini dihasilkan mesin presensi digital yang dapat menyimpan data keluar/masuk dari 10 orang karyawan. Data tersebut dapat diakses selama 1 tahun. Untuk mengakses alat ini masing-masing karyawan dilengkapi dengan nomor PIN dan nomor PIN ini dapat diganti setiap saat.

TAKARIR

Akumulator

Register yang digunakan untuk menyimpan semua proses aritmatika

ASCII (*American Standarts Committee for Information Interchange*)

Kode ini merupakan kode *alphanumeric* yang masing masing-kode berisi 7-bit

Assembler

Bahasa pemrograman mikrokontroler MCS51

Atmel

Sebuah perusahaan pembuat *flash memory* beraristektur MCS51

Bit

Bit adalah ukuran terkecil data digital. Bit biasanya hanyalah merupakan pilihan antara 0 dan 1. Dimana 0 biasanya berarti 'Off' dan 1 berarti 'On'.

Byte

Byte adalah merupakan kumpulan beberapa bit (1 *Byte* = 8 bit)

Bouncing

Lentangan yang biasa terjadi pada saklar mekanis

Chip

Sebuah kepingan IC

Clock

Sinyal kotak untuk menentukan kecepatan kerja dari perangkat digital

Compiler

Software untuk merubah ekstensi *file* tertentu kedalam ekstensi *file* yang lainnya

Cycle

Kecepatan siklus mesin program

Delay

Waktu tunda

Digital

Data dalam bentuk angka 0 dan 1

Downloader

Perangkat yang digunakan untuk mengisi program dari komputer ke mikrokontroler

EEPROM (*Electrical Erasable Programmable Read Only Memory*)

Memori yang dapat diisi ulang secara elektronik

Emulator

Software untuk simulator perangkat keras

Ground

Titik referensi tegangan biasanya untuk menentukan 0 V

Heksadesimal

Penulisan angka dalam format 16-an

IC (*Integrated Circuit*)

Sebuah alat yang didalamnya terdapat rangkaian elektronis dengan fungsi tertentu

Input

Masukan bagi alat atau sistem

Interupsi

Sela atau pemberhentian sesuatu untuk sementara waktu

Keypad

Papan tombol yang terdiri dari beberapa tombol

MCS51

Keluarga mikrokontroler Atmel AT89xx

MHz (*Mega Hertz*)

Jutaan gerak per detik

Mikrokontroler

Sebuah alat atau IC kecil yang dapat digunakan untuk mengendalikan sebuah system

I2C

Teknologi komunikasi serial yang hanya menggunakan 2 buah jalur I/O yaitu SCL dan SDA

Output

Keluaran dari alat atau sistem

PIN (*Personal Identification Number*)

Nomor/angka yang menunjukkan identitas seseorang

Port

Sebuah jalur atau pintu yang dapat digunakan sebagai masukan atau keluaran

RAM (*Random Access Memory*)

Memori yang dapat diolah secara acak, biasanya digunakan sebagai penyimpan data untuk sementara waktu

Regiser

Sebuah kumpulan data digital dalam mikrokontroler, dapat digunakan untuk mengatur atau melihat keadaan mikrokontroler

RST (*Reset*)

Keadaan awal dari sistem

ROM (*Read Only Memory*)

Memori yang hanya dapat dibaca, biasanya digunakan untuk menyimpan data program yang akan dijalankan pada mikrokontroler

SCL

Input serial *clock* untuk serial EEPROM

SDA

Input serial data untuk serial EEPROM

SFR (*Special Function Register*)

Merupakan register yang memiliki fungsi-fungsi khusus

SPI (*Serial Peripheral Interface*)

Antar muka untuk pemrograman serial

Software

Perangkat lunak, berupa pemrograman komputer

Timer/Counter

Aplikasi mikrokontroler untuk pewaktu dan pengitung yang dibedakan dari pemberian input *clock*

VCC

Tegangan referensi positif (+)

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING	ii
HALAMAN PENGESAHAN PENGUJI	iii
KATA PENGANTAR	iv
ABSTRAKSI	vi
TAKARIR	vii
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Sistematika Penulisan	3
BAB II DASAR TEORI	5
2.1. Mikrokontroler AT89S8252	5
2.1.1. Memori	6
2.1.1.1. Memori Data	7
2.1.1.2. Memori Program	7

2.1.1.3. EEPROM	7
2.1.2. <i>Timer/Couter</i>	9
2.2. Serial EEPROM	11
2.2.1. Operasi Penulisan	13
2.2.2. Operasi Pembacaan	14
2.2.2.1. Pembacaan saat ini	14
2.2.2.2. Pembacaan Acak	15
2.2.2.3. Pembacaan Beruntun	15
2.3. <i>Keypad</i> matriks 3 x 4	16
2.4. Penampil LCD	18
2.4.1. Register	19
2.4.2. <i>Busy Flag</i> (BF)	20
2.4.3. <i>Address Counter</i> (AC)	20
2.4.4. <i>Display Data RAM</i> (DDRAM)	20
2.4.5. <i>Character Generator ROM</i> (CGROM)	20
2.4.6. <i>Character Generator RAM</i> (CGRAM)	21
BAB III PERANCANGAN ALAT	22
3.1. Perangkat Keras	22
3.1.1. Rangkaian Mikrokontroler AT89S8252	23
3.1.2. Rangkaian <i>keypad</i> 3 x 4	25
3.1.3. Rangkaian Serial EEPROM	25
3.2. Perangkat Lunak	26
3.2.1. Cara Pembuatan Program	26

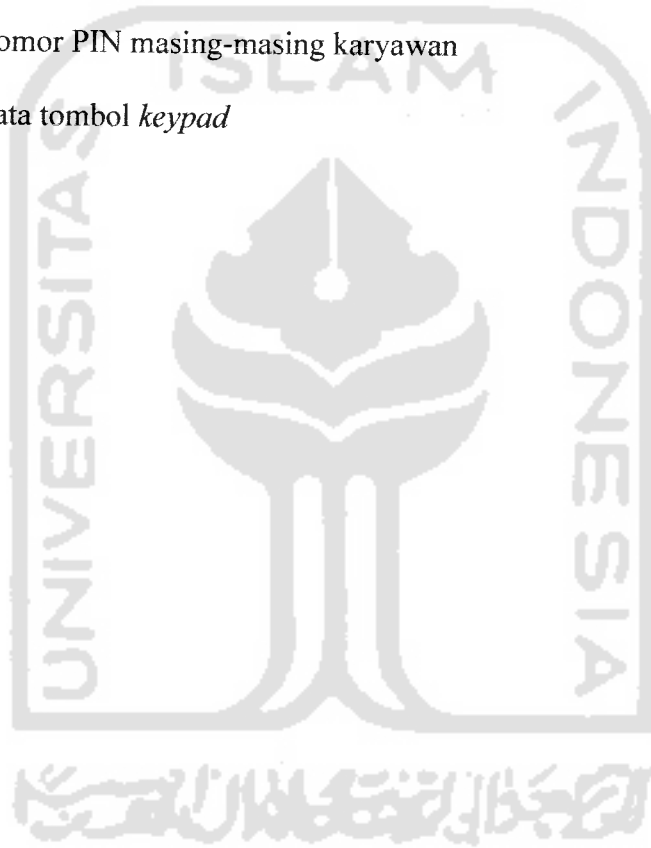
3.2.2. Perancangan Program	29
3.2.3. <i>Downloader</i> ke mikrokontroler AT89S8252	31
3.2.4. Implementasi Diagram Alir	32
3.2.4.1. Inisialisasi LCD	32
3.2.4.2. Inisialisasi internal EEPROM	32
3.2.4.3. Inisialisasi serial EEPROM	32
3.2.4.4. Inisialisasi interupsi	33
3.2.4.5. Inisialisasi Timer 1	33
3.2.4.6. Program menampilkan karakter pada LCD	33
3.2.4.7. Program membaca dan menulis internal EEPROM	34
3.2.4.8. Program membaca dan menulis Serial EEPROM	36
3.2.4.9. Program Pembacaan keypad	38
3.2.4.10. Program Pengecekan nomor PIN	39
3.2.4.11. Program Pergantian nomor PIN	40
3.2.4.12. Program Jam digital	41
3.2.4.13. Program Tanggal	42
BAB IV ANALISIS DAN PEMBAHASAN	44
4.1. Cara Penggunaan Alat	44
4.2. Pembahasan Memori program, internal EEPROM dan serial EEPROM	45
4.3. Pembahasan Program delay	47
4.4. Rutin Program Serial EEPROM	48
4.5. Pembahasan <i>keypad</i>	49

4.5. Pembahasan program interupsi jam digital	49
BAB V PENUTUP	51
5.1. Kesimpulan	51
5.2. Saran	51
DAFTAR PUSTAKA	
LAMPIRAN	



DAFTAR TABEL

	Halaman
Tabel 2.1 <i>Register</i> WMCON	8
Tabel 2.2 Konfigurasi Pin EEPROM AT24C256	11
Tabel 3.1 Hubungan Pin mikrokontroler dengan port printer	31
Tabel 4.1 Nomor PIN masing-masing karyawan	46
Tabel 4.2 Data tombol <i>keypad</i>	49



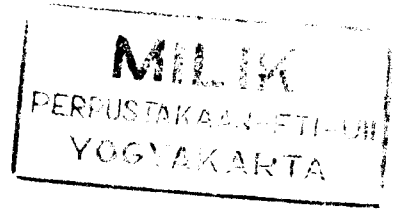
DAFTAR GAMBAR

	<i>Halaman</i>
Gambar 2. 1 Diagram blok mikrokontroler AT89S8252	6
Gambar 2. 2 Konsep dasar <i>timer/counter</i> pada AT89S8252	9
Gambar 2. 3 Konfigurasi kaki AT24C256	11
Gambar 2. 4 Penulisan secara <i>byte</i>	13
Gambar 2. 5 Penulisan secara <i>page</i>	13
Gambar 2. 6 Operasi Pembacaan Alamat saat ini	14
Gambar 2. 7 Operasi Pembacaan Acak	15
Gambar 2. 8 Operasi Penulisan Berurutan	16
Gambar 2. 9 <i>Keypad</i> matrik 3 x 4	17
Gambar 2. 10 Tampilan efek <i>Bouncing</i>	17
Gambar 2. 11 Diagram Blok LCD <i>dot matriks</i>	18
Gambar 3. 1. Diagram Blok Mesin presensi Digital	22
Gambar 3.2. Rangkaian sistem minimum AT89S8252	23
Gambar 3.3 Rangkaian <i>Power on Reset</i>	24
Gambar 3.4 Gambar <i>keypad</i> matrik 3 x 4 dan rangkaian tombol	25
Gambar 3.5 Rangkaian serial EEPROM	26
Gambar 3.6 Tampilan <i>compiler</i> HB 2000	27
Gambar 3.7 <i>Emulator</i> PDS-51	27
Gambar 3.8 Diagram Alir program	30

Gambar 3. 9	Tampilan <i>software</i> Aec_isp	31
Gambar 3. 10	Diagram Alir menampilkan karakter pada LCD	34
Gambar 3. 11	Diagram alir pembacaan dan penulisan pada internal EEPROM	35
Gambar 3. 12	Diagram alir pembacaan data pada serial EEPROM	36
Gambar 3. 13	Diagram alir penulisan data pada serial EEPROM	37
Gambar 3. 14	Diagram Alir pembacaan keypad	38
Gambar 3. 15	Diagram Alir pengecekan nomor PIN	39
Gambar 3. 16	Diagram alir pergantian nomor PIN	40
Gambar 3. 17	Diagram alir jam digital	41
Gambar 3. 18	Diagram alir penghitung tanggal	43
Gambar 4. 1	Diagram Blok Sistem Mesin presensi Digital	44

BAB I

PENDAHULUAN



1.1. Latar Belakang Masalah

Dewasa ini perkembangan dunia teknologi elektronika khususnya teknologi mikrokontroler mengalami kemajuan yang pesat, hal itu dapat diketahui dari banyaknya peralatan-peralatan elektronik yang mengaplikasikan teknologi mikrokontroler. Perancangan mesin presensi ini juga menggunakan aplikasi dari mikrokontroler khususnya mikrokontroler ATMEL. Pemilihan judul tersebut karena sistem presensi karyawan pada beberapa perusahaan/instansi baik pemerintah maupun swasta saat ini masih menggunakan sistem manual. Sistem manual ini sangat rentan akan terjadinya kesalahan pada penyimpanan data masuk dan keluar dari masing-masing karyawan yang disebabkan karena faktor lupa, atau kertas presensi rusak/hilang.

Dengan adanya perancangan mesin presensi digital ini diharapkan permasalahan-permasalahan pada sistem presensi manual dapat diatasi. Hal itu dapat dilakukan karena pada perancangan ini dilengkapi dengan memori penyimpan secara permanen yang menggunakan teknologi serial EEPROM (*Electrical Erasable Programmable Read Only Memory*). EEPROM dapat menyimpan data masuk dan keluar dari masing-masing karyawan selama 1 tahun. Pengendali alat ini menggunakan mikrokontroler AT89S8252. Mikrokontroler AT89S8252 merupakan mikrokontroler 8 bit, penggunaan mikrokontroler ini didasarkan pada

kebutuhan dan harga yang ekonomis. Kelebihan mikrokontroler AT89S8252 dengan mikrokontroler ATMEL keluarga AT 89Cxx adalah fasilitas ISP dan internal EEPROM sebesar 2 kbyte. Fasilitas ISP dapat digunakan untuk melakukan pemrograman ke *chip* langsung secara serial dan internal EEPROM digunakan untuk menyimpan data nomor PIN untuk 10 karyawan.

1.2. Rumusan Masalah

Dalam pembuatan alat ini masalah yang dibahas yaitu bagaimana cara kerja alat Mesin presensi Digital Berbasis Mikrokontroler AT89S8252 yang mana penyimpanan data presensi dari masing-masing karyawan dilakukan setiap saat dan dapat ditampilkan ke LCD, penyimpanan data presensi karyawan dilakukan selama 1 tahun karena biasanya perusahaan/instansi melakukan evaluasi terhadap karyawannya setiap 1 tahun.

1.3. Batasan Masalah

Penulisan laporan Tugas Akhir dengan judul “Mesin Presensi Digital Berbasis Mikrokontroler AT89S8252” ini dibatasi pada masalah-masalah sebagai berikut:

1. Jumlah karyawan yang dapat mengakses alat ini sebanyak 10 orang karyawan.

2. Sistem ini dirancang untuk mencatat/menyimpan data masuk dan keluar setiap saat, data yang disimpan dalam EEPROM hanya bisa ditampilkan untuk data selama satu tahun
3. Pengendali alat ini menggunakan mikrokontroler AT89S8252.

1.4. Sistematika Penulisan

Sistematika penulisan Tugas Akhir dengan judul “Mesin Presensi Digital berbasis Mikrokontroler AT89S8252” ini adalah sebagai berikut:

BAB I Pendahuluan

Bab ini berisi latar belakang masalah, rumusan masalah, batasan masalah, dan sistematika penulisan.

BAB II Dasar teori

Bab ini membahas semua teori yang merupakan acuan perancangan dan pembuatan rangkaian dijelaskan pada bab II ini, yang meliputi penjelasan tentang gambar alat, sifat, karakteristik, dan kegunaan komponen-komponen pendukung.

BAB III Perancangan alat

Bab ini membahas tentang diagram blok sistem dan prinsip kerja masing-masing bagian alat yang dirancang dijelaskan secara lengkap dalam bab III, baik perancangan perangkat keras maupun perangkat lunaknya.

BAB IV Analisa dan pembahasan

Pada bab IV ini membahas tentang hasil-hasil pengamatan yang diperoleh dari uji coba peralatan serta perancangan perangkat lunak (*software*) yang digunakan untuk pemantau hasil pengukuran alat.

BAB V Penutup

Bab ini merupakan bab akhir penyusunan laporan tugas akhir, yang berisi tentang kesimpulan dari hasil pengamatan dan saran-saran dari penyusun guna perbaikan dan pengembangan alat ini.



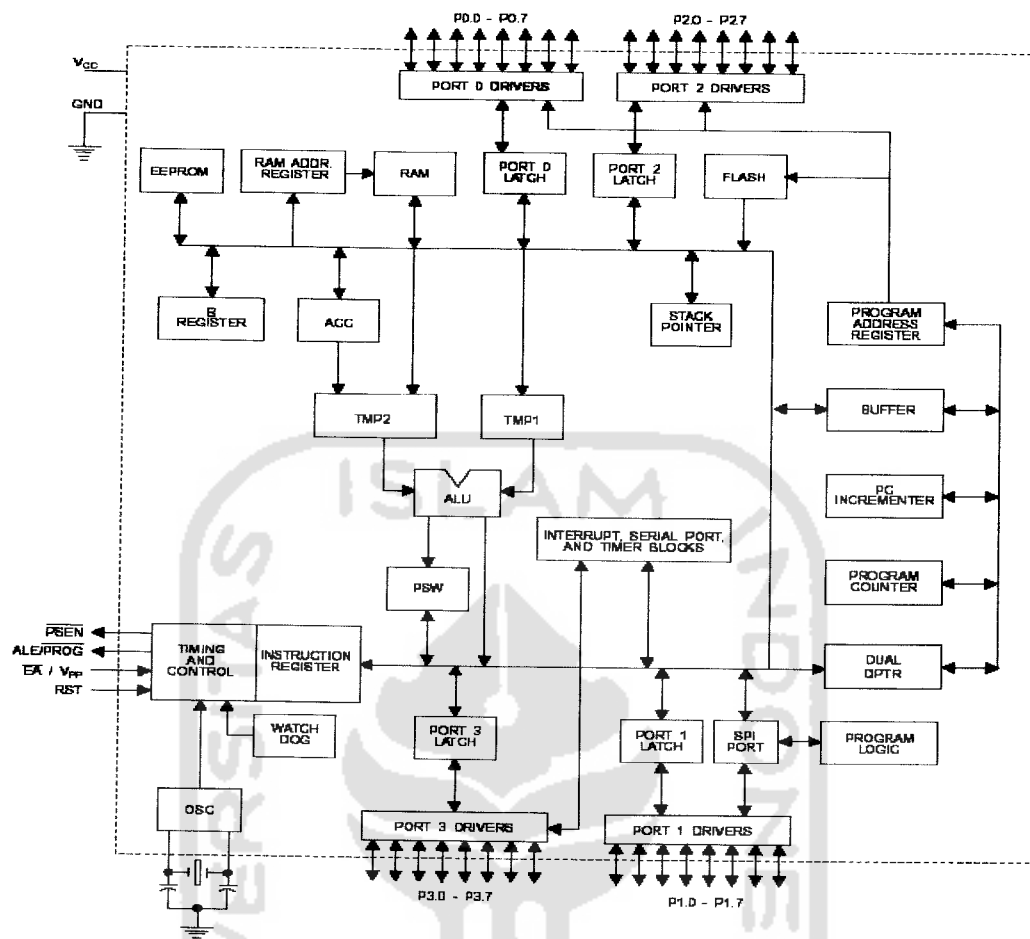
BAB II

DASAR TEORI

Mesin presensi digital memiliki beberapa bagian yaitu masukan, pengendali, penyimpanan dan keluaran. Bagian masukan terdiri dari papan tombol *keypad* 3 x 4, bagian pengendali menggunakan mikrokontroler AT89S8252, bagian penyimpanan menggunakan serial EEPROM AT24C256 dan bagian keluaran atau penampil berupa LCD 16 x 2 karakter.

2.1. Mikrokontroler AT89S8252

Mikrokontroler AT89S8252 merupakan mikrokontroler 8 bit yang memiliki spesifikasi sebagai berikut 8 kbytes *Flash memory*, 256 byte memori data, 2 kbytes EEPROM, 32-bit *Programmable I/O*, 3 buah *Timer/Counter* 16-bit, *Programmable UART serial channel*, *SPI Serial Interface*. Diagram blok mikrokontroler AT89S8252 ditunjukkan pada Gambar 2.1 berikut



Gambar 2. 1. Diagram blok mikrokontroler AT89S8252

2.1.1. Memori

Memori merupakan serangkaian piranti elektronis yang digunakan untuk menyimpan informasi data maupun program dari saluran masukan suatu sistem. Pada mikrokontroler AT89S8252 terdapat tiga bagian memori yaitu memori data, memori program dan EEPROM

2.1.1.1. Memori Data

Memori ini memiliki kapasitas 256 byte terbagi atas 128 byte RAM bawah dan 128 byte RAM atas. RAM bawah dipetakan pada alamat 00H – 7FH dan dapat diakses menggunakan pengalamatan langsung serta pengalamatan tidak langsung, sedangkan RAM atas dipetakan pada 80H – FFH dan hanya dapat diakses dengan pengalamatan tidak langsung. Alamat RAM atas sama dengan alamat SFR (*Special Function Register*)

2.1.1.2. Memori Program

Memori program sering juga disebut dengan *Flash Memory* dengan kapasitas sebesar 8 kbyte yang hanya digunakan untuk membaca saja. Memori ini dapat diakses pada alamat 0000H – 1FFFH.

2.1.1.3. EEPROM

EEPROM dan memori data eksternal pada AT89S8252 hanya dapat diakses dengan pengalamatan tidak langsung dengan perintah MOVX data 8 bit dan 16 bit. Memori EEPROM berada pada alamat 000H-7FFH. Register WMCON (96H) merupakan *Special Function Register* yang digunakan untuk mengakses memori EEPROM, memori data eksternal dan *watchdog timer*. Untuk mengakses memori EEPROM, register bit EEMEN diset 1 sedangkan untuk mengakses memori data eksternal, register bit EEMEN = 0. untuk penulisan

EEPROM register bit EEMWE diset = 1 seperti yang ditunjukkan pada Tabel 2.1 dibawah ini.

Tabel 2. 1. Register WMCON

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

WMCON alamat = 96H

Reset Value = 00000010B

Keterangan:

1. PS2, PS1, PS0

Bit Prescaler untuk *watchdog timer*, jika tiga bit ini bernilai '0' periode *watchdog timer*-nya 16 ms sedangkan jika '1' periodenya 2048 ms.

2. EEMWE

EEPROM Data Memory Write Enable bit, digunakan untuk penulisan EEPROM set '1' sebelum inisialisasi untuk penulisan data pada EEPROM dengan instruksi MOVX dan '0' setelah penulisan EEPROM.

3. EEMEN

Internal EEPROM Access Enable, jika bit EEMEN = 1 instruksi MOVX mengakses data *internal memory* dan jika bit EEMEN = 0 mengakses data *external memory*.

4. DPS

Data Pointer Register Select, bit DPS = 0 berarti *select register bank* yang pertama dan bit DPS = 1 *select* yang kedua.

5. WDTRST RDY/BSY

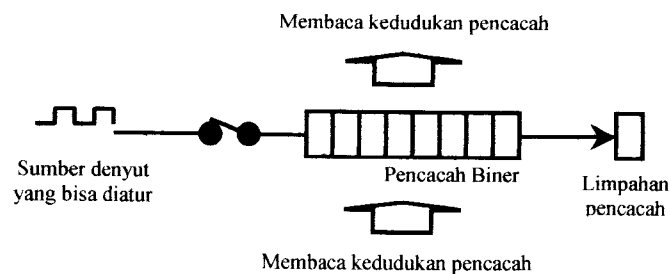
Watchdog Timer Reset and EEPROM Ready/Busy Flag, kondisi awal bit ini '1' untuk *reset watchdog timer* kemudian secara otomatis menjadi '0' pada instruksi selanjutnya, dan saat penulisan EEPROM RDY/BSY = 1 berarti EEPROM siap menulis data sedangkan RDY/BSY = 0 berarti operasi program sibuk dan secara otomatis mereset ke '1' saat program sudah selesai.

6. WDTEN

Watchdog Timer Enable Bit, WDTEN = 1 *Enable Watchdog Timer*, dan WDTEN = *Disable Watchdog Timer*.

2.1.2. Timer/Counter

Pada dasarnya sarana masukan yang satu ini merupakan seperangkat pencacah biner yang terhubung langsung ke saluran data mikrokontroler sehingga mikrokontroler bisa membaca kondisi pencacah dan bila diperlukan dapat pula mengubah kondisi pencacah tersebut. Seperti layaknya pencacah biner saat sinyal *clock* yang diberikan sudah melebihi kapasitas pencacah maka pencacah akan memberikan sinyal *overflow*, seperti yang ditunjukkan pada Gambar 2.2 berikut:



Gambar 2. 2. Konsep dasar *Timer/Counter* pada AT89S8252

Sinyal *clock* yang diberikan ke pencacah dibedakan menjadi 2 macam yaitu sinyal *clock* dengan frekuensi tetap dan sinyal *clock* dengan frekuensi bervariasi. Jika sebuah pencacah bekerja dengan frekuensi tetap pencacah tersebut bekerja sebagai *timer* atau pewaktu, karena pencacah tersebut setara dengan waktu yang bisa ditentukan secara pasti sedangkan jika sebuah pencacah bekerja dengan frekuensi bervariasi, pencacah tersebut bekerja sebagai *counter* atau pencacah.

Mikrokontroler AT89S8252 memiliki 3 perangkat *timer/counter* yaitu *timer 0*, *timer 1*, dan *timer 2*. Untuk mengakses *timer/counter* digunakan register khusus yang tersimpan dalam SFR. Pencacah biner *timer 0* diakses melalui register TL0 (*Timer 0 low byte*, pada alamat 6Ah) dan register TH0 (*Timer 0 high byte*, pada alamat 6Ch). Pencacah biner *timer 1* diakses melalui register TL1 (*Timer 1 low byte*, pada alamat 6Bh) dan register TH1 (*Timer 1 high byte*, pada alamat 6Dh). Untuk mengatur kerja *timer/counter* tersebut digunakan 2 register tambahan yaitu register TCON (*Timer control register* pada alamat 88h) dan register TMOD (*Timer mode register* pada alamat 89h).

Timer 2 merupakan sebuah *timer/counter* 16 bit. Register-register yang digunakan untuk mengakses *timer 2* adalah TL2 dan TH2, register kontrol *timer* T2CON dan *register-register capture*: RCAP2L dan RCAP2H.

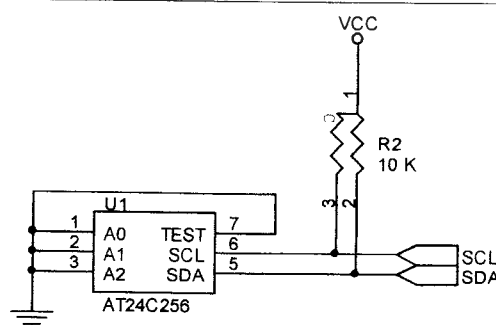
2.2. Serial EEPROM

Serial EEPROM yang digunakan dalam tugas akhir ini adalah ATMEL serial EEPROM AT24C256 yang merupakan memori serial dengan teknologi I2C

yang memiliki kapasitas sebesar 32 kbyte. Tipe I2C adalah teknologi komunikasi serial yang ditemukan oleh Philips pada tahun 1992 dan direvisi hingga versi 2.1 yang terbaru pada tahun 2000. Teknologi ini hanya menggunakan 2 buah jalur I/O yaitu SDA dan SCL. SDA merupakan jalur data pada komunikasi I2C sedangkan SCL merupakan jalur *clock* di mana sinyal *clock* akan selalu muncul untuk setiap *bit* dari pengiriman data. Komunikasi I2C diciptakan oleh Philips tidak hanya untuk serial EEPROM melainkan juga diperuntukkan bagi komponen-komponen lain yang mempunyai kemampuan untuk diakses dengan 2 buah jalur I/O SDA dan SCL. Oleh karena itu, untuk membedakan antara Serial EEPROM dengan komponen-komponen yang lain digunakan *Slave Address* yang menunjukkan identitas dari komponen tersebut. Dalam hal ini Serial EEPROM mempunyai kode 1010. Tabel 2.2 dan Gambar 2.3 berikut menunjukkan konfigurasi Pin EEPROM AT24C256.

Tabel 2. 2. Konfigurasi Pin EEPROM AT24C256

Nama Pin	Difinisi
Vcc	Power supply
Gnd	Ground
SCL	Input serial clock
SDL	Input/output serial data
NC	No conection



Gambar 2. 3. Konfigurasi kaki AT24C256

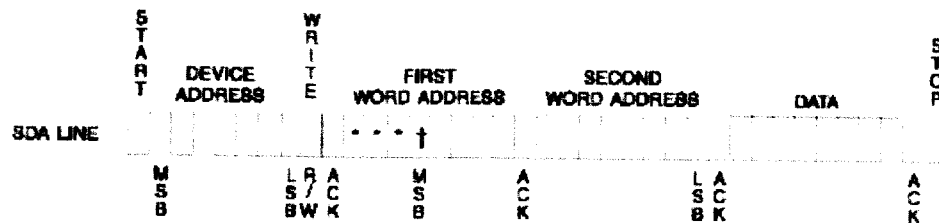
Serial EEPROM I2C berdasarkan pengalamatannya terdiri dari 2 jenis yaitu pengalamatan 8 bit yang digunakan untuk Serial EEPROM dengan kapasitas memori sebesar 128 byte hingga 2Kb dan pengalamatan 16 bit untuk Serial EEPROM 4Kb hingga 512Kb. Pada pengalamatan 16 bit terdapat 3 buah paket 8 bit data yang harus dikirimkan ke Serial EEPROM yaitu: *Control byte*, *High Byte Address* dan *Low Byte Address*, sedangkan pada pengalamatan 8 bit hanya diperlukan 2 buah paket 8 bit data yaitu *Control Byte* dan *Byte Address* saja. *Control byte* terdiri dari *Slave Address*, *Device Select* Bus bit-bit pengatur alamat dari Serial EEPROM dalam satu jalur *bus*, R/W bit penentu proses penulisan atau pembacaan data dari Serial EEPROM.

