

SKRIPSI

PERANCANGAN DAN PEMBUATAN SISTEM PENGELOLAAN PERPUSTAKAAN MENGGUNAKAN KARTU RFID, MIKROKONTROLLER AT89S52 DAN PC



Disusun Oleh :
DONY KRISTIawan
NIM : 03.17.078

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

MARET 2008

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN SISTEM
PENGELOLAAN PERPUSTAKAAN MENGGUNAKAN
KARTU RFID, MIKROKONTROLLER AT89S52 DAN PC**

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Elektronika Strata Satu (S-1)*

Disusun Oleh :

**DONY KRISTIAWAN
NIM : 03.17.078**

Diperiksa dan Disetujui

Dosen Pembimbing I

Dosen Pembimbing II

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

**Dr. Cahyo Crysdian, Msc
NIP.Y. 103040412**

Mengetahui

Ketua Jurusan Teknik Elektro S-1



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

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

MARET 2008



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

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

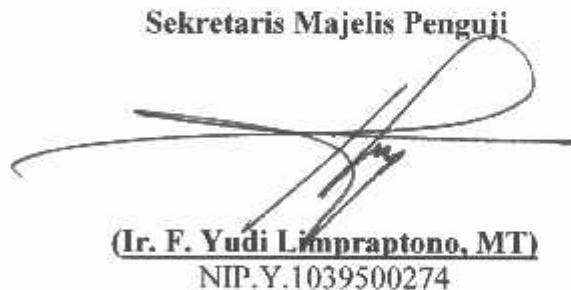
Nama : Dony Kristiawan
NIM : 03.17.078
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan dan Pembuatan Sistem Pengelolaan Perpustakaan
Menggunakan Kartu RFID, Mikrokontroller AT89S52 Dan PC

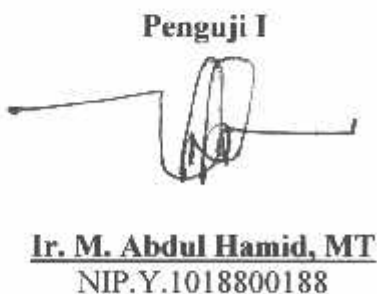
Dipertahankan di hadapan majelis penguji Skripsi jenjang Strata satu (S-1) pada :

Hari : Sabtu
Tanggal : 15 Maret 2008
Dengan Nilai : 85,44 (A) *84*

Ketua Majelis Penguji

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

Sekretaris Majelis Penguji

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

Penguji I

Ir. M. Abdul Hamid, MT
NIP.Y.1018800188

Penguji II

(I Komang Somawirata, MT)
NIP.P.1030100361

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran-Mu Ya Allah yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat menyelesaikan skripsi yang berjudul **“Perancangan Dan Pembuatan Sistem Pengelolaan Perpustakaan Menggunakan Kartu RFID, Mikrokontroller AT89S52 Dan PC”** ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan penyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penulis menyampaikan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
2. Bapak Ir. F. Yudi Limpraptono, MT selaku Dosen Pembimbing I
3. Bapak Dr. Cahyo Crysdiyan Msc, selaku Dosen Pembimbing II
4. Bapak dan Ibu serta saudara-saudara kami yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
5. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu Penulis mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan dan pembaca.

Malang, Maret 2008

Penulis



Persembahkan

وَالْعَصْرِ

إِنَّ الْإِنْسَانَ لَقِي خُسْرٍ

إِلَّا الَّذِينَ ءَامَنُوا وَعَمِلُوا الصَّالِحَاتِ وَتَوَّصُوا بِالحَقِّ وَتَوَّصُوا بِالصَّبْرِ

*"Demi masa ...
Sesungguhnya manusia itu benar-benar dalam kerugian,
kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya
mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran"*

syukur alhamdulillah, atas segala rahmad dan hidayah-Nya, penulis dapat menyelesaikan skripsi ini sesuai dengan rencana.

*....aku persembahkan skripsi ini kepada kedua orang tua ku
[kasnadi & susminingsih] yang telah dengan sabar,
mendidik, membimbing, membiayai, mendorong aku sehingga menjadi seperti
sekarang ini: (pak : buk !! akhir'e doni lulus juga) tak lupa kakakku [mbak
deni], [mak lin, mak siti almh, mbak um, intan, bang yus] ponakanku [daffa] yang
slalu ngriwuki disela-sela ngerjakan TA, keep smart le' kemampuanmu luar
biasa, juga [mas dodix] mas enek lowongan g??!! "ntar nyusul ksana..."*


Penulis juga mengucapkan terima kasih kepada :

Seluruh civitas & dosen elektronika ITN Malang...

pak yudi [terima kasih pak: bimbingan skripsinya selama ini], pak

cahyo [terima kasih pak: ternyata nasehat bapak ada benarnya]..pak

Joseph [selaku dosen wali selama penulis duduk dibangku kuliah] juga kepada
dosen-dosen lainnya yang tidak mungkin penulis sebutkan satu persatu...



Untuk teman2 seperjuangan ELEKTRONIKA Angkatan
2003...

prend smoga te2p kompak sampai esok kelak...

Untuk :

C_ndra[kencrut_AE] : prend thk u,meskipun agak g'teli temen yang atu ini dah banyak b'ntu mulai ngeprint laporan,tumpangan ke kmpus ampek info kerja pokok'e mantep tenan... prend moga-2 usaha golek gawean slama ini ada hasile..amien. ...\$@ndy[g'teli_crocodile land] : bro mana aweweknya???dah dead line neh g ada perkembangan link te2p aja kosong so jojoba...broo...ntar kalo dah gawe ojo lali ama temennya... @gu\$[hermwan] : bener-2 prend yang satu ini klo dah berargumen yang laen lwt,nothing lose lah,keep like this bro.meskipun begitu prend yang atu ini dah banyak tak repoti mulai dari flash disk n infrared yang dah beberapa semester g' balik-2 ampek skarang [buat kenang-2 aja y broo..he-22..] emank g' ada loe g' rame...,bro ntar klo dah gawe neng luar negri ajak-2 temennya.thon/hacking[wawar/emc]:le kapan garap TA22 ojo ngenet wae, & ilmune dibagi-bagi neng konco-koncane

[k_GM] : true prend..[maklum masa sulit n seneng slalu ngancani n masuk cari awewek..he-2x] n dy ama awewek jangan posisif klo udah pada kabur gigit jari loe[just joke],prend ntar klo lu2s n masuk PLN diajak juga duruk juga mau broo... [] : ho kapan chat lagi??meskipun agak

lemot tetep buat download dapet juga kok!! [cahyo]: nung enek lowongan g neng kantor mu,q lagi butuh kerjaan neh..,[yudi]:enek info gawean maneh yud.. [sepvian{kapan ngopi lagi??},suryo,yossi,harry,tangguh{thanks

tanamannya}, erik{thanks sgala bantuannya dalam ngerjakan TA}, andreas{broo ada info software baru g??thnks sgala bantuannya}

iyik{thanks nyuci PCB&bonnya yik},nery, abieh{kapan garap TA bleh??},edi thanks tumpangan mobilnya waktu seminar hasil, dan... (fagg{thanks atas sgala bantuannya n sorry waktu itu kefempatmu minta bantuan malem-2}, andi,neo,shito,okigondes,alis andi,s,alfat,mul alias

kusul,ay,Erwin,innuddin, dan...
wan n sorry klo ada yang lom tak sebut] thks broo atas sgala bantuannya & sorry klo slama kul q ada salah-2 kata n perbuatan...to my secret girl[okta,dina,dia,gum,tata, neme,me4a and always dea]thks udah menemani disela-sela kesibukan kuliah...juga tak lupa untuk op saluler yang udah n...berikan sedikit bandwidth nya untuk cari info di internet "gratis" : XL, 3

IM3, AXIS/NTS

Powered by :



ABSTRAKSI

PERANCANGAN DAN PEMBUATAN SISTEM PENGELOLAAN PERPUSTAKAAN MENGUNAKAN KARTU RFID, MIKROKONTROLLER AT89S52 DAN PC

Dony Kristiawan

03.17.078

Jurusan Teknik Elektro S1 – Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang
inod_dj@yahoo.com

Dosen Pembimbing : I. Ir. F.Yudi Limpraptono, MT.
II. Dr.Cahyo Crysdiان Msc.

Kata Kunci :Perpustakaan,RFID,Mikrokontroller AT89S52,FT232BM PC-Link USB.

ITN Malang sebagai lembaga institusi pendidikan ternyata tidak terlepas dari adanya perpustakaan,hal ini karena perpustakaan sebagai salah satu tempat yang menyediakan akses mudah dalam mencari literatur dari suatu ilmu pengetahuan akan tetapi proses pelayanan dalam perpustakaan ITN Malang saat ini masih dilakukan secara manual sehingga proses ini kurang efektif.Hendaknya pihak lembaga membuat suatu sistem pengelolaan perpustakaan yang terkomputerisasi sehingga memudahkan tugas seorang pustakawan.

Pada sistem pengelolaan perpustakaan yang telah dibuat dalam skripsi ini menggunakan Tag rfid sebagai kartu tanda anggota, kartu ini berisi ID yang nantinya akan dibaca oleh perangkat reader serta menggunakan Databasedekstop, yakni sebuah *Development Enviroment* dari Delphi yang dipakai untuk membuat database. Database yang digunakan berbentuk table dengan type Paradox, sedang tampilan aplikasinya menggunakan Delphi 7.0 Pelayanan perpustakaan terbatas hanya pada proses peminjaman dan pengembalian buku saja.Pada pengujian, tag dapat dibaca dengan jarak maksimal 7 cm dengan jeda waktu pembacaan tag berikutnya 8 detik. Serta memanfaatkan modul *FT232BM PC Link-USB* sebagai proses interface ke PC melalui port USB.

DAFTAR ISI

LEMBAR PERSETUJUAN	i
KATA PENGANTAR	ii
ABSTRAKSI	iv
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Batasan Masalah	3
1.5. Metodologi Penulisan	3
1.6. Sistematika Penulisan	4
BAB II TEORI DASAR	6
2.1. Pendahuluan	6
2.2. RFID (<i>Radio Frequency Identification</i>)	6
2.2.1. Teknologi Dasar RFID	7
2.2.1.1. Sistem RFID Aktif	7
2.2.1.2. Sistem RFID Pasif	8
2.2.2. <i>Tag Reader</i>	10
2.2.3. Mekanisme RFID	11
2.2.3.1. Prinsip Kerja <i>Reader</i> dan	

<i>Tag/Transpoder</i>	11
2.1.3.2. Pembacaan Format RFID	13
2.3. Metode Transmisi Data Digital	15
2.3.1. Transmisi Data Paralel	15
2.3.2. Transmisi Data Serial	15
2.3.2.1. Transmisi Data Serial Asinkron	16
2.3.2.2. Half Duplex	17
2.3.2.3. Full Duplex	18
2.4. USB (Universal Serial Bus)	18
2.4.1. Karakteristik Elektris USB	19
2.4.2. Proses Pengiriman Data	19
2.5. Mikrokontroler AT89S52	21
2.5.1. Perangkat Keras Mikrokontroler AT89S52	21
2.5.2. Arsitektur AT89S52	22
2.5.3. Konfigurasi Pin-pin Mikrokontroler AT89S52	23
2.5.4. Organisasi Memori	26
2.5.5. SFR (Special Function Register)	27
2.5.6. Metode Pengalamatan Memori	29
2.6. Display LCD (Liquid Crystal Display) M1632	30
2.7. Modul FT232BM (PC-Link USBer)	32
2.8. IDE (<i>Integrated Development Enviroment</i>)	35
BAB III PERANCANGAN DAN PEMBUATAN ALAT	38
3.1. Pendahuluan	38
3.2. Perencanaan Perangkat Keras (<i>Hardware</i>)	38

3.2.1. Blok Diagram Sistem	39
3.2.2. Prinsip Kerja	40
3.2.3. Perencanaan Rangkaian Reader ID-12	40
3.2.4. Perancangan Minimum Sistem AT89S52	42
3.2.4.1. Perancangan Rangkaian Reset	42
3.2.4.2. Perancangan Rangkaian clock	44
3.2.4.3. Perencanaan Pengaturan <i>Port</i>	44
3.2.4.3. Perencanaan Komunikasi Serial	45
3.2.5. Rangkaian switch data	47
3.2.6. Perancangan Rangkaian LCD M1632	48
3.3. Perancangan Perangkat Lunak (software)	50
3.3.1. Perencanaan DataBase	50
3.3.2. Flowchart	53
BAB IV PENGUJIAN ALAT	56
4.1. Pendahuluan	56
4.2. Pengujian RFID Reader	56
4.2.1. Tujuan	56
4.2.2. Peralatan yang digunakan	56
4.2.3. Prosedur Pengujian	56
4.2.4. Hasil Pengujian Pembacaan RFID	58
4.3. Pengujian Modul FT232BM PC-Link USBer	60
4.3.1. Tujuan	60
4.3.2. Peralatan Yang dibutuhkan	60
4.3.3. Prosedur Pengujian	60

4.4. Pengujian Tampilan LCD	66
4.4.1. Tujuan	66
4.4.2. Peralatan Yang Digunakan	66
4.4.3. Prosedur Pengujian	66
4.4.4. Hasil Pengujian	69
4.5. Pengujian Sistem Secara Keseluruhan.....	69
4.5.1. Tujuan	69
4.5.2. Peralatan Yang Digunakan	69
4.5.3. Prosedur Pengujian	69
4.5.4. Hasil Pengujian	70
4.5.4. Proses Identifikasi	70
4.6. Spesifikasi Alat	76
BAB V PENUTUP	77
5.1. Kesimpulan	77
5.2. Saran	77
DAFTAR PUSTAKA	78
LAMPIRAN-LAMPIRAN	

4.4. Pengujian Tampilan LCD	66
4.4.1. Tujuan	66
4.4.2. Peralatan Yang Digunakan	66
4.4.3. Prosedur Pengujian	66
4.4.4. Hasil Pengujian	69
4.5. Pengujian Sistem Secara Keseluruhan.....	69
4.5.1. Tujuan	69
4.5.2. Peralatan Yang Digunakan	69
4.5.3. Prosedur Pengujian	69
4.5.4. Hasil Pengujian	70
4.5.4. Proses Identifikasi	70
4.6. Spesifikasi Alat	76
BAB V PENUTUP	78
5.1. Kesimpulan	78
5.2. Saran	78
DAFTAR PUSTAKA	80
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

2-1. Bagian-bagian Tag RFID	10
2-2. Reader ID-12	10
2-3. RFID tag dengan silikon chip dan antenna	12
2-4. Komunikasi antara reader dan transponder	13
2-5. Metode Transmisi Data Digital	15
2-6. Transmisi Data Serial	16
2-7. Transmisi Data Serial Asinkron	17
2-8. Komunikasi Half Duplex	17
2-9. Komunikasi Full Duplex	18
2-10. Soket pin pada USB type A&B	19
2-11. Urutan Encode Data	20
2-12. Blok diagram mikrokontroler AT89S52	22
2-13. Konfigurasi pin-pin AT89S52	23
2-14. Diagram Blok LCD	30
2-15. <i>Pin-pin</i> kaki dari LCD M1632	31
2-16. Modul FT232BM PC-Link USBer	33
2-17. Tata letak komponen	34
2-18. Alokasi pinJ2 dan J3	34
2-19. IDE (<i>Integrated Development Enviroment</i>)	35
3-1. Diagram Blok Sistem	39
3-2. Rangkaian <i>ReaderID-12</i>	40
3-3. Rangkaian <i>Reset</i>	43

3-4. Rangkaian <i>Clock</i>	44
3-5. Perencanaan Pengaturan Port	45
3-7. Alokasi Pin J2	47
3-8 Rangkaian Switch data	48
3-9. Perencanaan Rangkaian Liquid Crystal Display (LCD)	49
3-10 Lembar Kerja DataBase Dekstop	50
3-10. Membuat Table Baru	51
3-11. Menentukan Type Table	51
3-12. Contoh Rancangan Tabel	52
3-13. Diagram Alir Program Di Mikrokontroller.....	53
3-14. Diagram Alir Program di Komputer.....	54
4-1. Kotak Dialog Connection Description	57
4-2. Kotak Dialog Connect To	57
4-3. Kotak Dialog COM 1 Properties	58
4-4. Identifikasi Reader Terhadap kartu	58
4-5. Jarak Pembacaan Reader Terhadap kartu	60
4-6. Rangkaian Pengujian Modul FT232BM	60
4-7. Proses Penjumperan Rx dan Tx	61
4-8. Panel Found New Hardware Wizard	61
4-9. Tampilan proses Bantuan Instalasi driver	62
4-10. Tampilan Browse untuk mencari lokasi driver berada	62
4-11. Tampilan Komfirmasi proses Hardware Installation	63
4-12. Proses Instalasi sedang berjalan	63
4-13. Tampilan indikasi modul terhubung ke PC	64

DAFTAR TABEL

2-1. Fungsi Pin <i>ID-12 Reader</i> (ASCII).....	11
2-2. Format data <i>ASCII</i>	13
2-3. Data Output Serial dalam Hexa.....	14
2-4. Proses Penggabungan data.....	14
2-5. Keterangan <i>Pin</i> atau kaki.....	19
2-6. Fungsi Khusus Port.....	24
2-7. Fungsi Khusus Port 3.....	25
2-8. <i>Special Function Register</i>	27
2-9. Fungsi pin-pin LCD M1632.....	31
4-1. Hasil Pengujian Pembacaan tag <i>RFID</i>	59

BAB I

PENDAHULUAN

1.1.Latar Belakang

Perkembangan teknologi yang canggih saat ini tidak terlepas dari peran ilmu pengetahuan, dengan ilmu pengetahuan kita dapat menciptakan suatu teknologi baru yang canggih dan *inovatif*. Oleh karena itu dengan membaca, ilmu pengetahuan kita akan bertambah.

Seiring dengan tingginya minat baca masyarakat saat ini peran perpustakaan sebagai tempat penyedia buku-buku ilmu pengetahuan menjadi sangat penting keberadaannya. Karena dengan perpustakaan kita mendapatkan akses ilmu pengetahuan yang mudah, murah atau bahkan tidak mengeluarkan biaya sama sekali.

Sebagai suatu contoh yakni Perpustakaan ITN Malang. Selama ini pengelolaan perpustakaan ITN Malang dilakukan secara konvensional atau manual terutama dalam hal pelayanan peminjaman dan pengembalian buku, serta proses registrasi anggota baru yang cukup lama, hal ini ironis sekali mengingat perkembangan zaman saat ini yang semakin canggih. Sudah saatnya sistem konvensional ini dirombak dengan memanfaatkan teknologi baru, sehingga memudahkan tugas seorang pustakawan dalam proses pelayanannya.

Dengan memanfaatkan sistem *radio frequency identification*, serta PC dalam skripsi ini akan diciptakan suatu sistem pengelolaan perpustakaan secara sistematis dan terdistribusi.

Sistem kendali utamanya adalah sebuah tag *RFID* yang didalamnya berisi data dalam format ASCII, serta sebuah reader *RFID* ID-12 yang berfungsi sebagai pembaca datanya, mikrokontroler AT89S52 dan sebuah modul *FT232BM (PC Link USBer)*. Sebuah modul yang memungkinkan kita berkomunikasi serial melalui port *USB* PC. Kita ketahui bersama penggunaan port serial serta parallel port saat ini mulai ditinggalkan, sebagai contoh perkembangan laptop dewasa ini. Serta kalau kita perhatikan sebagian besar peralatan saat ini mulai dikembangkan menggunakan port *USB* sebagai proses *interfacenya*. Sifatnya yang *autoload driver* memudahkan kita dalam mengkoneksikannya dengan PC. Dengan mudah PC secara otomatis akan mengenali peralatan yang dikoneksikan.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka permasalahan yang dapat dirumuskan adalah :

1. Bagaimana merancang sistem pengelolaan perpustakaan memanfaatkan teknologi *rfid*.
2. Bagaimana proses komunikasi serial melalui port *USB* dengan menggunakan sebuah modul *FT232BM PC-Link USBer*.

Sehubungan dengan hal itu, maka pada skripsi ini diberi judul :

“PERANCANGAN DAN PEMBUATAN SISTEM PENGELOLAAN PERPUSTAKAAN MENGGUNAKAN KARTU *RFID*, MIKROKONTROLLER AT89S52 DAN PC”

1.3. Tujuan

Adapun tujuan dari skripsi ini adalah merancang dan membuat sistem pengelolaan perpustakaan dengan menggunakan kartu *rfid*, PC, mikrokontroler

AT89S52 serta penggunaan modul *FT232BM PC-Link USBer* sebagai *interface* komunikasi serial melalui port *USB*.

1.4. Batasan Masalah

Dalam menyusun skripsi ini diperlukan suatu batasan masalah agar tidak menyimpang dari ruang lingkup yang akan dibahas. Adapun batasan masalahnya adalah sebagai berikut :

- a) Menggunakan mikrokontroler AT89S52 untuk pemrosesan data.
- b) PC digunakan sebagai media penyimpanan data identitas anggota dengan kodenya masing-masing.
- c) Koneksi peralatan dengan komputer via usb menggunakan modul *FT232BM (PC Link USBer USBer)*.
- d) Proses pelayanan terbatas hanya dalam peminjaman dan pengembalian buku saja.

1.5. Metodologi Penelitian

Metodologi penelitian yang digunakan dalam pembuatan skripsi ini adalah:

1. Study literatur

Untuk mengumpulkan data literatur berisikan teori-teori yang berhubungan dengan perancangan alat dipadukan dengan teori yang didapat di bangku kuliah.

2. Perancangan perangkat keras dan perangkat lunak

Untuk merancang dan membuat sistem pengelolaan perpustakaan menggunakan kartu *rfid* ini maka diperlukan suatu perangkat lunak dan perangkat keras. Perancangan perangkat keras meliputi minimum sistem mikrokontroler AT89S52, Reader ID-12, tag RFID, modul FT232BM dan

LCD (*Liquid Crystal Display*). Sedangkan untuk perancangan perangkat lunaknya adalah menggunakan bahasa pemrograman delphi sebagai program aplikasi di PC.

3. Pembuatan dan pengukuran/pengujian perangkat keras dan perangkat lunak.

Dari hasil perancangan dilakukan realisasi/pembuatan baik perangkat keras maupun pada perangkat lunak, dan dilakukan pengukuran /pengujian masing-masing bagian (sub-sistem) dari perangkat-perangkat tersebut sebelum dilakukan integrasi.

4. Pengujian Alat dan Analisa.

Sistem yang sudah dibangun, terintegrasi dianggap memadai sehingga dapat dipastikan apakah alat dan perangkat lunak tersebut sesuai yang direncanakan dan dilanjutkan dengan melakukan analisa terhadap keseluruhan dari sistem yang telah dibuat sehingga keakuratan dari sistem yang telah dibuat dapat diketahui..

1.6.Sistematika Penulisan

Penulisan Skripsi ini terbagi menjadi lima Bab dengan sistematika sebagai berikut:

BAB I PENDAHULUAN

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

BAB II TEORI DASAR

Membahas teori – teori dasar penunjang perancangan dan pembuatan alat.

Membahas teori – teori dasar penunjang perancangan dan pembuatan alat.

BAB III PERANCANGAN DAN PEMBUATAN ALAT

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

BAB IV PENGUJIAN ALAT

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

BAB V PENUTUP

Berisi kesimpulan dan saran.

BAB II

TEORI DASAR

2.1. Pendahuluan

Seiring perkembangan, sistem identifikasi banyak dipergunakan dalam industri untuk menyediakan info mengenai manusia, hewan, kendaraan dan lain-lain. Salah satu teknik adalah *RFID (Radio Frequency Identification)* dimana tidak diperlukan hubungan secara fisik antara obyek yang diidentifikasi dengan unit pembacanya, karena *RFID* telah menggunakan teknologi frekuensi radio.

Untuk dapat memahami alat yang akan dirancang maka dalam bab ini akan dijelaskan mengenai teori dasar yang akan berkaitan dengan sistem yang dipergunakan pada perancangan dan pembuatan alat.