Pembacaan maupun penulisan data ke dalam Serial EEPROM, selalu diawali dengan pengiriman *Control Byte* dan *Address Byte*. Hanya pada penulisan data akan dilanjutkan dengan pengiriman data 8 bit, sedangkan sebaliknya pada pembacaan akan dilanjutkan dengan pengambilan data 8 bit. Bit R/W pada *Control Byte* akan berlogika 1 untuk pembacaan data dan berlogika 0 untuk penulisan data

2.2.1. Operasi Penulisan

Penulisan data pada Serial EEPROM I2C dapat dilakukan secara *byte* maupun secara *page*. Pada penulisan secara *byte* dilakukan dengan mengirimkan *control byte*, alamat tujuan dan data sedangkan pada penulisan secara *page* dilakukan hanya dengan mengirimkan alamat tujuan awal saja yang kemudian

dilanjutkan dengan 32 *byte* data yang akan menempati lokasi secara berurutan mulai dari alamat tujuan awal.



Gambar 2. 4. Penulisan secara *byte*



Gambar 2. 5. Penulisan secara *page*

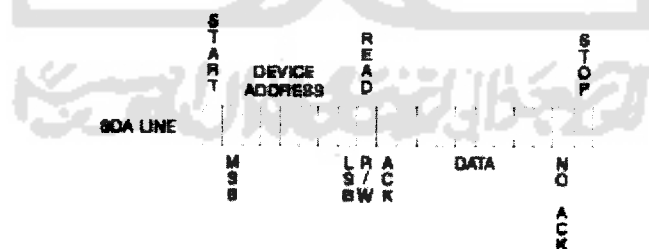
2.2.2. Operasi Pembacaan

Pembacaan data dapat dilakukan secara *Current Address Read* (pembacaan alamat saat ini) maupun *random read* (pembacaan secara acak). Pada *Current Address Read*, data yang dibaca adalah data pada alamat yang terakhir kali diakses saat itu, sedangkan pada pembacaan secara acak dilakukan dengan mengirimkan *Control Byte* dan alamat tujuan terlebih dahulu. Untuk pembacaan secara *sequential*, dilakukan dengan *control byte* dan dilanjutkan dengan data-data yang berada mulai dari alamat yang terakhir diakses saat itu berturut-turut hingga sinyal *stop bit* dikeluarkan.

2.2.2.1. Pembacaan alamat saat ini

Di dalam EEPROM terdapat *counter* alamat yang menyimpan informasi alamat terakhir yang diakses baik oleh operasi pembacaan data maupun operasi penulisan data. *Counter* alamat ini selalu bertambah satu dan akan tetap valid selama catu daya terpasang. *Counter* ini akan menggulung ke alamat *byte* awal pada *page* awal dari *byte* akhir pada *page* terakhir pada saat operasi pembacaan data. Untuk operasi penulisan, *counter* alamat ini akan menggulung ke alamat *byte* awal pada *page* yang sama.

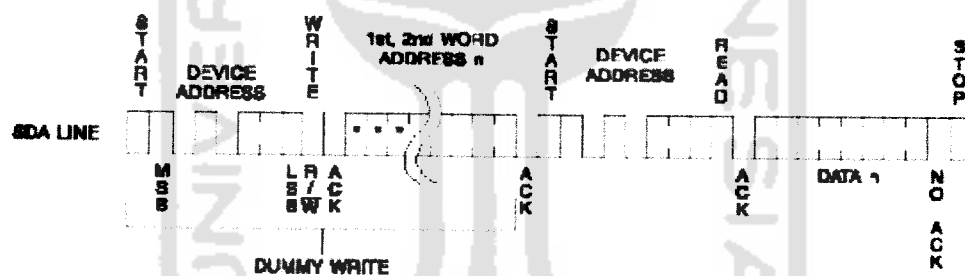
Jika pada piranti EEPROM ini diberikan perintah untuk pembacaan data, maka data *byte* yang ada pada alamat aktual yang ditunjukkan oleh *counter* alamat akan dikirim keluar oleh EEPROM. Dalam hal ini, mikrokontroler tidak akan merespon dengan mengirim bit *acknowledge* nol, namun akan langsung membangkitkan bit kondisi *stop*. Untuk lebih jelasnya, dapat dilihat pada Gambar di bawah ini.



Gambar 2. 6. Operasi Pembacaan Alamat saat ini

2.2.2.2. Pembacaan Acak

Proses pembacaan acak memerlukan urutan penulisan urutan *Control Byte* untuk menyimpan word alamat dari *byte* data yang ingin dibaca. Setelah serial EEPROM menerima urutan *Control Byte*, maka dia akan mengirimkan bit *acknowledge*, mikrokontroler harus mengirimkan bit kondisi *start* dan kemudian mengirimkan word alamat yang sama dengan bit *read* logika 1. Karena *counter* alamat sudah diset sebelumnya, maka pembacaan akan terjadi pada alamat tersebut. Setelah data *byte* dikirimkan, mikrokontroler tidak merespon dengan bit *acknowledge* nol, namun langsung membangkitkan bit kondisi *stop*. Untuk lebih jelasnya, dapat dilihat pada Gambar di bawah ini.

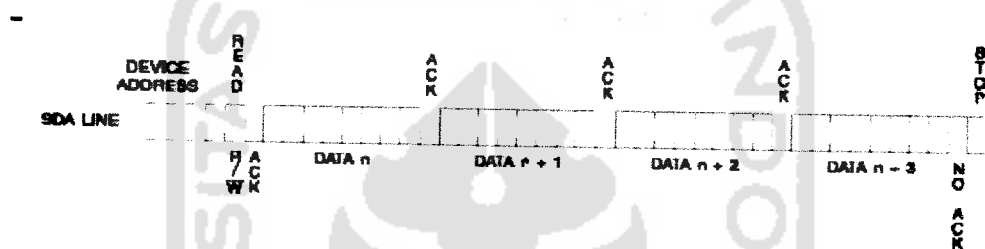


Gambar 2. 7. Operasi Pembacaan Acak

2.2.2.3. Pembacaan Berurutan

Proses pembacaan berurutan dinisialisasi dengan pembacaan alamat aktual (*current address read*) atau pembacaan acak (*random read*). Setelah mikrokontroler menerima sebuah word data, dia akan merespon dengan mengirimkan bit *acknowledge*.

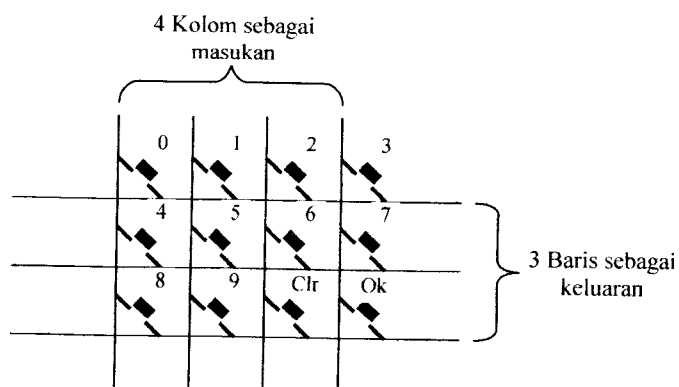
Apabila EEPROM menerima bit *acknowledge* ini, dia akan menambah satu *word* alamat dan secara serial akan mengirimkan word-word data. Jika batas alamat memori dicapai, word alamat akan menggulung ke awal dan kemudian meneruskan proses pembacaan ini. Proses pembacaan secara berurutan ini baru akan berhenti jika mikrokontroler tidak merespon data yang diterima dengan bit *acknowledge* nol namun langsung mengirimkan bit kondisi *stop*. Untuk lebih jelasnya, dapat dilihat pada Gambar di bawah ini.



Gambar 2. 8. Operasi Penulisan Berurutan

2.3. Keypad matriks 3 x 4

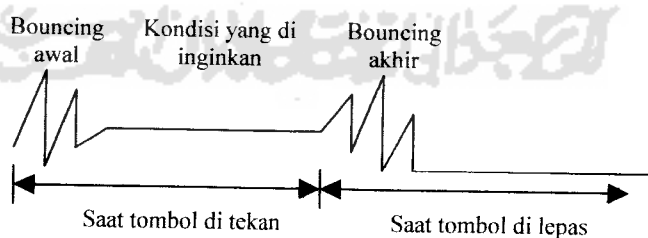
Keypad adalah kumpulan dari beberapa tombol yang tersusun sedemikian rupa sehingga membentuk baris dan kolom yang masing-masing dihubungkan dengan Port I/O mikrokontroler. Tombol-tombol tersebut dihubungkan sedemikian rupa sehingga apabila salah satu saklar ditekan maka akan menghubungkan salah satu baris dengan salah satu kolomnya. Seperti pada Gambar 2.6 dibawah ini



Gambar 2. 9. Keypad matrik 3 x 4

Pada saklar mekanis hal yang perlu dipertimbangkan adalah lentingan (*Bouncing*). Pada saat saklar ditekan dan dilepas terjadi *bouncing* yang menyebabkan logika saklar tidak bisa diperkirakan.

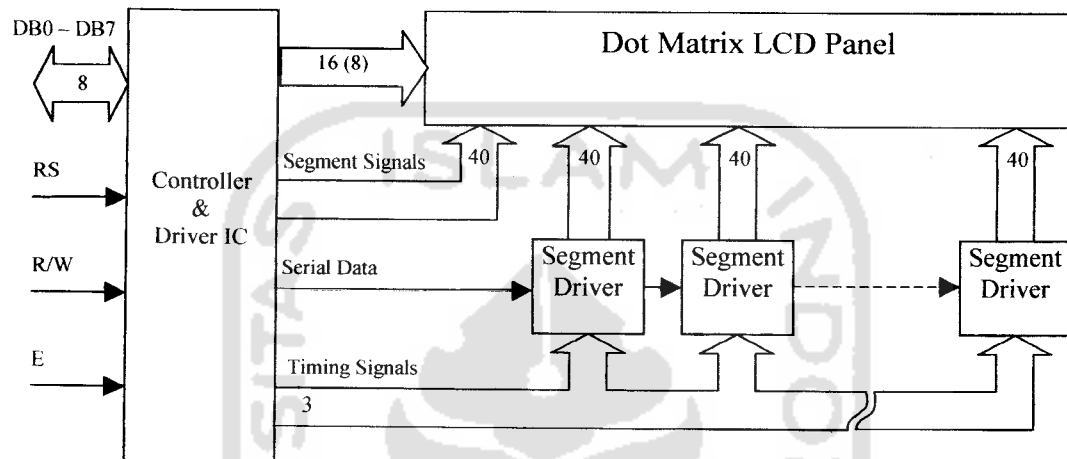
Misalkan seperti Gambar di atas saat penekanan tombol kondisi yang diharapkan adalah '0' tetapi karena adanya efek *bouncing* kondisi yang diharapkan tersebut tidak dapat diperkirakan. Untuk mengatasi hal tersebut maka pembacaan data saklar dilakukan setelah selang waktu tertentu. Untuk saklar mekanis biasanya sekitar 10 – 20 ms. Gambar 2.10 di bawah ini menunjukkan efek *bouncing* dari saklar mekanis.



Gambar 2. 10. Tampilan efek *Bouncing*

2.4. Penampil LCD

LCD adalah suatu *display* dari bahan cairan kristal yang pengoperasiannya menggunakan sistem *dot matriks* Gambar 2.8 di bawah ini menunjukkan diagram blok LCD *dot matriks*



Gambar 2. 11. Diagram Blok LCD *dot matriks*

Dari diagram blok LCD diatas dapat dijelaskan bagian-bagian LCD sebagai berikut :

DB0 – DB7 : sebagai I/O data

E : sebagai input, diaktifkan *high* kemudian *low* untuk meng-*clock*-kan satu data.

R/W : sebagai input, R=1, W=0, diaktifkan *low*.

RS : sebagai input, diaktifkan *low*.

0 : *register* instruksi (*write*); *busy flag* dan alamat *counter* (*read*).

1 : data *register* (*write* dan *read*)

LCD yang digunakan dalam penelitian ini adalah LCD *type* HD44780U yang memiliki beberapa bagian sebagai berikut :

2.4.1. Register

LCD HD44780U memiliki dua register 8 bit yaitu register instruksi (IR) dan register data (DR), IR menyimpan kode instruksi, menampilkan *clear* dan pergeseran *cursor* dan informasi alamat untuk *display* data RAM (DDRAM) dan pembangkit karakter RAM (CGRAM). Register instruksi IR hanya bisa dibaca dari MPU, sedangkan register data (DR) digunakan untuk menyimpan data yang akan ditulis dan dibaca pada DDRAM atau CGRAM.

Penulisan data ke register perintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur *Address Counter* maupun *Address Data*. Kondisi RS berlogika 0 menunjukkan akses data ke Register Perintah. RW berlogika 0 yang menunjukkan proses penulisan data. Sedangkan Penulisan data pada Register Data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke Register Data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Pengiriman data dari data bus DB0 – DB7 diawali dengan pemberian pulsa logika 1 pada E Clock. dan diakhiri dengan pulsa logika 0 pada E clock

2.4.2. *Busy Flag (BF)*

Ketika *busy flag* bernilai 1 LCD menjalankan operasi internal saat $RS = 0$ dan $RW = 1$ *busy flag* merupakan keluaran ke DB7 dan untuk menuliskan instruksi selanjutnya dilakukan setelah *busy flag* bernilai 0.

2.4.3. *Address Counter (AC)*

AC berfungsi untuk menghitung alamat dari DDRAM dan CGRAM. Saat alamat dari *instruksi* dituliskan ke IR maka alamat informasi dikirim dari IR ke AC, setelah pembacaan alamat dari DDRAM atau CGRAM, *address counter* secara otomatis alamatnya langsung ditambah satu.

2.4.4. *Display Data RAM (DDRAM)*

DDRAM digunakan untuk menyimpan karakter yang akan ditampilkan. Misalnya, karakter 'A' atau 41h yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD

2.4.5. *Character Generator ROM (CGROM)*

CGROM merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi.

2.4.6. Character Generator RAM (CGRAM)

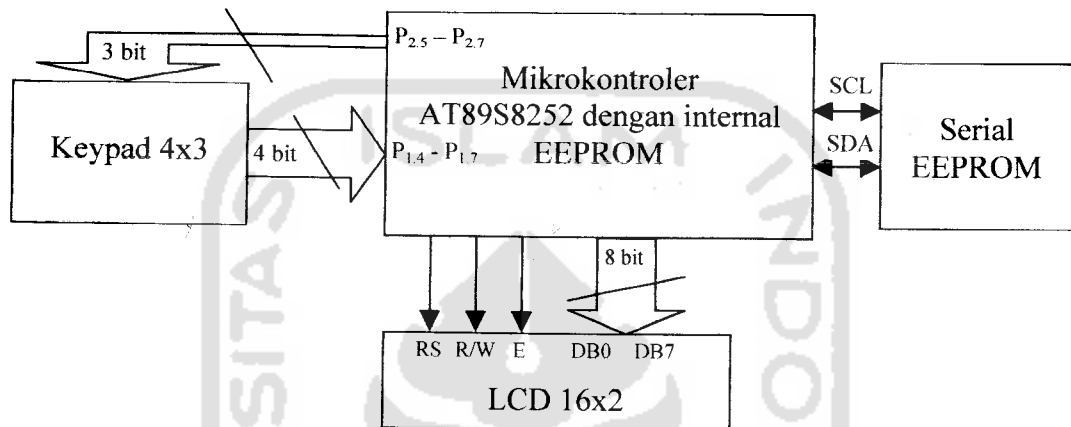
Memori ini fungsinya sama dengan CGROM tetapi pola karakter yang disimpan pada memori ini dapat diubah-ubah sesuai keinginan.



BAB III

PERANCANGAN ALAT

3.1. Perancangan Perangkat Keras

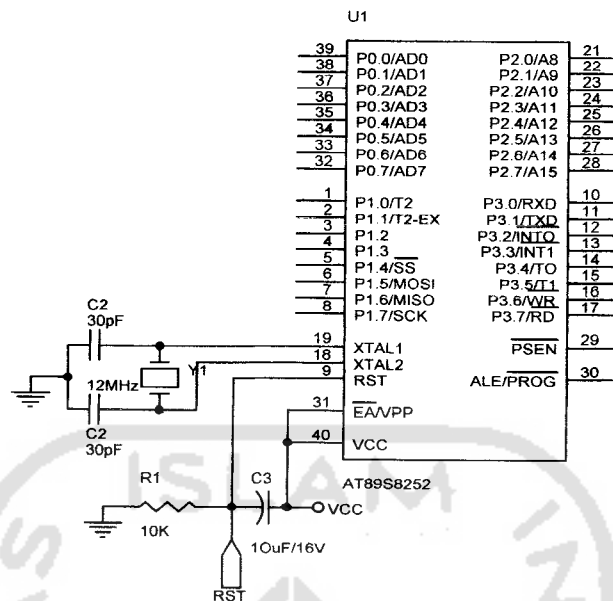


Gambar 3. 1. Diagram blok mesin presensi digital

Perangkat keras dalam perancangan tugas akhir ini meliputi perancangan rangkaian mikrokontroler AT89S8252, rangkaian *keypad* 3 x 4 dan rangkaian serial EEPROM AT24C256.

3.1.1. Rangkaian Mikrokontroler AT89S8252

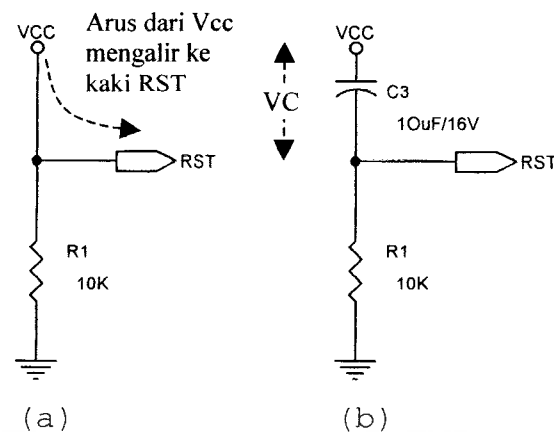
Rangkaian utama dari mikokontroler terdiri dari rangkaian osilator dan rangkaian *power on reset*. Pada Gambar 3.2 digambarkan rangkaian sistem minimum mikrokontroler AT89S8252.



Gambar 3. 2. Rangkaian sistem minimum AT89S8252

Semua mikrokontroler MCS-51 ATMEL memiliki osilator *on-chip*, yang dapat digunakan sebagai sumber detak (*clock*) ke CPU. Untuk menggunakannya hubungkan sebuah resonator kristal atau keramik di kaki-kaki XTAL1 dan XTAL2 pada mikrokontroler dan hubungkan kapasitornya ke ground. Pada rangkaian osilator ini digunakan kristal 12 MHz dengan dua buah kapasitor 30 pf. Pembangkit clock internal menentukan kecepatan siklus mesin mikrokontroler. Untuk satu siklus mesin paling lama dikerjakan dalam 12 periode osilasi atau jika digunakan osilator kristal 12 MHz periodenya sebesar 1 μ s.

.Reset pada mikrokontroler terjadi dengan adanya logika 1 (*high*) pada kaki RST. Setelah kondisi pin RST kembali 0 (*low*), mikrokontroler akan mulai menjalankan program dari alamat 0000h. Gambar 3.3 merupakan gambar rangkaian *power on reset*.



Gambar 3. 3. Rangkaian *power on reset*

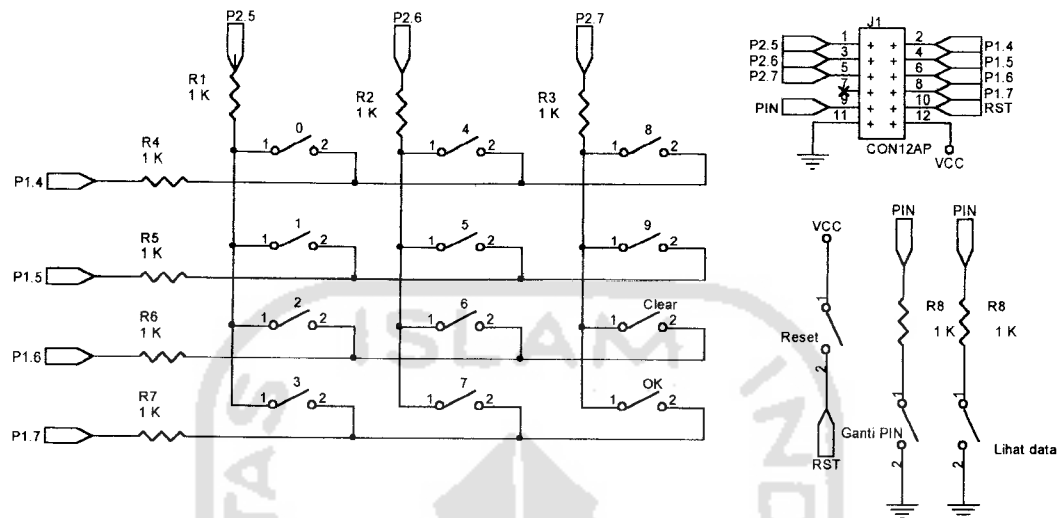
Pada saat sumber tegangan diaktifkan kapasitor terhubung singkat sehingga arus mengalir dari Vcc langsung ke kaki RST sehingga reset berlogika 1 seperti pada Gambar 3.3a, kemudian kapasitor terisi hingga tegangan pada kapasitor sama dengan Vcc pada saat ini kapasitor penuh (Gambar 3.3b). Dengan demikian tegangan reset akan turun menjadi 0 sehingga kaki RST berlogika 0.

3.1.2. Rangkaian *keypad* 3 x 4

Keypad merupakan rangkaian saklar yang berfungsi sebagai masukan. Pada alat ini digunakan papan tombol (*keypad*) matrik 3 x 4 yang mana 3 bit barisnya dihubungkan dengan, P_{2.5} - P_{2.7}, sedangkan 4 bit kolomnya dihubungkan dengan P_{1.4}–P_{1.7}. *Keypad* matrik tersebut tersusun dalam 12 tombol.

Cara kerja dari papan tombol matrik ini adalah dengan *system scanning* tombol satu persatu. Keluaran mikrokontroler akan memberikan kondisi '0' kepada salah satu port keluaran secara bergantian sehingga apabila pada saat yang bersamaan ada penekanan tombol maka pada salah satu port masukan akan

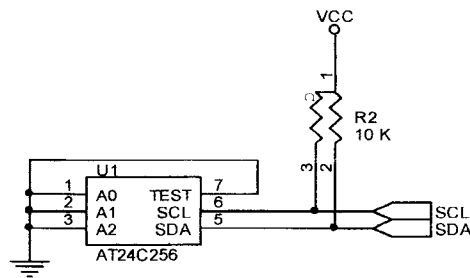
memiliki kondisi yang sama yaitu '0'. Rangkaian *keypad* matrik 3 x 4 ditunjukkan pada Gambar 3.4



Gambar 3. 4. Gambar *keypad* matrik 3 x 4 dan rangkaian tombol

3.1.3. Rangkaian Serial EEPROM

Seperti yang telah dijelaskan pada bab II bahwa dalam perancangan tugas akhir ini digunakan serial EEPROM AT24C256 yang menggunakan 2 buah jalur I/O yaitu SDA dan SCL. Kaki SDA dan SCL secara normal di-*pullup* dengan resistor. Data pada kaki SDA ini hanya boleh berubah pada saat periode rendah (*low*) dari *clock*. Jika perubahan data ini terjadi pada saat *clock* tinggi, maka ini akan mengindikasikan kondisi *start* dan *stop* dari transmisi data serial. Rangkaian serial EEPROM ditunjukkan pada Gambar 3.5 di bawah ini :



Gambar 3. 5. Rangkaian serial EEPROM

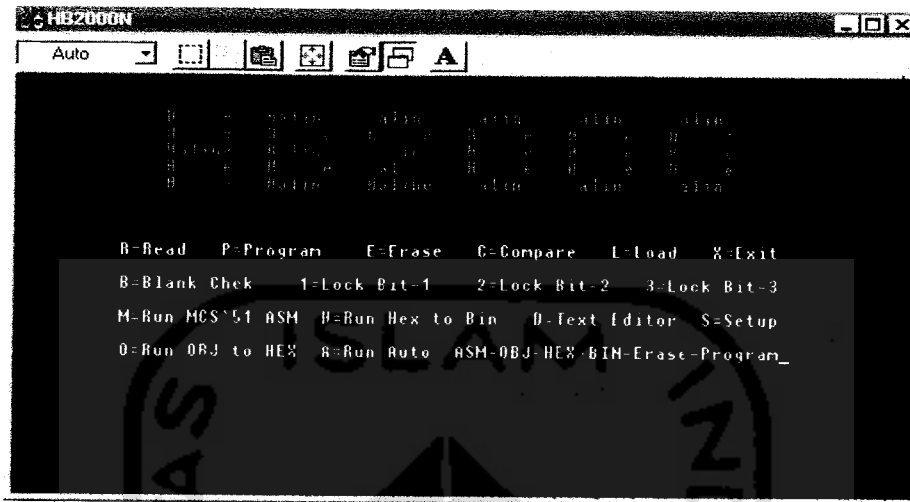
3.2. Perangkat Lunak

Perancangan perangkat lunak dalam tugas akhir ini terdiri dari cara pembuatan program, perancangan program dan perancangan *downloader* ke mikrokontroler AT89S8252.

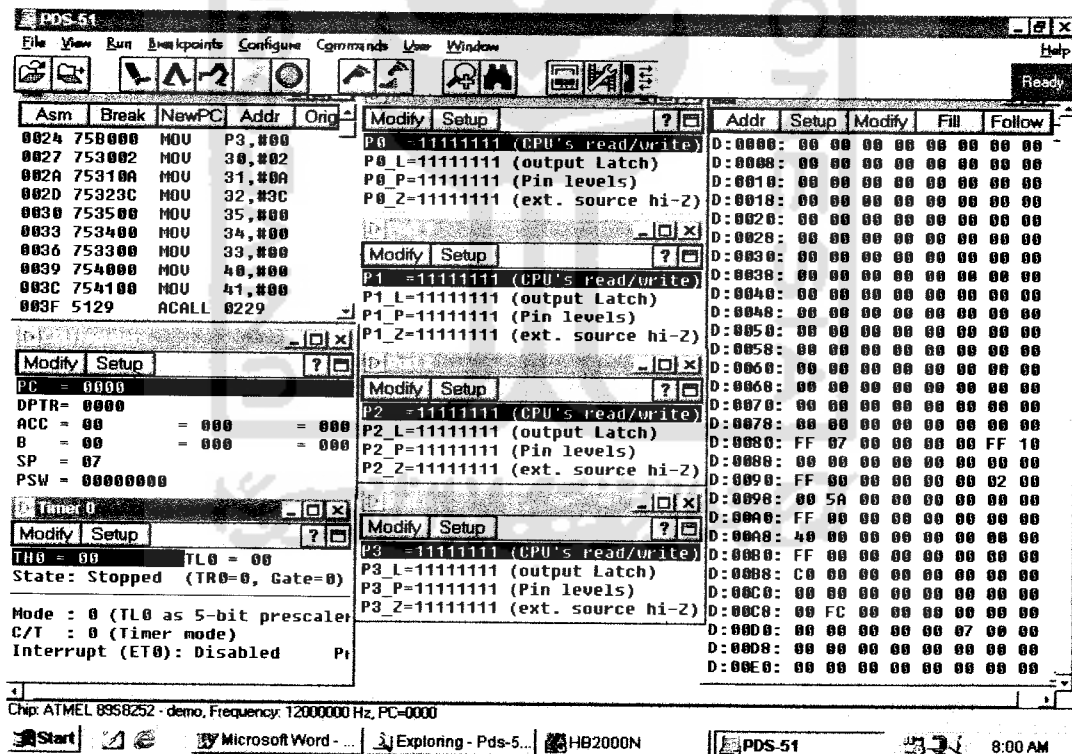
3.2.1. Cara Pembuatan Program

Program mesin presensi digital dibuat dalam bahasa *assembler* keluarga MCS-51 yang diketik dengan sembarang editor teks dan disimpan dengan ekstensi *.asm atau *.h51. Setelah nama filenya tersimpan maka lakukan kompilasi dengan HB 2000 seperti tampilan pada Gambar 3.6 di bawah. Untuk melakukan kompilasi tekan 'M' kemudian ketik nama file dalam ekstensi *.asm atau *.h51 kemudian enter, jika ada kesalahan akan ditunjukkan dengan membuka nama file yang berekstensi *.lst. Untuk mengubah ke format file *.obj e format heksa dilakukan dengan menekan 'O' ketik nama file *.obj sehingga diperoleh file dengan ekstensi *.hex, untuk program *easy programmer* yang dibutuhkan adalah file dalam format intel *heksa* sehingga hanya diperlukan file yang berekstensi

- *.hex saja. Selain cara di atas dapat juga dilakukan dengan menekan 'A' ketik file
- *.h51 lalu *enter* maka didapatkan semua format yang diperlukan (lst-obj-hex-bin).



Gambar 3. 6. Tampilan *compiler* HB 2000



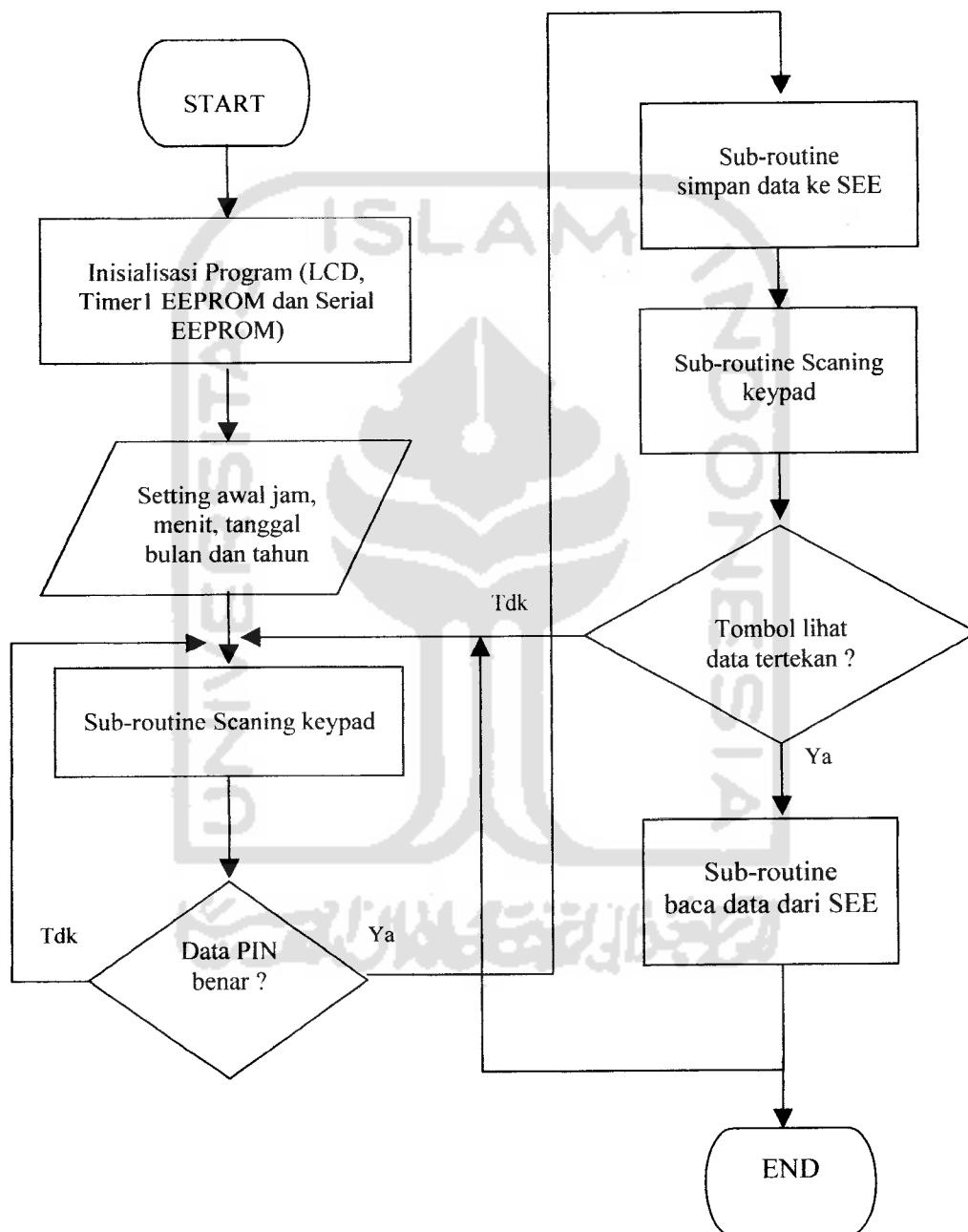
Gambar 3. 7. Emulator PDS-51

Setelah melakukan langkah-langkah tersebut di atas maka untuk melakukan simulasi program yang dibuat dapat menggunakan suatu simulator perangkat keras (*Emulator*) dengan men-*download* file *.hex, dengan simulasi perangkat keras tersebut dapat diketahui jalannya program yang dibuat dan juga dapat diketahui isi *accumulator*, *program counter*, *stack pointer* serta *register-register* lain yang digunakan dalam program yang dibuat.



3.2.2. Perancangan Program

Perancangan program mesin presensi digital dapat dilihat dalam diagram alir berikut :



Gambar 3. 8. Diagram alir program

Dari diagram alir di atas program diawali dengan inisialisasi untuk tampilan LCD, *internal* EEPROM dan serial EEPROM. Kemudian setelah inisialisasi program meminta masukan data jam, menit, tanggal, bulan dan tahun dari *keypad* dan diakhiri dengan penekanan tombol *OK*. Sehingga program siap untuk menerima masukan nomor PIN. Jika nomor pin yang dimasukkan salah, LCD akan menampilkan tulisan 'PIN Anda Salah' kemudian kembali ke program yang meminta masukan nomor PIN, sedangkan jika PIN yang dimasukkan benar LCD akan menampilkan nama *user*/karyawan 'masuk' saat masukan nomor PIN pertama dan 'keluar' untuk masukan nomor PIN kedua dan pada saat itu juga dilakukan penyimpanan data jam dan menit ke serial EEPROM yang disesuaikan dengan nomor dari masing-masing *user*/karyawan, misalnya untuk *user* pertama data disimpan mulai alamat 0000h. Apabila terjadi penekanan tombol lihat data, program meminta pemasukan tanggal, bulan dan tahun data yang ingin dilihat. Penampilan data *user* dimulai dari *user* pertam. Setelah menampilkan data 10 *user* program akan kembali pada permintaan memasukan nomor PIN, jika *user* memasukan nomor PIN untuk ketiga kalinya maka data tersebut akan diabaikan.

Dalam perancangan program juga dilengkapi dengan tombol untuk mengganti nomor PIN urutan pergantian pinnya adalah jika terjadi penekanan tombol ganti pin program meminta pemasukan PIN lama kemudian pemasukan PIN baru dan terakhir ulangi pemasukan pin yang baru.

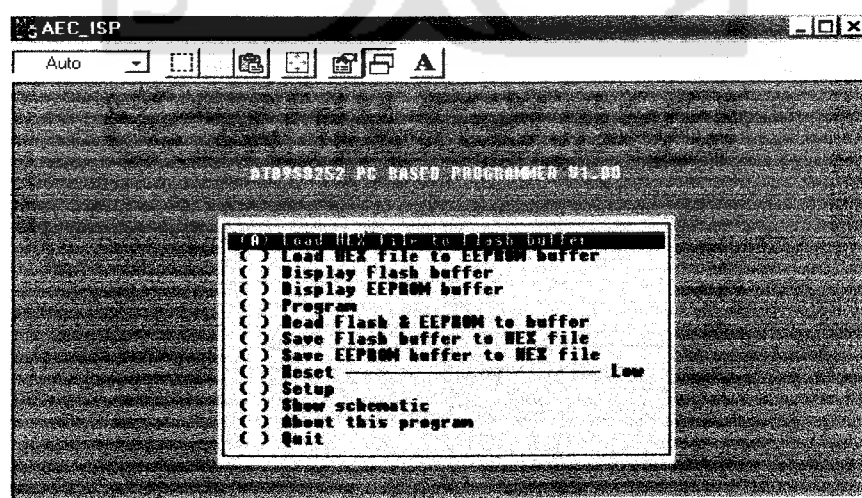
3.2.3. Downloader ke mikrokontroler AT89S8252

Mikrokontroler AT89S8252 merupakan mikrokontroler teknologi baru karena dilengkapi dengan fasilitas SPI (*Serial Peripheral Interface*) yang terdiri dari SCK/P1.7 (*serial clock*), MOSI/P1.5 (*input*) dan MISO/P1.6 (*output*) sehingga memungkinkan untuk mengisikan/men-download program secara langsung. Hubungan antara port printer dengan pin SPI mikrokontroler ditunjukkan pada Tabel 3.1 berikut

Tabel 3. 1. Hubungan Pin mikrokontroler dengan port printer

Port Printer	Pin SPI Mikrokontroler
6 (D4)	9 (Reset)
7 (D5)	6 (P1.5 – MOSI)
8 (D6)	8 (P1.7 – SCK)
10 (ACK)	7 (P1.6 – MISO)
18-20 (GND)	20 (GND)

Software untuk serial programming menggunakan software aec_isp yang memiliki tampilan seperti Gambar 3.9 berikut



Gambar 3. 9. Tampilan software Aec_isp

3.2.4. Implementasi Diagram Alir

3.2.4.1. Inisialisasi LCD

Program berikut merupakan urutan inisialisasi untuk mempersiapkan LCD bekerja 8 bit dan 16 x 2 karakter

```

mov    R1,#01h                ;menghapus tampilan LCD
lcall  write_inst
mov    R1,#00111000b          ;LCD bekerja 8 bit 2 line
lcall  write_inst
mov    R1,#00001110b          ;geser kursor
lcall  write_inst
mov    R1,#00000010b          ;mulai dari alamat 0
lcall  write_inst
mov    R1,#00000110b          ;geser cursor kekiri
lcall  write_inst

```

3.2.4.2. Inisialisasi internal EEPROM

Program inisialisasi internal EEPROM berisi pemetaan SFR WMCON yang merupakan *register control* untuk internal EEPROM seperti berikut :

WMCON	DATA	96h	; alamat register WMCON di 96h
EEMEN	Equ	00001000b	; aktifkan register EEMEN
EEMWE	Equ	00010000b	; aktifkan register EEMWE
WDTRST	Equ	00000010b	; aktifkan register WDTRST
DPS	Equ	00000100b	; aktifkan register DPS

3.2.4.3. Inisialisasi serial EEPROM

Inisialisasi serial EEPROM seperti berikut :

DSEG

Slave_AddrSEE: Ds 1 ;inisialisasi Serial EEPROM

```

DataSEE2:    Ds    1
SDA          Bit    P1.3
SCL          Bit    P1.2

```

3.2.4.4. Inisialisasi interupsi

Inisialisasi ini bertujuan untuk menentukan sumber-sumber interupsi seperti program berikut :

```

org    3h                ;External Interrupt 0 Vector
RETI
org    0Bh               ;Timer 0 Interrupt Vector
RETI
org    13h               ;External Interrupt 1 Vector
RETI
org    1Bh               ;Timer 1 Interrupt Vector
LJMP   Timer1inter
org    23h               ;Serial Interrupt Vector
RETI

```

3.2.4.5. Inisialisasi Timer 1

Inisialisasi timer 1 bertujuan untuk mengaktifkan timer 1 seperti berikut :

```

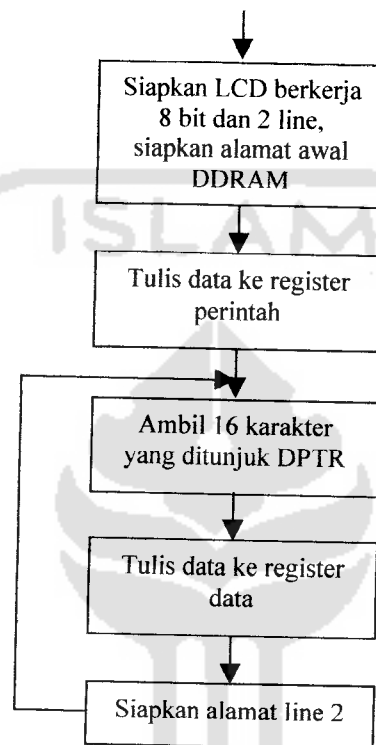
mov    TMOD,#00010000b    ;timer 1 bekerja pada mode 1
MOV    TH1,#high (-50000)  ; siapkan waktu tunda 50 mili detik
MOV    TL1,#low (-50000)
setb   ET1                ; aktifkan interupsi timer 1
SETB   EA                 ; aktifkan sistim interupsi
setb   TR1                ; jalankan timer 1

```

3.2.4.6. Program menampilkan karakter pada LCD

Untuk menuliskan data ke register perintah pada LCD pin RS '0', R/W berlogika '0' dan pin E diberi *clock* dengan program *delay* sedangkan untuk menuliskan data ke register data RS '1', R/W '0' dan pin E diberi *clock* dengan

program *delay*. Pada perancangan ini LCD diaplikasikan bekerja 8 bit untuk 2 line. Sehingga data dapat langsung dituliskan pada pin DB0-DB7 melalui port 3 mikrokontroler. Diagram alir untuk menampilkan karakter pada LCD seperti Gambar 3.10 berikut



Gambar 3. 10. Diagram alir menampilkan karakter pada LCD

3.2.4.7. Program membaca dan menulis pada internal EEPROM

Pembacaan dan penulisan pada internal EEPROM menggunakan register WMCON yang terdapat pada alamat 96h pada SFR. Pada saat pembacaan data bit EEMEN (bit ke-3 register WMCON) harus dalam kondisi '1' seperti yang ditunjukkan pada program berikut :

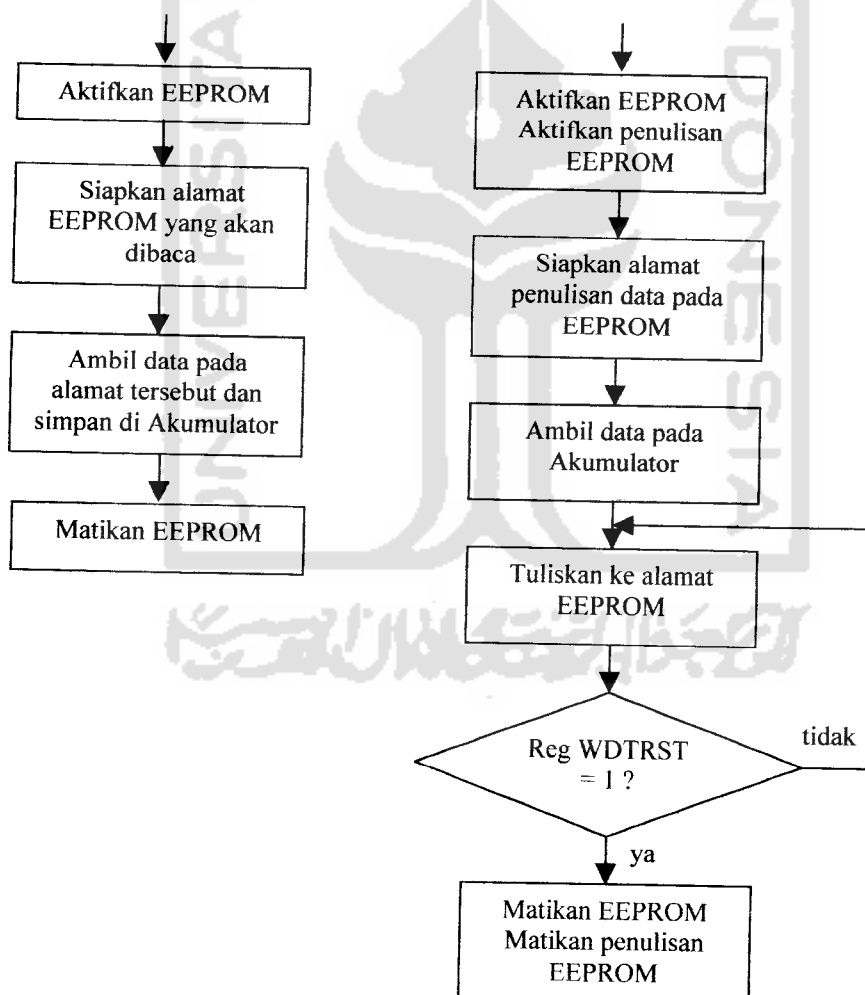
```
orl    WMCON,#EEMEN           ;aktifkan EEPROM
```

Nilai awal (*reset value*) dari register WMCON adalah 00000010b sehingga jika di-OR-kan dengan isi EEMEN 00001000b maka isi WMCON menjadi 00001010b itu berarti bit ke-3 WMCON bernilai '1', dan untuk mengembalikan kondisi WMCON pada nilai semula dilakukan dengan meng-XOR-kan WMCON dengan EEMEN seperti program berikut :

```
xrl    WMCON,#EEMEN    ;non aktifkan EEPROM
```

Sedangkan untuk penulisan data bit EEMEN dan EEMWE harus berlogika '1'.

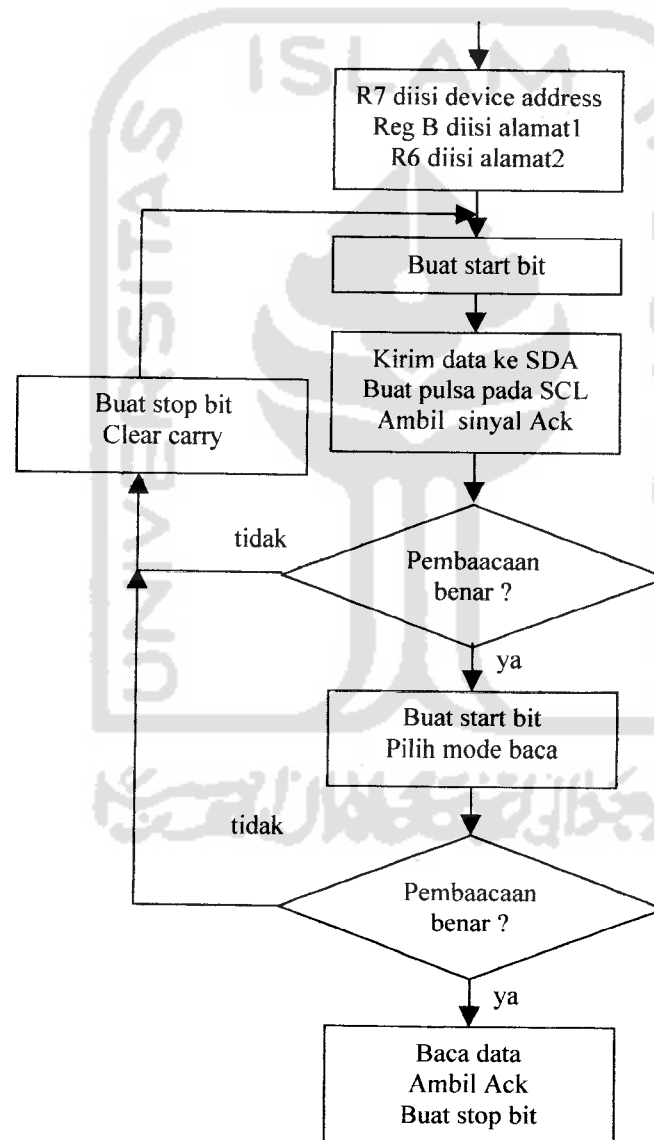
Berikut diagram alir penulisan dan pembacaan data ke internal EEPROM



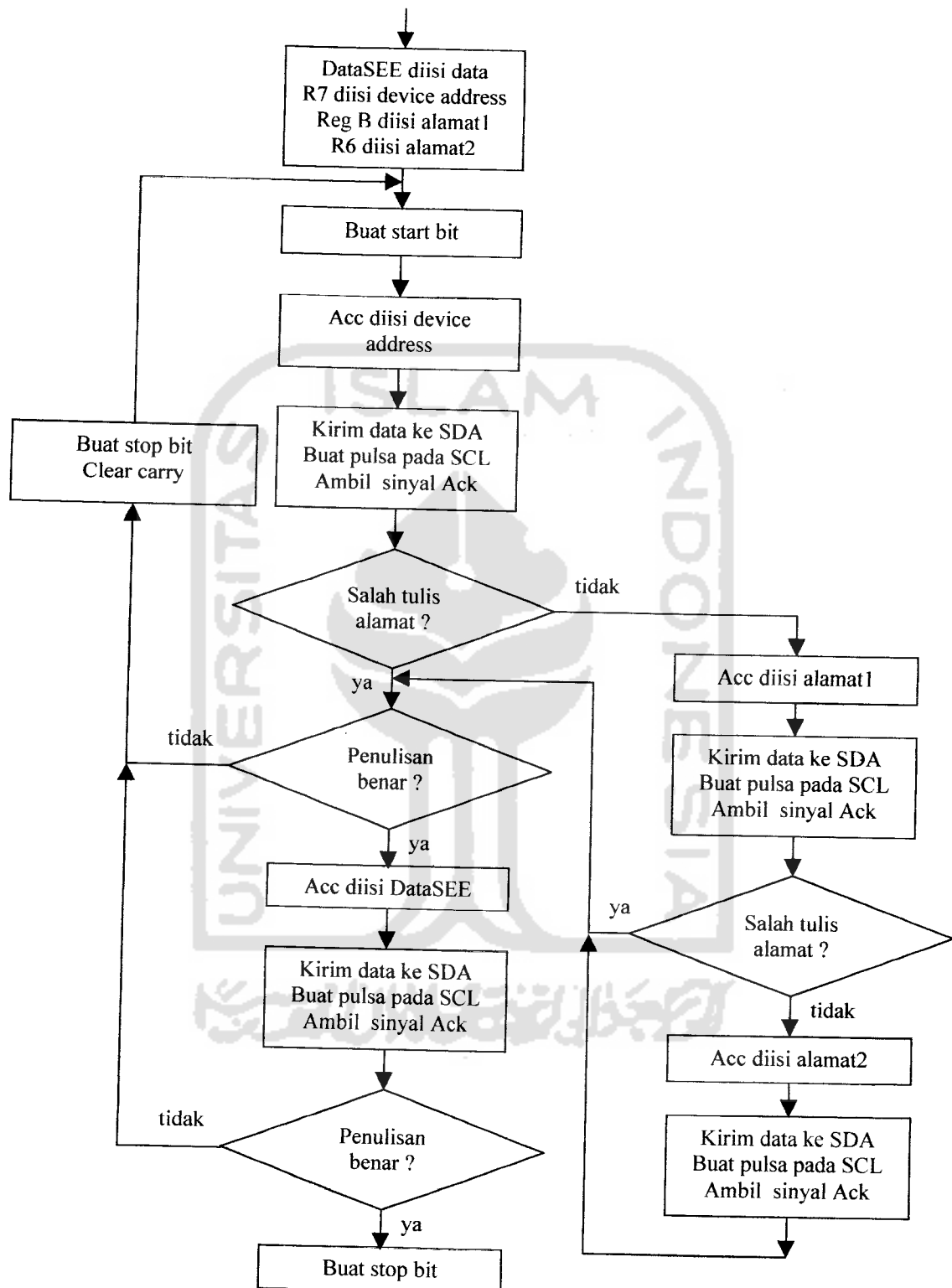
Gambar 3. 11. Diagram alir pembacaan dan penulisan pada internal EEPROM

3.2.4.8. Program Membaca dan Menulis Serial EEPROM

Pembacaan dan penulisan data pada serial EEPROM ini menggunakan sistem acak dengan DPTR sebagai pointer, Serial EEPROM yang digunakan merupakan serial EEPROM 16 bit dengan kapasitas 32 kbyte. Berikut diagram alir pembacaan dan penulisan data pada serial EEPROM



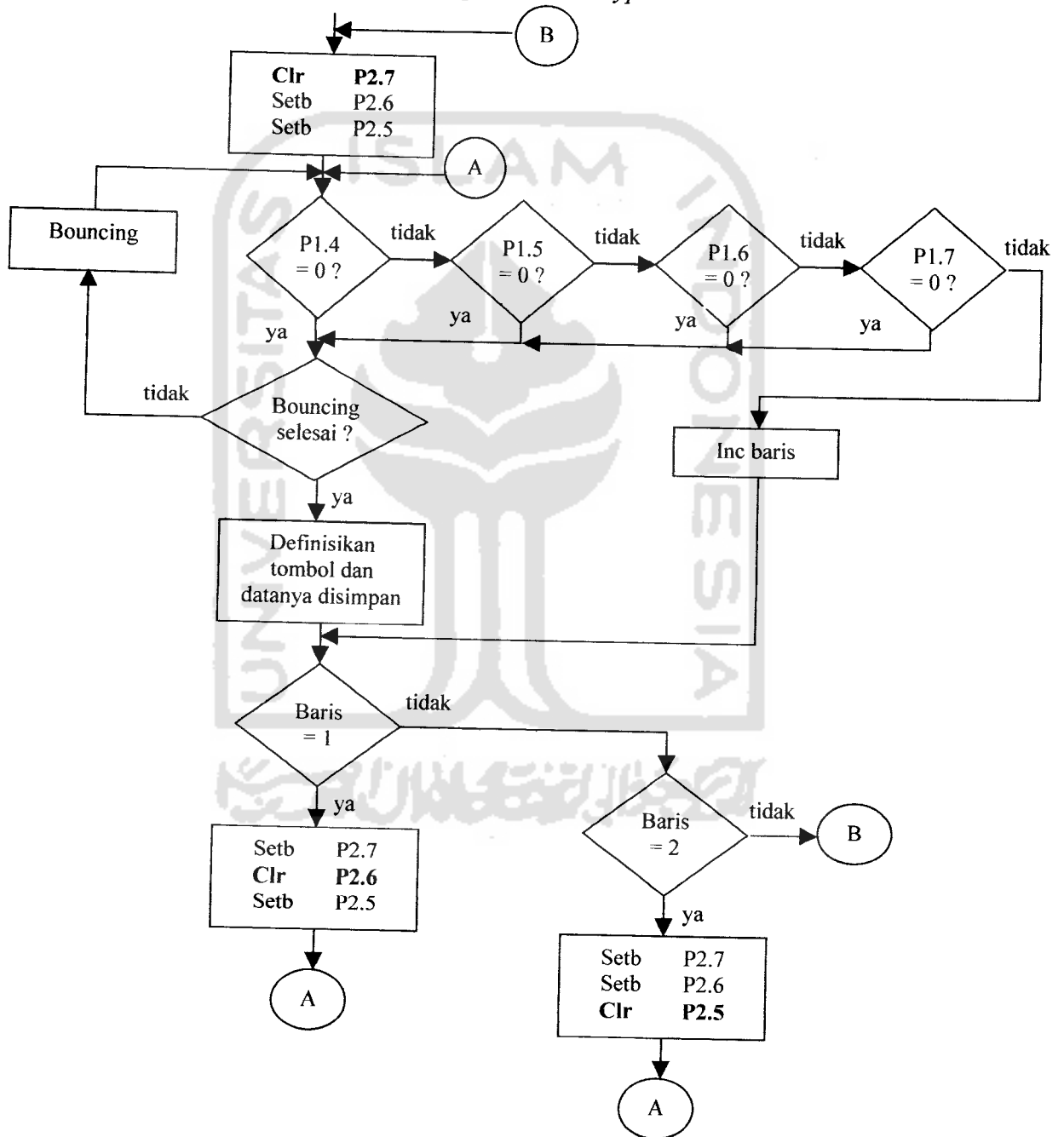
Gambar 3. 12. Diagram alir pembacaan data pada serial EEPROM