2.2. Radio Frequency Identification (RFID)

Radio Frequency Identification (RFID) adalah istilah umum yang digunakan untuk mendeskripsikan sebuah sistem yang mampu mengirimkan identitas (dalam bentuk nomor urut yang unik) sebuah objek secara nirkabel dengan menggunakan gelombang radio. RFID ini termasuk dalam golongan teknologi *Automatic Identification (Auto-ID)*. Teknologi lain yang termasuk dalam Auto-ID adalah barcode, pembaca karakter optis, dan teknologi biometri. Kelompok lain Auto-ID masih memerlukan campur tangan manusia untuk menangkap data, tetapi RFID dapat menangkap data tanpa campur tangan manusia. Secara umum, kartu RFID terdiri dari sebuah mikrochip yang mempunyai sebuah antena. Didalam kartu RFID tersebut dapat disimpan data yang ukurannya sebesar 2 kilobyte. Informasi

ini bisa berisi identitas dari sebuah objek (identifikasi unik sebuah objek) dan informasi tambahan dari objek (tanggal pembuatan barang atau tanggal pengiriman barang pada kasus *supply chain*). Untuk membaca data dari kartu RFID ini diperlukan sebuah piranti pembaca yang akan memancarkan gelombang radio dan menangkap signal yang dipancarkan oleh kartu RFID.

2.2.1. Teknologi Dasar RFID

Terdapat banyak jenis tipe teknologi RFID, maka diperlukan pertimbangan yang tepat untuk memilih tipe yang sesuai dengan aplikasi yang akan dibangun. Hal ini penting agar system berbasis RFID yang dibangun dapat mengurangi biaya dan meningkatkan efisiensi. Pembagian tipe teknologi RFID dapat didasarkan pada jenis frekuensi yang digunakan dan kemampuan untuk mengirim sinyal. Jenis frekuensi yang digunakan dapat dibagi menjadi *low-frequency*, *high-frequency*, dan *ultra-high frequency* (UHF). Sedangkan dari kemampuan untuk mengirim sinyal dapat dibedakan menjadi sistem RFID aktif dan sistem RFID pasif.

2.2.1.1. Sistem RFID Aktif

Pada sistem RFID aktif ini kartu RFID mempunyai sumber daya sendiri dan mempunyai *transmitter*. Sumber daya yang digunakan bisa berasal dari baterai atau tenaga surya. Karena mempunyai sumber daya sendiri, RFID jenis ini mempunyai jangkauan yang lebih luas, yaitu antara 20 meter sampai 100 meter. Kartu ini akan melakukan *broadcast* sinyal untuk mengirimkan data dengan menggunakan *transmitter* yang dimilikinya. RFID jenis ini biasanya beroperasi pada frekuensi 455 MHz, 2.45 GHz, atau 5.8 GHz. Kartu jenis ini digunakan pada

aset bernilai besar (kargo, kontainer atau mobil) karena kartu jenis ini berharga relatif mahal.

2.1.1.2. Sistem RFID Pasif

Pada sistem RFID pasif, kartu tidak mempunyai *transmitter* maupun sumber daya. Kartu jenis ini juga tidak membutuhkan perawatan. *Transponder* RFID terdiri dari microchip yang menempel pada antena. Karena ukurannya yang kecil, transponder bisa saja dibungkus dalam berbagai macam bentuk, seperti di dalam lipatan kertas, di dalam kertas berlabel barcode, atau di dalam kartu plastik. Bentuk pembungkus yang digunakan tergantung pada jenis karakteristik aplikasi yang menggunakan RFID ini. Kartu RFID pasif ini dapat menggunakan *low frequency* (124 kHz, 125 kHz, atau 135 kHz), *high frequency* (13,56MHz), atau UHF (860 MHz-960 MHz). Jenis frekuensi yang digunakan juga sangat bergantung pada karakteristik aplikasi karena tiap rentang frekuensi mempunyai karakteristik tertentu. Pada rentang frekuensi tertentu gelombang radio tidak dapat menembus benda logam atau air, rentang frekuensi juga mempunyai karakteristik jarak maksimum pancaran gelombang radio yang berbeda-beda. Perusahaan pengguna RFID umumnya banyak menggunakan RFID pasif berfrekuensi UHF dibandingkan dengan *low frequency* atau *high frequency*. Hal ini karena kartu RFID pasif yang menggunakan UHF berharga lebih murah dan jangkauannya lebih luas (jangkauannya sampai dengan 3.33 meter). Banyak aplikasi biasanya membutuhkan kartu RFID yang dapat dibaca pada jarak minimal 3 meter dari piranti pembaca. Aplikasi jenis ini misalnya aplikasi pengelolaan barang gudang yang memerlukan kartu dapat dibaca ketika masuk pintu, dan jangkauan kartu tentu saja minimal 3 meter. Sedangkan kartu RFID yang menggunakan *low*

frequency hanya dapat dibaca pada jarak maksimal 0.3 meter dari piranti pembaca, sedang untuk *high frequency* dapat dibaca pada jarak 1 meter.

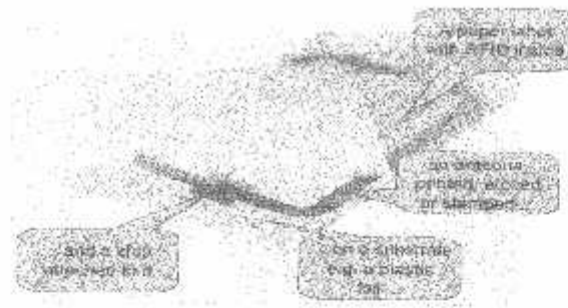
Metode pengiriman data kartu RFID pasif ke piranti pembaca dapat dibagi menjadi 2 macam, yaitu:

1. Inductive Coupling

Gulungan tembaga pada piranti pembaca membangkitkan medan elektromagnetik, kemudian gulungan yang ada di kartu RFID terinduksi oleh medan ini, hasil induksi inilah yang menjadi sumber tenaga bagi kartu RFID untuk mengirimkan kembali sinyal yang berisi data ke piranti pembaca. Karena menggunakan prinsip induksi ini, maka jarak antara kartu RFID dengan piranti pembaca juga harus pendek agar induksi dapat ditangkap. *Inductive coupling* ini digunakan pada kartu RFID dengan *low frequency* dan *high frequency*.

2. Propagation Coupling

Pada sistem ini, energi yang digunakan berasal dari energi elektromagnetik (gelombang radio) yang dipancarkan oleh piranti pembaca. Kartu RFID kemudian akan mengumpulkan energi elektromagnetik ini untuk digunakan sebagai sumber daya mengirimkan data yang dimilikinya ke piranti pembaca. Mekanisme ini disebut dengan *backscatter*. Modulasi bit data ke frekuensi bisa menggunakan *amplitude shift keying*, *phase shift keying*, atau *frequency shift keying*.



Gambar 2-1. Bagian - bagian *Tag RFID*^[1]

2.2.2. Tag Reader

Tag reader berfungsi untuk membaca data yang ada pada *tag* melewati *RF interface*. Untuk menambah fungsi *reader* dilengkapi dengan *internal storage*, dan aplikasi perangkat lunak untuk menyimpan data pada *server database*. Pada prakteknya *tag reader* dapat berupa perangkat keras yang terletak pada suatu tempat yang tetap. Pada aplikasinya *tag reader* dapat membaca sendiri *tag* yang dideteksi (*smart self*). *Tag reader smart self* dapat mendeteksi ketika ada penambahan *tag* atau ada *tag* yang keluar. Pada dasarnya *tag reader* merupakan suatu peralatan yang sederhana dan dapat digabungkan kedalam perlengkapan *mobile* seperti telepon selular atau PDAs.

Saluran (*channel*) dari *reader* ke *tag* disebut dengan saluran *forward* (*forward channel*), saluran dari *tag* ke *reader* disebut dengan saluran *backward* (*backward channel*).



Gambar 2-2. *Reader ID-12*^[2]

Karakteristik ID-12 :

- Read Range : 12+ cm.
- Dimensions : 26 mm x 25 mm x 7 mm.
- Frequency : 125 kHz.
- Card Format : EM 4001 atau kompetibel.
- Encoding : Manchester 64-bit, modulus 64.
- Power Requirement : 5 VDC @ 30mA nominal.
- Voltage Supply Range : 4.6V through 5.4V.

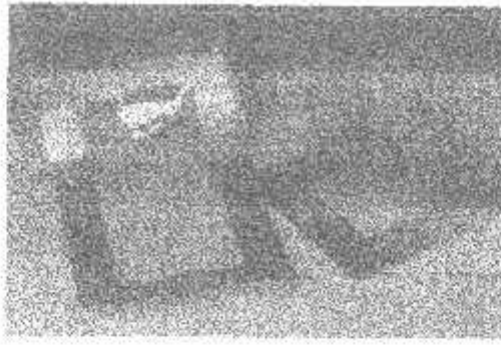
Tabel 2-1. Fungsi Pin *ID-12 Reader* (ASCII)^[21]

Pin No.	Description	ASCII
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V
Pin 2	Strap to +5V	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna
Pin 4	To External Antenna	Antenna
Pin 5	Card Present	No function
Pin 6	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND
Pin 8	Data 1	CMOS
Pin 9	Data 0	TTL Data (inverted)
Pin 10	3.1 kHz Logic	Beeper / LED
Pin 11	DC Voltage Supply	+5V

2.2.3.Mekanisme RFID

2.2.3.1.Prinsip Kerja Reader dan tag Transponder

Suatu transponder secara induktif yang tergabungkan terdiri atas suatu data elektronik didalam suatu microchip yang pada umumnya tunggal dan suatu coil area besar yang berfungsi sebagai suatu antenna.

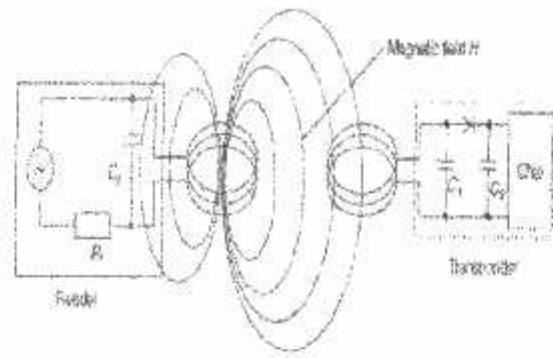


Gambar 2-3. *RFID tag dengan silicon chip dan antenna*^[1]

Secara induktif transponder dioperasikan dengan pasif yaitu bahwa semua energi yang diperlukan untuk operasi *microchip* harus disajikan oleh pembaca reader. *Coil* antenna pembaca menghasilkan suatu bidang elektromagnet frekuensi tinggi yang menembus penampang-lintang area coil dan area disekitar coil itu, karena panjang gelombang cakupan frekuensi menggunakan *low* frekuensi (125 kHz-135Khz). Suatu bidang elektromagnet dipancarkan menembus coil antenna transponder yang mana saat terinduksi suatu tegangan dihasilkan coil antenna transponder. Tegangan ini berfungsi sebagai power untuk pengaktifan data didalam *microchip*.

Suatu kapasitor yang dihubungkan parallel dengan *coil* antenna pembaca berkombinasi dengan *induktans coil* antenna untuk membentuk suatu rangkaian resonansi parallel, frekuensi resonan yang bersesuaian dengan frekuensi transmisi pembaca yang dihasilkan didalam coil antenna pembaca akan meningkatkan rangkaian resonan parallel tersebut, sehingga dapat digunakan untuk menghasilkan kekuatan bidang elektromagnet.

Coil antenna transponder dan kapasitor membentuk suatu rangkaian resonan dan mengatur kesesuaian frekuensi transmisi pembaca. Tegangan di transponder coil akan meningkatkan rangkaian frekuensi resonan paralel pada tag.



Gambar 2-4. Komunikasi antara *reader* dan *transponder*^[11]

2.2.3.2. Pembacaan Format RFID

RFID reader mempunyai banyak sekali type, antara lain : ID-10, ID-12, ID-20, dan sebagainya. RFID reader ini memiliki dua bentuk output serial yakni : ASCII dan Wiegand26. Paling umum banyak digunakan adalah output dengan format ASCII, karena output ini sangat mudah untuk dihubungkan pada mikrokontroler atau PC.

❖ Format Data ASCII

Output yang memiliki format-Serial ASCII 9600, N,8,1 memiliki struktur sebagai berikut :

Tabel 2-2. Format data *ASCII*^[8]

02 (1byte)	10 Data karakter ASCII (10byte)	Checksum (2byte)	CR (1byte)	LF (1byte)	03 (1byte)
---------------	------------------------------------	---------------------	---------------	---------------	---------------

Total paket yakni 16 bytes.

Sebagai suatu contoh : menggunakan *Hyperterminal* 9600 baud, N, 8, 1(Flow control=none)

●041A21EE34E5

♥

Keterangan : E5 = checksum

● = 02 ASCII

♥ = 03 ASCII

Untuk lebih jelasnya tentang cara pembacaan format ASCII, lihat contoh berikut. Misalnya data output serial (dalam hexa) yang kita tangkap adalah sebagai berikut :

Tabel 2-3. Data Output Serial dalam Hexa^[8]

02	30	34	36	32	30	31	44	37	36	43	44	43	0D	0A	03
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Langkah pertama adalah merubah semua nilai diatas menjadi karakter ASCII. misal 30H menjadi karakter "0", 34H menjadi "4", dst. Langkah kedua adalah menyusun data-data tersebut kedalam format data ASCII seperti Tabel 2-2. Kemudian ambil 10 data karakter ASCII. Dalam contoh ini berarti data tersebut adalah :

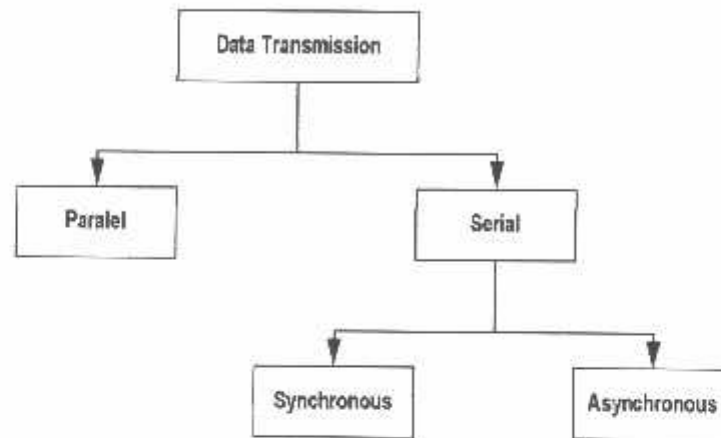
Tabel 2-4. Proses Penggabungan data^[8]

30	34	36	32	30	31	44	37	36	43	Data Hexa
		6	2	0	1	D	7	6	C	Data ASCII

Untuk data dengan warna hijau merupakan data untuk jenis kartu dan tidak digunakan dalam proses konversi, yang akan digunakan disini adalah data yang ke-3 s/d ke-10. Hasil konversi dari data heksa ke dalam data ASCII adalah "6201D76C" setelah itu gabungkan data ASCII menjadi bilangan hexadesimal, kemudian konversikan bilangan hexadesimal tersebut ke dalam desimal. Hasilnya sebagai berikut 6201D76C H menjadi 1644287852 (ini merupakan nomor kartu sebenarnya yang tertera pada badan kartu tersebut). Cara ini berlaku untuk kartu yang tidak dienkripsi.

2.3. Metode Transmisi Data Digital

Pengiriman data biner melalui suatu saluran dapat dilakukan dengan dua mode yaitu mode pentransmisiian paralel dan mode pentransmisiian serial.



Gambar 2-5. *Metode Transmisi Data Digital*⁵¹

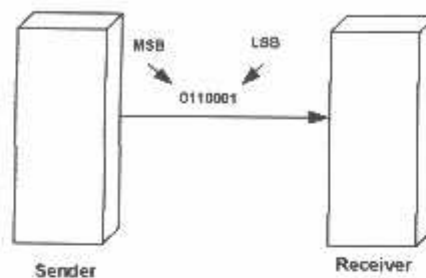
2.3.1. Transmisi Data Paralel

Transmisi data paralel adalah transmisi sejumlah n bit data yang dapat dikirimkan dan diterima dalam waktu yang bersamaan dengan menggunakan n saluran. Jadi setiap bit data mempunyai 1 saluran. Keuntungan dari transmisi paralel yaitu pengiriman data menjadi lebih cepat, sedangkan kerugiannya yaitu biaya yang lebih banyak karena membutuhkan jumlah saluran yang lebih banyak. Karena permasalahan biaya menyebabkan metode ini hanya digunakan untuk komunikasi jarak dekat.

2.3.2. Transmisi Data Serial

Transmisi data serial adalah transmisi data yang pengiriman dan penerimaan datanya berurutan tiap bitnya. Jadi kita hanya membutuhkan satu saluran untuk mengirimkan data antar dua perangkat komunikasi. Keuntungan transmisi serial yaitu biaya lebih murah karena hanya membutuhkan satu saluran

saja sehingga banyak digunakan untuk komunikasi jarak jauh. Sedangkan kerugiannya adalah kecepatan pengiriman data lebih rendah dibandingkan dengan transmisi paralel.

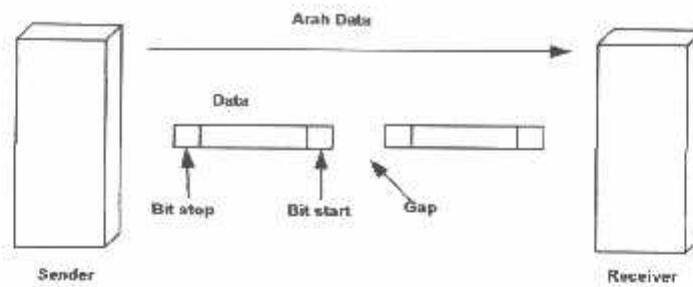


Gambar 2-6. *Transmisi Data Serial*^[5]

Pengiriman data akan dimulai dari LSB (*Least Significant Bit*) dan diakhiri dengan MSB (*Most Significant Bit*). Setiap karakter yang dikirimkan, disusun sesuai dengan suatu urutan dengan bit tertentu. Berdasarkan formatnya, salah satu jenis komunikasi serial yaitu :

2.3.2.1. Transmisi Data Serial Asinkron

Pada pengiriman data asinkron (tidak sinkron), setiap karakter dikirimkan sebagai satu kesatuan (*entity*) bebas yang berarti bahwa waktu antara pengiriman bit terakhir dari sebuah karakter dan bit pertama dari sebuah karakter berikutnya tidak tetap. Pengiriman data asinkron lebih sederhana daripada pengiriman sinkron karena hanya isyarat data saja yang dikirimkan. Detak penerima dibangkitkan secara lokal di dalam penerima dan tetap dijaga agar sesuai dengan detak pengirim yang menggunakan bit awal (*start bit*) dan bit akhir (*stop bit*) yang dikirimkan setiap karakter dan data yang satu dengan data selanjutnya dipisahkan oleh gap. Penyesuaian detak pengiriman dan penerima terjadi karakter per karakter.



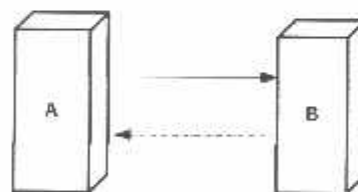
Gambar 2-7. *Transmisi Data Serial Asinkron*^[51]

Gambar di atas menunjukkan sinkronisasi awal-akhir, bit awal dan akhir tidak membawa informasi, tetapi hanya menunjukkan awal dan akhir setiap karakter. Dari gambar dapat dilihat bahwa bit kedelapan disebut bit paritas, diikutsertakan dalam bentuk gelombang tersebut. Bit ini akan dipasang pada 1 atau 0 untuk meyakinkan cacah bit pada setiap karakter adalah genap untuk paritas genap, atau ganjil untuk paritas ganjil.

Berdasarkan arah komunikasinya, pengiriman data serial dibedakan menjadi :

2.3.2.2. Half Duplex

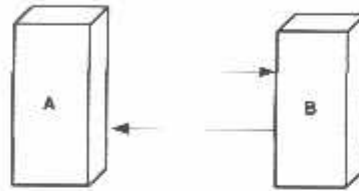
Merupakan sistem komunikasi yang mengirimkan data dalam satu arah. Sistem ini tidak dapat mengirimkan data secara bersamaan sehingga perlu saling menunggu secara bergantian untuk berkomunikasi. Pada gambar di bawah menunjukkan komunikasi half duplex.



Gambar 2-8. *Komunikasi Half Duplex*^[51]

2.3.2.3. Full Duplex

Merupakan sistem komunikasi yang mengirimkan data dalam dua arah. Sistem ini dapat mengirimkan data secara bersamaan sehingga tidak perlu saling menunggu secara bergantian untuk komunikasi. Pada gambar di bawah menunjukkan komunikasi full duplex.

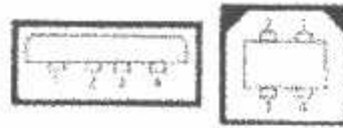


Gambar 2-9. Komunikasi *Full Duplex*^[5]

2.4. USB (*Universal Serial Bus*)

Universal Serial Bus (USB) adalah standar bus berseri untuk perangkat penghubung, biasanya kepada komputer namun juga digunakan di peralatan lainnya seperti konsol permainan dan PDA. Sistem USB mempunyai desain yang asimetris, yang terdiri dari pengontrol host dan beberapa peralatan terhubung yang berbentuk pohon dengan menggunakan peralatan *hub* yang khusus. Desain USB ditujukan untuk menghilangkan perlunya penambahan expansion card ke ISA komputer atau bus PCI, dan memperbaiki kemampuan plug-and-play (pasang-dan-mainkan) dengan memperbolehkan peralatan-peralatan ditukar atau ditambah ke sistem tanpa perlu mereboot komputer. Ketika USB dipasang, ia langsung dikenal sistem komputer dan memproses *device driver* yang diperlukan untuk menjalankannya. Kecepatan transfer data USB dibagi menjadi tiga, antara lain:

- High speed data dengan frekuensi clock 480 Mb/s.
- Full speed data dengan frekuensi clock 12 Mb/s.
- Low speed data dengan frekuensi clock 1.5 Mb/s.



Gambar 2-10. Soket pin pada USB type A&B^[6]

Tabel 2-5. Keterangan *Pin atau kaki*^[6]

1	Vbus (4.75 – 5.25V)
2	D-
3	D+
4	GND (Ground)

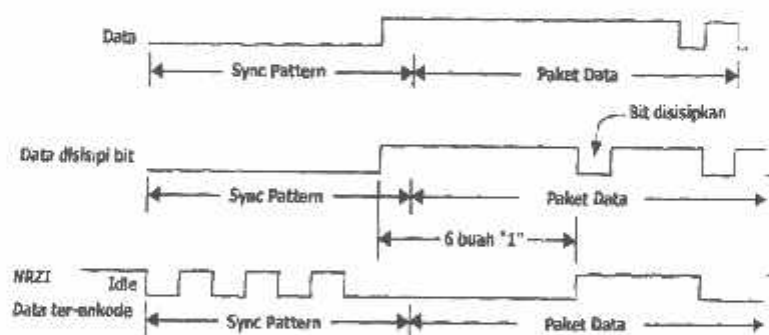
2.4.1. Karakteristik Elektris USB

Rentang tegangan kerja sinyal USB adalah 0.3 Volt hingga 3.6 Volt (pada beban 1.5 k Ω). Logika tinggi didapat jika tegangan sudah melebihi 2.8 Volt terhadap ground pada beban 15 k Ω . Pada piranti USB yang berkecepatan rendah dan penuh, diferensial “1” dikirim dengan menarik D+ hingga lebih besar dari 2.8 volt dengan sebuah resistor 15 k Ω terhubung ke ground dan sekaligus menarik D- hingga di bawah 0.3 volt dengan sebuah resistor 1.5 k Ω terhubung ke 3.6 volt. Hal yang sama diferensial “0” adalah D- lebih besar dari 2.8 volt dan D+ lebih rendah dari 0.3 volt dengan resistor pull-up dan pull down yang sama. Pada USB berkecepatan tinggi (480 Mbit/s) digunakan sumber arus tetap 17.78 mA untuk mengurangi noise.

2.4.2. Proses Pengiriman Data

Untuk mengirimkan paket data, USB menerapkan encode data NRZI (Non Return to Zero Invert). Dalam NRZI ini, logika “1” berarti tidak ada perubahan level tegangan dan logika “0” ditunjukkan dengan adanya perubahan level

tegangan. Beberapa literatur menyebutnya dengan NRZS (S=Space). Paket data dikirimkan ke bus USB berurutan dari bit yang berbobot paling rendah (LSB, Least Significant Bit), diikuti LSB berikutnya, dan terakhir adalah bit yang berbobot paling tinggi (MSB, Most Significant Bit). Penyisipan bit digunakan untuk memastikan kesempurnaan transisi sinyal. Sebuah bit logika “0” disisipkan setiap ditemui logika 1 sebanyak 6 kali berurutan sebelum di-encode ke NRZI. Tujuannya agar penerima mendapat transisi data (logika) setidaknya sekali setiap 7 bit : untuk kepastian *clock control* dan data. Penyisipan bit dimulai dengan *Sync Pattern* dan seterusnya hingga akhir transmisi. Data “1” di akhir *Sync Pattern* dianggap sebagai sebuah awal logika “1”. Penyisipan bit ini tetap dilakukan walau itu diakhir sinyal paket data. Penerima kemudian harus mendekode data NRZI ini, mengenali sisipan bit (jika ada) dan membuangnya. Jika logika “1” ditemukan sebanyak 7 kali berurutan maka dinyatakan ada *error* penyisipan bit dan data yang diterima akan diabaikan.