Gambar 3. 13. Diagram alir penulisan data pada serial EEPROM

3.2.4.9. Program Pembacaan keypad

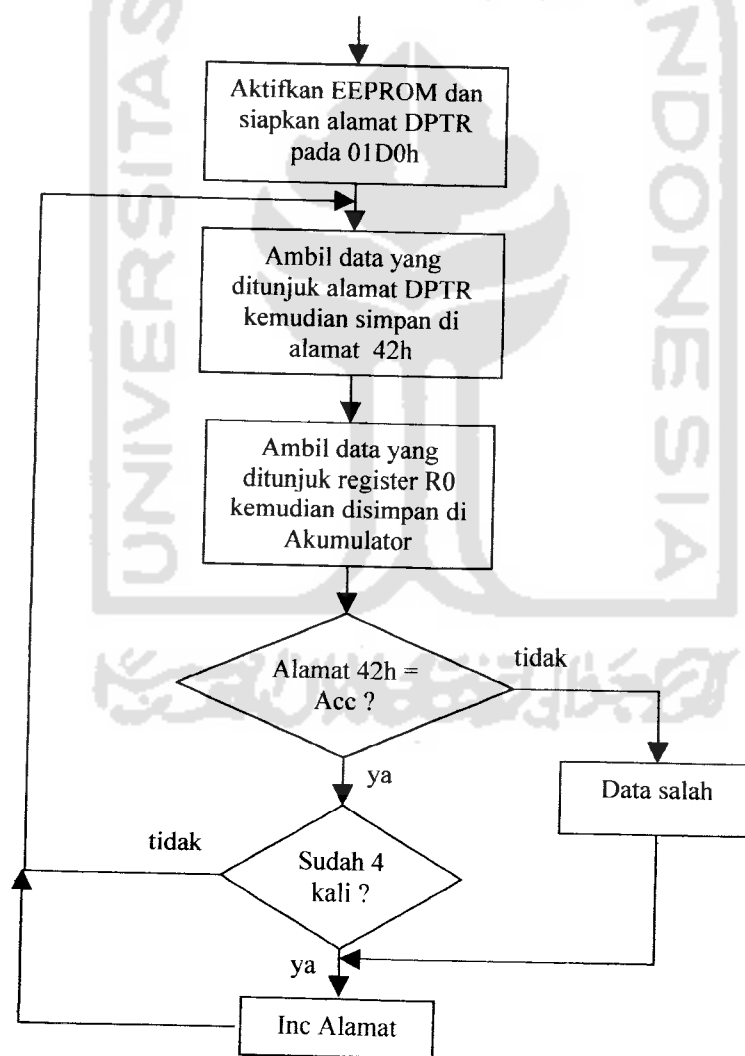
Pembacaan keypad dilakukan dengan sistim *scanning pad* yang mana P_{2.5} – P_{2.7} sebagai keluaran mikrokontroler dan P_{1.4} – P_{1.7} sebagai masukan mikrokontroler, berikut diagram alir pembacaan keypad :



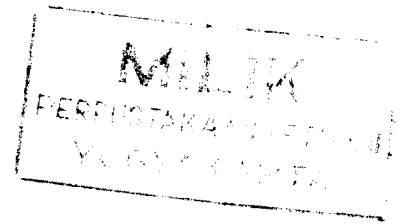
Gambar 3. 14. Diagram alir pembacaan keypad

3.2.4.10. Program Pengecekan nomor PIN

Kode PIN yang digunakan untuk mengakses alat ini terdiri dari 4 digit angka yang telah disimpan dalam internal EEPROM sehingga data PIN akan selalu ada biarpun mikrokontroler tidak diberi sumber tegangan. Proses pengecekannya dilakukan dengan membandingkan data pin yang dimasukkan dari *keypad* yang disimpan pada alamat yang ditunjuk register R0 dengan data pin yang tersimpan di EEPROM dari alamat 0000h seperti diagram alir berikut ini :

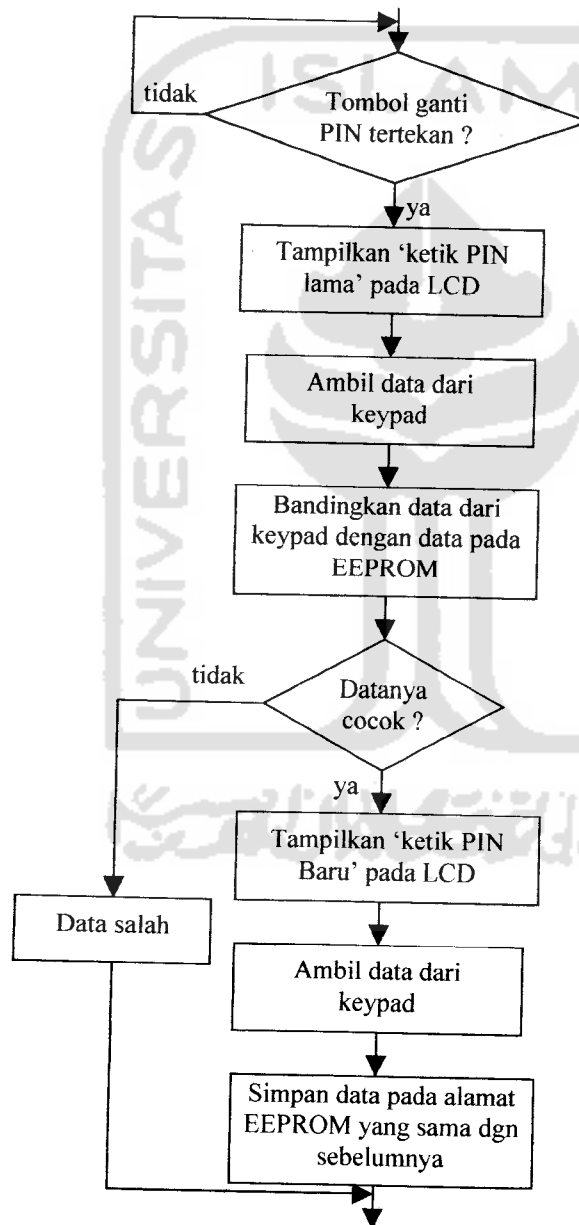


Gambar 3. 15. Diagram alir pengecekan nomor PIN



3.2.4.11. Program Pergantian nomor PIN

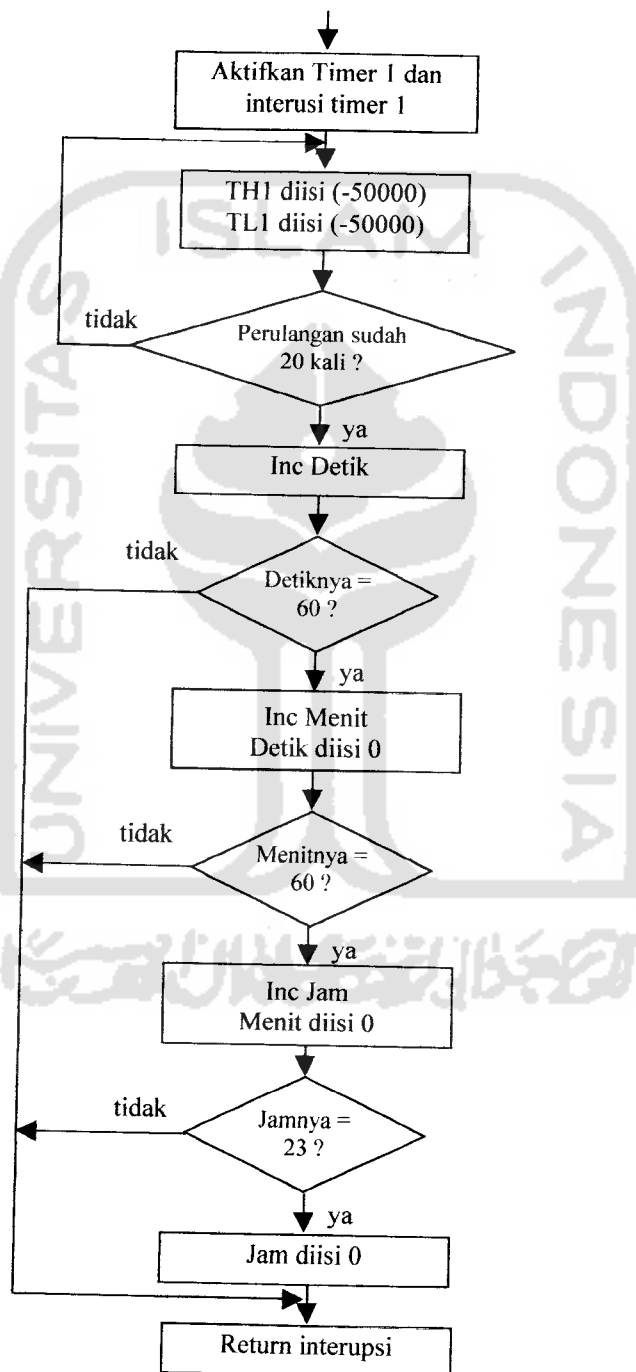
Program pergantian nomor PIN akan dikerjakan jika terjadi penekanan tombol ganti PIN. Proses pergantian PIN pertama kali meminta pemasukan nomor PIN lama kemudian pemasukan nomor PIN yang baru dan terakhir ulangi pemasukan PIN baru seperti diagram alir berikut :



Gambar 3. 16. Diagram alir pergantian nomor PIN

3.2.4.12. Program Jam digital

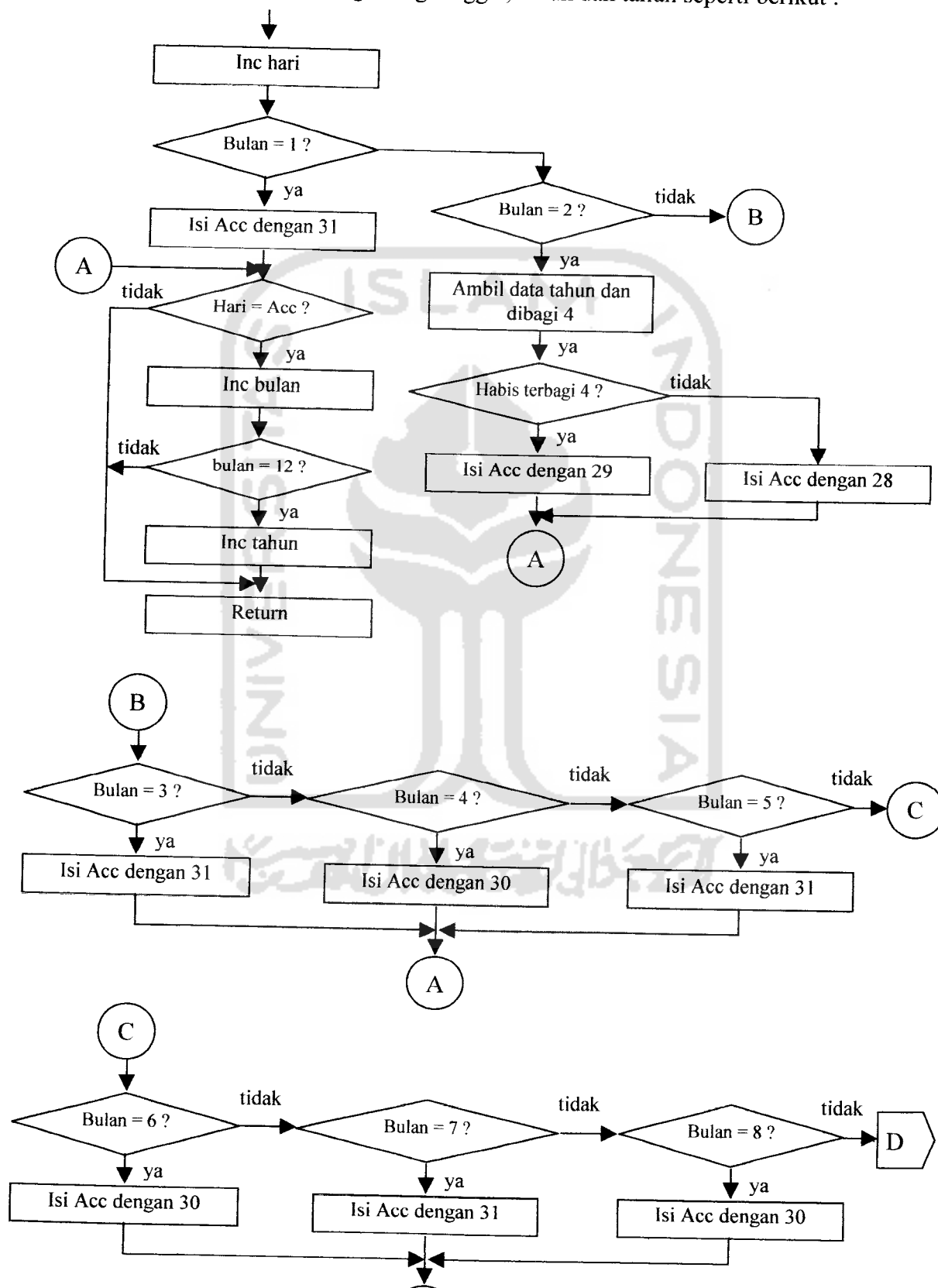
Program ini menggunakan fasilitas timer 1 untuk menghitung jam digital, program ini disimpan dalam *subroutine* yang diakses setiap 1 detik, seperti program berikut :

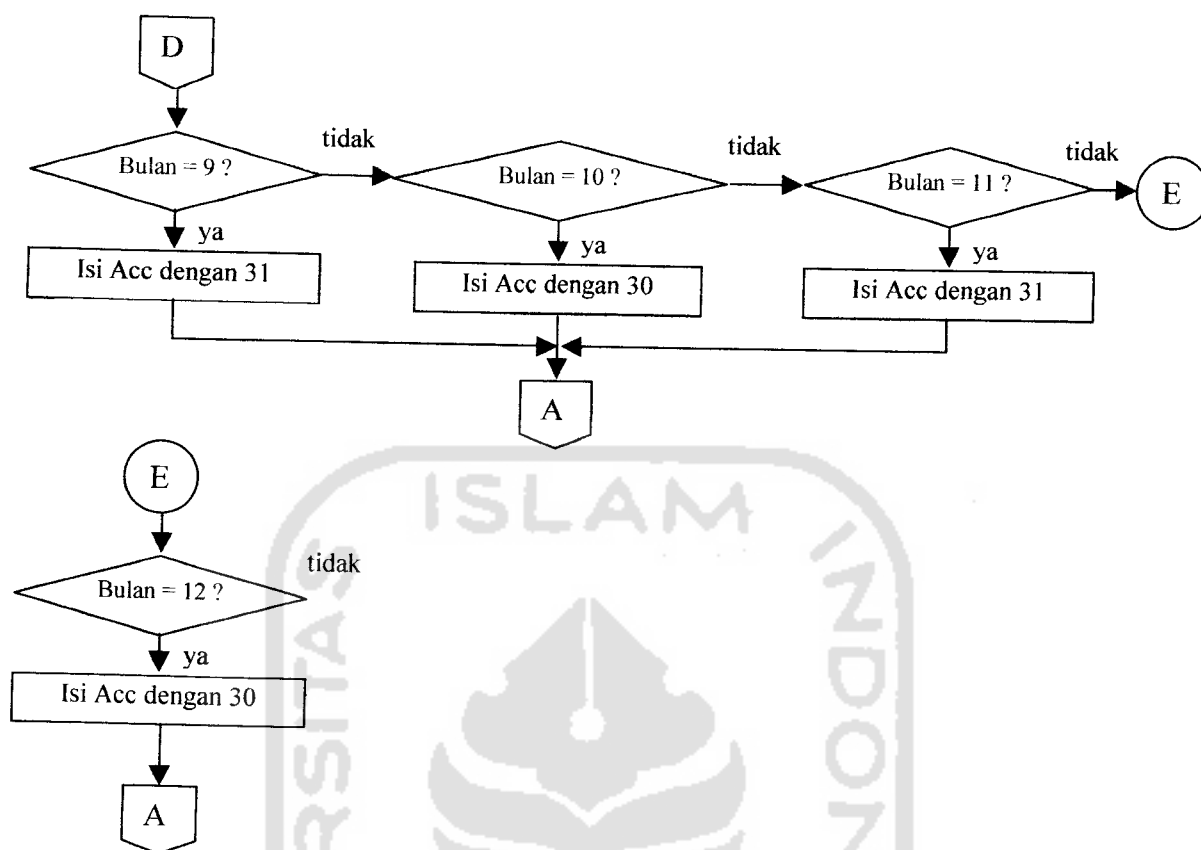


Gambar 3. 17. Diagram alir jam digital

3.2.4.13. Program Tanggal

Program tanggal menghitung tanggal, bulan dan tahun seperti berikut :





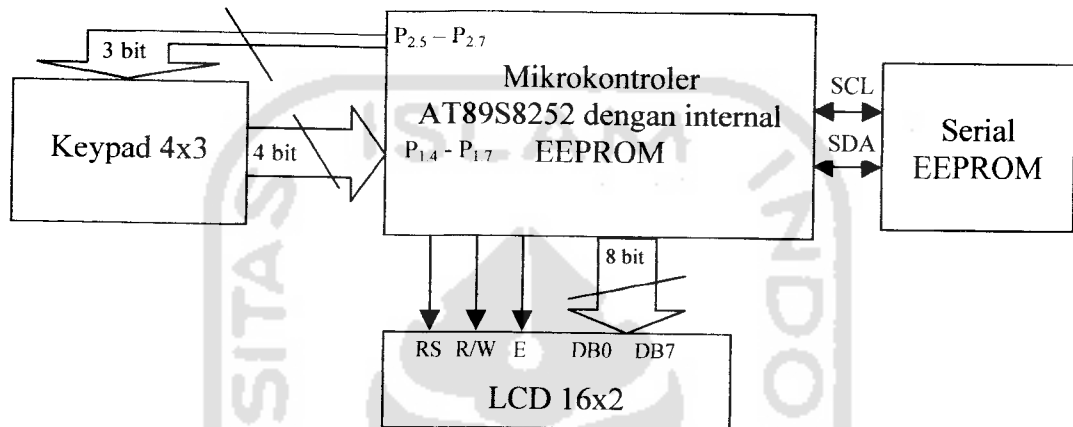
Gambar 3. 18. Diagram alir penghitung tanggal

BAB IV

ANALISIS DAN PEMBAHASAN

4.1. Cara Penggunaan Alat

Diagram blok mesin absensi digital seperti Gambar 4.1. berikut ini:



Gambar 4. 1. Diagram Blok Sistem Mesin Absensi Digital

Cara mengoperasikan alat ini adalah, pertama kali setelah program dijalankan program meminta pengesetan jam dan menit saat itu dan kemudian program meminta memasukkan nomor *user* (PIN) karyawan Untuk melihat data yang tersimpan dalam EEPROM digunakan tombol data, penekanan pertama untuk melihat data masuk dan keluar *user* pertama dan seterusnya, sedangkan tombol ganti PIN digunakan untuk mengganti PIN bila diinginkan. Urutan pergantian PIN adalah sebagai berikut :

1. Pertama masukkan nomor PIN lama, bila nomor PINnya benar maka,
2. Masukkan nomor PIN baru
3. Ulangi lagi pemasukan nomor baru.

Bila nomor yang dimasukkan benar program melakukan penyimpanan data, dan jika salah program menampilkan informasi “PIN anda salah”.

4.2. Pembahasan Memori program, internal EEPROM dan serial EEPROM

AT89S8252 mempunyai 8 kbyte memori program dengan teknologi *flash* EPROM (*Erasable Programmable Read Only Memory*) yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah perangkat pemrograman. *Flash* EPROM dalam AT89S8252 menggunakan *Atmel's High-Density Non Volatile Technology* yang mempunyai kemampuan untuk ditulis ulang hingga 1000 kali dan berisikan perintah standart MCS-51.

Memori program digunakan untuk menyimpan program, program yang ada pada *flash* EPROM akan dijalankan pada saat sistem di-*reset*, pin EA/VP berlogika ‘1’ sehingga mikrokontroler aktif berdasarkan program yang ada pada *flash* EPROM. AT89S8252 selain memiliki memori *flash* EPROM juga memiliki internal EEPROM (*Electrical Erasable Programmable Read Only Memory*) dengan kapasitas 2 kbyte. Memori ini digunakan untuk menyimpan data-data tampilan pada LCD seperti berikut : 'Set Jam dan Tgl ', 'Ketik PIN : ', 'PIN Anda salah ', 'PIN Lama : ', 'PIN Baru : ', ' Lihat data ', 'VERI MASUK', 'KIA MASUK', 'RINI MASUK', 'DB YENI MASUK', 'ROMI MASUK', 'DB ADI MASUK', 'LIA MASUK', 'NIA MASUK', 'NANA MASUK', 'TIA MASUK', 'VERI KELUAR', 'KIA KELUAR', 'RINI KELUAR', 'YENI KELUAR', 'ROMI KELUAR', 'ADI KELUAR', 'LIA

KELUAR', 'NIA KELUAR', 'NANA KELUAR', 'TIA KELUAR',
 'Tanggal:', 'Veri', 'Kia ', 'Rini', 'Yeni', 'Romi', 'Adi ', 'Lia ', 'Nia ', 'Nana', 'Tia ',

dan juga data nomor PIN masing-masing karyawan seperti pada Tabel 4.1 berikut

:Tabel 4. 1. Nomor PIN masing-masing karyawan

No	Nama User	Nomor PIN
1.	Veri	0123
2.	Kia	6789
3.	Rini	1346
4.	Yeni	1798
5.	Romi	8912
6.	Adi	3610
7.	Lia	2468
8.	Nia	7981
9.	Nana	3456
10.	Tia	6751

Sedangkan serial EEPROM digunakan untuk menyimpan data-data jam masuk dan keluar 10 karyawan selama 1 tahun dengan pembagian alamat memori sebagai berikut. Untuk masing-masing karyawan dibutuhkan memori penyimpanan sebesar 32 bit setiap harinya yang digunakan untuk menyimpan data jam, menit masuk dan data jam, menit keluar, misalnya Veri sebagai karyawan pertama untuk tanggal 1, data jam masuk disimpan pada alamat 0000h, data menit masuk disimpan pada alamat 0001h, data jam keluar disimpan pada alamat 0002h dan data menit keluar disimpan pada alamat 0003h. setiap alamat memiliki kapasitas 8 bit sehingga untuk setiap karyawan membutuhkan $4 \times 8 \text{ bit} = 32 \text{ bit}$ dalam 1 hari, sehingga untuk 10 karyawan sehari-hari membutuhkan memori $10 \times$

32 bit = 320 bit. Untuk menyimpan data 10 karyawan dalam 1 tahun dibutuhkan memori sebesar $(320 \text{ bit} \times 31 \text{ hari}) \times 12 \text{ bulan} = 119.040 \text{ bit}$, bila $1 \text{ byte} = 8 \text{ bit}$ maka besarnya memori yang dibutuhkan adalah : $119.040 \text{ bit} / 8 = 14,880 \text{ Kbyte}$. Penyimpanan data setelah 1 tahun dimulai dari alamat 0000h.

4.3. Pembahasan Program *delay*

Program *delay* seperti yang dijelaskan pada bab III berfungsi untuk membuat pulsa pada pin E. Untuk mengetahui periode dari pulsa yang dihasilkan program *delay* dapat dihitung sebagai berikut : 1 siklus mesin program (1 *cycle*) membutuhkan 12 periode osilasi dan karena menggunakan osilator kristal 12 Mhz maka besarnya periode 1 *cycle* adalah :

$$F_{\text{Cycle}} = \frac{12\text{Mhz}}{12} = 1\text{Mhz}, \text{ sehingga}$$

$$T_{\text{Cycle}} = \frac{1}{F_{\text{Cycle}}} = \frac{1}{1\text{Mhz}} = 1 \mu\text{s}$$

jadi untuk 1 mesin *cycle* periodenya adalah $1 \mu\text{s}$

1.	delay:	mov	R6,#30h	1 cycle
2.	delay1:	mov	R5,#10h	1 cycle
3.		djnz	R5,\$	2 cycle
4.		djnz	R6,delay1	2 cycle
5.		ret		2 cycle

sehingga periode program *delay* diatas dapat dihitung sebagai berikut. Program pada baris pertama memiliki 12 periode osilasi sehingga siklus mesinnya 1 *cycle*. Program baris ke-3 memiliki 2 *cycle* dan program ini berulang sebanyak isi dari R5 yaitu 10 h sehingga siklus mesinnya adalah $(2 \times 10h) = 32 \text{ cycle}$. Program

baris 2 sampai baris 4 diulang sebanyak isi R6 yaitu 30 h sehingga siklus mesinnya adalah $(1 + 32 + 2) \times 30h = 1680 \text{ cycle}$ dan total siklus mesin untuk program *delay* di atas adalah $1 + 1680 + 2 = 1683 \text{ cycle}$ dan periodenya sebesar : $1683 \times 1 \mu s = 1,683 \text{ ms}$.

4.4. Rutin Program Serial EEPROM

Berikut ini rutin-rutin program pembacaan dan penulisan data 16 bit serial EEPROM :

4.4.1. Sub-rutin Tulis2_SEE16

Sub-rutin ini terletak pada alamat 06C5H yang berfungsi untuk menulis data ke Serial EEPROM dengan menggunakan DPTR sebagai pointer. Urutan programnya adalah:

1. R7 diisi Device Address
2. Register B diisi Word Address 1
3. R6 diisi Word Address 2
4. Variabel DataSEE diisi data yang akan ditulis

4.4.2. Sub-rutin Baca2_SEE16b

Sub-rutin ini terletak pada alamat 0299H yang berfungsi untuk membaca data ke Serial EEPROM 16 bit dengan urutan program seperti berikut :

1. R7 diisi Device Address
2. Register B diisi Word Address 1
3. Akumulator berisi data yang telah dibaca

4.5. Pembahasan keypad

Tabel 4.2 berikut menunjukkan hubungan antara data tombol dengan data baris dan kolom.

Tabel 4. 2. Data tombol keypad

No	Data Tombol	Data Baris (P _{2.7} – P _{2.5})	Data Kolom (P _{1.7} – P _{1.4})
1.	0	011	1110
2.	1		1101
3.	2		1011
4.	3		0111
5.	4	101	1110
6.	5		1101
7.	6		1011
8.	7		0111
9.	8	110	1110
10.	9		1101
11.	Clear		1011
12.	Ok		0111

4.6. Pembahasan program interupsi jam digital

Program ini menggunakan fasilitas timer 1 untuk menghitung jam digital, program ini disimpan dalam *subroutine* yang diakses setiap 1 detik, seperti program berikut :

Timer1inter:

```

Push    PSW
Push    sp
MOV     TH1,#high (-50000)    ; siapkan waktu tunda 50 mikro detik
MOV     TL1,#low (-50000)
DJNZ    3Bh,SelesaiInterupsi  ;Pencacah turun
MOV     3Bh,#10                ;Nilai awalnya 10
DJNZ    3Ah,SelesaiInterupsi

```

```

Mov    3Ah,#2
; Terjadi sekali dalam waktu 1 detik
ACALL  JamDigital          ;Penambahan waktu
DJNZ   3Dh,SelesaiInterupsi ;Pencacah turun
MOV    3Dh,#60             ;Nilai awalnya 60

```

SelesaiInterupsi:

```

Pop     sp
Pop     PSW
RETI

```

Perhitungan dari program di atas agar program ini diaktikan setiap 1 detik adalah sebagai berikut : alamat 3B diisi 10 dan alamat 3A diisi 2 sedangkan TL1 dan TH2 diisi -50000, karena 1 detik = 1.000 milidetik = 1.000.000 mikrodetik sehingga jika digunakan tunda waktu 50.000 mikrodetik dikali (10 x 2) perulangan isi 3B dan 3A sama dengan 1×10^6 mikrodetik.

BAB V

PENUTUP

5.1. Kesimpulan

Dari perancangan dan pembahasan pada bab III dan IV dapat disimpulkan beberapa hal yaitu :

1. Alat absensi digital ini mampu menyimpan data selama 1 tahun karena dilengkapi dengan serial EEPROM yang berkapasitas 32 kbyte
2. Pergantian PIN dimaksudkan untuk memberikan keamanan terhadap kerahasiaan nomor PIN setiap user.

5.2. Saran

Beberapa saran yang dapat diberikan untuk pengembangan dari alat ini adalah :

1. Untuk memperbanyak user/karyawan yang dapat mengakses alat ini diperlukan *flash* memori dan serial EEPROM yang memiliki kapasitas lebih besar .
2. Supaya tampilan dari alat ini lebih komunikatif lagi dapat diganti dengan LCD grafik

DAFTAR PUSTAKA

- [AGF02] Agfianto, P. E. *Belajar Mikrokontroler AT89C51/52/55(teori dan aplikasi)*. Yogyakarta: Gava Media, 2002.
- [MAL03] Malik, Ibnu Moh. *Belajar Mikrokontroler AT89S8252*. Yogyakarta: Gava Media, 2003.
- [NAL03] Nalwan, A. P. *Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C5*. Jakarta: PT Elex Media komputindo, 2003.
- [NAL03] Nalwan, A. P. *Teknik Akses I2C Serial EEPROM oleh Modul DST-51/2 dan Modul SEE-24*. <http://www.delta-electronic.com>, 2003.
- [ATM01] Atmel. *Interfacing AT24CXX Serial EEPROMs*, <http://www.atmel.com>. 2001
- [ATM01] Atmel. *AT89S8252 Architectural Overview*, <http://www.atmel.com>. 2001
- [ATM01] Atmel. *AT89S8252 In-System Programming*, <http://www.atmel.com>. 2001
- [ATM01] Atmel. *MCS-51 Instructions Set*, <http://www.atmel.com>. 2001
- [ATM00] Atmel. *LCD Moduls HD44780*, <http://www.Hitachi.com>. 2000



DOS 7.10 (038-N) MCS-51 MACRO ASSEMBLER, V2.2
OBJECT MODULE PLACED IN BAYUAH.OBJ
ASSEMBLER INVOKED BY: C:\MCS51~1\SIM51E~1\ASM51.EXE BAYUAH.H51

LOC	OBJ	LINE	SOURCE
3CB0		1	aduh Equ -50000 ;variabel timer
0096		2	WMCON DATA 96h ;inisialisasi EEPROM
0008		3	EEMEN Equ 00001000b
0010		4	EEMWE Equ 00010000b
0002		5	WDTRST Equ 00000010b
0004		6	DPS Equ 00000100b
0000		7	ROM EQU 0h
		8	;
0000		9	ORG ROM ;Reset Vector
0000 020024		10	LJMP Start
		11	
0003		12	org 3h ;External Interrupt 0 Vector
0003 32		13	RETI
		14	;
000B		15	org 0Bh ;Timer 0 Interrupt Vector
000B 32		16	reti
		17	;
0013		18	org 13h ;External Interrupt 1 Vector
0013 32		19	RETI
		20	
001B		21	org 1Bh ;Timer 1 Interrupt Vector
001B 020987		22	LJMP Timerlinter
23			
0023		24	org 23h ;Serial Interrupt Vector
0023 32		25	RETI
		26	;
----		27	DSEG
0000		28	Slave_AddrSEE: Ds 1 ;inisialisasi Serial EEPROM
0001		29	DataSEE2: Ds 1
0093		30	SDA Bit P1.3
0092		31	SCL Bit P1.2
		32	;
0030		33	org 30h
----		34	CSEG
0024 753A02		35	Start: Mov 3Ah,#2 ;alamat 3Ah diisi 02
0027 753B0A		36	MOV 3Bh,#10 ;alamat 3Bh diisi 10
002A 753D3C		37	MOV 3Dh,#60 ;alamat 3Dh diisi 60
002D 7C12		38	MOV R4,#18
002F 7858		39	mov R0,#58h ;alamat 58h-6A diisi 00
0031 7600		40	snol: mov @R0,#0
0033 08		41	inc R0
0034 DCFB		42	djnz R4,snol
0036 7901		43	mov R1,#01h ;menghapus tampilan LCD
0038 1208D0		44	lcall write_inst
003B 120071		45	lcall disp ;subroutin inisialisasi LCD
003E 7902		46	mov R1,#00000010b ;mulai dari alamat 0
0040 1208D0		47	lcall write_inst
0043 7906		48	mov R1,#00000110b ;geser cursor kekiri
0045 1208D0		49	lcall write_inst
0048 900000		50	mov DPTR,#0000h ;set amalal DPTR di 0000h

LOC	OBJ	LINE	SOURCE
004B	7B10	51	mov R3,#16
004D	1162	52	acall tulis ;panggil subroutin tulis
004F	79C0	53	lagi: mov R1,#11000000b ;aktifkan line 2 LCD
0051	1208D0	54	lcall write_inst
0054	1209D8	55	lcall isitampilan ;panggil subroutin isitampilan
0057	1209F9	56	lcall isitanggal ;panggil subroutin isitanggal
005A	12076C	57	lcall tampilkan ;panggil subroutin tampilkan
005D	12007C	58	lcall mulai ;panggil subroutin mulai
0060	80ED	59	sjmp lagi ; lompat ke label lagi
		60	;
		61	;
		62	Subroutin Tulis untuk mengambil data dari EEPROM
		63	-----
0062	439608	63	tulis: orl WMCON,#EEMEN ;aktifkan EEPROM
0065	E0	64	movx A,@DPTR ;Acc diisi data dari EEPROM
0066	F9	65	mov R1,A ;pindahkan isi Acc ke Reg 1
0067	A3	66	inc DPTR ;DPTR+1
0068	1208DF	67	lcall write_data ;panggil subroutin write_data
006B	DBF5	68	djnz R3,tulis ;kurangi Reg 3,jika tdk nol,lompat ke tulis
006D	639608	69	xrl WMCON,#EEMEN ;non aktifkan EEPROM
0070	22	70	ret
		71	;
		72	;
		73	Subroutin Disp untuk inisialisasi LCD
		74	-----
0071	7938	74	disp: mov R1,#00111000b ;LCD bekerja pada 2 line
0073	1208D0	75	lcall write_inst
0076	790E	76	mov R1,#00001110b ;geser kursor
0078	1208D0	77	lcall write_inst
007B	22	78	ret
		79	;
007C	7830	80	mulai: mov R0,#30h ;reg 0 diisi data 30h
007E	753C0A	81	mulai0: mov 3Ch,#10 ;alamat 3C diisi data 10
0081	754300	82	mov 43h,#0 ;alamat 43h diisi data 0
0084	C2A5	83	clr P2. ;nolkan Port 2.5
0086	C2A6	84	clr P2.6 ;nolkan Port 2.6
0088	C2A7	85	clr P2.7 ;nolkan Port 2.7
		86	lomlagi:
008A	30940A	87	jnb P1.4,lompat ;lompat jika Port 1.4 nol
008D	309507	88	jnb P1.5,lompat ;lompat jika Port 1.5 nol
0090	309604	89	jnb P1.6,lompat ;lompat jika port 1.6 nol
0093	309701	90	jnb P1.7,lompat ;lompat jika Port 1.7 nol
0096	22	91	ret ;return
		92	;
		93	lompat:
0097	D53CF0	94	djnz 3Ch,lomlagi ;ulangi pembacaan 10 kali
009A	753C0A	95	mov 3Ch,#10
		96	scaning:
009D	754200	97	mov 42h,#0 ;alamat 42h diisi data 0
00A0	753F00	98	mov 3Fh,#0 ;alamat 3Fh diisi data 0
00A3	C2A7	99	clr P2.7 ;nolkan Port 2.7
00A5	D2A6	100	setb P2.6 ;Port 2.6 dibuat 1
00A7	D2A5	101	setb P2.5 ;Port 2.5 dibuat 1
00A9	20941A	102	jb P1.4,key0 ;lompat jika Port 1.4 satu
00AC	D53CFA	103	djnz 3Ch,balik ;ulangi pembacaan 10 kali
00AF	753C0A	104	ambil: mov 3Ch,#10
00B2	E542	105	mov A,42h ;Acc diisi data dari alamat 42h

LOC	OBJ	LINE	SOURCE
00B4	B40A03	106	cjne A,#0Ah,lanjut ;lompat jika data tidak 0Ah
00B7	020109	107	ljmp clear0 ;lompat label clear0
00BA	B40B03	108	lanjut: cjne A,#0Bh,lanjut1 ;lompat jika data tidak 0Bh
00BD	020144	109	ljmp enter ;jika sama lompat ke enter
		110	lanjut1:
00C0	1201AA	111	lcall tulis_data ;panggil subroutin tulis_data
00C3	02009D	112	ljmp scaning ;lompat ke scaning
00C6	0542	113	key0: inc 42h ;alamat 42h+1
00C8	209505	114	bounc0: jb P1.5,key1 ;
00CB	D53CFA	115	djnz 3Ch,bounc0
00CE	80DF	116	sjmp ambil
00D0	0542	117	key1: inc 42h
00D2	209605	118	bounc1: jb P1.6,key2
00D5	D53CFA	119	djnz 3Ch,bounc1
00D8	80D5	120	sjmp ambil
00DA	0542	121	key2: inc 42h
00DC	209705	122	bounc2: jb P1.7,key3
00DF	D53CFA	123	djnz 3Ch,bounc2
00E2	80CB	124	sjmp ambil
00E4	053F	125	key3: inc 3Fh
00E6	E53F	126	mov A,3Fh
00E8	B4010B	127	cjne A,#1,next
00EB	D2A7	128	Setb P2.7
00ED	C2A6	129	clr P2.6
00EF	D2A5	130	setb P2.5
00F1	0542	131	inc 42h
00F3	0200A9	132	ljmp balik
00F6	E53F	133	next: mov A,3Fh
00F8	B4020B	134	cjne A,#2,next1
00FB	D2A7	135	setb P2.7
00FD	D2A6	136	setb P2.6
00FF	C2A5	137	clr P2.5
0101	0542	138	inc 42h
0103	0200A9	139	ljmp balik
0106	02009D	140	next1: ljmp scaning
0109	E568	141	clear0: mov A,68H
010B	600B	142	jz hapus1
010D	7B03	143	mov R3,#3
010F	7871	144	mov R0,#71H
0111	7600	145	hapus2: MOV @R0,#0
0113	08	146	INC R0
0114	DBFB	147	DJNZ R3,hapus2
0116	8009	148	sjmp dihapus
0118	7858	149	hapus1: mov R0,#58H
011A	7B06	150	MOV R3,#6 ;Reg 3 diidi data 06
011C	7600	151	hapus3: MOV @R0,#0
011E	08	152	INC R0
011F	DBFB	153	DJNZ R3,hapus3
		154	dihapus:
0121	79C0	155	mov R1,#11000000b
0123	1208D0	156	lcall write_inst
0126	1209D8	157	lcall isitampilan
0129	1209F9	158	lcall isitanggal
012C	120A08	159	lcall isitanggal1
012F	E568	160	mov A,68H

LOC	OBJ	LINE	SOURCE
0131	6009	161	jz halo1
0133	1207B9	162	lcall tertampil
0136	755774	163	mov 57h,#74H
0139	02009D	164	ljmp scaning
013C	12076C	165	halo1: lcall tampilan
013F	7830	166	mov R0,#30H
0141	02009D	167	ljmp scaning
0144	E568	168	enter: mov A,68h
0146	6004	169	jz halo2
0148	7874	170	mov R0,#74h
014A	8002	171	sjmp carilagi
014C	7830	172	halo2: mov R0,#30h
		173	carilagi:
014E	E6	174	mov A,@R0
014F	9430	175	Subb A,#30h
0151	F5F0	176	mov B,A
0153	0543	177	inc 43h
0155	08	178	inc R0
0156	E6	179	mov A,@R0
0157	9430	180	Subb A,#30h
0159	FA	181	mov R2,A
015A	740A	182	mov A,#10
015C	A4	183	mul AB
015D	2A	184	ADD A,R2
015E	FA	185	mov R2,A
015F	08	186	inc R0
0160	E568	187	mov A,68h
0162	6002	188	jz wow
0164	8029	189	sjmp tambahan
0166	E543	190	wow: mov A,43h
0168	B40104	191	cjne A,#1,tambah
016B	8A5D	192	mov 5Dh,R2
016D	80DF	193	sjmp carilagi
016F	B40204	194	tambah: cjne A,#2,tambah1
0172	8A5C	195	mov 5Ch,R2
0174	80D8	196	sjmp carilagi
		197	tambah1:
0176	B40304	198	cjne A,#3,tambah2
0179	8A58	199	mov 58h,R2
017B	80D1	200	sjmp carilagi
		201	tambah2:
017D	B40404	202	cjne A,#4,tambah3
0180	8A59	203	mov 59h,R2
0182	80CA	204	sjmp carilagi
		205	tambah3:
0184	B40505	206	cjne A,#5,next2
0187	8A5A	207	mov 5Ah,R2
0189	020310	208	ljmp cek_pin
018C	02009D	209	next2: ljmp scaning
		210	;
		211	tambahan:
018F	E543	212	mov A,43h
0191	B40104	213	cjne A,#1,tambahan0
0194	8A71	214	mov 71h,R2
0196	80B6	215	sjmp carilagi

LOC	OBJ	LINE	SOURCE
		216	tambahan0:
0198	B40204	217	cjne A,#2,tambahan1
019B	8A72	218	mov 72h,R2
019D	80AF	219	sjmp carilagi
		220	tambahan1:
019F	B40305	221	cjne A,#3,tambahan2
01A2	8A73	222	mov 73h,R2
01A4	0201C9	223	ljmp tampil_data
		224	tambahan2:
01A7	02009D	225	ljmp scanning
		226	;
		227	tulis_data:
01AA	79C0	228	mov R1,#11000000b
01AC	1208D0	229	lcall write_inst
01AF	E568	230	mov A,68h
01B1	600D	231	jz halo
01B3	A857	232	mov R0,57h
01B5	E542	233	mov A,42h
01B7	120A1F	234	lcall Ruas1
01BA	8857	235	mov 57h,R0
01BC	1207B9	236	lcall tertampil
01BF	22	237	ret
01C0	E542	238	halo: mov A,42h
01C2	120A1F	239	lcall Ruas1
01C5	12076C	240	lcall tampilkan
01C8	22	241	ret
		242	;
		243	tampil_data:
01C9	0569	244	inc 69h
01CB	900000	245	mov DPTR,#0000h
01CE	7953	246	mov R1,#53h
01D0	E572	247	mov A,72h
01D2	756C01	248	mov 6Ch,#1
		249	pusing1:
01D5	7C9B	250	mov R4,#155
01D7	7D08	251	mov R5,#8
01D9	B56C13	252	cjne A,6Ch,pindah2
01DC	E571	253	mov A,71h
01DE	756A01	254	mov 6Ah,#1
01E1	7C28	255	pusing: mov R4,#40
01E3	B56A02	256	cjne A,6Ah,haril
01E6	801B	257	sjmp datake0
01E8	056A	258	haril: inc 6Ah
		259	caridptr:
01EA	A3	260	inc DPTR
01EB	DCFD	261	djnz R4,caridptr
01ED	80F2	262	sjmp pusing
01EF	056C	263	pindah2: inc 6Ch
01F1	A3	264	cdptr: inc DPTR
01F2	DCFD	265	djnz R4,cdptr
01F4	7C9B	266	mov R4,#155
01F6	DDF9	267	djnz R5,cdptr
01F8	80DB	268	sjmp pusing1
		269	datakel:
01FA	056A	270	inc 6Ah

LOC	OBJ	LINE	SOURCE
01FC	A3	271	reg2: inc DPTR
01FD	DAFD	272	djnz R2,reg2
01FF	7A04	273	mov R2,#4
0201	800A	274	sjmp ayo
		275	datake0:
0203	E569	276	mov A,69h
0205	756A01	277	mov 6Ah,#1
0208	7A04	278	mov R2,#4
020A	756E04	279	mov 6Eh,#4
020D	B56AEA	280	ayo: cjne A,6Ah,datake1
0210	12029C	281	ayol: lcall DPTR2SEE16bit
0213	F7	282	mov @R1,A
0214	09	283	inc R1
0215	A3	284	inc DPTR
0216	D56EF7	285	djnz 6Eh,ayol
0219	120280	286	lcall datasama
021C	E569	287	mov A,69h
021E	B40106	288	cjne A,#1,pil1
0221	1207F0	289	lcall tampilveri
0224	020274	290	ljmp puter
0227	B40206	291	pil1: cjne A,#2,pil2
022A	1207FC	292	lcall tampilkia
022D	020274	293	ljmp puter
0230	B40306	294	pil2: cjne A,#3,pil3
0233	120808	295	lcall tampilrini
0236	020274	296	ljmp puter
0239	B40406	297	pil3: cjne A,#4,pil4
023C	120814	298	lcall tampilveni
023F	020274	299	ljmp puter
0242	B40506	300	pil4: cjne A,#5,pil5
0245	120820	301	lcall tampilromi
0248	020274	302	ljmp puter
024B	B40606	303	pil5: cjne A,#6,pil6
024E	12082C	304	lcall tampiladi
0251	020274	305	ljmp puter
0254	B40706	306	pil6: cjne A,#7,pil7
0257	120838	307	lcall tampilia
025A	020274	308	ljmp puter
025D	B40806	309	pil7: cjne A,#8,pil8
0260	120844	310	lcall tampilnia
0263	020274	311	ljmp puter
0266	B40905	312	pil8: cjne A,#9,pil9
0269	120850	313	lcall tampilnana
026C	8006	314	sjmp puter
026E	B40A09	315	pil9: cjne A,#10,datake10
0271	12085C	316	lcall tampiltia
0274	757004	317	puter: mov 70h,#4
0277	020394	318	ljmp tertekan
		319	datake10:
027A	756900	320	mov 69h,#0
027D	02031F	321	ljmp lagian0
		322	;
		323	datasama:
0280	7901	324	mov R1,#00000001b
0282	1208D0	325	lcall write_inst

LOC	OBJ	LINE	SOURCE
0285	7906	326	mov R1,#00000110b
0287	1208D0	327	lcall write_inst
028A	7902	328	data1: mov R1,#00000010b
028C	1208D0	329	lcall write_inst
028F	1209F9	330	lcall isitanggal
0292	1209D8	331	lcall isitampilan
0295	1209E3	332	lcall isitglmasuk
0298	1209EE	333	lcall isitglkeluar
029B	22	334	ret
		335	DPTR2SEE16bit:
029C	7FA0	336	Mov R7,#0A0H
029E	8583F0	337	Mov B,DPH
02A1	AE82	338	Mov R6,DPL
02A3	51A6	339	Acall Baca2_SEE16b
02A5	22	340	Ret
		341	Baca2_SEE16b:
02A6	120705	342	Lcall Siapkan16bAlamatSEE
02A9	400C	343	Jc Wrong_Read
02AB	120759	344	Lcall Buat_StartBit
02AE	1202BD	345	Lcall ModeBacaSEE ;Kirim Device Address
02B1	4004	346	Jc Wrong_read
02B3	1202CC	347	Lcall BacaDataSEE
02B6	22	348	Ret
		349	Wrong_Read:
02B7	120728	350	Lcall Buat_StopBit
02BA	C3	351	Clr C
02BB	41A6	352	Ajmp Baca2_SEE16b
		353	ModeBacaSEE:
02BD	C0E0	354	Push ACC
02BF	EF	355	Mov A,R7
02C0	D2E0	356	Setb ACC.0
02C2	F500	357	Mov Slave_AddrSEE,A
02C4	D0E0	358	Pop ACC
02C6	E500	359	Mov A,Slave_AddrSEE
02C8	120713	360	Lcall KirimDataSEE
02CB	22	361	Ret
		362	BacaDataSEE:
02CC	C0F0	363	Push B
02CE	75F008	364	Mov B,#08H
02D1	E4	365	Clr A
		366	LoopBacaSEE16b:
02D2	C0F0	367	Push B
02D4	23	368	Rl A
02D5	D293	369	Setb SDA
02D7	D292	370	Setb SCL
02D9	C3	371	Clr C
02DA	A293	372	Mov C,SDA
02DC	92E0	373	Mov ACC.0,C
02DE	C292	374	Clr SCL
02E0	D0F0	375	Pop B
02E2	D5F0ED	376	Djnz B,LoopBacaSEE16b
02E5	120762	377	Lcall Ambil_Ack
02E8	120762	378	Lcall Ambil_Ack
02EB	120728	379	Lcall Buat_StopBit
02EE	D0F0	380	Pop B

LOC	OBJ	LINE	SOURCE
02F0	22	381	Ret
		382	;
		383	ceklagiah:
02F1	7901	384	mov R1,#00000001b
02F3	1208D0	385	lcall write_inst
02F6	7906	386	mov R1,#00000110b
02F8	1208D0	387	lcall write_inst
02FB	22	388	ret
		389	ceklagi:
02FC	7902	390	mov R1,#00000010b
02FE	1208D0	391	lcall write_inst
0301	1209D8	392	lcall isitampilan
0304	1209F9	393	lcall isitanggal
0307	12076C	394	lcall tampilkan
030A	79C0	395	mov R1,#11000000b
030C	1208D0	396	lcall write_inst
030F	22	397	ret
		398	cek_pin:
0310	758910	399	mov TMOD,#00010000b
0313	758D3C	400	MOV TH1,#high aduh
0316	758BB0	401	MOV TL1,#low aduh
0319	D2AB	402	setb ET1
031B	D2AF	403	SETB EA
031D	D28E	404	setb TR1
		405	lagian0:
031F	51F1	406	acall ceklagiah
0321	51FC	407	lagian: acall ceklagi
0323	900010	408	mov DPTR,#0010h
0326	7B0B	409	mov R3,#11
0328	120062	410	lcall tulis
032B	757000	411	mov 70h,#0
032E	020394	412	ljmp tertekan
		413	;
0331	D53C69	414	ganti0: djnz 3Ch,gantilagi
0334	753C0A	415	mov 3Ch,#10
0337	02034A	416	ljmp ganti_pin
033A	D53C60	417	ganti1: djnz 3Ch,gantilagi
033D	753C0A	418	mov 3Ch,#10
0340	E569	419	mov A,69h
0342	6003	420	jz langsung
0344	0201C9	421	ljmp tampil_data
		422	langsung:
0347	02035F	423	ljmp lihat_data
		424	;
		425	ganti_pin:
034A	0546	426	inc 46h
034C	51F1	427	acall ceklagiah
		428	lagian1:
034E	1202FC	429	lcall ceklagi
0351	900030	430	mov DPTR,#0030h
0354	7B0B	431	mov R3,#11
0356	120062	432	lcall tulis
0359	757002	433	mov 70h,#2
035C	0203A3	434	ljmp cek_tombol
		435	;

;bekerja pada mode 1

LOC	OBJ	LINE	SOURCE
		436	lihat_data:
035F 757100		437	mov 71h,#0
0362 757200		438	mov 72h,#0
0365 757300		439	mov 73h,#0
0368 0568		440	inc 68h
036A 7901		441	mov R1,#00000001b
036C 1208D0		442	lcall write_inst
036F 7906		443	mov R1,#00000110b
0371 1208D0		444	lcall write_inst
0374 7902		445	mov R1,#00000010b
0376 1208D0		446	lcall write_inst
0379 900050		447	mov DPTR,#0050h
037C 7B10		448	mov R3,#16
037E 120062		449	lcall tulis
		450	lihatlagi:
0381 79C0		451	mov R1,#11000000b
0383 1208D0		452	lcall write_inst
0386 120A08		453	lcall isitanggal
0389 1207B9		454	lcall tertampil
038C 755774		455	mov 57h,#74h
038F 12007E		456	lcall mulai0
0392 80ED		457	sjmp lihatlagi
		458	;
		459	tertekan:
0394 754A00		460	mov 4AH,#0
0397 754600		461	mov 46H,#0
039A 753C0A		462	mov 3CH,#10
		463	gantilagi:
039D 309091		464	jnb P1.0,ganti0
03A0 309197		465	jnb P1.1,gantil
		466	cek_tombol:
03A3 75487A		467	mov 48h,#7Ah
03A6 754701		468	mov 47h,#1
		469	scan_tombol:
03A9 754200		470	mov 42h,#0
03AC 753F00		471	mov 3Fh,#0
03AF C2A7		472	clr P2.7
03B1 D2A6		473	setb P2.6
03B3 D2A5		474	setb P2.5
		475	balikah:
03B5 20942A		476	jb P1.4,tombol0
03B8 D53CFA		477	djnz 3Ch,balikah
03BB 753C0A		478	mov 3Ch,#10
		479	ambilah:
03BE 757001		480	mov 70h,#1
03C1 A848		481	mov R0,48h
03C3 A642		482	mov @R0,42h
03C5 0548		483	inc 48h
03C7 E542		484	mov A,42h
03C9 B40A03		485	cjne A,#0Ah,trus
03CC 020442		486	ljmp clear
03CF B40B0A		487	trus: cjne A,#0Bh,trus1
03D2 E54A		488	mov A,4Ah
03D4 7003		489	jnz ubahlah
03D6 0204BD		490	ljmp banding_pin

LOC	OBJ	LINE	SOURCE
		491	ubahlah:
03D9 02048B		492	ljmp cari_pin
03DC 12045D		493	trus1: lcall ketik
03DF 0203A9		494	ljmp scan_tombol
		495	tombol0:
03E2 0542		496	inc 42h
		497	bouncing0:
03E4 209505		498	jb P1.5,tombol1
03E7 D53CFA		499	djnz 3Ch,bouncing0
03EA 80D2		500	sjmp ambilah
		501	tombol1:
03EC 0542		502	inc 42h
		503	bouncing1:
03EE 209605		504	jb P1.6,tombol2
03F1 D53CFA		505	djnz 3Ch,bouncing1
03F4 80C8		506	sjmp ambilah
		507	tombol2:
03F6 0542		508	inc 42h
		509	bouncing2:
03F8 209705		510	jb P1.7,tombol3
03FB D53CFA		511	djnz 3Ch,bouncing2
03FE 80BE		512	sjmp ambilah
		513	tombol3:
0400 053F		514	inc 3FH
0402 E53F		515	mov A,3FH
0404 B4010B		516	cjne A,#1,buin
0407 D2A7		517	setb P2.7
0409 C2A6		518	clr P2.6
040B D2A5		519	setb P2.5
040D 0542		520	inc 42H
040F 0203B5		521	ljmp balikah
0412 E53F		522	buin: mov A,3FH
0414 B4020B		523	cjne A,#2,buin1
0417 D2A7		524	setb P2.7
0419 D2A6		525	setb P2.6
041B C2A5		526	clr P2.5
041D 0542		527	inc 42H
041F 0203B5		528	ljmp balikah
0422 E570		529	buin1: mov A,70h
0424 B40103		530	cjne A,#1,gnt
0427 0203A9		531	ljmp scan_tombol
042A B40203		532	gnt: cjne A,#2,lama
042D 02034E		533	ljmp lagian1
0430 B40303		534	lama: cjne A,#3,lht
0433 02047A		535	ljmp lagian2
0436 B40406		536	lht: cjne A,#4,ulangjam
0439 12028A		537	lcall data1
043C 020274		538	ljmp puter
		539	ulangjam:
043F 020321		540	ljmp lagian
		541	;
0442 7904		542	clear: mov R1,#00000100b
0444 1208D0		543	lcall write_inst
0447 7920		544	mov R1,#'
0449 1208DF		545	clear1: lcall write_data

LOC	OBJ	LINE	SOURCE
044C	D547FA	546	djnz 47H,clear1
044F	754701	547	mov 47H,#1
0452	79CB	548	mov R1,#11001011b
0454	1208D0	549	lcall write_inst
0457	75487A	550	mov 48H,#7Ah
045A	0203A9	551	ljmp scan_tombol
		552	;
045D	7906	553	ketik: mov R1,#00000110b
045F	1208D0	554	lcall write_inst
0462	792A	555	mov R1,#''
0464	1208DF	556	lcall write_data
0467	1208FD	557	lcall Ldelay
046A	1208FD	558	lcall Ldelay
046D	0547	559	inc 47H
046F	22	560	ketik1: ret
		561	;
		562	samapin:
0470	E546	563	mov A,46H
0472	7001	564	jnz pinlama
0474	22	565	ret
		566	;
		567	pinlama:
0475	054A	568	inc 4AH
0477	1202F1	569	lcall ceklagiah
		570	lagian2:
047A	1202FC	571	lcall ceklagi
047D	900040	572	mov DPTR,#0040H
0480	7B0B	573	mov R3,#11
0482	120062	574	lcall tulis
0485	757003	575	mov 70h,#3
0488	0203A3	576	ljmp cek_tombol
		577	;
		578	cari_pin:
048B	787A	579	mov R0,#7Ah
048D	754504	580	mov 45H,#4
0490	E544	581	mov A,44H
0492	75F004	582	mov B,#4
0495	A4	583	mul AB
0496	F549	584	mov 49H,A
0498	439608	585	orl WMCON,#EEMEN
049B	439610	586	orl WMCON,#EEMWE
049E	9001D0	587	mov DPTR,#01D0h
04A1	6004	588	jz bedain
04A3	A3	589	beda: inc DPTR
04A4	D549FC	590	djnz 49H,beda
04A7	E6	591	bedain: mov A,@R0
04A8	F0	592	movx @DPTR,A
04A9	E596	593	loop2: mov A,WMCON
04AB	5402	594	anl A,#WDTRST
04AD	60FA	595	jz loop2
04AF	08	596	inc R0
04B0	A3	597	inc DPTR
04B1	D545F3	598	djnz 45H,bedain
04B4	639610	599	xrl WMCON,#EEMWE
04B7	639608	600	xrl WMCON,#EEMEN

LOC	OBJ	LINE	SOURCE
04BA	02031F	601	ljmp lagian0
		602	;
		603	banding_pin:
04BD	7D04	604	MOV R5,#4
04BF	756A01	605	mov 6Ah,#1
04C2	754504	606	mov 45H,#4
04C5	754400	607	mov 44H,#0
04C8	787A	608	mov R0,#7Ah
04CA	7F0A	609	mov R7,#10
04CC	439608	610	orl WMCON,#EEMEN
04CF	9001D0	611	mov DPTR,#01D0h
04D2	E0	612	jump: movx A,@DPTR
04D3	A3	613	inc DPTR
04D4	F542	614	mov 42H,A
04D6	E6	615	mov A,@R0
04D7	B54225	616	cjne A,42H,salah
04DA	08	617	inc R0
04DB	D545F4	618	djnz 45H,jump
04DE	639608	619	xrl WMCON,#EEMEN
04E1	754504	620	mov 45H,#4
04E4	E544	621	mov A,44H
04E6	B4004F	622	cjne A,#0,veri
04E9	120470	623	lcall samapin
04EC	055E	624	inc 5EH
04EE	12066A	625	lcall line2
04F1	E55E	626	mov A,5EH
04F3	B40136	627	cjne A,#1,lomveri
04F6	900060	628	mov DPTR,#0060H
04F9	02051E	629	ljmp nama
		630	;
04FC	A3	631	salah0: inc DPTR
04FD	8002	632	sjmp salah1
		633	;
04FF	0544	634	salah: inc 44h
0501	787A	635	salah1: mov R0,#7Ah
0503	D545F6	636	djnz 45h,salah0
0506	754504	637	mov 45h,#4
0509	787A	638	mov R0,#7Ah
050B	DFC5	639	djnz R7,jump
050D	12066A	640	lcall line2
0510	900020	641	mov DPTR,#0020h
0513	7B10	642	mov R3,#16
0515	120062	643	lcall tulis
0518	1208EE	644	lcall FLdelay
051B	02031F	645	ljmp lagian0
		646	;
051E	7B10	647	nama: mov R3,#16
0520	120062	648	lcall tulis
0523	120678	649	lcall simpan_data
0526	1208EE	650	lcall FLdelay
0529	02031F	651	ljmp lagian0
		652	;
		653	lomveri:
052C	B40206	654	cjne A,#2,lomveri1
052F	900100	655	mov DPTR,#0100H