Gambar 2-11. Urutan encode Data^[6]

2.5. Mikrokontroler AT89S52

Perbedaan mendasar antara mikrokontroler dan mikroprosesor adalah mikrokontroler selain memiliki CPU juga dilengkapi dengan memori *I/O* yang merupakan kelengkapan sebagai suatu *minimum system* mikrokomputer sehingga mikrokontroler dapat dikatakan sebagai mikrokomputer dalam keping tunggal (*single chip microcomputer*) yang dapat berdiri sendiri.

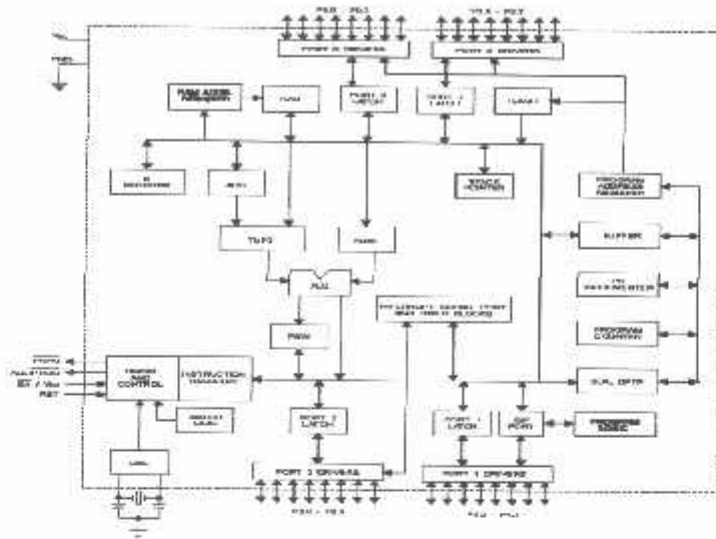
Mikrokontroler AT89S52 adalah mikrokontroler ATMEL *kompatibel* penuh dengan mikrokontroler keluarga MCS-51, dengan *supply* daya rendah, memiliki *performance* yang tinggi, dan merupakan mikrokontroler 8 bit yang dilengkapi 8 Kbyte EPROM (*Enable and programmable read Only Memory*) dan 256 Byte RAM *internal*. Program memori dapat diprogram ulang dalam sistem.

Dalam sistem mikrokontroler terdapat dua hal yang mendasar, yaitu: perangkat keras dan perangkat lunak yang keduanya saling terkait dan mendukung.

2.5.1. Perangkat keras Mikrokontroler AT89S52

Mikrokontroler AT89S52 secara umum memiliki struktur :

- CPU 8 bit
 - *Memory*
 - *Port I/O*
 - *Timer dan Counter*
 - Sumber *Interrupt*
 - *Program Serial* yang dapat diprogram
 - Osilator dan *Clock*
-



Gambar 2-12. Blok diagram Mikrokontroler AT89S52^[3]

2.5.2. Arsitektur AT89S52

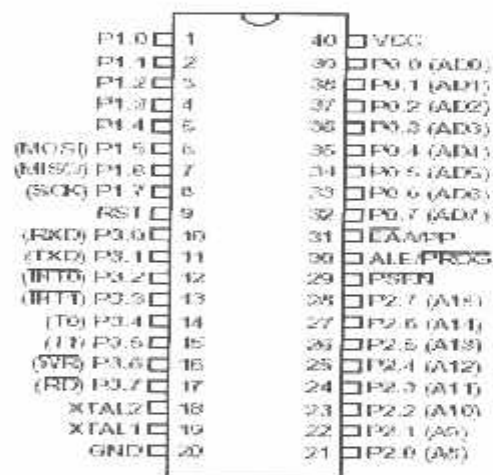
Arsitektur mikrokontroler AT89S52 adalah sebagai berikut :

1. CPU (*Central Processing Unit*) 8-bit dengan *register A* (*accumulator*) dan B.
2. 16-Bit *Program Counter (PC)* dan *Data Pointer (DPTR)*.
3. 8 Bit *Program Status Word (PSW)*.
4. 4 Bit *Stack Pointer (SP)*.
5. 8 Kbyte *internal EPROM*.
6. 256 byte *internal RAM*.
 - 4 *bank register*, masing-masing berisi 8 byte
 - 16 byte alamat serbaguna yang dapat diakses sebagai *byte* atau bit, tergantung *software* yang digunakan.
 - 80 byte *general purpose memory data*.
7. 32 pin *input-output* tersusun atas P0-P3, masing-masing 8-bit.

8. 2 buah 16-bit *Timer/Counter*,
9. 2 buah *port serial full duplex*
10. *Control Register*, antara lain : TCON, SCON, PCON, IP, dan IE
11. 5 buah sumber interupsi (2 buah sumber interup eksternal dan 3 buah sumber *internal*).
12. Osilator dan *Clock Internal*.
13. *Watch Dog Programmable Timer*.
14. *ISP Port* ,

2.4.3. Konfigurasi Pin-pin Mikrokontroler AT89S52

Konfigurasi kaki-kaki Mikrokontroler AT89S52 terdiri dari 40 pena (pin), seperti pada gambar dibawah ini :



Gambar 2-13. Konfigurasi pin-pin AT89S52^[3]

Fungsi dari tiap-tiap pena adalah sebagai berikut :

1. VCC, *Pin 40*

Merupakan pin positif sumber tegangan 5 volt DC.

2. GND (ground), *Pin 20*

Merupakan pin grounding sumber tegangan.

3. Port 0, *Pin 32 – 39*

Merupakan port input dua arah dan dikonfigurasi sebagai *multiplex* dua bus alamat rendah (A0-A7) dan data selama pengaksesan program memori dan data internal.

4. Port 1, *Pin 1 – 8*

Merupakan port input dua arah dengan pull-up dan juga menerima *Low-order address byte* selama memprogram dan verifikasi dari flash. Pada mikrokontroler AT89S52 port 1 memiliki 3 pin dengan fungsi khusus.

Tabel 2-6. Fungsi Khusus Port^[3]

Port Pin	Alternative Functions
P1.5	MOSI (<i>used for In-system Programming</i>)
P1.6	MOSI (<i>used for In-system Programming</i>)
P1.7	SCK (<i>used for In-Programming</i>)

5. Port 2, *Pin 21 - 28*

Merupakan port I/O dengan *internal pull-up*. Mengeluarkan *address* tinggi selama pengambilan (*fetching*) program memori external. Selama pengaksesan ke external data memori, port 2 mengeluarkan isi SFR (*Special Function Register*). Menerima *address* dan beberapa sinyal control selama pemrograman.

6. Port 3, *Pin 10 - 17*

Merupakan port I/O dengan *internal pull-up*. Port 3 juga memiliki fungsi khusus, yaitu:

Tabel 2-7. Fungsi Khusus Port 3^[3]

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

7. Reset, *Pin 9*

Perubahan taraf tegangan dari rendah ke tinggi akan mereset AT89S52.

8. ALE/PROG, *Pin 30*

Pulsa output ALE digunakan untuk proses-proses 'latching' *byte address* rendah (A0-A7) selama pengaksesan ke eksternal memori. Pin ini juga digunakan untuk memasukkan pulsa program (prog) selama pemrograman.

9. PSEN, *Pin 29*

Merupakan strobe baca ke program memori eksternal.

10. EA/VPP, *Pin 31*

External Address Enable (EA) digroundkan jika mengakses memori eksternal, akan dihubungkan ke VCC jika digunakan untuk mengakses memori internal.

11. X-TALI: 1 dan X-TALL 2, *Pin 19 dan Pin 18*

Kaki ini dihubungkan dengan kristal bila menggunakan osilator internal. XTAL1.1 merupakan input inverting osilator amplifier sedangkan XTAL1.2 merupakan output inverting osilator amplifier.

2.5.4. Organisasi Memori

Organisasi memori pada mikrokontroler AT89S52 dapat dibagi menjadi dua bagian besar yaitu memori program dan memori data. Pembagian tersebut didasarkan atas fungsi dari penyimpanan data maupun program. Memori program digunakan untuk menyimpan instruksi-instruksi yang akan diajarkan oleh mikrokontroler, sedangkan memori data digunakan sebagai tempat menyimpan instruksi-instruksi yang sedang diolah mikrokontroler. Program mikrokontroler disimpan dalam memori program berupa ROM. Mikrokontroler AT89S52 dilengkapi dengan ROM internal, sehingga untuk menyimpan program tidak digunakan ROM eksternal yang terpisah dari mikrokontroler. Agar tidak menggunakan memori program eksternal, EA (*External Address enable*) dihubungkan dengan Vcc. Memori program mikrokontroler menggunakan alamat 16 bit mulai 0000_H-0FFF_H, sehingga kapasitas penyimpanan program maksimal adalah 8Kbyte. Sinyal PSEN (*Program Strobe Enable*) tidak digunakan jika menggunakan memori internal.

Selain program mikrokontroler AT89S52 juga memiliki data internal 256 *byte* dan mampu mengakses memori data eksternal sebesar 64Kb. Semua memori data internal dapat dialamati dengan data langsung atau tidak langsung. Untuk membaca data digunakan sinyal RD sedangkan untuk menulis data digunakan sinyal WR.

2.4.5. SFR (*Special Function Register*)

Register Fungsi Khusus (Special Function Register) terletak pada 256 byte bagian atas memori data internal dan berisi *register-register* untuk pelayanan *latch port, timer, program status words, control peripheral* dan sebagainya.

Alamat register fungsi khusus ditunjukkan pada tabel dibawah ini:

Tabel 2-8. *Special Function Register*^[3]

Simbol	Nama Register	Alamat
ACC	<i>Accumulator</i>	E0 _H
B	<i>Register B</i>	F0 _H
PSW	<i>Program Status Word</i>	D0 _H
SP	<i>Stack Pointer</i>	81 _H
DPTR	<i>Data Pointer 2 Byte</i>	
DPL	Bit Rendah	82 _H
DPH	Bit Tinggi	83 _H
P0	<i>Port 0</i>	80 _H
P1	<i>Port 1</i>	90 _H
P2	<i>Port 2</i>	A0 _H
P3	<i>Port 3</i>	B0 _H
IP	<i>Interupt Priority Control</i>	D8 _H
IE	<i>Interupt Enable Control</i>	A8 _H
TMOD	<i>Timer/Counter Mode Control</i>	89 _H
TCON	<i>Timer/Counter Control</i>	88 _H
TH0	<i>Timer/Counter 0 High byte</i>	8C _H
TL0	<i>Timer/Counter 0 Low byte</i>	8A _H
TH1	<i>Timer/Counter 1 High byte</i>	8D _H
TL1	<i>Timer/Counter 1 Low</i>	8B _H
SCON	<i>Serial Control</i>	98 _H
SBUF	<i>Serial Data Buffer</i>	99 _H
PCON	<i>Power Control</i>	87 _H

Beberapa macam *register* fungsi khusus yang sering digunakan adalah sebagai berikut :

- *Accumulator* (ACC) merupakan *register* untuk penambahan dan pengurangan. Perintah *mnemonic* untuk mengakses akumulator disederhanakan sebagai A.
 - *Register B* merupakan *register* khusus yang berfungsi melayani operasi perkalian dan pembagian.
 - *Register R* merupakan delapan set *register* yang dinamakan R0, R1, R2, R3, R4, R5, R6 dan R7, fungsi dari *register-register* ini adalah sebagai *register* yang membantu penyimpanan data yang menggunakan banyak operasi. *Register-register* ini yang membantu akumulator dalam melakukan operasi antara dua operan.
 - *Stack Pointer* (SP) merupakan *register* 8 bit yang dapat diletakkan di alamat manapun pada RAM *internal*.
 - *Data Pointer* (DPTR) terdiri dari dua *register*, yaitu *register* untuk byte tinggi (*Data Pointer High*, DPH) dan *register* untuk byte rendah (*Data Pointer Low*, DPL) yang berfungsi untuk mengunci alamat 16 bit. DPTR berfungsi untuk menunjuk suatu lokasi data, namun pada beberapa perintah DPTR digunakan untuk mengakses memori eksternal.
 - *PC (Program Counter)* merupakan alamat 16 bit yang menginstruksikan AT89S52 alamat instruksi yang selanjutnya akan dilaksanakan. Saat inisialisasi AT89S52, PC terisi dengan 00000h dan akan bertambah satu setiap kali instruksi telah dilaksanakan. Harga PC tidak dapat langsung dirubah
-

dengan menggunakan perintah MOV PC,2340h, namun dengan perintah LJMP 2340 yang akan mengisi PC dengan 2340h..

- Port 0 sampai Port 3 merupakan *register* yang berfungsi untuk membaca dan mengeluarkan data pada port 0,1,2 dan 3. Masing-masing *register* ini dapat dialamati per byte maupun per bit.
- *Control Register* terdiri dari register yang mempunyai fungsi kontrol. Untuk mengontrol sistem interupsi, terdapat dua *register* khusus yaitu *register* IP (*Interrupt Priority*) dan *register* IE (*Interrupt Enable*). Untuk mengontrol *timer/counter* terdapat dua *register* khusus yaitu register TCON (*Timer Counter Control*) serta *port serial* menggunakan *register* SCON (*Serial Port Control*).

2.5.6. Metode Pengalamatan

1). Pengalamatan Langsung

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu *register* secara langsung. Untuk melaksanakan pengalamatan langsung digunakan tanda #.

Contoh : MOV A, #0A

2). Pengalamatan Tak Langsung

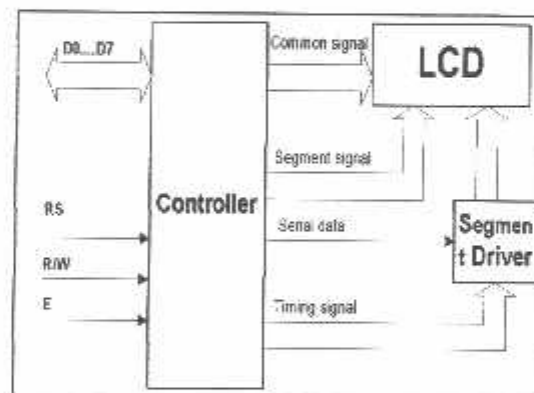
Operand pengalamatan tak langsung menunjuk ke sebuah *register* yang berisi lokasi alamat memori yang akan digunakan dalam operasi. Lokasi yang nyata tergantung pada isi *register* saat instruksi dijalankan. Untuk melaksanakan pengalamatan tak langsung digunakan simbol @.

Contoh : ADD A, @RO

2.6. Display LCD

Modul display yang digunakan dalam aplikasi ini adalah LCD modul M1632. Modul LCD ini membutuhkan daya yang kecil dan dilengkapi dengan panel LCD dengan tingkat kontras yang cukup tinggi serta pengendali LCD CMOS yang terpasang dalam modul tersebut. Pengendali mempunyai pembangkit karakter ROM/RAM dan display data RAM. Semua fungsi display diatur oleh instruksi-instruksi, sehingga modul LCD ini dapat dengan mudah dihubungkan dengan unit MCU. LCD tipe ini tersusun sebanyak dua baris dengan 16 karakter.

Masukan yang diperlukan untuk mengendalikan modul berupa bus data yang masih termultiplek dengan bus alamat serta 3 bit sinyal kontrol. Sementara pengendalian LCD dilakukan secara internal oleh kontroler yang sudah terpasang dalam modul LCD. Diagram blok untuk LCD dapat dilihat dalam Gambar berikut:



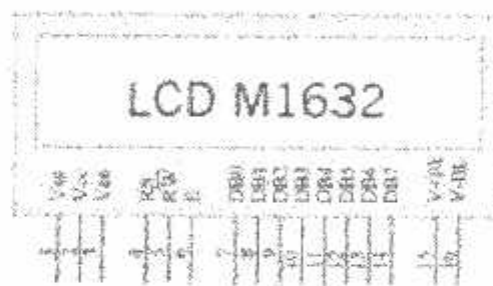
Gambar 2-14. Diagram blok LCD M1632^[7]

LCD modul M1632 mempunyai spesifikasi perangkat keras sebagai berikut:

1. LCD ini terdiri dari 32 karakter dengan 2 baris masing-masing 16 karakter.
2. Karakter generator ROM dengan 192 tipe karakter.
3. Karakter generator RAM dengan 8 tipe karakter.

4. 80 x 8 bit display data RAM.
5. Dapat diinterfocekan ke MPU 8 atau 4.
6. Dilengkapi fungsi tambahan : *display clear, cursor home, display On/Off, display character blink cursor shift, display shift*.
7. Internal data.
8. Internal otomatis, *reset* pada On.
9. 5 Volt PSU tunggal.

Berikut menunjukkan gambaran pin-pin kaki dari LCD M1632 :



Gambar 2-15.Pin-Pin kaki dari LCD M1632^[7]

LCD M1632 mempunyai 16 pin atau penyemat yang mempunyai fungsi-fungsi seperti ditunjukkan dalam Tabel.

Tabel 2-9. Fungsi pin-pin LCD M1632^[7]

No	Nama Penyemat	Fungsi
1	Vss	Terminal ground
2	Vcc	Tegangan catu +5 volt
3	Vee	Drive LCD
4	RS	Sinyal pemilih register 0: Instruksi register (tulis) 1: Data Register (tulis dan baca)
5	R/W	Sinyal seleksi tulis atau baca

		0: Tulis 1: Baca
6	E	Sinyal operasi awal, sinyal ini mengaktifkan data tulis dan baca
7 – 14	DB0-DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan
15	V+ BL	Pengendali kecerahan latar belakang LCD 4 - 4,42 V dan 50 – 500 mA
16	V-BL	Pengendali kecerahan latar belakang LCD 0 V

2.7. Modul FT232BM (PC-Link USBer)

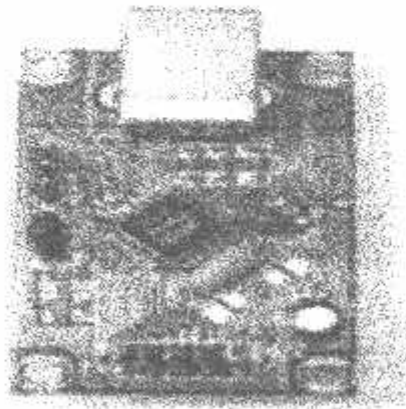
PC-Lik USBer ini merupakan modul *development tools* untuk USB device tipe FT232BM. Modul ini biasanya digunakan untuk aplikasi seperti USB <> UART TTL, USB <> UART RS-232, USB <> UART RS-422, USB <> UART RS-485, antarmuka mikrokontroler dengan USB, antarmuka modem UART dengan USB, dll Modul ini sangat kompetibel dengan banyak perangkat dan aplikasi yang ingin digunakan dan salah satunya diaplikasikan pada sistem pengelolaan perpustakaan menggunakan kartu *rfid*.

Adapun spesifikasi dari modul ini adalah sebagai berikut:

- 1) Tegangan kerja 4,4 – 5,25 Volt DC.
- 2) Tersedia 2 LED untuk indikator Tx dan Rx data pada komunikasi serial.
- 3) Sinyal antarmuka yang mendukung komunikasi dengan modem UART.
- 4) Mendukung format UART dengan 7 atau 8 bit data, 1 atau 2 Stop bit, dan Odd, Even, Mark, Space, No parity.

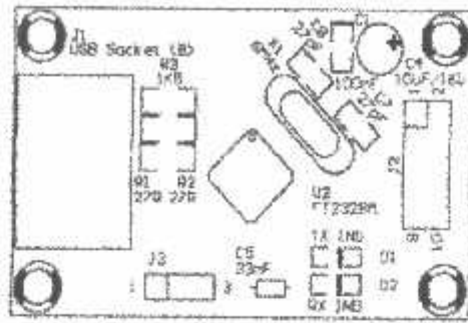
- 5) Memiliki baudrate 3Mbps (TTL), 1Mbps (RS-232), 3Mbps (RS- 422 / RS-485).
- 6) Kompatibel dengan USB 1.1 dan USB 2.0.
- 7) Kompatibel dengan UHCI / OHCI / EHCI host controller.
- 8) Mendukung kondisi USB Suspend / Resume lewat pin SLEEP dan RI.
- 9) Memiliki output dengan level TTL 5 Volt.
- 10) Memiliki EEPROM eksternal untuk menyimpan data PID, VID, nomor serial, dan deskripsi produk. Pengisian datanya melalui USB.
- 11) Virtual COM port driver (VCP) dan D2XX (USB Direct Drivers + DLL S/W Interface) untuk Windows® 98, 98SE, ME, 2000, dan XP.

Modul ini menggunakan *self powered* yang artinya untuk mengaktifkan modul ini, dibutuhkan catu daya sendiri. Adapun gambar fisik dari modul FT232BM PC-Link USBer ini dapat digambarkan sebagai berikut:



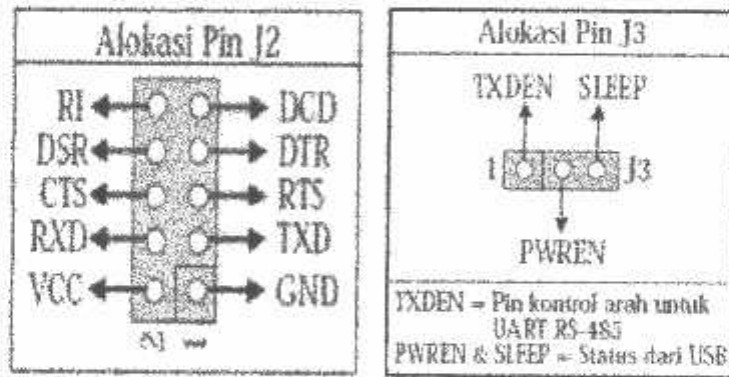
Gambar 2-16. Modul FT232 BM PC-Link USBer^[3]

Berikut gambaran dari tata letak komponen modul ini.



Gambar 2-17. Tata letak komponen^[4]

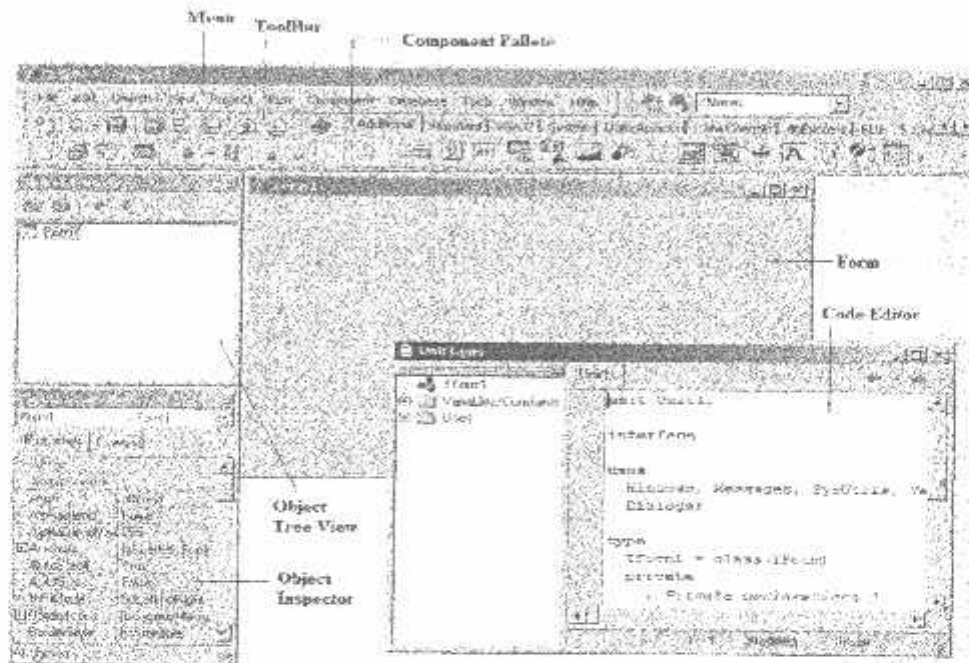
Antarmuka dari/ke modul USB menggunakan level TTL dan terhubung ke header dengan label J2. Dengan modul ini, kita juga bisa gunakan sebagai konverter dari *usb* ke RS232 dengan mengontrol pin TXDEN pada *header* dengan label J3. Akan tetapi dalam skripsi ini yang digunakan adalah level TTL karena sesuai dengan data keluaran pada mikrokontroller sehingga kita tidak perlu mengaktifkan *header* J3. Berikut gambaran dari alokasi pin *header*.



Gambar 2-18. Alokasi Pin J2 dan Pin J3^[4]

2.8. IDE (*Integrated Development Enviroment*)

IDE adalah sebuah lingkungan yang berisi tool – tool yang diperlukan untuk desain, menjalankan dan mengetes sebuah aplikasi, disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Di Delphi, *IDE* terdiri dari :



Gambar 2-19. IDE (*Integrated Development Enviroment*)^[9]

a. Main Window

Main Window adalah bagian utama dari *IDE*. Main Window mempunyai semua fungsi utama dari program – program Windows lainnya.

b. Menu Utama

Menu utama dipakai untuk membuka atau menyimpan file, memanggil wizard, menampilkan jendela lain, mengubah option dan lain sebagainya.

c. Toolbar

Dengan menu toolbar dapat melakukan beberapa operasi pada menu utama yang setiap tombol berisi informasi mengenai fungsi dari tombol tersebut.

d. *Form Designer*

Jendela kosong yang digunakan untuk merancang aplikasi Windows.

e. *Code Editor*

Merupakan bagian yang terpenting di lingkungan Delphi. Jendela ini dipakai untuk menuliskan program Delphi.

f. *Code Explorer*

Code explorer digunakan untuk memudahkan navigasi didalam file unit.

g. *Object Treeview*

Merupakan daftar dari komponen-komponen apa saja yang telah kita pergunakan dan juga merupakan peta dari program yang kita buat.

h. Menu Borland Delphi

1. *Menu File*

Berisi fasilitas untuk membuat Project baru, menyimpan Project, membuka Project, dan keluar dari IDE Delphi.

2. *Menu Edit*

Berisi fasilitas untuk melakukan *editing* atau perubahan pada kode program, juga pengaturan form dan unit (ukuran, penempatan, kontrol, dsb).

3. *Menu Search*

Berisi Fasilitas untuk melakukan pencarian atau penggantian kata dalam tubuh kode program (unit) dan juga mencari letak kesalahan program.

4. *Menu View*

Berisi fasilitas untuk mengatur tampilan IDE Delphi. Misalnya Object Inspector, daftar komponen, pengaturan *Toolbar*, *Form*, dan *Unit*.

5. *Menu Project*

Berisi fasilitas yang berkaitan dengan properti dari *Project*, misalnya menambahkan atau memisahkan *Form* dan *Unit* dari sebuah *Project*.

6. *Menu Run*

Berisi fasilitas untuk Kompiler Delphi, yang terpenting adalah *Run* dan *Reset*.

7. *Menu Component*

Berisi fasilitas untuk mengatur properti *Component Palette* dan instalasi komponen baru.

8. *Menu Database*

Berisi fasilitas yang berkaitan dengan pembuatan aplikasi data.

9. *Menu Tools*

Berisi fasilitas untuk melakukan pengaturan direktori, *library*, *path* penyimpanan file-file penting dalam Delphi, dan tools yang bekerjasama dengan Delphi.

10. *Menu Window*

Berisi fasilitas untuk berpindah dari satu jendela kerja ke jendela kerja yang lain dalam IDE Delphi.

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan dibahas mengenai perencanaan dan pembuatan alat, Sistem Pengelolaan Perpustakaan Menggunakan Kartu RFID, Mikrokontroler AT89S52 Dan PC. Pembahasan dilakukan pada setiap blok rangkaian yang terdiri atas : cara kerja masing-masing blok rangkaian, fungsi masing-masing blok rangkaian, prinsip kerja sistem secara keseluruhan.

Secara garis besar terdapat dua bagian Perangkat yang ada, yaitu :

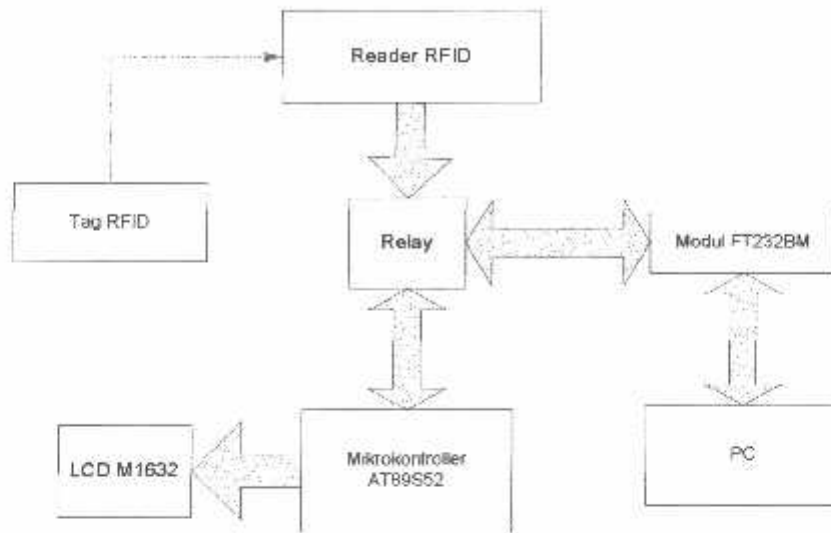
- Perencanaan perangkat keras (Hardware)
- Perencanaan perangkat lunak (Software)

Pada perencanaan perangkat keras meliputi seluruh peripheral yang digunakan pada sistem ini. Sedang pada perencanaan perangkat lunak meliputi : *flowchart* dan *software* secara umum, tetapi pada kenyataannya kedua perangkat tersebut dalam kerjanya akan saling menunjang satu sama lain.

3.2. Perencanaan Perangkat Keras (hardware)

Dalam skripsi ini, perancangan dan pembuatan sistem pengelolaan perpustakaan menggunakan kartu RFID, Mikrokontroler AT89S52 dan PC adapun diagram bloknya adalah sebagai berikut.

3.2.1. Blok Diagram Sistem



Gambar 3-1. Blok Diagram Sistem

Sumber : perancangan hardware

Dari gambar blok diagram diatas, maka prinsip kerja dari masing-masing blok adalah sebagai berikut :

1. **Tag RFID** : Menyimpan kode ASCII dengan nilai tertentu berfungsi sebagai identifikasi seseorang atau objek yang mampu menampilkan identitas member yang tersimpan di Database PC.
 2. **RFID Reader** : Menerima kode ASCII dari tag RFID kemudian mengirimkannya ke mikrokontroler.
 3. **Mikrokontroler AT89S52** : Menerjemahkan data ASCII dari Reader RFID dan mengirimkannya ke PC, selain itu menerima perintah dari PC untuk menampilkan teks di LCD.
 4. **Modul FT232BM(PC Link USBer USB-UART Converter)** : Sebagai interface peralatan ke PC melalui port USB.
 5. **LCD M1632** : Menampilkan teks.
-

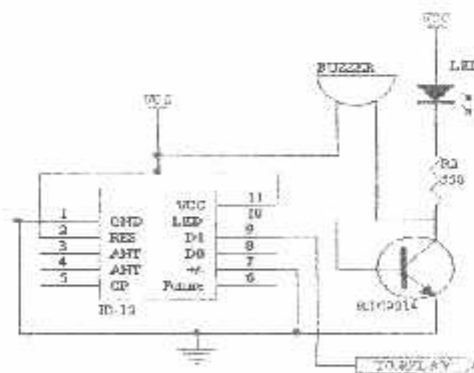
6. **PC** : Berfungsi sebagai database, menerima data dari mikrokontroler sekaligus mengirimkan perintah ke mikrokontroler.

3.2.2. Prinsip Kerja

RFID reader memancarkan frekuensi 125 KHz, kemudian sinyal ini diterima oleh *RFID tag* sehingga *tag* akan terdeteksi dan mentransmisikan informasi yang tersimpan di dalam microchip dari *tag* kepada *reader*. Informasi tersebut dikirim ke Mikrokontroler AT89S52 untuk diterjemahkan. Kemudian setelah diterjemahkan, Mikrokontroler mengirimkannya ke PC untuk dicocokkan dengan database. Jika data tersebut valid maka PC akan mengupdate database jika tidak proses akan berulang seperti awal. Selain itu PC juga memerintah Mikrokontroler untuk menampilkan teks di LCD. Sebagai *interface*, peralatan ke PC melalui port *USB* digunakan modul FT232BM.

3.2.3. Perencanaan Rangkaian Reader ID-12

Dalam skripsi ini *RFID Reader* yang digunakan adalah jenis ID-12, mempunyai jangkauan baca maksimal 12 cm, frekuensi kerja 125 kHz.



Gambar 3-2. Rangkaian *Reader ID-12*

Sumber : Perancangan rangkaian reader

❖ Parameter Input dan Output

1. Perencanaan parameter input :

- ✓ Pulsa TTL dengan frekuensi 3.1 KHz dari pin 9 module ID-12.
- ✓ Beban buzzer dengan resistansi 100Ω tegangan 5 volt.

2. Perencanaan parameter output :

- ✓ Agar buzzer tidak rusak diberi tegangan bias sebesar 2.5 volt, karena buzzer dipicu oleh pulsa 3.1 KHz sehingga terjadi disipasi daya ke beban.

$$I_{Bz} = \frac{V_{Bz}}{R_{Bz}}$$

$$= \frac{2.5}{100} = 25 \text{ mA}$$

$$R_{Bz} = \frac{V_s - V_{Bz}}{I_{Bz}}$$

$$= \frac{5 - 2.5}{25 \cdot 10^{-3}} = 100 \text{ } \Omega$$

- ✓ $R_{LED} = 560\Omega$, karena tegangan input 5 volt maka arus yang mengalir ke LED sebesar :

$$I_{led} = \frac{V}{R_{led}}$$

$$= \frac{5V}{560\Omega} = 8.92 \text{ mA}$$

Sehingga arus total $I_t = I_{led} + I_{Bz}$

$$= 8.92 + 25$$

$$= 33.92 \text{ mA}$$

Dari proses perencanaan dan gambar rangkaian *reader* diatas dapat ditarik kesimpulan bahwa pin-pin kaki yang digunakan adalah :

Pin 1 berfungsi sebagai (GND); pin2 sebagai Reset Bar yang terhubung ke Vcc berguna untuk mengembalikan *reader* pada kondisi semula; pin11 sebagai sumber tegangan suplay pada *reader* (Vcc); pin10 sebagai *pendrive* transistor; pin9 sebagai keluaran pulsa TTL, karena format keluaran data yang diinginkan berbentuk ASCII maka pada pin7 dihubungkan ke *ground*.

3.2.4. Perancangan minimum sistem AT89S52

Disini mikrokontroller berfungsi sebagai pengolah data, agar dapat melakukan prosesnya harus didukung oleh beberapa komponen tambahan, yakni berupa rangkaian clock dan rangkaian reset.

Pada bagian ini perancangan yang akan dibahas meliputi :

- Perancangan rangkaian reset.
- Perancangan clock.
- Perancangan pengaturan port.

3.2.4.1. Perancangan Rangkaian Reset

Rangkaian reset ini diperlukan agar AT89S52 dapat di-reset secara otomatis pada saat pertama kali power diaktifkan, atau disebut power-on reset. Saat catu daya dinyalakan rangkaian reset akan menahan logika tinggi pada pin RST untuk jangka waktu tertentu. Jangka waktu tersebut ditentukan oleh pengosongan muatan pada kondensator, waktu penundaan yang diperlukan adalah $2 \mu\text{s}$ (2 kali siklus).

ditambah dengan waktu penundaan saat kristal mulai on. Karena kristal yang digunakan mempunyai frekuensi sebesar 11.0592 MHz maka satu periode membutuhkan waktu :

$$T = \frac{1}{f_{kristal}} = \frac{1}{11.0592}$$

$$= 9.042 \times 10^{-8} \text{s}$$

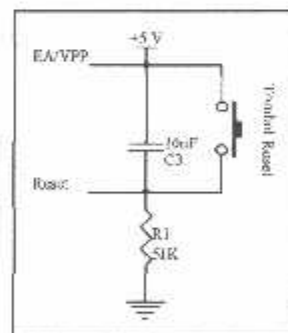
Sehingga waktu minimal logika tinggi yang dibutuhkan untuk mereset mikrokontroller adalah :

$$\text{Reset(min)} = T \times \text{Periode yang dibutuhkan}$$

$$= 9.042 \times 10^{-8} \text{s} \times 24$$

$$= 2.17 \mu\text{s}$$

Dari sini dapat disimpulkan bahwa mikrokontroller membutuhkan waktu minimal 2.17 μs untuk mereset, waktu minimal inilah dijadikan pedoman untuk menentukan nilai R dan C. Dalam perancangan digunakan R = 10 K, C = 10 μF , $V_{cc} = 5 \text{ V}$. Rangkaian reset ini juga ditambah dengan saklar agar dapat juga dilakukan secara manual. Rangkaian *reset* ditunjukkan dalam Gambar berikut.



Gambar 3-3. Rangkaian *Reset*

(Sumber: Perancangan rangkaian *reset*)

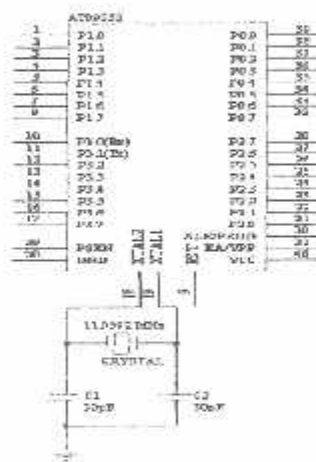
3.2.4.2. Perancangan Rangkaian Clock

Mikrokontroler AT89S52 ini memiliki internal clock generator yang berfungsi sebagai sumber clock, tetapi masih diperlukan rangkaian tambahan untuk membangkitkan clock tersebut. Rangkaian ini terdiri dari 2 buah kapasitor dan sebuah kristal dengan ketentuan :

$C1 \text{ dan } C2 = 30 \text{ PF} \pm 10 \text{ PF}$ untuk kristal

$= 40 \text{ PF} \pm 10 \text{ PF}$ untuk keramik resonator

Dalam perencanaan rangkaian mikrokontroler ini digunakan kapasitor sebesar 30 pF. Rangkaian clock ditunjukkan dalam Gambar berikut.



Gambar 3-4. Rangkaian Clock

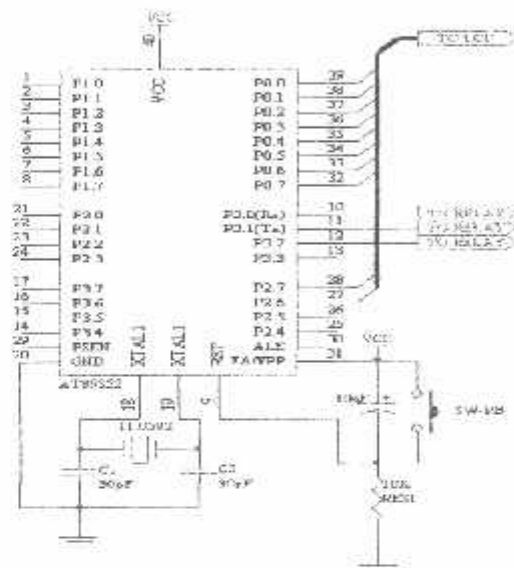
(Sumber : Perancangan rangkaian clock)

3.2.4.3. Perancangan Pengaturan Port

Pada perancangan pengaturan port akan dijelaskan port apa saja yang akan digunakan pada Mikokontroller AT89S52 yaitu:

- P0.0 ... P0.7 : Data Bus ke LCD M1632.
- P3.0 : Digunakan sebagai penerima data (Rx).
- P3.1 : Digunakan sebagai pengiriman data (Tx).

- d) P2.6 : sebagai kontrol sinyal tulis data ke LCD.
- e) P2.7 : sebagai kontrol sinyal baca data ke LCD.
- f) Pin 9 : sebagai masukan reset.
- g) Pin 20 : sebagai GND.
- h) Pin 18 : sebagai masukan clock (XTAL2).
- i) Pin 19 : sebagai masukan clock (XTAL1).



Gambar 3-5. Perencanaan Pengaturan Port

Sumber: Perancangan pengaturan port

3.2.4.4. Perencanaan Komunikasi Serial

Pada perencanaan komunikasi serial alat, digunakan sebuah modul FT232BM PC Link-USB sebagai *interface* komunikasi serial antara MCU dengan PC via port *USB*. Untuk melaksanakan proses komunikasi data antara keduanya diperlukan syarat-syarat yang harus dipenuhi yakni, pengaturan *baud rate* serta pengaturan secara *software*. Jenis data yang akan dikirim adalah dalam bentuk data biner (bit per bit transfer) dengan satuan *baud rate*. Serial control (SCON) merupakan register khusus pengontrol kerja port serial, diset untuk

mentransmisikan data 8 bit UART (Universal Asynron Reciever/Transmitter) yang merupakan standart komunikasi data dengan *baud rate* yang dapat diatur. Penentuan *baud rate* tersebut dilakukan dengan pengesettan. Timer atau *counter* 1 high bit (TH1) bila diinginkan *baud rate* 9600 bps dan SMOD = 0 maka:

$$\text{Baud rate} = \frac{2^{\text{SMOD}} \times f_{\text{Oscillator}}}{32 \times 12[256 - \text{TH1}]}$$

$$9600 = \frac{2^0 \times 11,059 \cdot 10^6}{32 \times 12[256 - \text{TH1}]}$$

$$[256 - (\text{TH1})] = \frac{1}{32} \times \frac{11,059 \cdot 10^6}{12 \times 9600}$$

$$[256 - (\text{TH1})] = 3$$

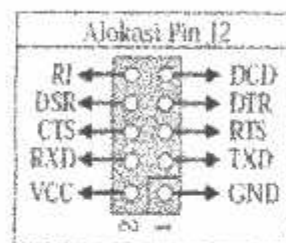
$$\text{TH1} = 256 - 3 = 253 \text{ atau } 0\text{xFD}_{\text{H}}$$

Jika *baud rate* besarnya adalah 9600 bps data membutuhkan waktu 0,104s. Karena satu karakter data terdiri atas 10 bit atau $10 \times 0,104 = 1,04$ ms. Sehingga kecepatan transfernya $1/1,04$ ms atau 0,96 karakter/ms.

Antarmuka serial menggunakan modul FT232BM. Modul ini memungkinkan kita berkomunikasi serial melalui port USB. Karena saluran data keluaran dari modul ini menggunakan standart TTL maka kita tinggal menghubungkan port Rx dan Tx modul ini dengan Mikrokontroller. Hal ini dapat dilakukan karena port serial AT89S52 juga menggunakan standart TTL yang merupakan level tegangan baku dalam rangkaian-rangkaian digital. Dalam standart TTL logic 0 (low) dinyatakan sebagai tegangan antara 0 volt sampai 0.8 volt, dan logic 1 (high) dinyatakan sebagai tegangan antara 2.5 volt sampai 5 volt. Berikut gambaran pin pada modul FT232BM yang digunakan :

Antarmuka dari atau ke modul FT232BM PC Link-USBer menggunakan level TTL dan terhubung ke header dengan label J2. Dalam implementasi perancangan alat pin yang digunakan hanya :

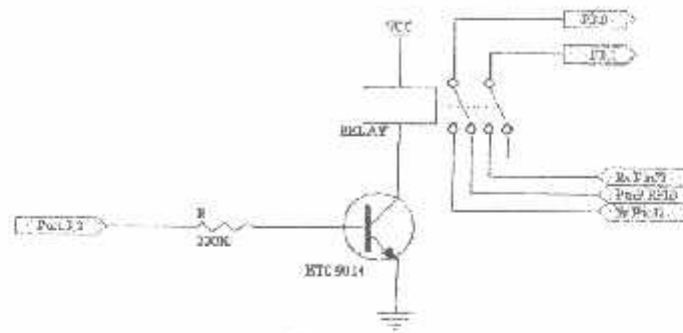
- ✓ Pin 1 : Ground.
- ✓ Pin 2 : Vcc (sebagai tegangan supply modul).
- ✓ Pin 3 : TXD (sebagai pengiriman data).
- ✓ Pin 4 : RXD (sebagai Penerima data).



Gambar 3-7. Pin J2
(Sumber : www.innovativeelectronic.com)

3.2.5. Rangkaian Switch Data

Rangkaian switch data disini digunakan untuk memilih jalur komunikasi data dengan menggunakan CS (chip select) pada port 3.2 pada mikrokontroller AT89S52 yang dihubungkan ke sebuah transistor type KTC 9014 dimana transistor tersebut berfungsi sebagai driver relay DPDT(*dual pole dual terminal*) dengan catu daya $V_{cc} = 5\text{ V}$. Seperti pada gambar berikut :



Gambar 3-8. Rangkaian switc data

(Sumber: Perancangan rangkaian switc data)

Data transistor KTC9014 yang diperoleh dari *datasheet* (KEC, 1994) adalah :

- $R_B = 220 \text{ K}\Omega$
- H_{fe} minimum = 60.
- H_{fe} maximum = 1000.

$$I_B = \frac{V_{cc}}{R_b}$$

$$I_B = \frac{5V}{220K\Omega} = 0,023 \text{ mA}$$

Setelah didapat nilai I_B , maka dapat ditentukan nilai I_C :

$$I_C = H_{fe} \cdot I_B$$

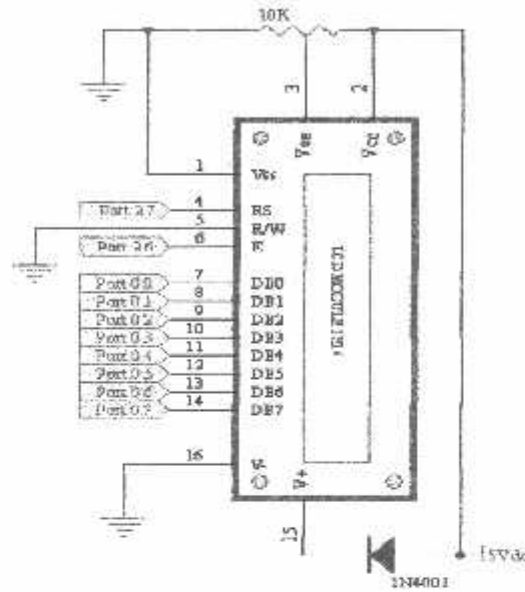
$$I_C = 60 \cdot 0,023 \text{ mA} = 1,38 \text{ mA}$$

Dengan didapatkannya nilai $I_C = 1,38 \text{ mA}$ arus inilah yang dipakai untuk memicu relay.

3.2.6. Perancangan rangkaian LCD M1632

LCD diperlukan untuk menampilkan nilai karakter input yang akan diproses dan data karakter output dari hasil pengukuran supaya hasil proses dan pengukuran bisa dipahami oleh manusia. LCD M1632 pada skripsi ini digunakan hanya sebagai tampilan data karakter input output saja.

pengukuran bisa dipahami oleh manusia. LCD M1632 pada skripsi ini digunakan hanya sebagai tampilan data karakter input output saja.



Gambar 3-9. Perencanaan Rangkaian LCD M1632

Sumber :Perancangan rangkaian LCD

Untuk nilai arus yang ada pada V_{in} LCD adalah sebesar 0,33mA yang di dapat dari :

$$V_{in} = 5 \text{ Volt}$$

$$R = 10 \text{ Kohm}$$

$$I_{LCD} = \frac{V_{in}}{R} = \frac{5 \text{ Volt}}{10 \times 10^3 \text{ ohm}} = \frac{5 \times 10^{-3} \text{ v}}{10 \text{ ohm}}$$

$$= 0,5 \text{ mA}$$

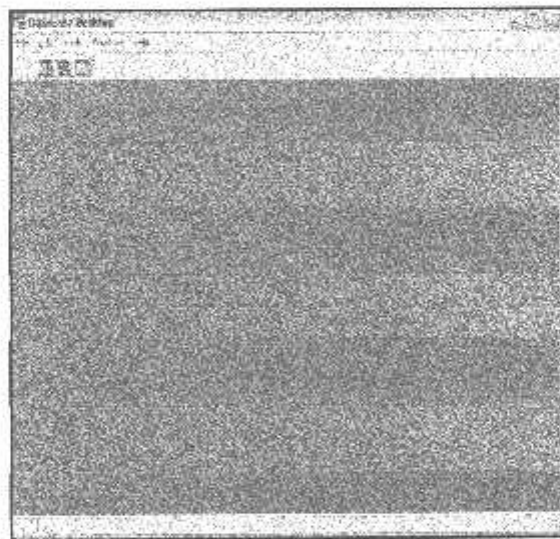
3.3. Perancangan Perangkat Lunak (software)

Setelah semua perangkat keras telah selesai dikerjakan pada tahap selanjutnya adalah pembuatan perangkat lunak (software) yang akan menangani sistem rangkaian. Pada perangkat lunak inilah kita dapat menentukan bagaimana sistem rangkaian ini bekerja, pada bagian ini juga kerja rangkaian ditentukan.