LOC	OBJ	LINE	SOURCE
0532	02051E	656	ljmp nama
		657	lomveril:
0535	02031F	658	ljmp lagian0
		659	;
0538	B4011F	660	veri: cjne A,#1,kia
053B	120470	661	lcall samapin
053E	055F	662	inc 5FH
0540	12066A	663	lcall line2
0543	E55F	664	mov A,5FH
0545	B40106	665	cjne A,#1,lomkia
0548	900070	666	mov DPTR,#0070H
054B	02051E	667	ljmp nama
054E	B40206	668	lomkia: cjne A,#2,lomkia1
0551	900110	669	mov DPTR,#0110H
0554	02051E	670	ljmp nama
		671	lomkia1:
0557	02031F	672	ljmp lagian0
		673	;
055A	B4021F	674	kia: cjne A,#2,rini
055D	120470	675	lcall samapin
0560	0560	676	inc 60H
0562	12066A	677	lcall line2
0565	E560	678	mov A,60H
0567	B40106	679	cjne A,#1,lomrini
056A	900080	680	mov DPTR,#0080H
056D	02051E	681	ljmp nama
		682	lomrini:
0570	B40206	683	cjne A,#2,lomrini1
0573	900120	684	mov DPTR,#0120h
0576	02051E	685	ljmp nama
		686	lomrini1:
0579	02031F	687	ljmp lagian0
		688	;
057C	B4031F	689	rini: cjne A,#3,yeni
057F	120470	690	lcall samapin
0582	0561	691	inc 61H
0584	12066A	692	lcall line2
0587	E561	693	mov A,61H
0589	B40106	694	cjne A,#1,lomyeni
058C	900090	695	mov DPTR,#0090H
058F	02051E	696	ljmp nama
		697	lomyeni:
0592	B40206	698	cjne A,#2,lomyeni1
0595	900130	699	mov DPTR,#0130H
0598	02051E	700	ljmp nama
		701	lomyeni1:
059B	02031F	702	ljmp lagian0
		703	;
059E	B4041F	704	yeni: cjne A,#4,romi
05A1	120470	705	lcall samapin
05A4	0562	706	inc 62H
05A6	12066A	707	lcall line2
05A9	E562	708	mov A,62H
05AB	B40106	709	cjne A,#1,lomromi
05AE	9000A0	710	mov DPTR,#00A0H

LOC	OBJ	LINE	SOURCE
05B1	02051E	711	ljmp nama
		712	lomromi:
05B4	B40206	713	cjne A,#2,lomromil
05B7	900140	714	mov DPTR,#0140H
05BA	02051E	715	ljmp nama
		716	lomromil:
05BD	02031F	717	ljmp lagian0
		718	;
05C0	B4051F	719	romi: cjne A,#5,adi
05C3	120470	720	lcall samapin
05C6	0563	721	inc 63H
05C8	12066A	722	lcall line2
05CB	E563	723	mov A,63H
05CD	B40106	724	cjne A,#1,lomadi
05D0	9000B0	725	mov DPTR,#00B0H
05D3	02051E	726	ljmp nama
05D6	B40206	727	lomadi: cjne A,#2,lomadi1
05D9	900150	728	mov DPTR,#0150H
05DC	02051E	729	ljmp nama
		730	lomadi1:
05DF	02031F	731	ljmp lagian0
		732	;
05E2	B4061F	733	adi: cjne A,#6,lia
05E5	120470	734	lcall samapin
05E8	0564	735	inc 64H
05EA	12066A	736	lcall line2
05ED	E564	737	mov A,64H
05EF	B40106	738	cjne A,#1,lomlia
05F2	9000C0	739	mov DPTR,#00C0H
05F5	02051E	740	ljmp nama
05F8	B40206	741	lomlia: cjne A,#2,lomlia1
05FB	900160	742	mov DPTR,#0160H
05FE	02051E	743	ljmp nama
		744	lomlia1:
0601	02031F	745	ljmp lagian0
		746	;
0604	B4071F	747	lia: cjne A,#7,nia
0607	120470	748	lcall samapin
060A	0565	749	inc 65H
060C	12066A	750	lcall line2
060F	E565	751	mov A,65H
0611	B40106	752	cjne A,#1,lomnia
0614	9000D0	753	mov DPTR,#00D0H
0617	02051E	754	ljmp nama
061A	B40206	755	lomnia: cjne A,#2,lomnia1
061D	900170	756	mov DPTR,#0170H
0620	02051E	757	ljmp nama
		758	lomnia1:
0623	02031F	759	ljmp lagian0
0626	B4081F	760	nia: cjne A,#8,nana
0629	120470	761	lcall samapin
062C	0566	762	inc 66H
062E	12066A	763	lcall line2
0631	E566	764	mov A,66H
0633	B40106	765	cjne A,#1,lomnana

LOC	OBJ	LINE	SOURCE
0636	9000E0	766	mov DPTR,#00E0H
0639	02051E	767	ljmp nama
		768	lomnana:
063C	B40206	769	cjne A,#2,lomnana1
063F	900180	770	mov DPTR,#0180H
0642	02051E	771	ljmp nama
		772	lomnana1:
0645	02031F	773	ljmp lagian0
0648	B4091C	774	nana: cjne A,#9,tia
064B	120470	775	lcall samapin
064E	0567	776	inc 67H
0650	12066A	777	lcall line2
0653	E567	778	mov A,67H
0655	B40106	779	cjne A,#1,lomtia
0658	9000F0	780	mov DPTR,#00F0h
065B	02051E	781	ljmp nama
065E	B40206	782	lomtia: cjne A,#2,tia
0661	900190	783	mov DPTR,#0190H
0664	02051E	784	ljmp nama
0667	02031F	785	tia: ljmp lagian0
		786	;
066A	79C0	787	line2: mov R1,#11000000b
066C	1208D0	788	lcall write_inst
066F	22	789	ret
		790	;
0670	056A	791	dapat: inc 6Ah
0672	A3	792	hah: inc DPTR
0673	DDFD	793	djnz R5,hah
0675	7D04	794	mov R5,#4
0677	22	795	ret
		796	;
		797	simpan_data:
0678	900000	798	mov DPTR,#0000h
067B	E558	799	mov A,58h
067D	756A01	800	mov 6Ah,#1
0680	756B00	801	mov 6Bh,#0
0683	785E	802	mov R0,#5Eh
0685	756C01	803	mov 6Ch,#1
0688	756D04	804	mov 6Dh,#04
		805	plushari:
068B	B56A2C	806	cjne A,6Ah,pertama
068E	E559	807	mov A,59h
0690	7E9B	808	plus1: mov R6,#155
0692	7F08	809	mov R7,#8
0694	B56C2D	810	cjne A,6Ch,bin1
0697	E573	811	mov A,73h
0699	B56D34	812	cjne A,6Dh,pindah
069C	E544	813	mov A,44h
069E	FE	814	mov R6,A
069F	600B	815	jz simpan
06A1	08	816	plusR0: inc R0
06A2	DEFD	817	djnz R6,plusR0
06A4	75F004	818	mov B,#4
06A7	A4	819	MUL AB
06A8	FE	820	mov R6,A

LOC	OBJ	LINE	SOURCE
06A9	A3	821	plus: inc DPTR
06AA	DEFD	822	djnz R6,plus
06AC	E6	823	simpan: mov A,@R0
06AD	B40103	824	cjne A,#1,in
06B0	D1D8	825	atas: acall menyimpan
06B2	22	826	out: ret
		827	;
06B3	B402FC	828	in: cjne A,#2,out
06B6	A3	829	inc DPTR
06B7	A3	830	inc DPTR
06B8	80F6	831	sjmp atas
		832	pertama:
06BA	056A	833	inc 6Ah
06BC	7E28	834	mov R6,#40
06BE	A3	835	sdnol: inc DPTR
06BF	DEFD	836	djnz R6,sdnol
06C1	02068B	837	ljmp plushari
06C4	056C	838	bln1: inc 6Ch
06C6	A3	839	sdnol1: inc DPTR
06C7	DEFD	840	djnz R6,sdnol1
06C9	7E9B	841	mov R6,#155
06CB	DFF9	842	djnz R7,sdnol1
06CD	020690	843	ljmp plus1
06D0	056D	844	pindah: inc 6Dh
06D2	900000	845	mov DPTR,#0000h
06D5	02068B	846	ljmp plushari
		847	menyimpan:
06D8	855D01	848	mov DataSEE2,5Dh
06DB	D1E4	849	acall Tulis2DPTRSEE16b
06DD	A3	850	inc DPTR
06DE	855C01	851	mov DataSEE2,5Ch
06E1	D1E4	852	acall Tulis2DPTRSEE16b
06E3	22	853	ret
		854	;
		855	Tulis2DPTRSEE16b:
06E4	7FA0	856	Mov R7,#0A0H
06E6	8583F0	857	Mov B,DPH
06E9	AE82	858	Mov R6,DPL
06EB	D1EE	859	acall Tulis2_SEE16b
06ED	22	860	Ret
		861	;
		862	Tulis2_SEE16b:
06EE	120705	863	Lcall Siapkan16bAlamatSEE
06F1	400B	864	Jc Wrong_Write
06F3	E501	865	Mov A,DataSEE2
06F5	120713	866	Lcall KirimDataSEE
06F8	4004	867	Jc Wrong_Write
06FA	120728	868	Lcall Buat_StopBit
06FD	22	869	Ret
		870	;
		871	Wrong_Write:
06FE	120728	872	Lcall Buat_StopBit
0701	C3	873	Clr C
0702	0206EE	874	Ljmp Tulis2_SEE16b
		875	;

LOC	OBJ	LINE	SOURCE
		876	Siapkan16bAlamatSEE:
0705	120731	877	Lcall KirimDeviceAddress
0708	4008	878	Jc SalahTulisAlamat
070A	12073D	879	Lcall Kirim1WordAddress
070D	4003	880	Jc SalahTulisAlamat
070F	120747	881	Lcall Kirim2WordAddress
		882	SalahTulisAlamat:
0712	22	883	Ret
		884	KirimDataSEE:
0713	C0F0	885	Push B
0715	75F008	886	Mov B,#8
		887	Send8_bitloop:
0718	33	888	Rlc A
0719	9293	889	Mov SDA,C
071B	120750	890	Lcall Pulse_SEE
071E	D5F0F7	891	Djnz B,Send8_bitloop
0721	D0F0	892	Pop B
0723	C3	893	Clr C
0724	120762	894	Lcall Ambil_Ack
0727	22	895	Ret
		896	Buat_StopBit:
0728	C293	897	Clr SDA
072A	D292	898	Setb SCL
072C	D293	899	Setb SDA
072E	C292	900	Clr SCL
0730	22	901	Ret
		902	KirimDeviceAddress:
0731	120759	903	Lcall Buat_StartBit
0734	C0E0	904	Push ACC
0736	EF	905	Mov A,R7 ;Device Address
0737	120713	906	Lcall KirimDataSEE ;
073A	D0E0	907	Pop ACC ;
073C	22	908	Ret ;
		909	
		910	Kirim1WordAddress:
073D	C0E0	911	Push ACC ;
073F	E5F0	912	Mov A,B ;First Word Address
0741	120713	913	Lcall KirimDataSEE ;
0744	D0E0	914	Pop ACC ;
0746	22	915	Ret ;
		916	
		917	Kirim2WordAddress:
0747	C0E0	918	Push ACC
0749	EE	919	Mov A,R6 ;Second Address
074A	120713	920	Lcall KirimDataSEE ;
074D	D0E0	921	Pop ACC
074F	22	922	Ret
		923	Pulse_SEE:
0750	C0F0	924	Push b
0752	D292	925	Setb SCL
0754	C292	926	Clr SCL
0756	D0F0	927	Pop B
0758	22	928	Ret
		929	Buat_StartBit:
0759	D293	930	Setb SDA

LOC	OBJ	LINE	SOURCE
075B	D292	931	Setb SCL
075D	C293	932	Clr SDA
075F	C292	933	Clr SCL
0761	22	934	Ret
		935	Ambil_Ack:
0762	C3	936	Clr C
0763	D293	937	Setb SDA
0765	D292	938	Setb SCL
0767	A293	939	Mov C,SDA
0769	C292	940	Clr SCL
076B	22	941	Ret
		942	;
		943	tampilkan:
076C	A930	944	mov R1,30h
076E	1208DF	945	lcall write_data
0771	A931	946	mov R1,31h
0773	1208DF	947	lcall write_data
0776	793A	948	mov R1,'#':
0778	1208DF	949	lcall write_data
077B	A932	950	mov R1,32h
077D	1208DF	951	lcall write_data
0780	A933	952	mov R1,33h
0782	1208DF	953	lcall write_data
0785	792F	954	mov R1,'#7'
0787	1208DF	955	lcall write_data
078A	A934	956	mov R1,34h
078C	1208DF	957	lcall write_data
078F	A935	958	mov R1,35h
0791	1208DF	959	lcall Write_data
0794	792D	960	mov R1,'#-'
0796	1208DF	961	lcall Write_data
0799	A936	962	mov R1,36h
079B	1208DF	963	lcall Write_data
079E	A937	964	mov R1,37h
07A0	1208DF	965	lcall Write_data
07A3	792D	966	mov R1,'#-'
07A5	1208DF	967	lcall Write_data
07A8	A938	968	mov R1,38h
07AA	1208DF	969	lcall Write_data
07AD	A939	970	mov R1,39h
07AF	1208DF	971	lcall Write_data
07B2	1208FD	972	lcall Ldelay
07B5	1208FD	973	lcall Ldelay
07B8	22	974	ret
		975	tertampil:
07B9	9001A0	976	mov DPTR,#01A0h
07BC	7B08	977	mov R3,#8
07BE	120062	978	lcall tulis
		979	tampilhari:
07C1	A974	980	mov R1,74h
07C3	1208DF	981	lcall write_data
07C6	A975	982	mov R1,75h
07C8	1208DF	983	lcall Write_data
07CB	792D	984	mov R1,'#-'
07CD	1208DF	985	lcall Write_data

LOC	OBJ	LINE	SOURCE
07D0	A976	986	mov R1,76h
07D2	1208DF	987	lcall Write_data
07D5	A977	988	mov R1,77h
07D7	1208DF	989	lcall Write_data
07DA	792D	990	mov R1,#'-'
07DC	1208DF	991	lcall Write_data
07DF	A978	992	mov R1,78h
07E1	1208DF	993	lcall Write_data
07E4	A979	994	mov R1,79h
07E6	1208DF	995	lcall Write_data
07E9	1208FD	996	lcall Ldelay
07EC	1208FD	997	lcall Ldelay
07EF	22	998	ret
		999	tampilveri:
07F0	9001A8	1000	mov DPTR,#01A8h
07F3	7B04	1001	mov R3,#4
07F5	120062	1002	lcall tulis
07F8	120868	1003	lcall tampilkandata
07FB	22	1004	ret
		1005	tampilkia:
07FC	9001AC	1006	mov DPTR,#01ACh
07FF	7B04	1007	mov R3,#4
0801	120062	1008	lcall tulis
0804	120868	1009	lcall tampilkandata
0807	22	1010	ret
		1011	tampilrini:
0808	9001B0	1012	mov DPTR,#01B0h
080B	7B04	1013	mov R3,#4
080D	120062	1014	lcall tulis
0810	120868	1015	lcall tampilkandata
0813	22	1016	ret
		1017	tampilyeni:
0814	9001B4	1018	mov DPTR,#01B4h
0817	7B04	1019	mov R3,#4
0819	120062	1020	lcall tulis
081C	120868	1021	lcall tampilkandata
081F	22	1022	ret
		1023	tampilromi:
0820	9001B8	1024	mov DPTR,#01B8h
0823	7B04	1025	mov R3,#4
0825	120062	1026	lcall tulis
0828	120868	1027	lcall tampilkandata
082B	22	1028	ret
		1029	tampiladi:
082C	9001BC	1030	mov DPTR,#01BCh
082F	7B04	1031	mov R3,#4
0831	120062	1032	lcall tulis
0834	120868	1033	lcall tampilkandata
0837	22	1034	ret
		1035	tampillicia:
0838	9001C0	1036	mov DPTR,#01C0h
083B	7B04	1037	mov R3,#4
083D	120062	1038	lcall tulis
0840	120868	1039	lcall tampilkandata
0843	22	1040	ret

LOC	OBJ	LINE	SOURCE
		1041	tampilnia:
0844	9001C4	1042	mov DPTR,#01C4h
0847	7B04	1043	mov R3,#4
0849	120062	1044	lcall tulis
084C	120868	1045	lcall tampilkandata
084F	22	1046	ret
		1047	tampilnana:
0850	9001C8	1048	mov DPTR,#01C8h
0853	7B04	1049	mov R3,#4
0855	120062	1050	lcall tulis
0858	120868	1051	lcall tampilkandata
085B	22	1052	ret
		1053	tampiltia:
085C	9001CC	1054	mov DPTR,#01CCh
085F	7B03	1055	mov R3,#3
0861	120062	1056	lcall tulis
0864	120868	1057	lcall tampilkandata
0867	22	1058	ret
		1059	tampilkandata:
0868	7954	1060	mov R1,#'T'
086A	1208DF	1061	lcall write_data
086D	7967	1062	mov R1,#'g'
086F	1208DF	1063	lcall write_data
0872	796C	1064	mov R1,#'l'
0874	1208DF	1065	lcall write_data
0877	793A	1066	mov R1,#':'
0879	1208DF	1067	lcall write_data
087C	1207C1	1068	lcall tampilhari
087F	79C0	1069	mov R1,#11000000b
0881	1208D0	1070	lcall write_inst
0884	794D	1071	mov R1,#'M'
0886	1208DF	1072	lcall write_data
0889	793D	1073	mov R1,#'='
088B	1208DF	1074	lcall write_data
088E	A94F	1075	mov R1,4Fh
0890	1208DF	1076	lcall write_data
0893	A950	1077	mov R1,50h
0895	1208DF	1078	lcall write_data
0898	793A	1079	mov R1,#':'
089A	1208DF	1080	lcall write_data
089D	A951	1081	mov R1,51h
089F	1208DF	1082	lcall write_data
08A2	A952	1083	mov R1,52h
08A4	1208DF	1084	lcall write_data
08A7	7920	1085	mov R1,#' '
08A9	1208DF	1086	lcall write_data
08AC	794B	1087	mov R1,#'K'
08AE	1208DF	1088	lcall write_data
08B1	793D	1089	mov R1,#'='
08B3	1208DF	1090	lcall write_data
08B6	A94B	1091	mov R1,4Bh
08B8	1208DF	1092	lcall write_data
08BB	A94C	1093	mov R1,4Ch
08BD	1208DF	1094	lcall write_data
08C0	793A	1095	mov R1,#':'

LOC	OBJ	LINE	SOURCE
08C2	1208DF	1096	lcall write_data
08C5	A94D	1097	mov R1,4Dh
08C7	1208DF	1098	lcall write_data
08CA	A94E	1099	mov R1,4Eh
08CC	1208DF	1100	lcall write_data
08CF	22	1101	ret
		1102	;
		1103	write_inst:
08D0	C2A0	1104	clr P2.0
08D2	C2A1	1105	clr P2.1
08D4	89B0	1106	mov P3,R1
08D6	D2A2	1107	setb P2.2
08D8	11F4	1108	acall delay
08DA	C2A2	1109	clr P2.2
08DC	11F4	1110	acall delay
08DE	22	1111	ret
		1112	;
		1113	write_data:
08DF	D2A0	1114	setb P2.0
08E1	C2A1	1115	clr P2.1
08E3	89B0	1116	mov P3,R1
08E5	D2A2	1117	setb P2.2
08E7	11F4	1118	acall delay
08E9	C2A2	1119	clr P2.2
08EB	11F4	1120	acall delay
08ED	22	1121	ret
		1122	;
08EE	7F03	1123	FLdelay:mov R7,#3
08F0	11FD	1124	FLd: acall Ldelay
08F2	DFFC	1125	djnz R7,FLd
		1126	;
08F4	7E30	1127	delay: mov R6,#30h
08F6	7D10	1128	delay1: mov R5,#10h
08F8	DDFE	1129	djnz R5,\$
08FA	DEFA	1130	djnz R6,delay1
08FC	22	1131	ret
		1132	;
08FD	7CF0	1133	Ldelay: mov R4,#0F0h
08FF	11F4	1134	Ld1: acall delay
0901	DCFC	1135	djnz R4,Ld1
0903	22	1136	ret
		1137	;
		1138	tanggal:
0904	0558	1139	inc 58h
0906	1209F9	1140	lcall isitanggal
0909	E559	1141	mov A,59h
		1142	januari:
090B	B4011B	1143	cjne A,#1,februari
090E	7420	1144	mov A,#32
		1145	tambahbulan:
0910	B55815	1146	cjne A,58h,selesai
0913	755801	1147	mov 58h,#01
0916	0559	1148	inc 59h
0918	1209F9	1149	lcall isitanggal
091B	740D	1150	mov A,#13

LOC	OBJ	LINE	SOURCE
091D	B55908	1151	cjne A,59h,selesai
0920	755901	1152	mov 59h,#01
0923	055A	1153	inc 5Ah
0925	1209F9	1154	lcall isitanggal
		1155	selesai:
0928	22	1156	ret
		1157	februari:
0929	B40213	1158	cjne A,#2,maret
092C	E55A	1159	mov A,5Ah
092E	75F004	1160	mov B,#04
0931	84	1161	div AB
0932	E5F0	1162	mov A,B
0934	B40004	1163	cjne A,#0,biasa
0937	741E	1164	mov A,#30
0939	80D5	1165	sjmp tambahbulan
093B	741D	1166	biasa: mov A,#29
093D	80D1	1167	sjmp tambahbulan
093F	B40304	1168	maret: cjne A,#3,april
0942	7420	1169	mov A,#32
0944	80CA	1170	sjmp tambahbulan
0946	B40404	1171	april: cjne A,#4,mei
0949	741F	1172	mov A,#31
094B	80C3	1173	sjmp tambahbulan
094D	B40504	1174	mei: cjne A,#5,juni
0950	7420	1175	mov A,#32
0952	80BC	1176	sjmp tambahbulan
0954	B40604	1177	juni: cjne A,#6,juli
0957	741F	1178	mov A,#31
0959	80B5	1179	sjmp tambahbulan
095B	B40704	1180	juli: cjne A,#7,agustus
095E	7420	1181	mov A,#32
0960	80AE	1182	sjmp tambahbulan
		1183	agustus:
0962	B40804	1184	cjne A,#8,september
0965	7420	1185	mov A,#32
0967	80A7	1186	sjmp tambahbulan
		1187	september:
0969	B40904	1188	cjne A,#9,oktober
096C	741F	1189	mov A,#31
096E	80A0	1190	sjmp tambahbulan
		1191	oktober:
0970	B40A04	1192	cjne A,#10,november
0973	7420	1193	mov A,#32
0975	8099	1194	sjmp tambahbulan
		1195	november:
0977	B40B04	1196	cjne A,#11,desember
097A	741F	1197	mov A,#31
097C	8092	1198	sjmp tambahbulan
		1199	desember:
097E	B40C04	1200	cjne A,#12,kembali
0981	7420	1201	mov A,#32
0983	808B	1202	sjmp tambahbulan
		1203	kembali:
0985	80A1	1204	sjmp selesai
		1205	;

LOC	OBJ	LINE	SOURCE
		1206	Timerlinter:
0987	C0D0	1207	Push PSW
0989	C0E0	1208	Push Acc
098B	758D3C	1209	MOV TH1,#high aduh
098E	758BB0	1210	MOV TL1,#low aduh
0991	D53B11	1211	DJNZ 3Bh,SelesaiInterupsi
0994	753B0A	1212	MOV 3Bh,#10
0997	D53A0B	1213	DJNZ 3Ah,SelesaiInterupsi
099A	753A02	1214	Mov 3Ah,#2
		1215	; Terjadi sekali dalam waktu 1 detik
099D	31AA	1216	ACALL JamDigital
099F	D53D03	1217	DJNZ 3Dh,SelesaiInterupsi
09A2	753D3C	1218	MOV 3Dh,#60
		1219	SelesaiInterupsi:
09A5	D0E0	1220	Pop Acc
09A7	D0D0	1221	Pop PSW
09A9	32	1222	RETI
		1223	;
		1224	JamDigital:
09AA	055B	1225	Inc 5Bh
09AC	743C	1226	Mov A,#60
09AE	B55B1C	1227	Cjne A,5Bh,isitampilan1
09B1	755B00	1228	Mov 5Bh,#0
		1229	SatuMenit:
09B4	055C	1230	INC 5Ch
09B6	743C	1231	MOV A,#60
09B8	B55C12	1232	CJNE A,5Ch,isitampilan1
09BB	755C00	1233	MOV 5Ch,#0
		1234	SatuJam:
09BE	055D	1235	INC 5Dh
09C0	7418	1236	MOV A,#24
09C2	B55D08	1237	CJNE A,5Dh,isitampilan1
09C5	31CE	1238	acall hapusdata
09C7	120904	1239	lcall tanggal
09CA	755D00	1240	MOV 5Dh,#0
		1241	isitampilan1:
09CD	22	1242	ret
		1243	;
		1244	hapusdata:
09CE	795E	1245	mov R1,#5Eh
09D0	7A0A	1246	mov R2,#10
09D2	7700	1247	hp: mov @R1,#0
09D4	09	1248	inc R1
09D5	DAFB	1249	djnz R2,hp
09D7	22	1250	ret
		1251	;
		1252	isitampilan:
09D8	7830	1253	MOV R0,#30h
09DA	E55D	1254	mov A,5Dh
09DC	5117	1255	ACALL Ruas2
09DE	E55C	1256	mov A,5Ch
09E0	5117	1257	ACALL Ruas2
09E2	22	1258	RET
		1259	;
		1260	isitglmasuk:

;Disimpan mulai dari RuasJam10

OC	OBJ	LINE	SOURCE
09E3	784F	1261	MOV R0,#4Fh ;Disimpan mulai dari RuasJami0
09E5	E553	1262	mov A,53h
09E7	5117	1263	ACALL Ruas2
09E9	E554	1264	mov A,54h
09EB	5117	1265	ACALL Ruas2
09ED	22	1266	RET
		1267	;
		1268	isitglkeluar:
09EE	784B	1269	MOV R0,#4Bh ;Disimpan mulai dari RuasJami0
09F0	E555	1270	mov A,55h
09F2	5117	1271	ACALL Ruas2
09F4	E556	1272	mov A,56h
09F6	5117	1273	ACALL Ruas2
09F8	22	1274	RET
		1275	;
		1276	isitanggal:
09F9	7834	1277	MOV R0,#34h ;Disimpan mulai dari RuasJami0
09FB	E558	1278	mov A,58h
09FD	5117	1279	ACALL Ruas2
09FF	E559	1280	mov A,59h
0A01	5117	1281	Acall Ruas2
0A03	E55A	1282	mov A,5Ah
0A05	5117	1283	ACALL Ruas2
0A07	22	1284	RET
		1285	isitanggal1:
0A08	7874	1286	MOV R0,#74h ;Disimpan mulai dari RuasJami0
0A0A	E571	1287	mov A,71h
0A0C	5117	1288	ACALL Ruas2
0A0E	E572	1289	mov A,72h
0A10	5117	1290	Acall Ruas2
0A12	E573	1291	mov A,73h
0A14	5117	1292	ACALL Ruas2
0A16	22	1293	RET
		1294	Ruas2:
0A17	75F00A	1295	MOV B,#10
0A1A	84	1296	DIV AB ;A := A/10, sisanya di B
0A1B	511F	1297	ACALL Ruas1 ;Nilai A dirubah jadi ruas
0A1D	E5F0	1298	MOV A,B ;Kini B yang dijadikan ruas
0A1F	900A29	1299	Ruas1: mov DPTR,#segmen
0A22	5127	1300	acall JadikanRuas ;Konversikan nilai ke ruas
0A24	F6	1301	MOV @R0,A ;Simpan untuk ditampilkan
0A25	08	1302	INC R0 ;Alamat berikutnya
0A26	22	1303	RET
		1304	;
		1305	JadikanRuas:
0A27	93	1306	MOVC A,@A+DPTR
0A28	22	1307	RET
		1308	;
		1309	segmen:
0A29	30	1310	DB '0'
0A2A	31	1311	DB '1'
0A2B	32	1312	DB '2'
0A2C	33	1313	DB '3'
0A2D	34	1314	DB '4'
0A2E	35	1315	DB '5'