3.3.1. Perencanaan Database

Program aplikasi yang digunakan di PC pada skripsi ini menggunakan program Delphi dimana database yang dirancang menggunakan table dengan type table Paradox7. proses perancangannya adalah sebagai berikut :

- 1) Pertama membuka lembar kerja Database Dekstop pada program Delphi, sehingga tampilannya akan seperti berikut :



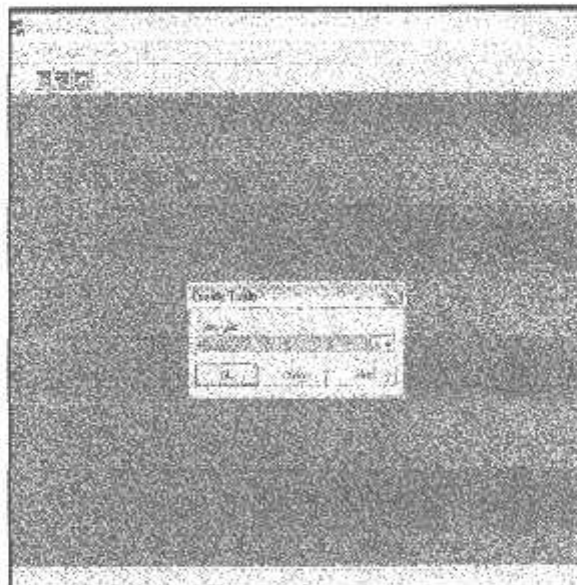
Gambar 3-10. Lembar Kerja DataBase Dekstop
(Sumber :Perancangan DataBase)

- 2) Setelah itu klik menu file, kemudian membuat Table baru dengan klik New-Table.



Gambar 3-11. Membuat Table Baru
(Sumber :Perancangan DataBase)

- 3) Kemudian tampilan akan beralih pada jenis atau type table yang akan digunakan pilih Paradox7, selanjutnya klik OK



Gambar 3-12. Menentukan Type Table
(Sumber :Perancangan DataBase)

- 4) Berikut contoh Table yang telah dibuat.

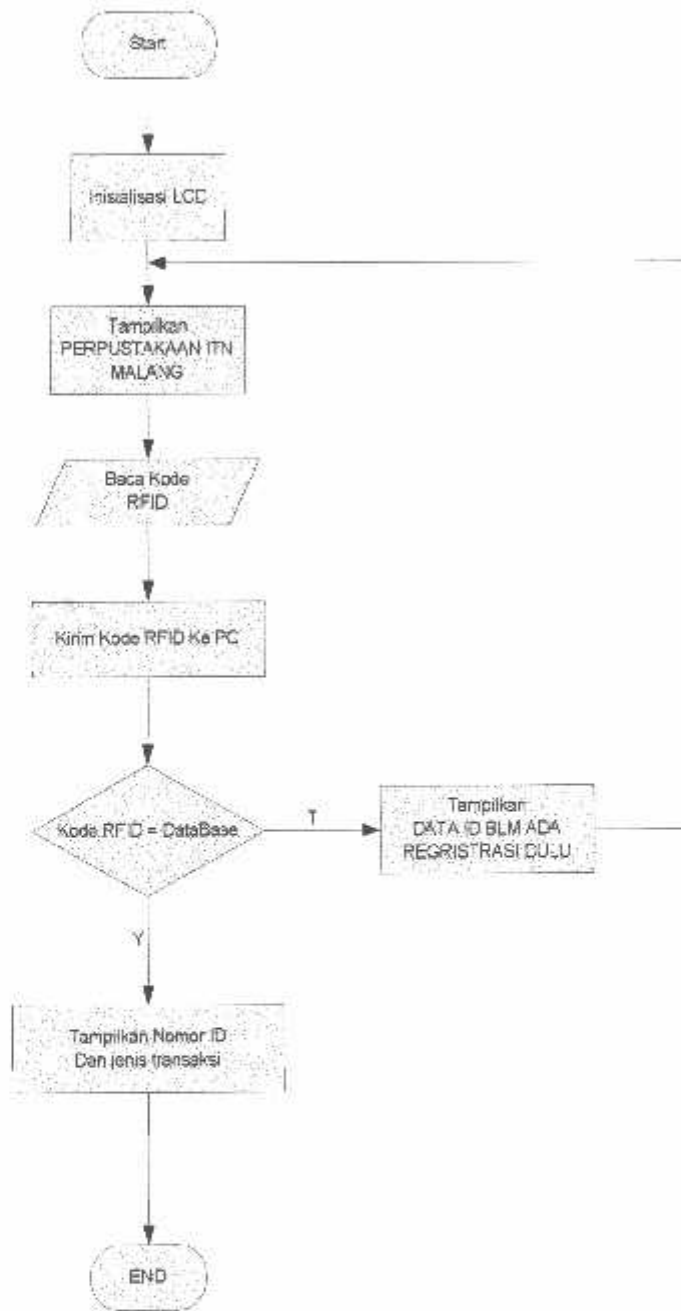


Gambar 3-13. Contoh Rancangan Table
(Sumber :Perancangan DataBase)

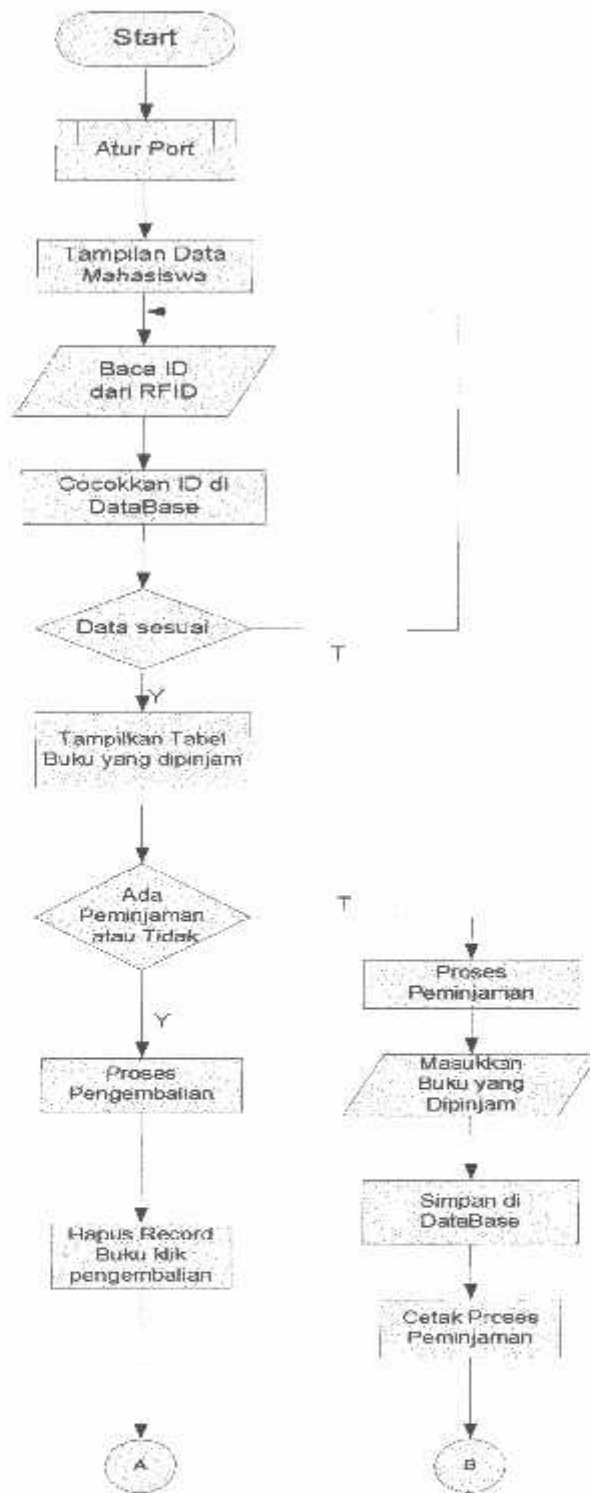
Dalam perancangan Database ini jumlah database yang dibuat ada 5 buah, yakni terdiri :

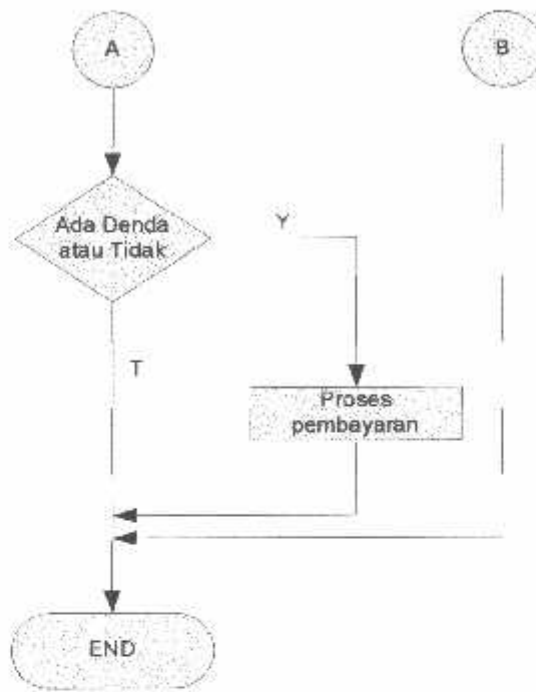
1. tbdenda : digunakan sebagai proses pengaturan denda dan terdiri atas dua field (denda dan durasi).
2. tbid : digunakan sebagai database inputan untuk data identitas anggota perpustakaan.
3. tbIDBUKU : digunakan sebagai database inputan buku.
4. tbJURUSAN : digunakan sebagai database jenis jurusan yang terdiri dari sebuah field (JURUSAN).
5. tbPINJAMAN : digunakan sebagai database proses inputan dari peminjaman buku-buku.

3.3.2. Flowchart



Gambar 3-14. Diagram Alir Program di Mikrokontroller
(Sumber : perancangan Software)





Gambar 3-15. Diagram Alir Program di Komputer

(Sumber : perancangan Software)

BAB IV

PENGUJIAN ALAT

4.1. Pendahuluan

Untuk mendapat hasil yang maksimal setelah melaksanakan perencanaan dan pembuatan alat, maka perlu dilakukan pengujian terhadap alat yang telah dibuat. Pengujian ini bertujuan untuk mengetahui apakah alat yang telah dibuat dapat bekerja sesuai dengan perencanaan.

Bagian-bagian yang diuji dari peralatan adalah :

- Pengujian RFID Reader.
- Pengujian Komunikasi Serial Modul FT232BM PC-Link USBer.
- Pengujian Tampilan LCD.
- Pengujian Keseluruhan sistem.

4.2. Pengujian RFID Reader

4.2.1. Tujuan

Untuk mengetahui apakah RFID Reader dapat membaca tag atau tidak.

4.2.2. Peralatan yang digunakan

1. Tag RFID.
2. Rangkaian Reader RFID.
3. PC.
4. Adaptor.

4.2.3. Prosedur Pengujian

1. Menghubungkan Rangkaian Reader ke port COM1 PC.

2. Untuk menguji pengiriman data ke PC digunakan Hyper Terminal prosesnya :

- Klik Start
- Programs
- Accessoris
- Communications
- Hyper Terminal

3. Maka akan tampil jendela connection description, ketikkan sebuah alias nama dalam hal ini TES.



Gambar 4 - 1. Kotak Dialog Connection Description

4. Setelah memberikan nama deskripsi koneksi, pilih setting komunikasi menggunakan COM1 .



Gambar 4 - 2. Kotak Dialog Connect To.

5. Pada Properties COM1 aturlah port settingnya seperti gambar berikut :



Gambar 4 - 3. Kotak Dialog COM 1 Properties

6. Setelah semua settingan di Hyper Terminal selesai, lakukan pengujian dengan mendekatkan tag ke reader. Sehingga akan tampil ID dari tag tersebut.



Gambar 4 - 4. Identifikasi Reader Terhadap kartu.

4.2.4. Hasil Pengujian Pembacaan RFID

Setelah dilakukan beberapa pengujian, didapatkan hasil bahwa reader dapat membaca tag. Berikut tabel hasil pengujiannya :

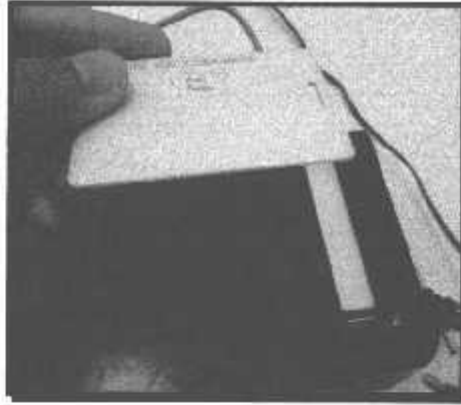
Tabel 4-1. Hasil Pengujian Pembacaan tag RFID

Jarak Pengukuran	PERCOBAAN														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
2 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
3 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
4 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
5 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
6 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
7 cm	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
8 cm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9 cm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10 cm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11 cm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12 cm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Keterangan : * Data terbaca

- Data tidak Terbaca

Dari tabel pengujian dapat ditarik kesimpulan bahwa setelah diadakan percobaan sebanyak 15 kali reader masih dapat membaca tag dengan baik dalam jarak maksimal 7 cm, selebihnya data tidak dapat terbaca. Jeda waktu untuk pembacaan tag berikutnya adalah 8 detik. Pembacaan tag dilakukan secara tegak lurus terhadap reader.



Gambar 4-5. Pengukuran Jarak baca Reader Terhadap kartu.

4.3. Pengujian Komunikasi Serial Modul FT232BM PC-Link USBer.

4.3.1. Tujuan

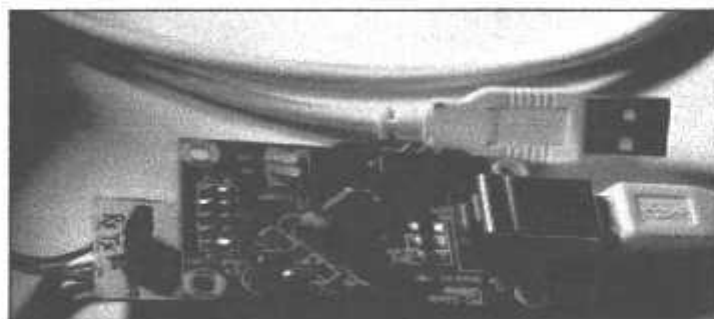
Untuk mengetahui apakah modul dapat mengirimkan data atau menerima data.

4.3.2. Peralatan yang dibutuhkan

1. PC.
2. Modul FT232BM PC-Link USBer.
3. Adaptor.
4. Jumper, konektor USB.

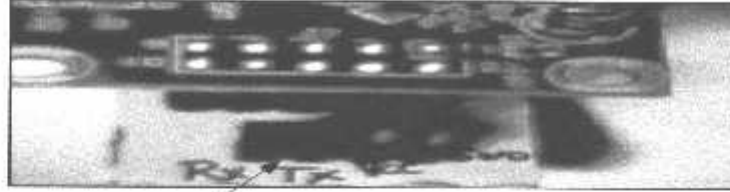
4.3.3. Prosedur Pengujian

1. Menyusun rangkaian seperti gambar 4-6. Hubungkan Rangkaian Modul FT232BM ke PC menggunakan Konektor USB.



Gambar 4-6. Rangkaian Pengujian Modul FT232BM

2. Dalam pengujian, pin kaki Rx; Tx pada modul FT232BM dihubungkan singkat. Sehingga ketika PC mengirim data ke modul FT232BM, data akan di-loop balik ke PC.



Bagian Jumper Rx
Dan Tx

Gambar 4-7. Proses Penjumperan Rx Dan Tx

3. Karena koneksi Modul ke PC menggunakan port USB maka kita harus menginstall *driver* modul FT232BM. Dalam hal ini kita menginstall *driver* VCP, dengan *driver* ini modul FT232BM akan dikenali PC sebagai *virtual com*. Langkah-langkahnya adalah sebagai berikut :

- 1) Kita hubungkan konektor USB ke port USB PC, secara otomatis sistem operasi windows di PC akan menampilkan Found New Hardware Wizard.



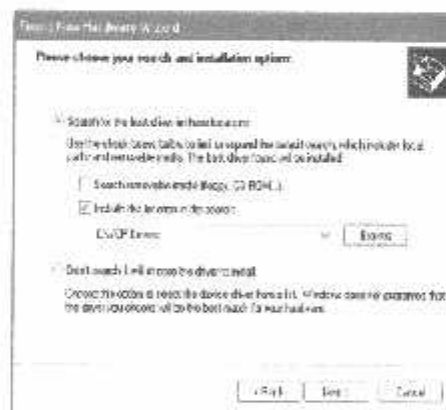
Gambar 4-8. Panel Found New Hardware Wizard

- 2) Setelah itu kita akan diarahkan untuk menginstal driver dari CD instalasi atau dari floppy.



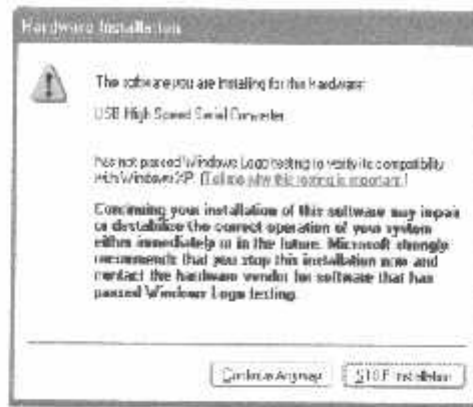
Gambar 4-9. Tampilan Proses Bantuan Penginstalan driver

- 3) Menentukan lokasi tempat driver yang akan diinstal berada, untuk mencari klik tombol browse.



Gambar 4-10. Tampilan Browse untuk mencari lokasi driver berada

- 4) Setelah sesuai proses instalasi akan dimulai, lakukan dengan mengklik tombol Continue Anyway.



Gambar 4-11. Tampilan Konfirmasi proses Hardware Installation

- 5) Proses instalasi akan berlanjut, tunggu sejenak sampai semua driver yang dibutuhkan terinstal semua.



Gambar 4-12. Proses Installation sedang berjalan

- 6) Setelah selesai dan klik finish, untuk mengecek modul kita terhubung atau tidak buka Device Manager dan klik option Port(COM&LPT) apabila proses instalasi berhasil maka tampilan akan seperti berikut :



Gambar 4-13. Tampilan indikasi modul terhubung ke PC

4. Pengujian dilakukan dengan menggunakan program Delphi, Sebagai indikasi terjadinya proses kirim atau terima data dapat dilihat nyala led indicator Rx, Tx pada modul FT232BM. Hal ini ditunjukkan pada gambar 4-16.

❖ Program Ujicoba Komunikasi FT232BM ke PC Menggunakan Program Delphi 7.0

```

procedure TForm1.Button1Click(Sender: TObject);
begin
ComPort1.WriteStr('Test')
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var
data:string;
begin
ComPort1.ReadStr(data,count);
Memo1.Text:=Memo1.Text+data;

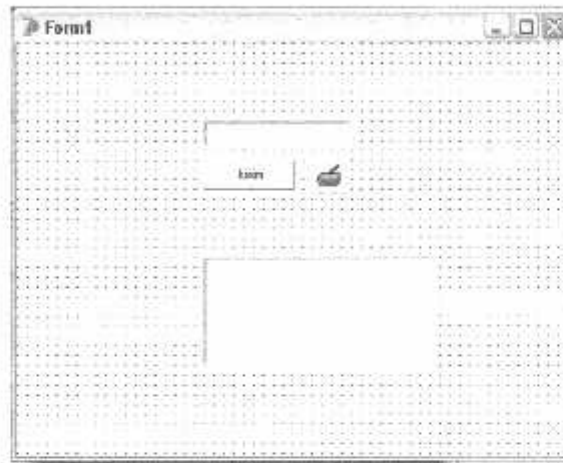
```

end;

end.

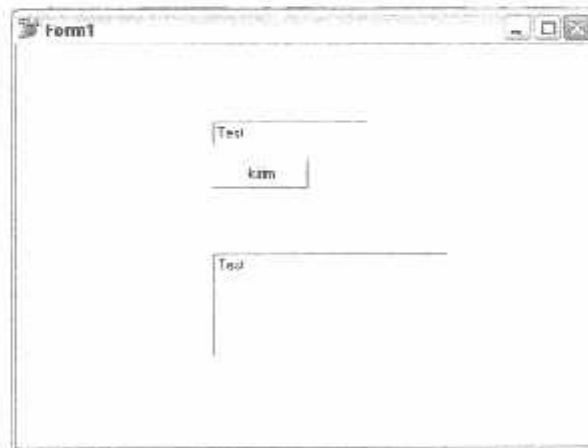
❖ Langkah Pembuatan Form di Delphi

- ✓ Buat form baru.
- ✓ Tambahkan beberapa jenis komponen palette : 1 buah button, 1 component, 1 buah edit, 1 buah memo.

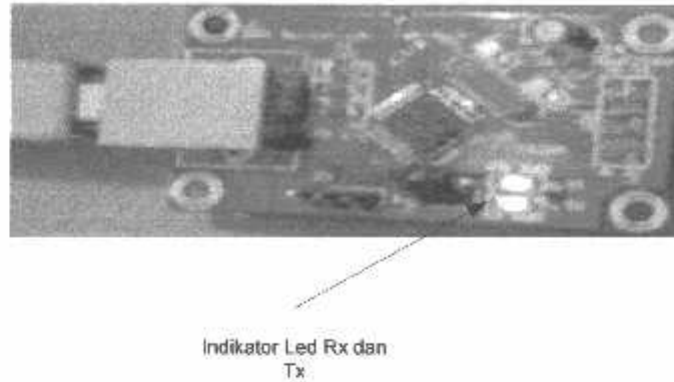


Gambar 4-14. Pembuatan form dalam program Delphi

- ✓ Setelah semua komponen selesai dibuat lakukan running dengan menekan tombol F9.



Gambar 4-15. Tampilan setelah running



Gambar 4-16. Tampilan Indikator led Rx dan Tx Modul FT232BM

4.4. Pengujian Tampilan LCD

4.4.1. Tujuan

Untuk mengetahui kemampuan rangkaian tampilan yang sudah dibuat apakah dapat mendukung sistem yang direncanakan dan untuk menampilkan teks pada LCD.

4.4.2. Peralatan yang digunakan

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

4.4.3. Prosedur Pengujian



Gambar 4 -17. Diagram Blok Pengujian Rangkaian LCD.

1. Menyusun rangkaian seperti gambar 4 - 17.
2. Menjalankan program untuk menampilkan tulisan ke LCD.
3. Mengamati keluaran pada LCD.

4. Isi memori program seperti dibawah ini yang menampilkan tulisan
"PERPUSTAKAAN ITN MALANG"

➤ Program Pengujian LCD

LCD_RS bit P2.7

LCD_CS bit P2.6

ORG 0011

```

START :   CALL INISIAL
          MOV DPTR,#BARIS_1
          MOV R3,#16
          MOV A,#80H
          CALL TULIS_INST

TULIS :   CLR A
          MOVC A,@A+DPTR
          INC DPTR
          CALL TULIS_DATA
          DJNZ R3,TULIS

          MOV DPTR,#BARIS_2
          MOV R3,#16
          MOV A,#0C0H
          CALL TULIS_INST

TULISI :  CLR A
          MOVC A,@A+DPTR
          INC DPTR
          CALL TULIS_DATA

```

```
                DJNZ R3,TULIS1
                JMP $
INISIAL :      MOV A,#03FH
                CALL TULIS_INST
                MOV A,#0DH
                CALL TULIS_INST
                MOV A,#01H
                CALL TULIS_INST
TULIS_INST :   CLR P2.7
                MOV P0,A
                SETB P2.6
                CLR P2.6
                CALL TUNDA
                RET
TULIS_DATA :   SETB P2.7
                MOV P0,A
                SETB P2.6
                CLR P2.6
                CALL TUNDA
                RET
TUNDA      :   MOV R0,#0FFH
TUNDA1     :   MOV R1,#50H
                DJNZ R1,$
                DJNZ R0,TUNDA1
```

RET

BARIS_1 : DB 'PERPUSTAKAAN ITN'

BARIS_2 : DB ' MALANG '

END

4.4.4. Hasil Pengujian

Setelah data diolah mikrokontroler maka hasil tampilan LCD berupa tulisan "PERPUSTAKAAN ITN MALANG"



Gambar 4-18. Hasil Pengujian Tampilan LCD.

4.5. Pengujian Sistem Secara Keseluruhan

4.5.1. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui apakah semua sistem yang telah dirancang berjalan dengan baik dan juga untuk mengetahui *error* yang terjadi.

4.5.2. Peralatan yang digunakan

- PC.
- Alat Secara Keseluruhan.
- Konektor USB.
- Adaptor.
- Tag RFID.

4.5.3. Prosedur Pengujian

1. Menghubungkan keseluruhan rangkaian sesuai dengan diagram blok

2. Menjalankan program Database Delphi
3. Melakukan setting Port COM yang digunakan dalam program delphi
4. Mendekatkan *Tag* RFID
5. Melakukan proses peminjaman atau pengembalian buku

4.5.4. Hasil Pengujian

1. Tampilan Informasi Pada LCD
 - a. Menampilkan Tulisan “ PERPUSTAKAAN ITN MALANG ”.
 - b. Menampilkan Tulisan “MAAF ID BLM ADA”.
 - c. Menampilkan Tulisan “ ID TAG PEMINJAMAN ”.
 - d. Menampilkan Tulisan “ ID TAG PENGEMBALIAN ”.
2. Tampilan Informasi Pada Program Delphi
 - a. Menampilkan Form Setting Port.
 - b. Menampilkan Form Data Mahasiswa Secara Otomatis.
 - c. Menampilkan Form Proses Peminjaman Buku.
 - d. Menampilkan Form Pengembalian Buku.
 - e. Menampilkan Form Data Buku.
 - f. Menampilkan Form Regristrasi atau pendaftaran member baru.
 - g. Menampilkan Form setting Denda.
 - h. Menampilkan Form Data Kontrol Peminjaman.

4.5.5. Proses Identifikasi

- a. Setelah mengkoneksikan peralatan dengan PC atau Laptop, pertama kita harus mensetting port dulu seperti gambar 4-19 sedangkan tampilan di LCD akan menampilkan tulisan seperti gambar 4-20.
-



Gambar 4 – 19. Setting Port Pada Program Delphi



Gambar 4 – 20. Tampilan LCD awal sebelum membaca tag

- b. Ketika tag didedkatkan ke reader ID akan dicek dan dicocokkan dengan database. Jika tag belum terdaftar dalam database PC, maka tampilan LCD akan menampilkan tulisan seperti berikut.



Gambar 4 – 22. Tampilan LCD ID belum terdaftar di Database.

- c. Sedangkan jika ID telah terdaftar di Database PC. Ada 2 proses identifikasi data oleh sistem, secara otomatis sistem akan mengidentifikasinya bahwa :

- 1) Kondisi pertama : jika sebelumnya belum pernah melakukan transaksi peminjaman sistem akan mengindikasikan bahwa ini merupakan proses peminjaman sehingga tampilan pada pada table buku yang dipinjam akan kosong sesuai gambar 4-23 sedangkan tampilan LCD akan menampilkan Nomor ID dan tulisan PEMINJAMAN seperti pada gambar 4-24 jika melakukan peminjaman tinggal klik button pinjam kemudian browse buku yang dipinjam diakhiri klik button simpan.

Gambar 4 – 23. Tampilan Form Proses Peminjaman Buku.



Gambar 4 – 24. Tampilan Proses Peminjaman Buku

- 2) Kondisi kedua : sebaliknya jika sebelumnya telah melakukan transaksi peminjaman maka dalam table buku yang dipinjam terdapat record buku, proses pengembalian buku dilakukan dengan

menghapus record buku yang dipinjam dengan cara mengklik button pengembalian, tampilan di LCD seperti gambar berikut



Gambar 4 – 25. Tampilan Proses Pengembalian Buku

 A screenshot of a web-based library management system. The window title is 'PERPUSTAKAAN ITH MAI ANS'. The page has a header 'Menu Tambah Data control'. Below this, there is a user profile section with fields for 'ID' (0000593873), 'NIK' (0012378), and 'NAMA' (Dewi Nurhasanah). To the right of these fields is a small portrait photo of a woman. Below the profile is a section titled 'BUKU YANG DIPINJAM'. This section contains a table with the following data:

ID BUKU	JUDUL	PINJAM	KEMBAL
M00001	Penggunaan Mikroprolel A19851 dengan DC- dan Koneksi	1/25/200	

 At the bottom of the interface, there is a section 'Buku yang tersedia' with a search input field and several buttons: 'PINJAM', 'PENGEMBALIAN', 'BATAL', 'Dine', and 'PENYUTRAN'.

Gambar 4 – 26. Tampilan Form Proses Pengembalian Buku

- d. Sistem juga dilengkapi dengan data record buku untuk memudahkan proses peminjaman buku. Serta disediakan juga form khusus untuk memasukkan identitas buku baru.

Gambar 4 – 27. Tampilan Form Input Data Buku Baru

ID	JUDUL
000001	Perancangan Mikrokontroler AT89C51 dengan C/C++ dan assembly
000002	Perancangan Sistem Dikita /
000003	Desain Perangkat Lunak dan Layout PCB dengan Protek 99SE
000004	Belajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi Edisi 2
000005	Linux For Beginner
000006	Perawatan dan Kalibrasi Oemotek
000007	Pembuat Sosis Cair Dari Hanes
000008	Kontrol robot variabel dalam robot
000009	Konversi Tenaga Listrik
000010	Belajar Komunikasi USB2
000011	Perancangan Sistem Deteksi Tidal

Gambar 4 – 28. Tampilan Form Data Kontrol dari Data Buku

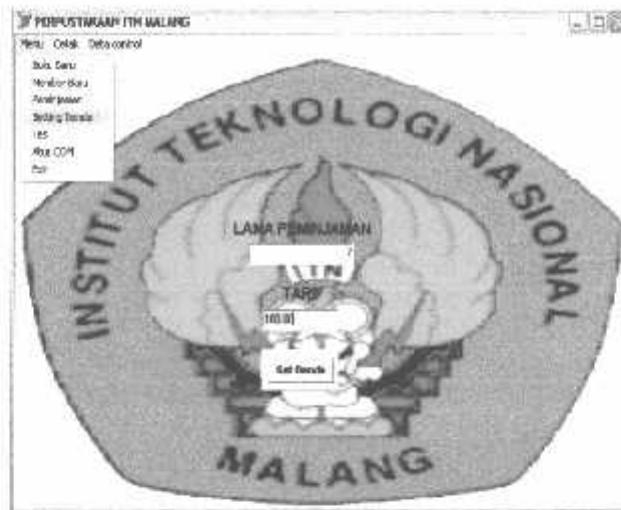
- e. Untuk kemudahan dalam proses registrasi anggota baru disediakan juga form khusus yang nantinya akan ditampilkan di Data Mahasiswa.

Gambar 4 – 29. Tampilan Form Registrasi atau pendaftaran member baru

ID	NIM	NAMA	ALAMAT
0000000004	0017070	Ukhaerul Khairi	Jl. Candi Pangung 4A, 2
0000000074	0017074	Tulus Arwani	Jl. Pejuang Depan 10

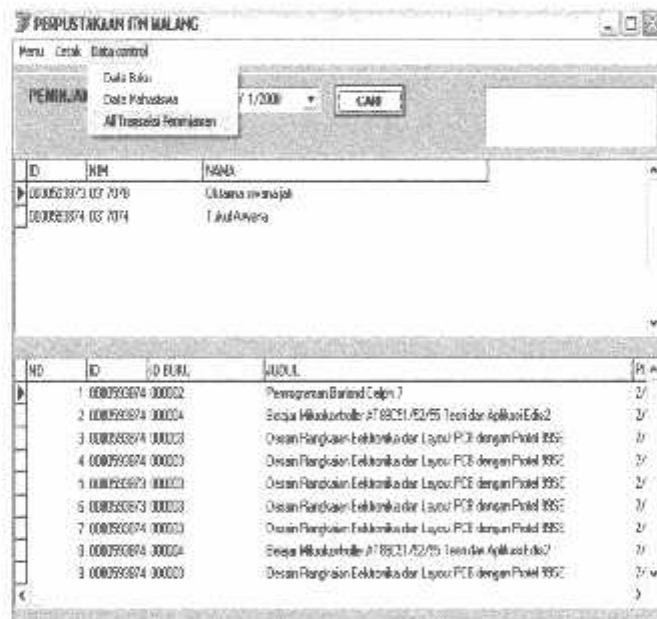
Gambar 4 – 30. Tampilan Form Data Mahasiswa

- f. Untuk proses penentuan denda jika terjadi keterlambatan pengembalian maka juga disediakan form setting denda yang ketentuannya disesuaikan dengan kebijakan perpustakaan. Dalam praktik untuk lama peminjaman buku disetting maksimal selama 7 hari dengan denda keterlambatan pengembalian Rp.100; per hari, per buku.



Gambar 4 – 31. Tampilan Form Setting Denda.

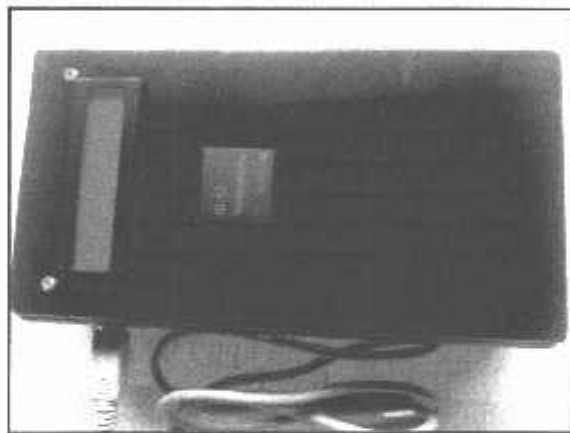
- g. Sebagai proses dokumentasi dari setiap transaksi peminjaman dan pengembalian buku yang dilakukan oleh anggota dalam sistem juga disediakan form khusus data kontrol peminjaman.



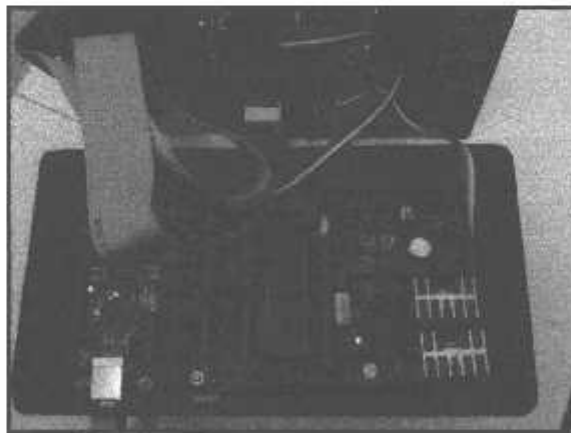
4.6. Spesifikasi Alat

➤ Sistem Pengelolaan Perpustakaan Menggunakan RFID

- Dimensi : 18.5 x 10 x 5.5 cm
- Power Supply : AC / DC 9-12 volt
- Modul FT232BM PC-Link USB
- Mikrokontroler AT89S52
- LCD M1632
- RFID ID-12
- *Buzzer & Led*



Gambar 4 - 33. Foto Alat



Gambar 4 - 34. Foto Komponen Alat

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan dari hasil pengujian alat maka dapat diambil suatu kesimpulan sebagai berikut :

1. Alat yang telah dirancang dapat berjalan sesuai dengan rencana.
2. Karena menggunakan koneksi lewat port *USB*, setelah terkoneksi ke PC alat akan dideteksi sebagai COM.
3. *RFID reader* dapat membaca tag dengan jarak maksimal hanya 7 cm, dengan jeda waktu pembacaan tag berikutnya adalah 8 detik.
4. Pada percobaan pembacaan tag sebanyak 15 kali tidak terdapat error relatif, artinya meskipun dilakukan pembacaan berulang-ulang reader dapat membaca dengan baik dengan syarat kondisi pembacaan tag tegak lurus dengan reader.
5. Untuk menghubungkan antara alat dengan program aplikasi yang akan digunakan sebelumnya harus dilakukan penyettingan Port COM nya.

5.2 Saran

Pengembangan lebih lanjut sistem yang sudah dibuat, penulis menyarankan sebaiknya tag *RFID* tidak difungsikan sebagai kartu member saja tapi juga dapat digunakan sebagai kunci dalam memasuki area perpustakaan. Dimana dalam implementasinya perangkat *reader* diintegrasikan sebuah *driver* motor yang nantinya digunakan sebagai pengontrol pintu secara otomatis. Serta penggunaan

antenna eksternal sehingga pengguna tidak harus selalu mendekatkan *tag* ke *readernya*.

DAFTAR PUSTAKA

- [1] <http://www.wileyurope.com>, RFID Handbook Fundamentals and Applications in Contactless Smart Cards and Identification Second Edition.
- [2] <http://www.alldatasheet.com>, Data Sheet RFID ID-12
- [3] <http://www.atmel.com>, Data Sheet AT89S52.
- [4] <http://www.innovative-electronics.com>
- [5] DC Green, Komunikasi Data, 1998 ; 27
- [6] Widodo Budiharto, I/O Bus & Motherboard, PT Elex Media Komputindo, Jakarta, 2004.
- [7] LCD Module User Manual.
- Madcoms.2002.*Pemrograman Borland Delphi7*.Yogyakarta:Penerbit C.V.ANDI OFFSET.
- Digital,Lab.Elektronika.2007.*PELATIHAN DELPHI DAN DATABASE*.
- Wahyudin,Didin.2007, "Belajar Mudah Mikrokontroller AT89S52 dengan Bahasa Basic Menggunakan BASCOM-8051",C.V ANDI OFFSET,Yogyakarta.
- Budioko,Totok.2005. *Belajar Dengan Mudah dan Cepat Bahasa C pada Mikrokontroller AT89x051/AT89C51/52*.Yogyakarta:Penerbit Gava Media.
- Putra,Agfianto Eko.2004.*Belajar Mikrokontroller AT89C51/52/55:Teori dan Aplikasi,Edisi2*.Yogyakarta:Penerbit Gava Media

LAMPIRAN



Formulir Perbaikan Ujian Skripsi

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

NAMA

Sony Kristiawan

NIM

05.17.078

Perbaikan meliputi

1) Rancangan di Hood wire. gunakan
di di panel.

2) perancangan di Kase. di perpanjang
* tabel ada corep?
* dst

Malang, 15-07-2008


Sony 5.15.45



FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Elektronika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

Nama : Dony Kristiawan
NIM : 03.17.078
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Masa Bimbingan : 09 November 2007 s/d 09 Mei 2008
Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengelolaan Perpustakaan Menggunakan Kartu RFID, Mikrokontroller AT89S52 Dan PC.

Tanggal	Uraian	Paraf
15 Maret 2008	❖ Rancangan di Hardware disesuaikan dengan dilaporan	
	❖ Perancangan Database Diperlengkap	

Mengetahui,

Dosen Pembimbing I

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

Dosen Pembimbing II

(Dr. Cahyo Crysdian, Msc)
NIP.Y. 103040412

Dosen Penguji,

Penguji I

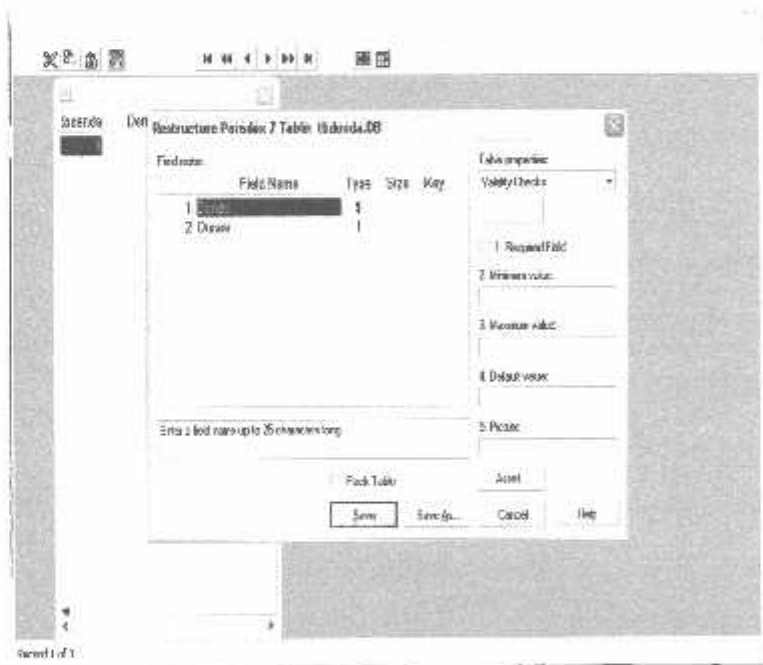
(Ir. M. Abdul Hamid, MT)
NIP.Y. 1018800188

Penguji II

(Ir. Komang Somawirata, S1, MT)
NIP.Y. 1030100361



Gambar struktur proses pembuatan table ID



Gambar struktur proses pembuatan table Denda



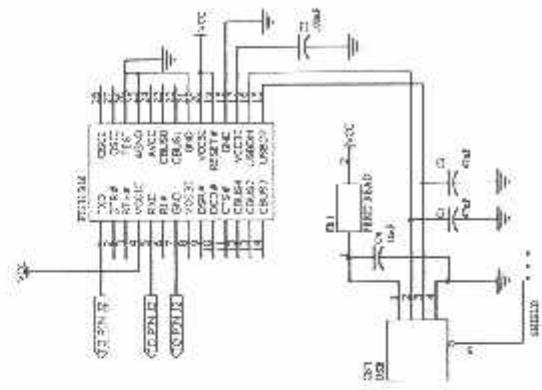
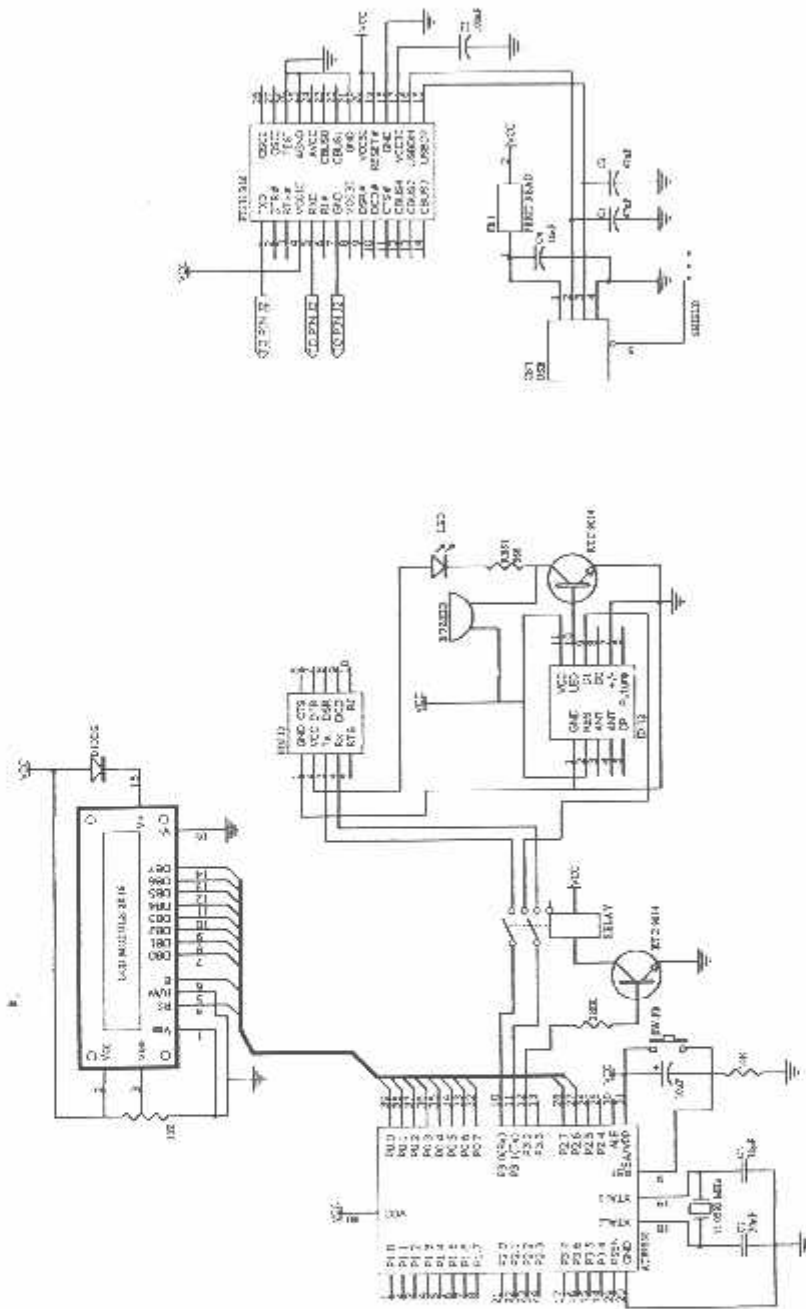
Gambar struktur proses pembuatan table ID Buku



Gambar struktur proses pembuatan table Jurusan



Gambar struktur proses pembuatan table Pinjaman



Tipe	BARUBALAS BERSALURAN SUTERA
Spesifikasi	RT2-M14
Material	RT2-M14
Tempo	1000000
Tempo	1000000
Tempo	1000000



FORMULIR BIMBINGAN SKRIPSI

Nama : Dony Kristiawan
 Nim : 03.17.078
 Masa Bimbingan : 09 November 2007 s/d 09 Mei 2008
 Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengelolaan Perpustakaan Menggunakan Kartu RFID, Mikrokontroler AT89S52 Dan PC

No	Tanggal	Uraian	Paraf Pembimbing
1	8/2008 1	Bab I + II Perbaiki	
2	2/2008 2	Bab I + II + III Perbaiki Kapasitas hard disk tambahan 100 USD data Base.	
3	11/2008 2	Bab I - III	
4	15/2008 2	Simpulan Kesimpulannya	
5	18/2008 2	Perbaikan Kesimpulan	
6	6/2008 3	Tambahan Kesimpulan	
7	10/2008 3	Kesimpulan	
8			
9			
10			

Malang,

Dosen pembimbing 1

Ir. F. Yudi L. Praptono, MT
 NIP. Y. 1039500274



FORMULIR BIMBINGAN SKRIPSI

Nama : Dony Kristiawan
Nim : 03.17.078
Masa Bimbingan : 09 November 2007 s/d 09 Mei 2008
Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengelolaan Perpustakaan Menggunakan Kartu RFID, Mikrokontroler AT89S52 Dan PC

No	Tanggal	Uraian	Paraf Pembimbing
1	9/1/2008	Bab I & II	
2	9/2/2008	Bab III, IV	
3	12/2/2008	Kesimpulan ok	
4	18/2/2008	Makalah Seminar Acc	
5	19/3/2008	Acc Compres.	
6			
7			
8			
9			
10			

Malang

Dosen pembimbing II

Dr. Cahya Crisdian, Msc
NIP. Y. 103040412

Form S-4b

LISTING
PROGRAM

```

#include <at89x51.h>
#include "Pending.c"
#include "lcdpar.c"
#include "myser1.c"

#define cs                P3_2
#define buzzer1         P0_1
unsigned char
j[16],i,trfid,k[20],a,b1,b2,b3,b4,b5,b6,b7,b8,tandap;
unsigned long Ndata;
bit t;
void SerialInterrupt (void) interrupt 4 using 1
{
    while(!RI){;}
    RI = Lo ;
    if (SBUF==2) {trfid=1;i=0;}
    else if(SBUF==3) {trfid=2;t=0;}
    else ;
    if (trfid==1)
    {
        j[i]=SBUF;
        i++;
    }
}
void cetakrfid()
{
    i=3;
    while(i<11)
    {
        konversi=j[i];
        dataout();
        i++;
    }
    t=1;
    //pos(2,1);
}
unsigned char kon(unsigned char n)
{
    if (n <= 0x3a) return(n-0x30);
    else return(n-0x37);
    /*else if(n=='A') return(0xa);
    else if(n=='B') return(0xb);
    else if(n=='C') return(0xc);
    else if(n=='D') return(0xd);
    else if(n=='E') return(0xe);
    else if(n=='F') return(0xf);
    else return(0);*/
}
void konrfid()
{
}

//Nurjannah=0x10000000*kon(j[3])+0x1000000*kon(j[4])+0x100000*kon(
j[5])+0x10000*kon(j[6])+0x1000*kon(j[7])+0x100*kon(j[8])+0x10*kon(
j[9])+kon(j[10]);

b1=kon(j[3]);b2=kon(j[4]);b3=kon(j[5]);b4=kon(j[6]);b5=kon(j[7]);b
6=kon(j[8]);
b7=kon(j[9]);b8=kon(j[10]);

Ndata=0x10000000*b1+0x1000000*b2+0x100000*b3+0x10000*b4+0x1000*b5+
0x100*b6+0x10*b7+b8;

//Nurjannah=2657107;
konversi=Ndata/1000000000+0x30;k[0]=konversi;