LOC	OBJ	LINE	SOURCE
0A2F	36	1316	DB '6'
0A30	37	1317	DB '7'
0A31	38	1318	DB '8'
0A32	39	1319	DB '9'
		1320	End

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND



MCS-51 MACRO ASSEMBLER EEPROM

07/28/:4 PAGE 1

DOS 7.10 (038-N) MCS-51 MACRO ASSEMBLER, V2.2

OBJECT MODULE PLACED IN EEPROM.OBJ

ASSEMBLER INVOKED BY: C:\MCS51~1\SIM51E~1\ASM51.EXE EEPROM.H51

LOC	OBJ	LINE	SOURCE
0000		1	org 000h
		2	awal:
0000 53657420		3	DB 'Set Jam dan Tgl '
0004 4A616D20			
0008 64616E20			
000C 54676C20			
		4	mpin:
0010 4B657469		5	DB 'Ketik PIN : '
0014 6B205049			
0018 4E203A20			
001C 20202020			
		6	salahpin:
0020 2050494E		7	DB ' PIN Anda salah '
0024 20416E64			
0028 61207361			
002C 6C616820			
0030 2050494E		8	gpin: DB ' PIN Lama : '
0034 204C616D			
0038 61203A20			
003C 20202020			
0040 2050494E		9	bpin: DB ' PIN Baru : '
0044 20426172			
0048 75203A20			
004C 20202020			
0050 2020204C		10	ltgl: DB ' Lihat data '
0054 69686174			
0058 20646174			
005C 61202020			
0060 56455249		11	veril: DB 'VERI MASUK'
0064 20202020			
0068 2020204D			
006C 4153554B			
0070 4B494120		12	kial: DB 'KIA MASUK'
0074 20202020			
0078 2020204D			
007C 4153554B			
0080 52494E49		13	rini1: DB 'RINI MASUK'
0084 20202020			
0088 2020204D			
008C 4153554B			
0090 59454E49		14	yenil: DB 'YENI MASUK'
0094 20202020			
0098 2020204D			
009C 4153554B			
00A0 524F4D49		15	romil: DB 'ROMI MASUK'
00A4 20202020			
00A8 2020204D			
00AC 4153554B			
00B0 41444920		16	adi1: DB 'ADI MASUK'

00B4 20202020

MCS-51 MACRO ASSEMBLER EEPROM

07/28/:4 PAGE 2

LOC	OBJ	LINE	SOURCE
00B8	2020204D		
00BC	4153554B		
00C0	4C494120	17	lia1: DB 'LIA MASUK'
00C4	20202020		
00C8	2020204D		
00CC	4153554B		
00D0	4E494120	18	nia1: DB 'NIA MASUK'
00D4	20202020		
00D8	2020204D		
00DC	4153554B		
00E0	4E414E41	19	nana1: DB 'NANA MASUK'
00E4	20202020		
00E8	2020204D		
00EC	4153554B		
00F0	54494120	20	tia1: DB 'TIA MASUK'
00F4	20202020		
00F8	2020204D		
00FC	4153554B		
0100	56455249	21	veri2: DB 'VERI KELUAR'
0104	20202020		
0108	20204B45		
010C	4C554152		
0110	4B494120	22	kia2: DB 'KIA KELUAR'
0114	20202020		
0118	20204B45		
011C	4C554152		
0120	52494E49	23	rini2: DB 'RINI KELUAR'
0124	20202020		
0128	20204B45		
012C	4C554152		
0130	59454E49	24	yeni2: DB 'YENI KELUAR'
0134	20202020		
0138	20204B45		
013C	4C554152		
0140	524F4D49	25	romi2: DB 'ROMI KELUAR'
0144	20202020		
0148	20204B45		
014C	4C554152		
0150	41444920	26	adi2: DB 'ADI KELUAR'
0154	20202020		
0158	20204B45		
015C	4C554152		
0160	4C494120	27	lia2: DB 'LIA KELUAR'
0164	20202020		
0168	20204B45		
016C	4C554152		
0170	4E494120	28	nia2: DB 'NIA KELUAR'
0174	20202020		
0178	20204B45		
017C	4C554152		
0180	4E414E41	29	nana2: DB 'NANA KELUAR'
0184	20202020		
0188	20204B45		

```

018C 4C554152
0190 54494120      30      tia2:  DB      'TIA      KELUAR'
MCS-51 MACRO ASSEMBLER  EEPROM
07/28/:4  PAGE      3

```

LOC	OBJ	LINE	SOURCE
0194	20202020		
0198	20204B45		
019C	4C554152		
01A0	54616E67	31	tgl1: DB 'Tanggal: '
01A4	67616C3A		
01A8	56657269	32	user1: DB 'Veri'
01AC	4B696120	33	user2: DB 'Kia'
01B0	52696E69	34	user3: DB 'Rini'
01B4	59656E69	35	user4: DB 'Yeni'
01B8	526F6D69	36	user5: DB 'Romi'
01BC	41646920	37	user6: DB 'Adi'
01C0	4C696120	38	user7: DB 'Lia'
01C4	4E696120	39	user8: DB 'Nia'
01C8	4E616E61	40	user9: DB 'Nana'
01CC	54696120	41	user10: DB 'Tia'
		42	
01D0	00	43	nomor: DB 00h,01h,02h,03h
01D1	01		
01D2	02		
01D3	03		
01D4	06	44	DB 06h,07h,08h,09h
01D5	07		
01D6	08		
01D7	09		
01D8	01	45	DB 01h,03h,04h,06h
01D9	03		
01DA	04		
01DB	06		
01DC	01	46	DB 01h,07h,09h,08h
01DD	07		
01DE	09		
01DF	08		
01E0	08	47	DB 08h,09h,01h,02h
01E1	09		
01E2	01		
01E3	02		
01E4	03	48	DB 03h,06h,01h,00h
01E5	06		
01E6	01		
01E7	00		
01E8	02	49	DB 02h,04h,06h,08h
01E9	04		
01EA	06		
01EB	08		
01EC	07	50	DB 07h,09h,08h,01h
01ED	09		
01EE	08		
01EF	01		
01F0	03	51	DB 03h,04h,05h,06h
01F1	04		
01F2	05		
01F3	06		

01F4 06 52 DB 06h,07h,05h,01h
 01F5 07
 01F6 05

□MCS-51 MACRO ASSEMBLER EEPROM

07/28/:4 PAGE 4

LOC OBJ LINE SOURCE

01F7 01

53

end

□MCS-51 MACRO ASSEMBLER EEPROM

07/28/:4 PAGE 5

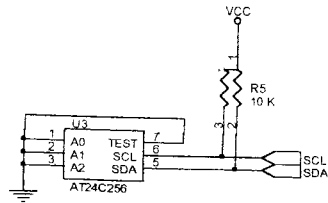
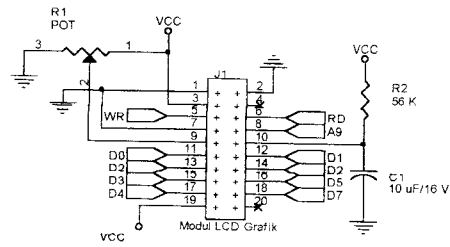
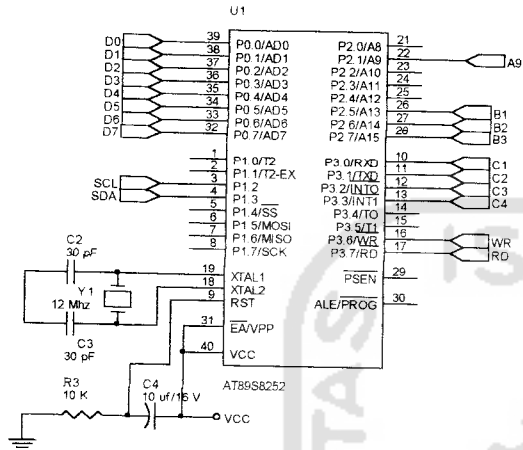
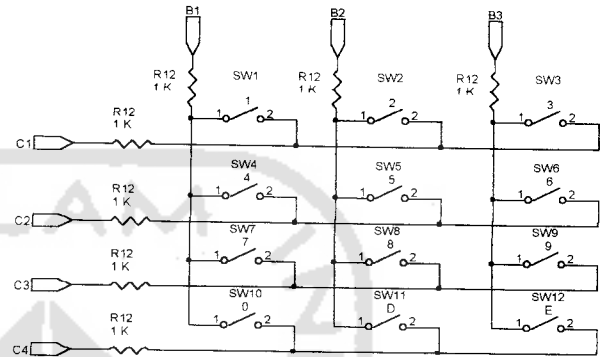
SYMBOL TABLE LISTING

N A M E	T Y P E	V A L U E	A T T R I B U T E S
ADI1 . . .	C ADDR	00B0H	A
ADI2 . . .	C ADDR	0150H	A
AWAL . . .	C ADDR	0000H	A
BPIN . . .	C ADDR	0040H	A
GPIN . . .	C ADDR	0030H	A
KIA1 . . .	C ADDR	0070H	A
KIA2 . . .	C ADDR	0110H	A
LIA1 . . .	C ADDR	00C0H	A
LIA2 . . .	C ADDR	0160H	A
LTGL . . .	C ADDR	0050H	A
MPIN . . .	C ADDR	0010H	A
NANA1. . .	C ADDR	00E0H	A
NANA2. . .	C ADDR	0180H	A
NIA1 . . .	C ADDR	00D0H	A
NIA2 . . .	C ADDR	0170H	A
NOMOR. . .	C ADDR	01D0H	A
RINI1. . .	C ADDR	0080H	A
RINI2. . .	C ADDR	0120H	A
ROMI1. . .	C ADDR	00A0H	A
ROMI2. . .	C ADDR	0140H	A
SALAHPIN	C ADDR	0020H	A
TGLL . . .	C ADDR	01A0H	A
TIA1 . . .	C ADDR	00F0H	A
TIA2 . . .	C ADDR	0190H	A
USER1. . .	C ADDR	01A8H	A
USER10 . .	C ADDR	01CCH	A
USER2. . .	C ADDR	01ACH	A
USER3. . .	C ADDR	01B0H	A
USER4. . .	C ADDR	01B4H	A
USER5. . .	C ADDR	01B8H	A
USER6. . .	C ADDR	01BCH	A
USER7. . .	C ADDR	01C0H	A
USER8. . .	C ADDR	01C4H	A
USER9. . .	C ADDR	01C8H	A
VERI1. . .	C ADDR	0060H	A
VERI2. . .	C ADDR	0100H	A
YENI1. . .	C ADDR	0090H	A
YENI2. . .	C ADDR	0130H	A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE, NO ERRORS FOUND





Features

Compatible with MCS[®]51 Products

8K Bytes of In-System Reprogrammable Downloadable Flash Memory

- SPI Serial Interface for Program Downloading
- Endurance: 1,000 Write/Erase Cycles

2K Bytes EEPROM

- Endurance: 100,000 Write/Erase Cycles

4V to 6V Operating Range

Fully Static Operation: 0 Hz to 24 MHz

Three-level Program Memory Lock

- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

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

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

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



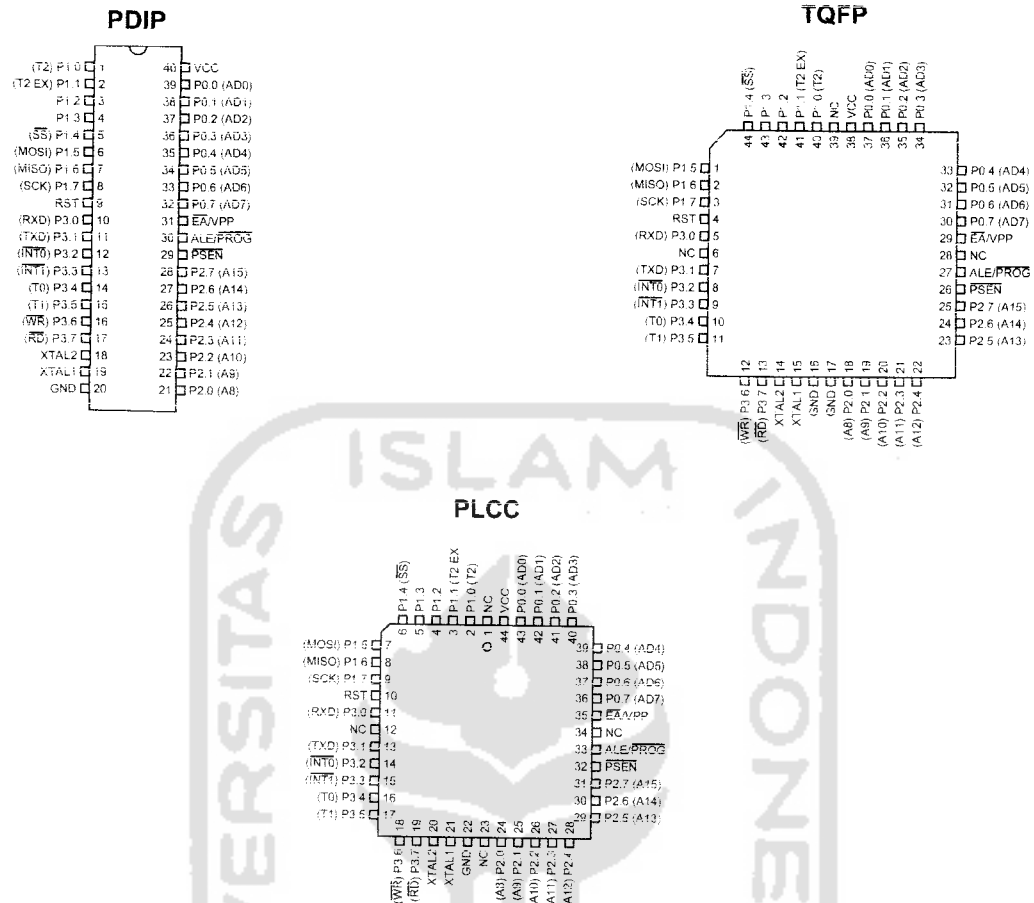
**8-bit
Microcontroller
with 8K Bytes
Flash**

AT89S8252

0401F-MICRO-11/03



in Configurations



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

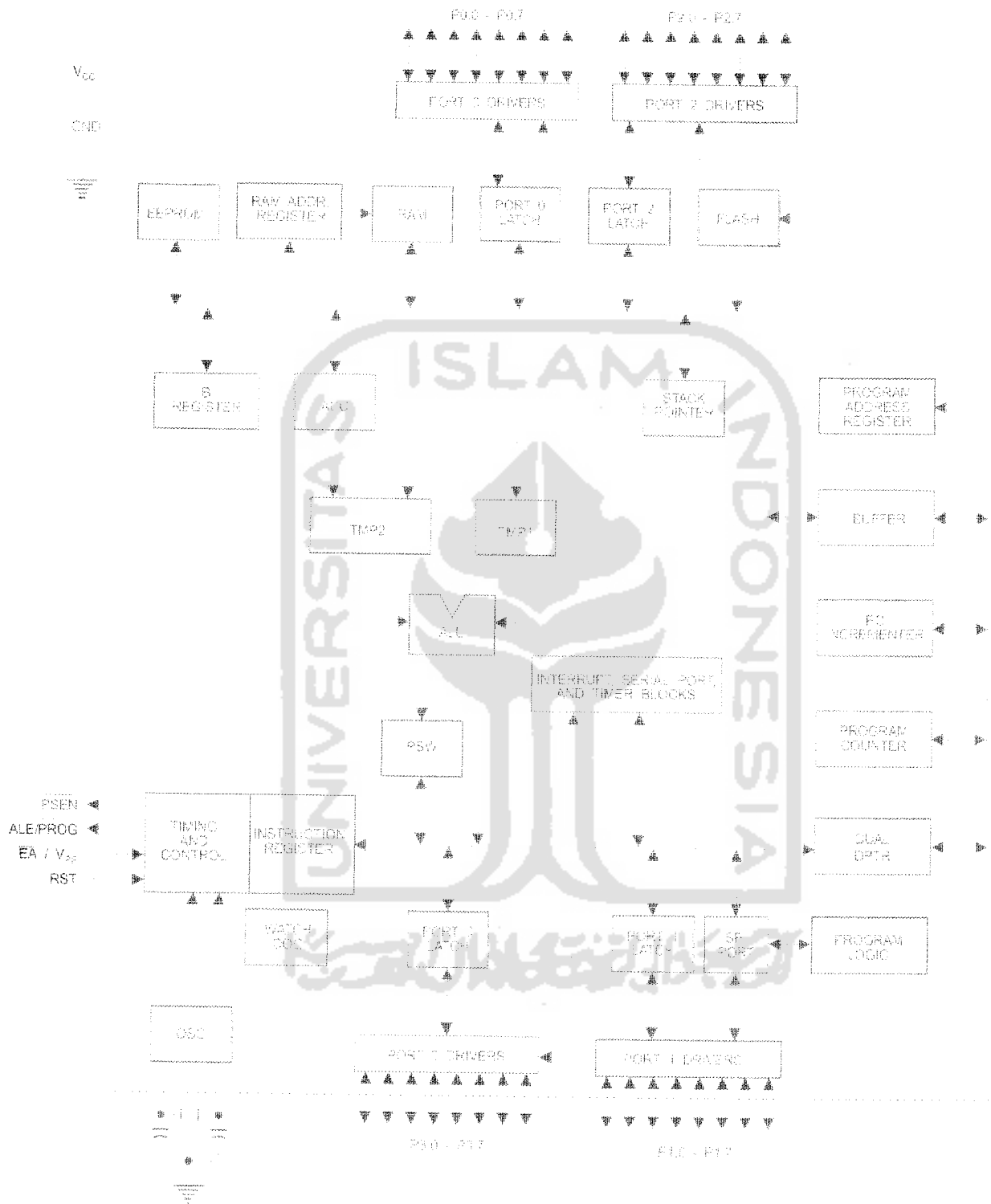
Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.

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

Port 1

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

Block Diagram



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

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

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

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

Port 2

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

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

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

Port 3

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	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	WR (external data memory write strobe)
P3.7	RD (external data memory read strobe)

RST

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

ALE/PROG

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

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

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

PSEN

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

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

EA/VPP

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

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

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

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

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

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

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

Table 1. AT89S8252 SFR Map and Reset Values

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

ble 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H

Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
	7	6	5	4	3	2	1	0

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



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

Figure 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H

Reset Value = 0000 0010B

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

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

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

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

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

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

Table 4. SPCR – SPI Control Register

SPCR Address = D5H

Reset Value = 0000 01XXB

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

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

ble 5. SPSR – SPI Status Register

PSR Address = AAH

Reset Value = 00XX XXXXB

	SPIF	WCOL	–	–	–	–	–	–
Bit	7	6	5	4	3	2	1	0

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

Table 6. SPDR – SPI Data Register

SPDR Address = 86H

Reset Value = unchanged

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

Data Memory – EEPROM and RAM

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

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

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

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

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

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

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

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

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

programming is still in progress and $\overline{\text{RDY/BSY}} = 1$ means EEPROM write cycle is completed and another write cycle can be initiated.

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

Programmable Watchdog Timer

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

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

Table 7. Watchdog Timer Period Selection

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

Timer 0 and 1

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

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/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.

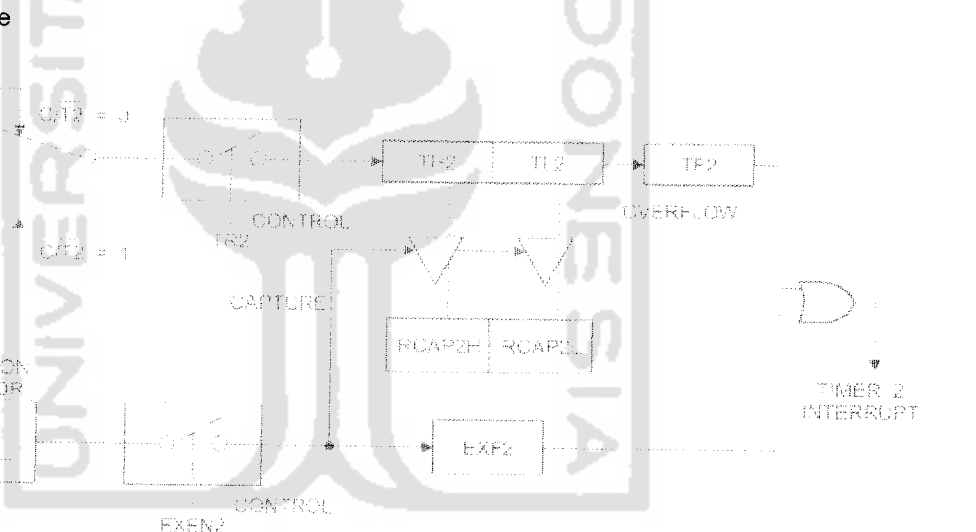
Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

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.

Table 8. Timer 2 Operating Modes

(continued)

In the capture mode, two options are selected by bit EXEN2 in T3CON. If EXEN2 = 0



Auto-reload (Up or Down Counter)

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

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

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

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

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

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



ble 9. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H

Reset Value = XXXX XX00B

Not Bit Addressable

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

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

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

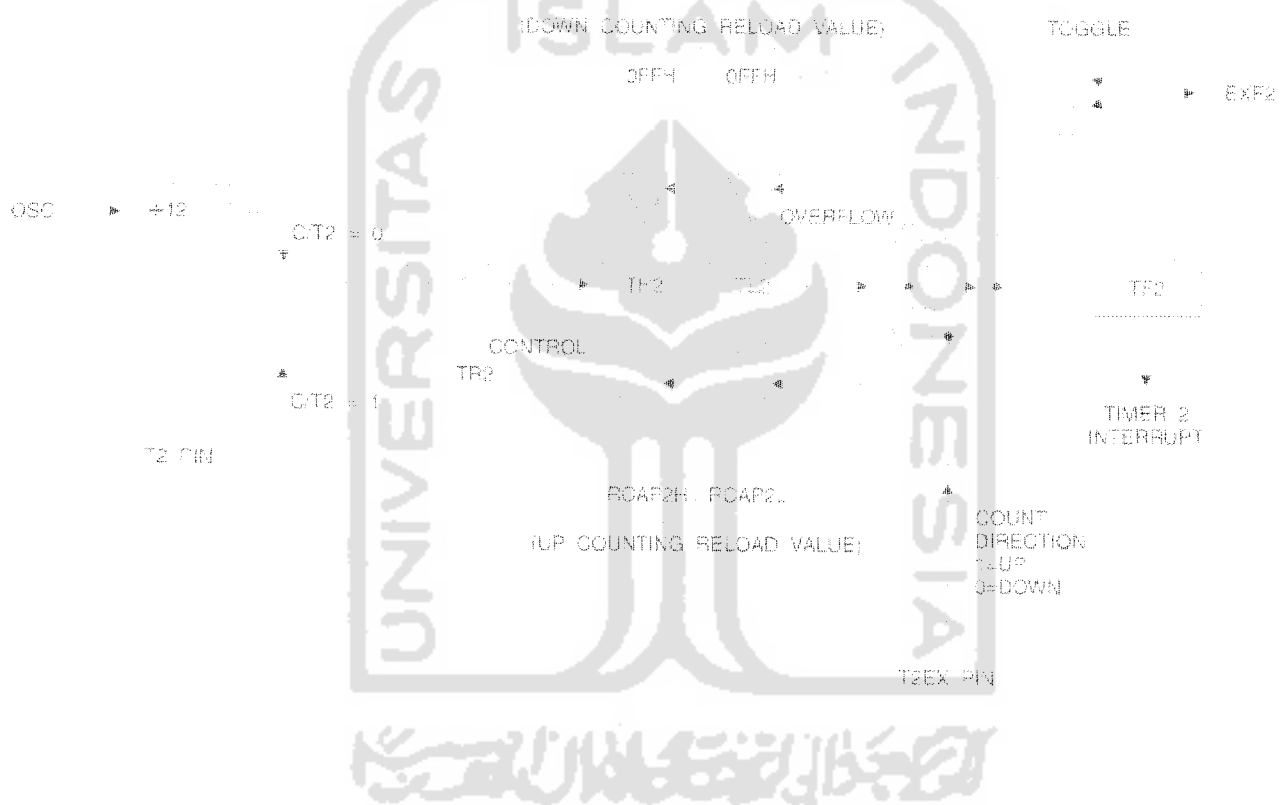
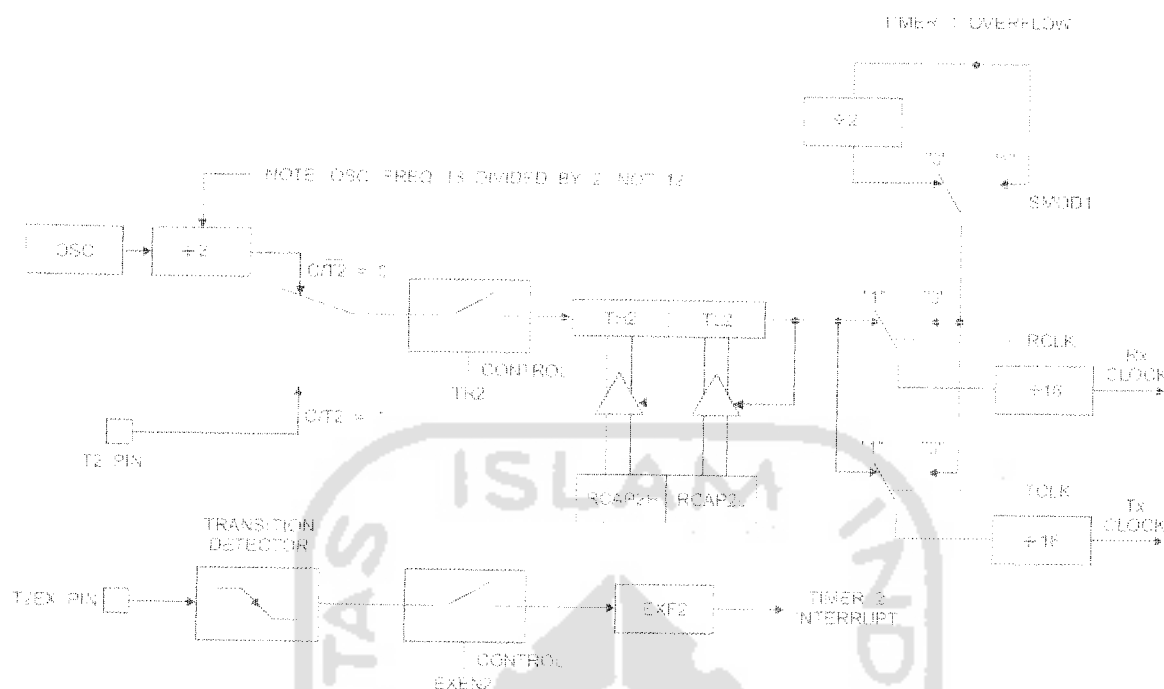


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

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

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

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

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

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($CP/\overline{T2} = 0$). 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.