```

```

dataout();
konversi=(Ndata/100000000)%10+0x30;k[1]=konversi;
dataout();
konversi=(Ndata/10000000)%10+0x30;k[2]=konversi;
dataout();
konversi=(Ndata/1000000)%10+0x30;k[3]=konversi;
dataout();
konversi=(Ndata/100000)%10+0x30;k[4]=konversi;
dataout();
konversi=(Ndata/10000)%10+0x30;k[5]=konversi;
dataout();
konversi=(Ndata/1000)%10+0x30;k[6]=konversi;
dataout();
konversi=(Ndata/100)%10+0x30;k[7]=konversi;
dataout();
konversi=(Ndata/10)%10+0x30;k[8]=konversi;
dataout();
konversi=Ndata%10+0x30;k[9]=konversi;
dataout();
t=1;
//pos(2,1);
}
void kirimkom()
{
i=0;
while(i<10)
{
konversi=k[i];
PutChar(konversi);
i++;
}
//i=0;
}
void tampil()
{
konversi=0;a=0;
while(konversi!='%')
{konversi=GetChar();
k[a]=konversi;a++;
}
a=0;konversi=0;
while(konversi!='%')
{konversi=k[a];
if(konversi=='%');
else dataout();
a++;
}
}
//-----
// Program Utama
//-----
void main ()
{
/* Begin of Main
//TMOD=0x10;
initser(0xfd);
initlcd();
cs=0;
//delay(1000);
i=0; tombol=1;buzer1=0;
t=1;//TANDA RFID SEDANG BERLANGSUNG
trfid=0;//TANDA RFID BERAKHIR
tekan=1;
Send_Text("test");
//r=1000;

```



```
while(1)/* Pengulangan Loop tanpa henti
{/* Begin of while
//hapus()

cetak(1,1,"PERPUSTAKAAN ITN");
cetak(2,1," M A L A N G ");

if (t==0)
{
hapus();cetak(1,1,"ID: ");konrfid();
hapus();
cs=1;
kirimkom();EA=0;
cs=1;
pos(1,1);
tampil();
pos(2,1);
tampil();
delay(3000);
cs=0;EA=1;hapus();
}

}/* End of while
}/* End of Main
```

```

#ifndef PENDING_C
#define PENDING_C

void Tunda_lms()
{
    _asm
    mov    r6,#2
xxx1:    mov    r7,#230
xxx2:    djnz   r7,xxx2
         djnz   r6,xxx1
         _endasm;
}

void Tunda_mili(int n)
{
    int i;
    for (i=0; i< n ; i++)
        Tunda_lms();
}

void Tunda_mikro(int n)
{
    int i;
    for (i=0; i< n ; i++)
        ;
}

#endif

```

```

#ifndef lcdku_C
#define lcdku_C

#define e          P1_1
#define rs         P1_0
#define con_lcd    P1_2
#define data_lcd   P1_3

unsigned char doni;

void tunda_lms()
{
    _asm
    mov    R6,#2
Liip:    mov    R7,#230
Loop:    djnz   R7,Loop
         djnz   R6,Liip
         _endasm;
}

void timelms()
{
    int i;
    for (i=0; i<150; i++)
        {;}
}

void delay(int n)
{
    int i;
    for (i=0; i< n ; i++)
        tunda_lms();//timelms();
}

```

```

void delayx(int n)
{
int i;
for (i=0; i< n ; i++)
;
}

void geser()
{
_asm
    mov     a, _doni
    clr     p1.2
    mov     b, #8
a1:
    rrc     a
    mov     p1.3, c
    nop
    nop
    nop
    setb    p1.2
    nop
    nop
    nop
    clr     p1.2
    djnz    b, a1
_endasm;
}

void callout()
{
    rs=0;
    e=1;
    geser();
    delayx(100);
    e=0;
}

void dataout()
{
    rs=1;
    e=1;
    geser();
    delayx(100);
    e=0;
}

void initlcd()
{
    doni=0x1;
    callout();
    doni=0x38;
    callout();
    doni=0x6;
    callout();
    doni=0xc;
    callout();
}

void pos(int i,int n)
{
if(i==1)
    doni=0x80+n-1;
else

```

```

        doni=0xc0+n-1;
        callout();
    }
void cetak(int i,int n,unsigned char *text)
{
    //a=0;
    //n=0;
    pos(i,n);
while(*text)
    {
        doni>(*text++);
        dataout();
        //n=n+1;
        //a=a+1;
    }
}

void hapus()
{
cetak(1,1,"          ");
cetak(2,1,"          ");
pos(1,1);
}

#endif
-----
#ifndef MYSER1_C
#define MYSER1_C

#define Hi          1
#define Lo          0

void Init_Serial()
{
    SCON    = 0x50      ;
    PCON    = 0x80      ;
    TMOD    = TMOD | 0x20 ;
    TL1 = -3; TH1 = -3; TR1 = 1; /* 19200bps with 11.059MHz
crystal */
    ES = 1;
}
void initser(char baud)
{
    EA=1;
    ES=1;
    TMOD    = TMOD | 0x20 ; //timer 1 mode 2 (autoreload)
    PCON=0x0 ; // SMOD =0
    SM0=0;
    SM1=1; //mode 1
    REN=1; //terima dibolehkan
    TH1=baud; //kecepatan 9600 bps (nilai reload)
    TL1=baud; //nilai awal timer
    TR1=1; // timer dijalankan
}

void PutChar(unsigned char Data_Kirim)
{
    ES    = 0 ;
    SBUF  = Data_Kirim;
    while(!TI){;}
    TI    = Lo;
}

```

```

    ES    = 1 ;
}

unsigned char GetChar(void)
{
    while(!RI){;}
    RI    = Lo    ;
    //C++;
    return SBUF    ;
}

void Send_Text(unsigned char *text)
{
    while( *text )           // while not end of text
    {
        PutChar(*text++);   // Write character and
    increment position
    }
    return;
}

void kirimasli(unsigned char *text)
{
    while( *text!='?' )     // while not end of text
    {
        PutChar(*text++);   // Write character and
    increment position
    }
    return;
}

void enter()
{
    //PutChar(0x0D);
    //PutChar(0x0A);
    PutChar(0x0D);
    PutChar(0x0A);
}

#endif

```

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, ExtCtrls, Menus, Buttons, Grids, DBGrids,  
ComCtrls, CPort, DBCtrls, Mask, jpeg, ExtDlgs;
```

```
type
```

```
TForm1 = class(TForm)  
  MainMenu1: TMainMenu;  
  file1: TMenuItem;  
  Cetak1: TMenuItem;  
  DataMahasiswa1: TMenuItem;  
  Masukkanbukubarui: TMenuItem;  
  MasukkanDataMahasiswa1: TMenuItem;  
  Cari1: TMenuItem;  
  Buku1: TMenuItem;  
  Mahasiswa1: TMenuItem;  
  Peminjam1: TMenuItem;  
  Notebook1: TNotebook;  
  TBIDMAHASISWA: TTable;  
  DSIDMAHASISWA: TDataSource;  
  TBIDBUKU: TTable;  
  TBPINJAM: TTable;  
  TBJURUSAN: TTable;  
  DSIDBUKU: TDataSource;  
  DSPINJAM: TDataSource;  
  DSJURUSAN: TDataSource;  
  DBGrid1: TDBGrid;  
  DBGrid2: TDBGrid;  
  GroupBox1: TGroupBox;  
  Label3: TLabel;  
  Cbcari: TComboBox;  
  Ecarimhs: TEdit;  
  Label4: TLabel;  
  Button1: TButton;  
  GroupBox2: TGroupBox;  
  Ecari: TEdit;  
  BitBtn1: TBitBtn;  
  Cbcari: TComboBox;  
  Label5: TLabel;  
  Label6: TLabel;  
  DBNavigator1: TDBNavigator;  
  DBNavigator6: TDBNavigator;  
  ComPort1: TComPort;  
  GroupBox3: TGroupBox;  
  Button2: TButton;
```

DateTimePicker1: TDateTimePicker;
Label1: TLabel;
DBGrid3: TDBGrid;
DBGrid4: TDBGrid;
TBIDMAHASISWA1: TTable;
TBIDBUKU1: TTable;
TBPINJAMI: TTable;
DSIDMAHASISWA1: TDataSource;
DSIDBUKU1: TDataSource;
DSPINJAMI: TDataSource;
DBEdit1: TDBEdit;
Label2: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
DBEdit2: TDBEdit;
DBEdit3: TDBEdit;
DBEdit4: TDBEdit;
DBComboBox1: TDBComboBox;
Label10: TLabel;
Button3: TButton;
Button4: TButton;
Label11: TLabel;
DBEdit5: TDBEdit;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit9: TDBEdit;
Button5: TButton;
Button6: TButton;
DBComboBox2: TDBComboBox;
Edit1: TEdit;
Peminjaman1: TMenuItem;
DBEdit8: TDBEdit;
Label16: TLabel;
DBEdit10: TDBEdit;
DBEdit11: TDBEdit;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
DBGrid5: TDBGrid;
Tbpinjams1: TTable;
Dspinjams1: TDataSource;
tes1: TMenuItem;
Panel1: TPanel;
DBLookupComboBox1: TDBLookupComboBox;

Label20: TLabel;
Button9: TButton;
Button10: TButton;
Button11: TButton;
Denda1: TMenuItem;
DBEdit12: TDBEdit;
RichEdit1: TRichEdit;
RichEdit2: TRichEdit;
Thdenda: TTable;
DSDenda: TDataSource;
Button13: TButton;
DBEdit14: TDBEdit;
Button14: TButton;
Label21: TLabel;
DBEdit13: TDBEdit;
Label22: TLabel;
Image1: TImage;
Bevel1: TBevel;
Bevel2: TBevel;
Exit1: TMenuItem;
Image2: TImage;
Image3: TImage;
Panel2: TPanel;
Panel3: TPanel;
Panel5: TPanel;
Panel6: TPanel;
Image4: TImage;
Button12: TButton;
OpenPictureDialog1: TOpenPictureDialog;
SavePictureDialog1: TSavePictureDialog;
Image5: TImage;
Button15: TButton;
Panel7: TPanel;
Panel9: TPanel;
Image7: TImage;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Image6: TImage;
Bevel3: TBevel;
Exit2: TMenuItem;
Print: TButton;
Peminjaman2: TMenuItem;
RichEdit3: TRichEdit;
procedure BitBtn1Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure MasukanDataMahasiswa1Click(Sender: TObject);

```
procedure Button4Click(Sender: TObject);
procedure Masukanbukubaru1Click(Sender: TObject);
procedure Buku1Click(Sender: TObject);
procedure Mahasiswa1Click(Sender: TObject);
procedure Peminjam1Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure kirimdata(data:string);
procedure Button7Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Button8Click(Sender: TObject);
procedure tes1Click(Sender: TObject);
procedure Peminjaman1Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
procedure DataMahasiswa1Click(Sender: TObject);
procedure Button12Click(Sender: TObject);
procedure Button13Click(Sender: TObject);
procedure Button14Click(Sender: TObject);
procedure Denda1Click(Sender: TObject);
procedure Exit1Click(Sender: TObject);
procedure Notebook1PageChanged(Sender: TObject);
procedure Button15Click(Sender: TObject);
procedure Exit2Click(Sender: TObject);
procedure PrintClick(Sender: TObject);
procedure Peminjaman2Click(Sender: TObject);
```

```
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  Form1: TForm1;
  denda:integer;
  GAMBAR:string;
```

```
implementation
```

```
{ $R *.dfm }
```

```
procedure TForm1.kirimdata(data:string);
var
  data_kirim:string;
  i:integer;
begin
```

```

i:=0;
repeat
  i:=i+1;
  data_kirim:=copy(data,i,1);
  ComPort1.WriteStr(data_kirim);
  Sleep(10);
until i=length(data);
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
  IF TBIDBUKU.RecordCount=0 THEN
    begin
      ShowMessage('DATA TIDAK ADA');
    end
  else
    begin
      if Cbcari.Text='JUDUL' then
        begin
          TBIDBUKU.Filtered:=FALSE;
          TBIDBUKU.Filter:='judul='+QuotedStr(Ecari.Text);
          TBIDBUKU.Filtered:=true;
          if TBIDBUKU.RecordCount=0 then ShowMessage('DATA TIDAK ADA')
          end
        ELSE
          BEGIN
            TBIDBUKU.Filtered:=FALSE;
            IF NOT TBIDBUKU.FindKey([Ecari.Text]) THEN ShowMessage('DATA
TIDAK ADA');
            END;
          end;
          Ecari.Text:='';
        end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  IF TBIDMAHASISWA.RecordCount=0 THEN
    begin
      ShowMessage('DATA TIDAK ADA');
    end
  else
    begin
      if Cbcarimhs.Text='NAMA' then
        begin
          TBIDMAHASISWA.Filtered:=FALSE;
          TBIDMAHASISWA.Filter:='judul='+QuotedStr(Ecarimhs.Text);
          TBIDMAHASISWA.Filtered:=true;
          if TBIDMAHASISWA.RecordCount=0 then ShowMessage('DATA TIDAK
ADA')

```

```

    end
ELSE
    BEGIN
        TBIDMAHASISWA.Filtered:=FALSE;
        IF NOT TBIDMAHASISWA.FindKey([Ecarimhs.Text]) THEN
ShowMessage('DATA TIDAK ADA');
        END;
    end;
    Ecari.Text:=";

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
TBIDMAHASISWA1.Edit;
Image4.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+
TBIDMAHASISWA1.fieldbyname('id').AsString+'.jpg');
if gambar<>" then TBIDMAHASISWA1.FieldByName('foto').AsString:='ada';
TBIDMAHASISWA1.Post;
TBIDMAHASISWA.Refresh;
Notebook1.ActivePage:='cari mahasiswa';
GAMBAR:=";
end;

procedure TForm1.MasukanDataMahasiswa1Click(Sender: TObject);
begin
TBIDMAHASISWA1.Append;
Notebook1.ActivePage:='masuk mahasiswa';
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
TBIDMAHASISWA.Cancel;
Notebook1.ActivePage:='cari mahasiswa';
end;

procedure TForm1.Masukanbukubaru1Click(Sender: TObject);
begin
TBIDBUKU1.Append;
Notebook1.ActivePage:='masuk buku';
end;

procedure TForm1.Buku1Click(Sender: TObject);
begin
Notebook1.ActivePage:='cari buku';
end;

procedure TForm1.Mahasiswa1Click(Sender: TObject);
begin

```

```

Notebook1.ActivePage:='cari mahasiswa';
end;

procedure TForm1.Peminjam1Click(Sender: TObject);
begin
Notebook1.ActivePage:='cari peminjam';
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
TBIDBUKU1.Edit;
TBIDBUKU1.Post;
TBIDBUKU1.Refresh;
Notebook1.ActivePage:='cari buku';
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
TBIDBUKU1.Cancel;
Notebook1.ActivePage:='cari buku';
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
ComPort1.ShowSetupDialog;
ComPort1.Open;
Notebook1.ActivePage:='cari mahasiswa';
Edit1.Text:='';
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
Notebook1.ActivePage:='awal';
denda:=0;
Image5.Visible:=false;
Image4.Visible:=false;
GAMBAR:='';
end;

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var data,DATA_KIRIM:string;
begin
ComPort1.ReadStr(data,count);
Edit1.Text:=Edit1.Text+data;
//ShowMessage(Edit1.Text);
if length(Edit1.Text)=10 then
begin
// ShowMessage(Edit1.Text);
if TBIDMAHASISWA1.FindKey([Edit1.Text]) then

```

```

begin
    kirimdata(TBIDMAHASISWA1.fieldbyname('ID').AsString+'%');
    Notebook1.ActivePage:='peminjaman';
    Button15.Click;
    if Tbpinjamsl.fieldbyname('kembali').IsNull and
(Tbpinjamsl.fieldbyname('pinjam').AsString<>") then
        begin
            DATA_KIRIM:='PENGEMBALIAN%';
        end
    else begin
        DATA_KIRIM:='PEMINJAMAN%';
    end;
    kirimdata(DATA_KIRIM);
end
else
begin
//ShowMessage('tes');
    kirimdata('MAAF ID BLM ADA%');
    kirimdata('REGISTRASI DULU%');
    end;
    Edit1.Text:='';
end
else if length(Edit1.Text)>10 then Edit1.Text:='';
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
    kirimdata('MAAF ID BLM ADA%');
    kirimdata('REGISTRASI DULU%');
end;

procedure TForm1.tes1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='tes';
end;

procedure TForm1.Peminjaman1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='peminjaman';
end;

procedure TForm1.Button9Click(Sender: TObject);
begin
    IF Button9.Caption='PINJAM' Then
        begin
            Tbpinjamsl.Append;
            Button9.Caption:='SIMPAN';
            DBLookupComboBox1.Enabled:=TRUE;
        end
end

```

```

else begin
    Tbpinjamsl.FieldByName('no').AsInteger:=TBPINJAM1.RecordCount+1;

    Tbpinjamsl.FieldByName('JUDUL').AsString:=TBIDBUKU1.FieldByName('JU
DUL').AsString;
    Tbpinjamsl.FieldByName('PINJAM').AsDateTime:=DATE;
    Tbpinjamsl.Post;
    Button9.Caption:='PINJAM';
    TBPINJAM.Refresh;
    TBPINJAM1.Refresh;
    DBLookupComboBox1.Enabled:=FALSE;
end;
end;

procedure TForm1.Button10Click(Sender: TObject);
var
    tgl,tgl1:TDate;
    i:integer;
begin
    tgl1:=Tbpinjamsl.FieldByName('pinjam').AsDateTime;
    Tbpinjamsl.Edit;
    Tbpinjamsl.FieldByName('KEMBALI').AsDateTime:=DATE;
    Tbpinjamsl.FieldByName('Keterangan').AsString:='kembali';
    Tbpinjamsl.Post;
    //ShowMessage(DateToStr(date-7));
    if tgl1<(date-Tbdenda.fieldbyname('DURASI').AsInteger) then
        begin

            i:=0;
            tgl:=Date-Tbdenda.fieldbyname('DURASI').AsInteger;
            //ShowMessage(IntToStr(denda)+' '+IntToStr(i));
            repeat
                i:=i+1;
                tgl:=tgl-1;
            //ShowMessage(DateToStr(tgl));
            until tgl1=tgl;
            //ShowMessage(IntToStr(denda)+' '+IntToStr(i));
            denda:=denda+i*Tbdenda.fieldbyname('denda').AsInteger;
            //ShowMessage('denda');
            ShowMessage('Denda '+TBIDMAHASISWA1.fieldbyname('nama').AsString+'
sebesar '+IntToStr(denda));
            end;
            TBPINJAM.Refresh;
            TBPINJAM1.Refresh;
        end;

procedure TForm1.Button11Click(Sender: TObject);
begin
    Tbpinjamsl.Cancel;

```

```

Notebook1.ActivePage:='cari peminjam';
end;

procedure TForm1.DataMahasiswa1Click(Sender: TObject);
VAR
    no:integer;
begin
    Notebook1.ActivePage:='print mhs';
    no:=1;
    TBIDMAHASISWA1.First;

    RichEdit2.Clear;
    RichEdit2.Lines.Add('DATA MAHASISWA');
    RichEdit2.Lines.Add("");

    REPEAT
    RichEdit2.Lines.Add(IntToStr(no)+'
'+TBIDMAHASISWA1.fieldbyname('id').AsString+' '+
TBIDMAHASISWA1.fieldbyname('NIM').AsString+'
'+TBIDMAHASISWA1.fieldbyname('NAMA').AsString+' '+
TBIDMAHASISWA1.fieldbyname('ALAMAT').AsString+'
'+TBIDMAHASISWA1.fieldbyname('JURUSAN').AsString);
    TBIDMAHASISWA1.Next;
    no:=no+1;
    UNTIL TBIDMAHASISWA1.Eof;
    RichEdit2.Print("");
    //RichEdit2.Clear;
end;

procedure TForm1.Button12Click(Sender: TObject);
begin
    if OpenPictureDialog1.Execute
    then begin
        Image4.Visible:=true;
        Image4.Picture.LoadFromFile (OpenPictureDialog1.FileName);
        // ShowMessage(OpenPictureDialog1.FileName);
        gambar:='ada';
    end;
end;

procedure TForm1.Button13Click(Sender: TObject);
// set ulang denda
begin
    denda:=0;
end;

procedure TForm1.Button14Click(Sender: TObject);

```

```

begin
Tbdenda.Edit;
Tbdenda.Post;

end;

procedure TForm1.Denda1Click(Sender: TObject);
begin
Notebook1.ActivePage:='DENDA';
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
ComPort1.ShowSetupDialog;
ComPort1.Open;

end;

procedure TForm1.Notebook1PageChanged(Sender: TObject);
begin
Image4.Visible:=false;
end;

procedure TForm1.Button15Click(Sender: TObject);
begin
IF TBIDMAHASISWA1.FieldByName('foto').IsNull then
  BEGIN
  //ShowMessage('tidak ada gambar');
  Image5.Visible:=FALSE;
  END
else begin
  Image5.Visible:=true;

Image5.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\
'+TBIDMAHASISWA1.fieldbyname('id').AsString+'.jpg');
  end;
end;

procedure TForm1.Exit2Click(Sender: TObject);
begin
close;
end;

procedure TForm1.PrintClick(Sender: TObject);
VAR
  no:integer;
begin
RichF.dit1.Lines.Add(TBIDMAHASISWA1.fieldbyname('id').AsString+'
'+TBIDMAHASISWA1.fieldbyname('nama').AsString);

```



```

RichEdit1.Lines.Add("");
RichEdit1.Lines.Add('PEMINJAMAN');
RichEdit1.Lines.Add("");
no:=1;
Tbpinjamsl.First;
REPEAT
RichEdit1.Lines.Add(IntToStr(no)+' '+Tbpinjamsl.fieldbyname('id
buku').AsString+' '+
Tbpinjamsl.fieldbyname('judul').AsString+'
'+Tbpinjamsl.fieldbyname('pinjam').AsString+' '+
DateToStr(Date+StrToInt(DBEdit13.Text)));
Tbpinjamsl.Next;
no:=no+1;
UNTIL Tbpinjamsl.Eof;
RichEdit1.Print("");
//RichEdit1.Clear;
end;
procedure TForm1.Peminjaman2Click(Sender: TObject);
VAR
    no:integer;
begin

no:=1;
TBPINJAM.First;
RichEdit3.Clear;
RichEdit3.Lines.Add(' ALL ');
RichEdit3.Lines.Add('Transaksi Peminjaman');
RichEdit3.Lines.Add("");
RichEdit3.Lines.Add("");