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

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

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

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

Programmable Clock Out

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

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

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

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

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

Figure 6. SPI Block Diagram



UART

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

Serial Peripheral Interface

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

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

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

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

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

Figure 7. SPI Master-slave Interconnection

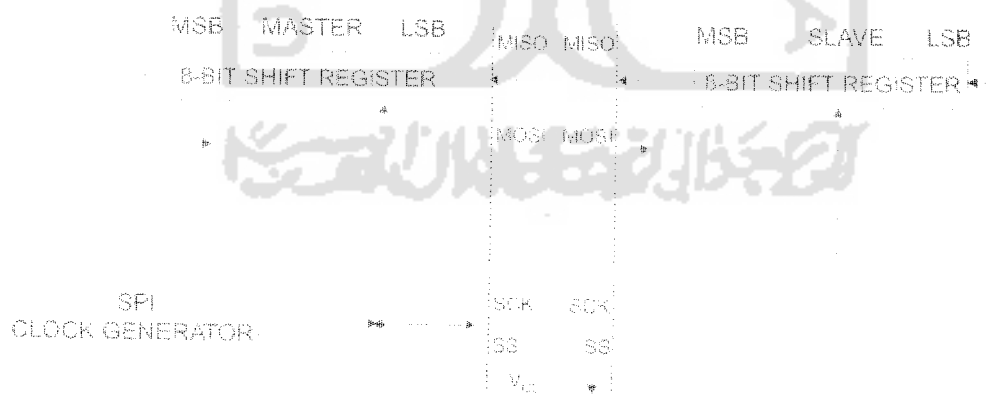
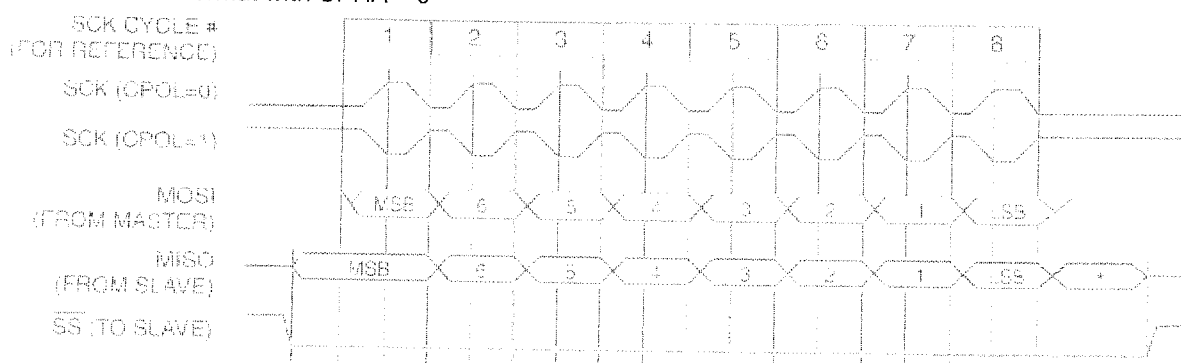
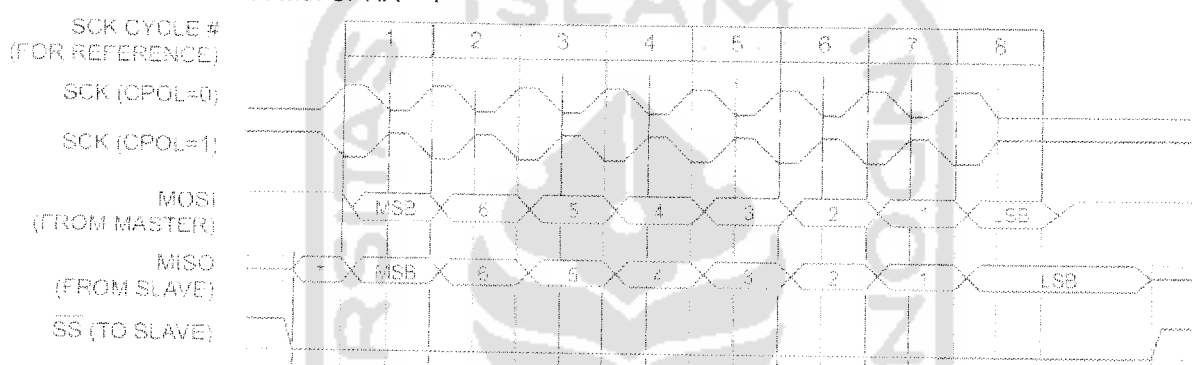


Figure 8. SPI Transfer Format with CPHA = 0



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

Figure 9. SPI Transfer Format with CPHA = 1



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

Interrupts

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

SB)(LSB)

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables the interrupt.

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

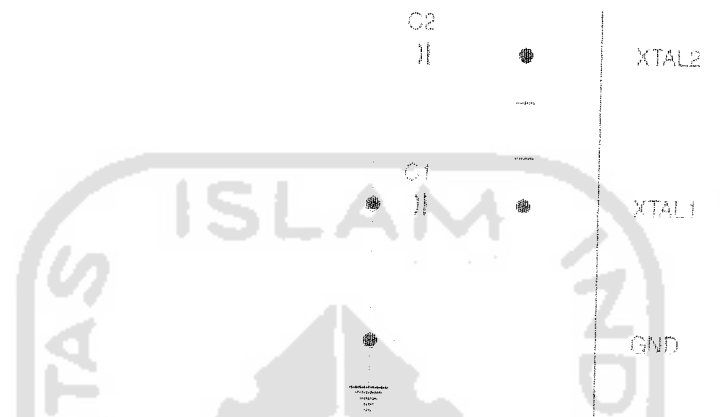
SPI and UART interrupt enable bit.
Timer 1 interrupt enable bit.
External interrupt 1 enable bit.
Timer 0 interrupt enable bit.
External interrupt 0 enable bit.

to unimplemented bits, because they may be used in future AT89C51

Oscillator Characteristics

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

Figure 11. Oscillator Connections



Note: C1, C2 = $30 \text{ pF} \pm 10 \text{ pF}$ for Crystals
 = $40 \text{ pF} \pm 10 \text{ pF}$ 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	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

Power-down Mode

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

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

Program Memory Lock Bits

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

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

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

Lock Bit Protection Modes⁽¹⁾⁽²⁾

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

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

Programming the Flash and EEPROM

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The AT89S8252 is shipped with the Serial Programming Mode enabled.

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

(030H) = 1EH indicates manufactured by Atmel

(031H) = 72H indicates 89S8252

Programming Interface

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

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

Serial Downloading

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

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

The Code and Data memory arrays have separate address spaces:

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

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

Serial Programming Algorithm

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

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

Serial Programming Instruction

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

Instruction Set

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

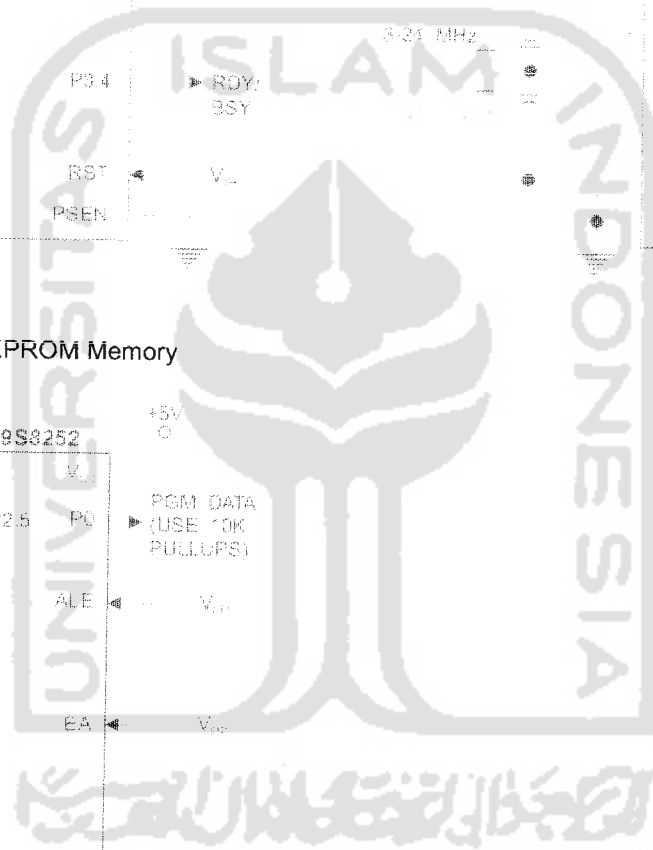
- Notes:
1. DATA polling is used to indicate the end of a byte write cycle which typically takes less than 2.5 ms at 5V.
 2. "aaaaa" = high order address.
 3. "x" = don't care.

Serial and EEPROM Parallel Programming Modes

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

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

Figure 15. Flash/EEPROM Serial Downloading



93C252

+5V

GND

PGM DATA
(USE 10K
PULLUPS)

ALE

EA

V_{cc}

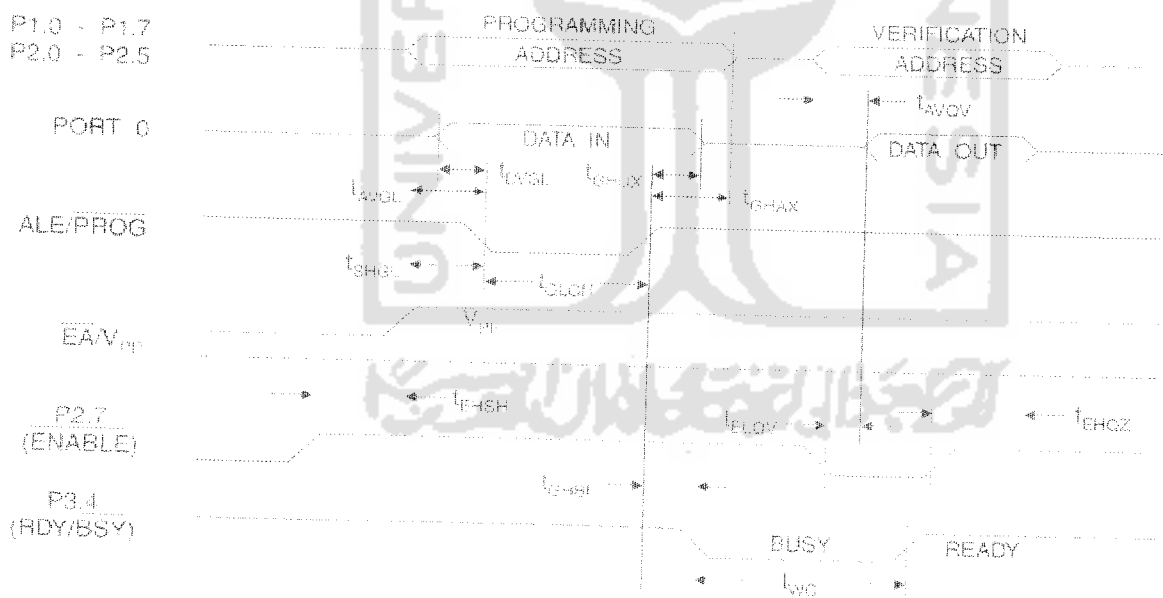
V_{ee}

Flash Programming and Verification Characteristics – Parallel Mode

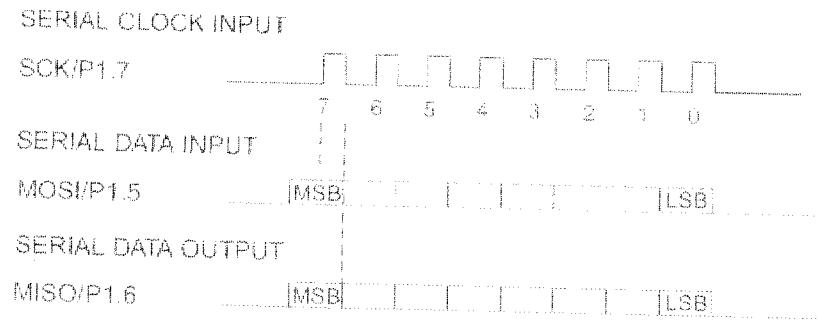
0°C to 70°C, $V_{CC} = 5.0V \pm 10\%$

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

Flash/EEPROM Programming and Verification Waveforms – Parallel Mode



Serial Downloading Waveforms



Serial Programming Characteristics

Figure 16. Serial Programming Timing

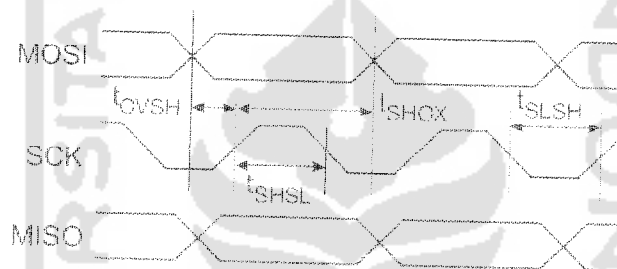


Table 11. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 6.0\text{V}$ (Unless Otherwise Noted)

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



Interfacing AT24CXX Serial EEPROMs with AT89CX051 Microcontrollers

Serial memory devices offer significant advantages over parallel devices in applications where lower data transfer rates are acceptable. In addition to requiring less board space, serial devices allow microcontroller I/O pins to be conserved. This is especially valuable when adding external memory to low pin count microcontrollers such as the Atmel AT89C1051 and AT89C2051.

This application note presents a suite of software routines which may be incorporated into a user's application to allow an AT89CX051 microcontroller to read and write AT24CXX serial EEPROMs. The software supports all members of the AT24CXX family, and may easily be modified for compatibility with any of the Atmel 8051-code compatible microcontrollers.

Hardware

A typical interconnection between an AT89CX051 microcontroller and an AT24CXX serial EEPROM is shown in Figure 1. As indicated in the figure, up to eight members of the AT24CXX family may share the bus, utilizing the same two microcontroller I/O pins. Each device on the bus must have its address inputs (A0, A1, A2) hard-wired to a unique address. In the figure, the first device recognizes address zero (A0, A1, A2 tied low), while the eighth recognizes address seven (A0, A1, A2 tied high). Not all members of the AT24CXX family recognize all three address inputs, limiting the number of some devices which

(continued)

Interfacing 24CXX Serial EEPROMs

Application Note

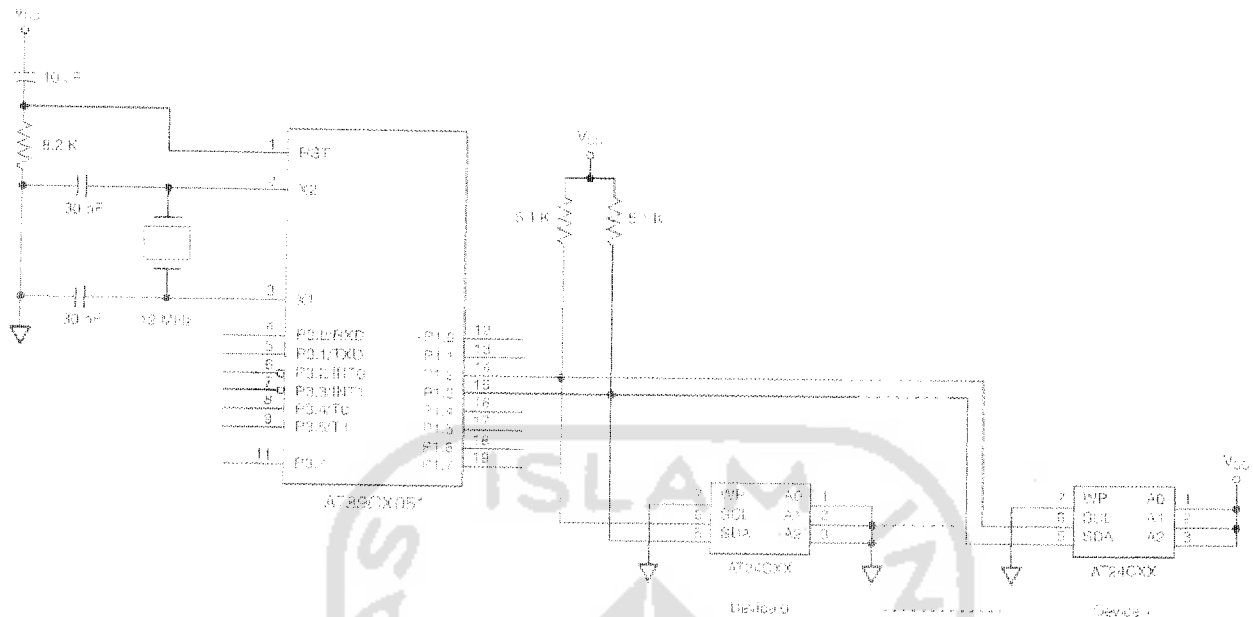
Table 1. Atmel's 2-Wire Serial EEPROM Family

Device	Size (Bytes)	Page Size (Bytes)	Max Per Bus	Addresses Used
AT24C01	1K	4	1	None
AT24C01A	1K	8	8	A0, A1, A2
AT24C02	2K	8	8	A0, A1, A2
AT24C04	4K	16	4	A1, A2
AT24C08	8K	16	2	A2
AT24C16	16K	16	1	None
AT24C164	16K	16	8	A0, A1, A2
AT24C32	32K	32	8	A0, A1, A2
AT24C64	64K	32	8	A0, A1, A2
AT24C128	128K	64	4	A0, A1
AT24C256	256K	64	4	A0, A1
AT24C512	512K	128	4	A0, A1
AT24C1024	1M	256	2	A1

Rev. 0507D-05/01



Figure 1. Typical Circuit Configuration



may be present to less than eight. The exact number of devices of each type which may share the bus is shown in Table 1.

Bidirectional Data Transfer Protocol

The Bidirectional Data Transfer Protocol utilized by the AT24CXX family allows a number of compatible devices to share a common 2-wire bus. The bus consists of a serial clock (SCL) line and a serial data (SDA) line. The clock is generated by the bus master and data is transmitted serially on the data line, most significant bit first, synchronized to the clock. The protocol supports bidirectional data transfers in 8-bit bytes.

In this application, the microcontroller serves as the bus master, initiating all data transfers and generating the clock which regulates the flow of data. The serial devices present on the bus are considered slaves, accepting or sending data in response to orders from the master.

The bus master initiates a data transfer by generating a start condition on the bus. This is followed by transmission of a byte containing the device address of the intended recipient. The device address consists of a 4-bit fixed portion and a 3-bit programmable portion. The fixed portion must match the value hard-wired into the slave, while the programmable portion allows the master to select between a maximum of eight slaves of similar type on the bus.

AT24CXX serial EEPROMs respond to device addresses with a fixed portion equal to '1010' and a programmable portion matching the address inputs (A0, A1, A2). Not all members of the AT24CXX family examine all three address inputs; Table 1 shows which of the three address inputs are valid for each member of the family.

The eighth bit in the device address byte specifies a write or read operation. After the eighth bit is transmitted, the master releases the data line and generates a ninth clock. If a slave has recognized the transmitted device address, it will respond to the ninth clock by generating an acknowledge condition on the data line. A slave which is busy when addressed may not generate an acknowledge. This is true for the AT24CXX when a write operation is in progress.

Following receipt of the slave's address acknowledgment, the master continues with the data transfer. If a write operation has been ordered, the master transmits the remaining data, with the slave acknowledging receipt of each byte. If the master has ordered a read operation, it releases the data line and clocks in data sent by the slave. After each byte is received, the master generates an acknowledge condition on the bus. The acknowledge is omitted following receipt of the last byte. The master terminates all operations by generating a stop condition on the bus. The master may also abort a data transfer at any time by generating a stop condition.

Refer to the AT24CXX family data sheets for detailed information on AT24CXX device operation and Bidirectional Data Transfer Protocol bus timing.

The software for this application may be obtained by downloading from Atmel's Web Site or BBS: (408) 436-4309. Consult the comment block at the beginning of the source code file for detailed information on features and operation.



Web Site: <http://www.atmel.com>

BBS: 1-(408) 436-4309

M1632 MODULE LCD 16 X 2 BARIS (M1632)

Deskripsi:

M1632 adalah merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGROM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory) dan DDRAM (Display Data Random Access Memory).

DDRAM

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Contoh, untuk karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD.

Display position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9
	BA	BB	BC	BD	BE	BF	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
	CA	CB	CC	CD	CE	CF	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3
	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	FA	FB	FC	FD
	FE	FF														

Gambar 1

DDRAM M1632 (diambil dari data sheet HD44780)

CGRAM

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

CGROM

CGROM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubahnya. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif.

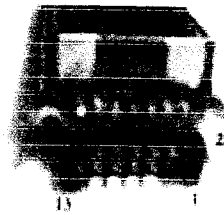
Pada gambar 2, tampak terlihat pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Pada saat HD44780 akan menampilkan data 41H yang tersimpan pada DDRAM, maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter A.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000 0000			0	1	A	Q	a	q			-	9	E	e	P	
0001 0001			!	2	B	R	b	r			.	7	F	f	5	q
0010 0010			"	3	C	S	c	s			'	8	H	h	x	
0011 0011			#	4	D	T	d	t			~	5	E	e	u	
0100 0100			\$	5	E	U	e	u			.	7	F	f	5	q
0101 0101			&	6	F	V	f	v			7	G	W	w		
0110 0110			'	7	G	W	w				7	G	W	w		
0111 0111			(	8	H	X	h	x			7	G	W	w		
1000 1000			)	9	I	Y	i	y			7	G	W	w		
1001 1001			*	:	J	Z	j	z			7	G	W	w		
1010 1010			+	:	K	L	k	l			7	G	W	w		
1011 1011			,	<	L	l					7	G	W	w		
1100 1100			-	=	M	J	m				7	G	W	w		
1101 1101			.	>	N	^	n				7	G	W	w		
1110 1110			/	?	O	_	o				7	G	W	w		
1111 1111											7	G	W	w		

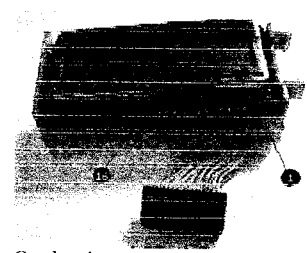
Gambar 2

Hubungan antara CGROM dan DDRAM (diambil dari data sheet HD44780)

Pin Out		Nama Pin		Deskripsi	
No					
1	VCC			+5V	
2	GND			0V	
3	VEE			Tegangan Kontras LCD	
4	RS			Register Select, 0 = Register Perintah, 1 = Register I	
5	R/W			1 = Read, 0 = Write	
6	E			Enable Clock LCD, logika 1 setiap kali pengiriman data	
7	D0			Data Bus 0	
8	D1			Data Bus 1	
9	D2			Data Bus 2	
10	D3			Data Bus 3	
11	D4			Data Bus 4	
12	D5			Data Bus 5	
13	D6			Data Bus 6	
14	D7			Data Bus 7	
15	Anoda (Kabel coklat untuk LCD Hitachi)			Tegangan positif backlight	
16	Katoda (Kabel merah untuk LCD Hitachi)			Tegangan negatif backlight	



Gambar 3
Pin Out M1632 LCD Hitachi



Gambar 4
Pin Out LCD M1632 Standard

Register

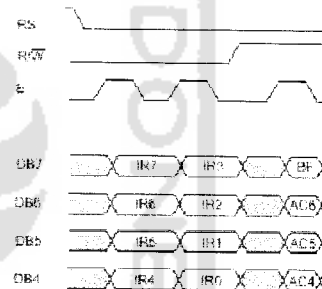
HD44780, mempunyai dua buah Register yang aksesnya diatur dengan menggunakan kaki. Pada saat RS berlogika 0, maka register yang diakses adalah Register Perintah dan pada saat RS berlogika 1, maka register yang diakses adalah Register Data.

Register Perintah

Register ini adalah register di mana perintah-perintah dari mikrokontroler ke HD44780 pada proses penulisan data atau tempat status dari HD44780 dapat dibaca pada saat pembacaan data.

Penulisan Data ke Register Perintah

Penulisan data ke Register Perintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi dan mengatur Address Counter maupun Address Data. Gambar 5 menunjukkan proses penulisan data ke Register Perintah. RW berlogika 0 yang menunjukkan proses penulisan data akan dilakukan. Nibble tinggi (bit 7 sampai bit 4) terlebih dahulu dikirimkan dengan diawali pulsa logika 1 pada E Clock. Kemudian Nibble rendah (bit 3 sampai bit 0) dikirimkan dengan diawali pulsa logika 1 pada E Clock. Untuk mode 8 bit interface, proses penulisan dapat langsung dilakukan secara 8 bit (bit 7 ... bit 0) diawali sebuah pulsa logika 1 pada E Clock.



Gambar 5

Timing diagram Penulisan Data ke Register Perintah Mode 4 bit Interface

Tabel 1

Perintah-perintah M1632

Perintah	D7	D6	D5	D4	D3	D2	D1	D0	Deskripsi
Hapus Display	0	0	0	0	0	0	0	1	Hapus Display dan DDRAM
Posisi Awal	0	0	0	0	0	0	1	X	Set Alamat DDRAM di 0
Set Mode	0	0	0	0	0	1	1/0	S	Atur arah pergeseran cursor dan display

DELTA ELECTRONIC

<http://www.delta-electronic.com>

Display On/OFF	0	0	0	0	1	D	C	B	Anur display (D) On/OFF, cursor ON/OFF, Blinking (B)
Geser Cursor/Display	0	0	0	1	S/C	R/L	X	X	Geser Cursor atau display tanpa mer alamat DDRAM
Set Fungsi	0	0	1	DL	N	F	X	X	Atur panjang data, jumlah baris tampil, dan font karakter
Set Alamat CGRAM	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Data dapat dibaca atau ditulis set alamat diatur
Set Alamat DDRAM	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Data dapat dibaca atau ditulis set alamat diatur

X = diabaikan

I/D 1=Increment, 0=Decrement

S 0=Display tidak geser

S/C 1=Display Shift, 0=Geser Cursor

R/L 1=Geser Kiri, 0=Geser Kanan

DL 1=8 bit, 0=4bit

N 1=2 baris, 0=1 baris

F 1=5x10, 0=5x8

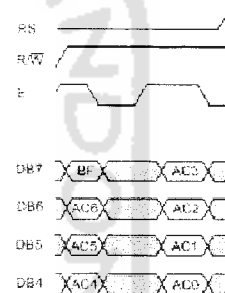
D 0=Display OFF, 1=Display ON

C 0=Cursor OFF, 1=Cursor ON

B 0=Blinking OFF, 1=Blinking ON

Pembacaan Data dari Register Perintah

Proses pembacaan data pada register perintah biasa digunakan untuk melihat status busy dari LCD atau membaca Address Counter. RS diatur pada logika 0 untuk akses ke Register Perintah, R/W diatur pada logika 1 yang menunjukkan proses pembacaan data. 4 bit nibble tinggi dibaca dengan diawali pulsa logika 1 pada E Clock dan kemudian 4 bit nibble rendah dibaca dengan diawali pulsa logika 1 pada E Clock. Untuk Mode 8 bit interface, pembacaan 8 bit (nibble tinggi dan rendah) dilakukan sekaligus dengan diawali sebuah pulsa logika 1 pada E Clock.



Gambar 6

Timing Diagram Pembacaan Register Perintah Mode 4 bit Interface

Register Data

Register ini adalah register di mana mikrokontroler dapat menuliskan atau membaca data ke dan dari DDRAM. Penulisan data pada register ini akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

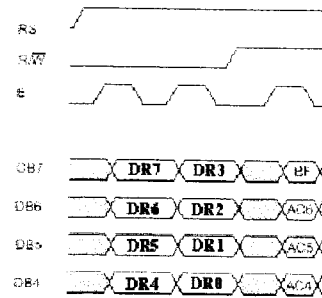
Penulisan Data ke Register Data

Penulisan data pada Register Data dilakukan untuk mengirimkan data yang akan ditampilkan pada LCD. Proses diawali dengan adanya logika 1 pada RS yang menunjukkan akses ke Register Data, kondisi R/W diatur pada logika 0 yang menunjukkan proses penulisan data. Data 4 bit nibble tinggi (bit 7 hingga

DELTA ELECTRONIC

<http://www.delta-electronic.com>

bit 4) dikirim dengan diawali pulsa logika 1 pada sinyal E Clock dan kemudian diikuti 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali pulsa logika 1 pada sinyal E Clock.

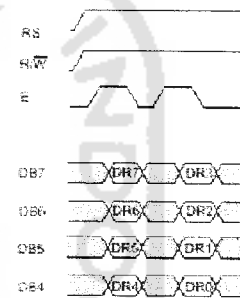


Gambar 7

Timing Diagram Penulisan Data ke Register Data Mode 4 bit Interface

Pembacaan Data dari Register Data

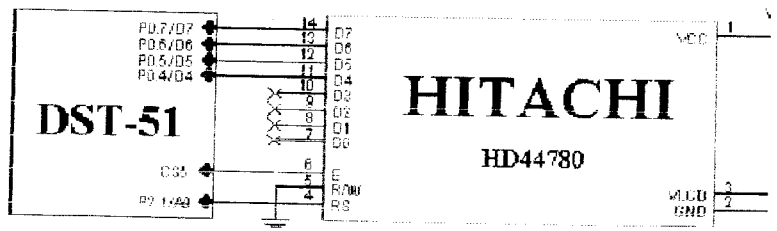
Pembacaan data dari Register Data dilakukan untuk membaca kembali data yang tampil p LCD. Proses dilakukan dengan mengatur RS pada logika 1 yang menunjukkan adanya akses ke Regi Data. Kondisi R/W diatur pada logika tinggi yang menunjukkan adanya proses pembacaan data. Data 4 nibble tinggi (bit 7 hingga bit 4) dibaca dengan diawali adanya pulsa logika 1 pada E Clock dan dilanjut dengan data 4 bit nibble rendah (bit 3 hingga bit 0) yang juga diawali dengan pulsa logika 1 pada E Clock



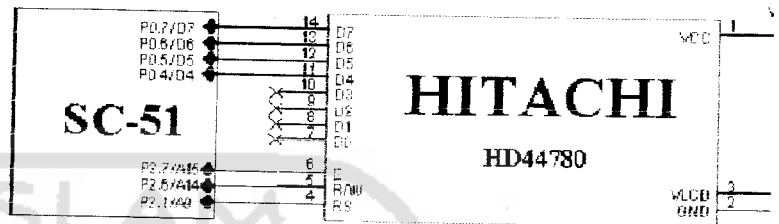
Gambar 8

Timing Diagram Pembacaan Data dari Register Data Mode 4 bit Interface

Antar muka LCD dengan mikrokontroler



Gambar 9
Antar muka dengan Modul DST-51



Gambar 10
Antar Muka dengan Modul SC-51 atau AT8951

Program

Rutin-rutin Program untuk DST-51 yang diassembly dengan ALDS atau ASM51

Rutin-rutin Program untuk SC-51/AT8951 yang diassembly dengan ALDS atau ASM51

Rutin delay yang diassembly dengan ALDS atau ASM51

Datasheet: HD44780