REPEAT
RichEdit3.Lines.Add(IntToStr(no)+'
'+TBPINJAM.fieldbyname('ID').AsString+' '+
TBPINJAM.fieldbyname('ID BUKU').AsString+'
'+TBPINJAM.fieldbyname('JUDUL').AsString+' '+
TBPINJAM.fieldbyname('PINJAM').AsString+'
'+TBPINJAM.fieldbyname('KEMBALI').AsString+' '+
TBPINJAM.FieldName('Keterangan').AsString);
TBPINJAM.Next;
no:=no+1;
UNTIL TBPINJAM.Eof;
RichEdit3.Print("");
RichEdit3.Clear;
end;

end.

```

Dot Matrix Liquid Crystal Display Modules

CHARACTER TYPE

• FEATURES :

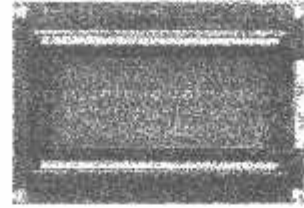
- Slim, light weight and low power consumption
- High contrast and wide viewing angle
- Built-in controller for easy interfacing
- LCD modules with built-in EL or LED backlight



M1641



L1642



L1614



M1632



L1652



L2012

• SPECIFICATIONS :

	Standard products			Products of optional specification				
Character Point (character x lines)	16 x 1	16 x 2	16 x 2	16 x 2	16 x 4	20 x 2		
Model	M1641	M1632	L1642	L1652	L1614	L2012		
Effective backlight	M1641DAS M1641DWS	M1632DAS M1632DWS	L1642DL000S L16422LJ000S	L1652DL000S L16522LJ200S	L1614DL000S L16142LJ000S	L2012DL000S L20122LJ000S		
Controller	M16417DYS	M16327DYS	L16429L3000S	L16529L3000S	L16148L3000S	L20128L3000S		
Alternative (wide temp.)	M16410CS	M16320CS	L16420DL000S	L16520DL200S	L16140DL000S	L201210L000S		
D backlight (wide temp.)	M16417JYS	M16327JYS	L16429L1000S	L16529L1000S	L16149L1000S	L20129L1000S		
Character font	3x5 dot matrix	3x5 dot matrix	3x5 dot matrix	3x5 dot matrix	3x5 dot matrix	3x5 dot matrix		
Module size (mm)	Reflective	80.0 x 35.0 x 11.3	85.0 x 30.0 x 10.1	80.0 x 35.0 x 11.3	122.0 x 44.0 x 11.3	87.0 x 80.0 x 11.5	115.0 x 37.0 x 11.3	
	EL Backlight	80.0 x 35.0 x 11.3	85.0 x 30.0 x 10.1	80.0 x 35.0 x 11.3	122.0 x 44.0 x 11.3	87.0 x 80.0 x 11.5	115.0 x 37.0 x 11.3	
Viewing angle (mm)	LED Backlight	80.0 x 35.0 x 15.8	80.0 x 30.0 x 15.8	80.0 x 35.0 x 15.8	122.0 x 44.0 x 15.8	87.0 x 80.0 x 15.8	115.0 x 37.0 x 15.8	
	Wiring area (mm)	54.5 x 13.5	52.0 x 16.0	54.5 x 13.5	99.0 x 24.0	61.5 x 25.2	83.0 x 18.6	
Label size (mm)	0.55 x 0.75						0.50 x 0.55	
Operating voltage (VDS-VSS)V	+5 V						-5 V	
Current consumption (mA)	1.5						2.0	
Capacitance (pF)	0.2						0.3	
Wiring method (duty)	1/15						1/15	
IC-IC	KS0066 or equivalent						MSM5839 or equivalent	
Operating temperature (°C)	normal temp.	0 to +50						0 to +50
	wide temp. *2	-20 to +70						-20 to +70
Storage temperature (°C)	normal temp.	-20 to +60						-20 to +60
	wide temp.	-30 to +80						-30 to +80
Viewing angle (°)	Reflective	25						40
	EL backlight	30						45
	LED Backlight	35						45
Power supply (V)	Model	5S						5A
	Power supply (V)	+5.0						-5.0
Current consumption (mA)	Model	5S						5A
	Power supply (V)	+5.0						-5.0
Forward current consumption (mA)	Model	10						45
	Power supply (V)	+5.0						-5.0
Forward input voltage (VDS)	Model	100						154
	Power supply (V)	+4.1						-4.1

connecting cursor H : Horizontal V : Vertical T : Thickness (max)

*1: External temperature compensation

*2: Using EL backlight

*3: Based on normal temperature range

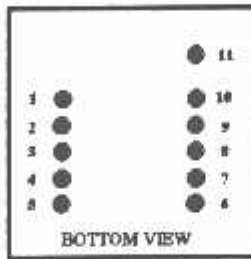
In policy is one of continuous improvements we reserve the right to change the specifications for the products in the catalogue without notice.

ID SERIES DATASHEET Feb 10 , 2004

ID-2 / ID-12 / ID-20

The ID2, ID12 and ID20 are similar to the ID0, ID10 and ID15 MK(ii) series devices, but they have extra pins which allow Magnetic Emulation output to be included in the functionality. The ID-12 and ID-20 come with internal antennas, and have read ranges of 12+ cm and 16+ cm, respectively. With an external antenna, the ID-2 can deliver read ranges of up to 25 cm. All three readers support ASCII, Wiegand26 and Magnetic ABA Track2 data formats.

ID2 / ID12 / ID20 PIN-OUT



1. GND
2. RES (Reset Bar)
3. ANT (Antenna)
4. ANT (Antenna)
5. CP
6. Future
7. +/- (Format Selector)
8. D1 (Data Pin 1)
9. D0 (Data Pin 0)
10. LED (LED / Beeper)
11. +5V



Operational and Physical Characteristics

Parameters	ID-2	ID-12	ID-20
Read Range	N/A (no internal antenna)	12+ cm	16+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64	Manchester 64-bit, modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 65mA nominal
I/O Output Current	+/-200mA PK	-	-
Voltage Supply Range	+4.6V through +5.4V	+4.6V through +5.4V	+4.6V through +5.4V

Pin Description & Output Data Formats

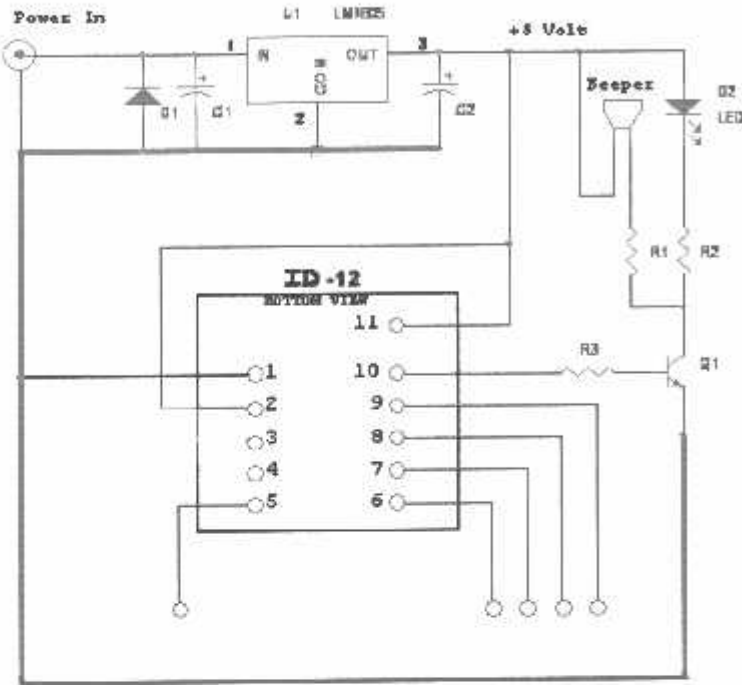
Pin No.	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	No function	Card Present	No function
Pin 6	Future	Future	Future	Future
Pin 7	Format Selector (+/-)	Strap to GND	Strap to Pin 10	Strap to +5V
Pin 8	Data 1	CMOS	Clock	One Output
Pin 9	Data 0	TTL Data (inverted)	Data	Zero Output
Pin 10	3.1 kHz Logic	Beeper / LED	Beeper / LED	Beeper / LED
Pin 11	DC Voltage Supply	+5V	+5V	+5V

ID Innovations

Advanced Digital Reader Technology

---Better by Design

Circuit Diagram for the ID-12



COMPONENT LIST

- R1 = 100R
- R2 = 1K
- R3 = 1K
- C1 = 100uF 16V
- C2 = 100uF 10V
- Beeper = 2.7-3.5KHz 100R
- D1 = 1N4001
- D2 = GREEN LED
- U1 = LM7805
- Q1 = UTC8050 (NPN)
- ID2 = ID Innovations ID2

* Please Note the ID2 has an internal tuning capacitor of 1.5nF and this makes the total tuning capacity = 2.5nF

The 3.1KHz Beeper Logic is centered for most Beepers in range 2.7-3.5KHz

PC-Link USBer

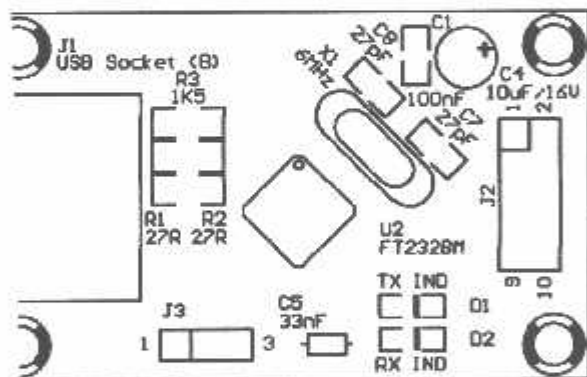
USB - UART Converter

USBer merupakan sebuah modul development tools untuk USB ke tipe FT232BM. Modul ini biasanya digunakan untuk aplikasi seperti USB ↔ UART TTL, USB ↔ UART RS-232, USB ↔ UART RS-422, USB ↔ UART RS-485, antarmuka mikrokontroler dengan USB, antarmuka modem UART dengan USB, dll.

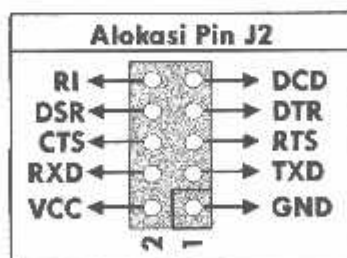
Spesifikasi

- Tegangan kerja 4,4 – 5,25 Volt DC.
- Tersedia 2 LED untuk indikator Tx dan Rx data pada komunikasi serial.
- Sinyal antarmuka yang mendukung komunikasi dengan modem UART.
- Mendukung format UART dengan 7 / 8 bit data, 1 / 2 Stop bit, dan Odd / Even / Mark / Space / No parity.
- Memiliki baudrate 3Mbps (TTL), 1Mbps (RS-232), 3Mbps (RS-422 / RS-485).
- Pin sinyal kontrol (arah) untuk komunikasi RS-485 yang bekerja secara otomatis.
- Kompatibel dengan USB 1.1 dan USB 2.0.
- Kompatibel dengan UHCI / OHCI / EHCI host controller.
- Mendukung kondisi USB Suspend / Resume lewat pin SLEEP dan RI.
- Memiliki output dengan level TTL 5 Volt.
- Memiliki EEPROM eksternal untuk menyimpan data PID, VID, nomor serial, dan deskripsi produk. Pengisian datanya melalui USB.
- Virtual COM port driver (VCP) dan D2XX (USB Direct Drivers + DLL S/W Interface) untuk Windows® 98, 98SE, ME, 2000, dan XP.
- Modul menggunakan konfigurasi daya Self Powered.

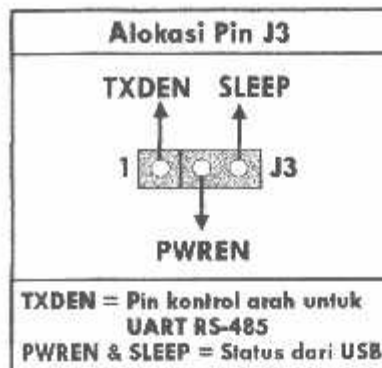
Letak Komponen



muka dari/ke modul USBer menggunakan level TTL dan langsung ke header dengan label J2.



Modul USBer menjadi sebuah konverter dari USB ke RS-485. Pin kontrol TXDEN, pin ini terdapat pada J3.



Penjelasan lebih lengkap meliputi fungsi pin, timing diagram, dan lain-lain terdapat pada FT232BM.PDF.

Penjelasan rutin pada library terdapat pada D2XX Programming Guide.PDF.

Isi CD

1. User Guide untuk instalasi driver dan penjelasan rutin library.
2. Driver untuk VCP dan D2XX.
3. MProg 2.3 untuk memprogram AT93C46 USBer.
4. Manual USBer dan gambar rangkaian LED.
5. Program Tester USBerTester.EXE.

Prosedur Instalasi dan Testing

Untuk petunjuk instalasi software dapat dilihat pada file **win98.pdf** (untuk OS Windows® 98, 98SE), **win200.pdf** (untuk OS windows 2000), **winXP.pdf** (untuk windows XP). Semua file ini terdapat pada CD yang terletak pada direktori **User Guide**. Pada file – file ini akan dijelaskan tentang prosedur instalasi untuk driver VCP dan D2XX beserta cara penyelesaian apabila di dalam melakukan instalasi terdapat kesalahan. Sedangkan untuk mengetahui cara pemrograman menggunakan driver D2XX, dapat dilihat pada file **D2XX Programming.pdf** yang juga terletak di direktori **User Guide**. Untuk mengetahui apakah modul USBer bekerja atau tidak, dapat menggunakan program **USBerTester.EXE** yang terdapat pada CD. Program ini akan melakukan 2 testing, yaitu bagian input dengan menggunakan push button, dan bagian output menggunakan LED. Untuk rangkaian lengkap LED dan push button dapat dilihat pada file **rangkain.bmp**. Sedangkan untuk proses pengujiannya sebagai berikut :

1. Lakukan proses instalasi untuk driver **D2XX** sampai selesai.
2. Jalankan program **USBerTester.EXE** pada CD-ROM.
3. Klik pada tombol **Connect**.
4. Hubungkan modul USBer pada rangkaian LED. lalu klik pada tombol **Testing Output (Start)**. Hasil dari testing ini adalah nyala LED yang bergeser satu persatu.
5. Klik pada tombol **Testing Output (Stop)**. Kemudian ganti rangkaian LED dengan rangkaian push button, lalu klik pada tombol **Testing Input (Start)**. Hasil dari prosedur testing yang kedua adalah nyala LED pada display software **USBerTester** sesuai dengan tombol yang ditekan.
6. Untuk menghentikan prosedur testing yang kedua klik pada tombol **Testing Input (Stop)**.

Trademark & Copyright

Windows is a registered trademark of Microsoft Corp.

Terima Kasih atas kepercayaan Anda menggunakan produk kami, bila ada kesulitan, pertanyaan atau saran mengenai produk ini silahkan menghubungi technical support kami :

support@innovativeelectronics.com



Future Technology Devices International Ltd.

FT232R USB UART I.C.

Incorporating Clock Generator Output
and FTDIChip-ID™ Security Dongle

FT232R is the latest device to be added to FTDI's range of USB UART interface Integrated Circuit Devices. The FT232R is a USB to serial UART interface with optional clock generator output, and the new FTDIChip-ID™ security dongle feature. In addition, asynchronous and synchronous bit bang interface modes are available. USB to serial interfacing using the FT232R have been further simplified by fully integrating the external EEPROM, clock circuit and resistors onto the device.

FT232R adds two new functions compared with its predecessors, effectively making it a "3-in-1" chip for some application areas. The internally generated clock (6MHz, 12MHz, 24MHz, and 48MHz) can be brought out of the device and used to drive a microcontroller or external logic. A unique number (the FTDIChip-ID™) is burnt into the device during manufacture and is readable over USB, thus forming the basis of a security dongle which can be used to protect customer application software from being copied.

FT232R is available in Pb-free (RoHS compliant) compact 28-Lead SSOP and QFN 32 packages

Features

1.1 Hardware Features

Single chip USB to asynchronous serial data transfer interface.

Entire USB protocol handled on the chip - No USB-specific firmware programming required.
UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity.
Fully assisted hardware or X-On / X-Off software handshaking.

Data transfer rates from 300 baud to 3 Megabaud (RS422 / RS485 and at TTL levels) and 300 baud to 1 Megabaud (RS232).

256 byte receive buffer and 128 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.

FTDI's royalty-free VCP and D2XX drivers eliminate the requirement for USB driver development in most cases.

In-built support for event characters and line break condition.

New USB FTDIChip-ID™ feature.

New configurable CBUS I/O pins.

Auto transmit buffer control for RS485 applications.
Transmit and receive LED drive signals.

New 48MHz, 24MHz, 12MHz, and 6MHz clock output signal Options for driving external MCU or FPGA.

FIFO receive and transmit buffers for high data throughput.

Adjustable receive buffer timeout.

Synchronous and asynchronous bit bang mode

interface options with RD# and WR# strobes.

New CBUS bit bang mode option.

- Integrated 1024 Bit internal EEPROM for storing USB VID, PID, serial number and product description strings, and CBUS I/O configuration.
- Device supplied preprogrammed with unique USB serial number.
- Support for USB suspend and resume.
- Support for bus powered, self powered, and high-power bus powered USB configurations.
- Integrated 3.3V level converter for USB I/O .
- Integrated level converter on UART and CBUS for interfacing to 5V - 1.8V Logic.
- True 5V / 3.3V / 2.8V / 1.8V CMOS drive output and TTL input.
- High I/O pin output drive option.
- Integrated USB resistors.
- Integrated power-on-reset circuit.
- Fully integrated clock - no external crystal, oscillator, or resonator required.
- Fully integrated AVCC supply filtering - No separate AVCC pin and no external R-C filter required.
- UART signal inversion option.
- USB bulk transfer mode.
- 3.3V to 5.25V Single Supply Operation.
- Low operating and USB suspend current.
- Low USB bandwidth consumption.
- UHCI / OHCI / EHCI host controller compatible
- USB 2.0 Full Speed compatible.
- -40°C to 85°C extended operating temperature range.
- Available in compact Pb-free 28 Pin SSOP and QFN-32 packages (both RoHS compliant).

1.2 Driver Support

Royalty-Free VIRTUAL COM PORT

*) DRIVERS for...

Windows 98, 98SE, ME, 2000, Server 2003, XP,
Windows Vista / Longhorn*
Windows XP 64-bit.*
Windows XP Embedded.
Windows CE.NET 4.2 & 5.0
MAC OS 8 / 9, OS-X
Linux 2.4 and greater

Royalty-Free D2XX Direct Drivers (USB Drivers + DLL S/W Interface)

- Windows 98, 98SE, ME, 2000, Server 2003, XP,
- Windows Vista / Longhorn*
- Windows XP 64-bit.*
- Windows XP Embedded.
- Windows CE.NET 4.2 & 5.0
- Linux 2.4 and greater

drivers listed above are all available to download for free from the FTDI website. Various 3rd Party Drivers are available for various other operating systems - see the [FTDI website](#) for details.

* Under Development. Contact FTDI for availability

1.3 Typical Applications

USB to RS232 / RS422 / RS485 Converters

Upgrading Legacy Peripherals to USB

Cellular and Cordless Phone USB data transfer cables and interfaces

Interfacing MCU / PLD / FPGA based designs to USB

USB Audio and Low Bandwidth Video data transfer

PDA to USB data transfer

USB Smart Card Readers

USB Instrumentation

- USB Industrial Control
- USB MP3 Player Interface
- USB FLASH Card Reader / Writers
- Set Top Box PC - USB interface
- USB Digital Camera Interface
- USB Hardware Modems
- USB Wireless Modems
- USB Bar Code Readers
- USB Software / Hardware Encryption Dongles

Enhancements

2.1 Device Enhancements and Key Features

This section summarises the enhancements and the key features of the FT232R device. For further details, consult the [device pin-out description](#) and [functional description](#) sections.

Integrated Clock Circuit - Previous generations of FTDI's USB UART devices required an external crystal or ceramic resonator. The clock circuit has now been integrated onto the device meaning that no crystal or ceramic resonator is required. However, if required, an external 12MHz crystal can be used as the clock source.

Integrated EEPROM - Previous generations of FTDI's USB UART devices required an external EEPROM if the device were to use USB Vendor ID (VID), Product ID (PID), serial number and product description strings other than default values in the device itself. This external EEPROM has now been integrated onto the FT232R chip meaning that all designs have the option to change the product description strings. A user area of the internal EEPROM is available for storing additional data. The internal EEPROM is programmable in circuit, over USB without any additional programming requirement.

Pre-programmed EEPROM - The FT232R is supplied with its internal EEPROM preprogrammed with a serial number which is unique to each individual device. This, in most cases, will remove the need to program the device EEPROM.

Integrated USB Resistors - Previous generations of FTDI's USB UART devices required two external series resistors on the USB DP and USB DM lines, and a 1.5 k Ω pull up resistor on USB DP. These three resistors have now been integrated onto the device.

Integrated AVCC Filtering - Previous generations of FTDI's USB UART devices had a separate AVCC pin - the supply to the internal PLL. This pin required an external R-C filter. The separate AVCC pin is now connected internally to VCC, and the filter has now been integrated onto the chip.

Reduced External Components - Integration of the crystal, EEPROM, USB resistors, and AVCC filter will substantially reduce the bill of materials cost for USB interface designs using the FT232R compared to its FT232BM predecessor.

Transmit and Receive Buffer Smoothing - The FT232R's 256 byte receive buffer and 128 byte transmit buffer utilise buffer smoothing technology to allow for high data throughput.

Configurable CBUS I/O Pin Options - There are now 5 configurable Control Bus (CBUS) lines. Options are **TXDEN** - transmit enable for RS485 designs, **PWREN#** - Power control for high power, bus powered designs, **TXLED#** - for pulsing an LED upon transmission of data, **RXLED#** - for pulsing an LED upon receiving data, **TX&RXLED#** - which pulse an LED upon transmission OR reception of data, **SLEEP#** - indicates that the device is going into USB suspend mode, **CLK48 / CLK24 / CLK12 / CLK6** - 48MHz, 24MHz, 12MHz, and 6MHz clock output signal options. There is also the option to bring out bit bang mode read and write strobes (see below). The CBUS lines can be configured with any one of these output options by setting bits in the internal EEPROM. The device is supplied with the most commonly used pin definitions preprogrammed - see [Section 10](#) for details.

Enhanced Asynchronous Bit Bang Mode with RD# and WR# Strobes - The FT232R supports FTDI's BM chip suspend mode. In bit bang mode, the eight UART lines can be switched from the regular interface mode to an 8-bit general purpose I/O port. Data packets can be sent to the device and they will be sequentially sent to the interface state controlled by an internal timer (equivalent to the baud rate prescaler). With the FT232R device this mode has been enhanced so that the internal RD# and WR# strobes are now brought out of the device which can be used with external logic to be clocked by accesses to the bit bang I/O bus. This option will be described more fully in a separate application note.

Synchronous Bit Bang Mode - Synchronous bit bang mode differs from asynchronous bit bang mode in that the device pins are only read when the device is written to. Thus making it easier for the controlling program to measure response to an output stimulus as the data returned is synchronous to the output data. The feature was previously only available on FTDI's FT2232C device. This option will be described more fully in a separate application note.

Bit Bang Mode - This mode allows four of the CBUS pins to be individually configured as GPIO pins, similar to asynchronous bit bang mode. It is possible to use this mode while the UART interface is being used, thus providing four general purpose I/O pins which are available during normal operation. An application note describing this feature is available separately from the [FTDI website](#).

Lower Supply Voltage - Previous generations of the chip required 5V supply on the VCC pin. The FT232R will work with a Vcc supply in the range 3.3V - 5.25V. Bus powered designs would still take their supply from the 5V on the USB connector, but for self powered designs where only 3.3V is available and there is no 5V supply there is no longer any need for an additional external regulator.

Integrated Level Converter on UART Interface and Control Signals - VCCIO pin supply can be from 1.8V to 5V. Connecting the VCCIO pin to 1.8V, 2.8V, or 3.3V allows the device to directly interface to 1.8V, 2.8V or 3.3V and other logic families without the need for external level converter I.C. devices.

3.3V / 2.8V / 1.8V Logic Interface - The FT232R provides *true* CMOS Drive Outputs and TTL level Inputs.

Integrated Power-On-Reset (POR) Circuit- The device incorporates an internal POR function. A RESET# pin is available in order to allow external logic to reset the FT232R where required. However, for many applications the RESET# pin can be left unconnected, or pulled up to VCCIO.

Lower Operating and Suspend Current - The device operating supply current has been further reduced to 15mA, and the suspend current has been reduced to around 70µA. This allows greater margin for peripheral designs to meet the USB suspend current limit of 500µA.

Low USB Bandwidth Consumption - The operation of the USB interface to the FT232R has been designed to use as little as possible of the total USB bandwidth available from the USB host controller.

High Output Drive Option - The UART interface and CBUS I/O pins can be made to drive out at three times the standard signal drive level thus allowing multiple devices to be driven, or devices that require a greater signal drive strength to be interfaced to the FT232R. This option is enabled in the internal EEPROM.

Power Management Control for USB Bus Powered, High Current Designs- The PWREN# signal can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. An option in the internal EEPROM makes the device gently pull down on its UART interface lines when the power is cut off (PWREN# is high). In this mode any residual voltage on external circuitry is bled to GND when power is removed, thus ensuring that external circuitry controlled by PWREN# resets reliably when power is restored.

UART Pin Signal Inversion - The sense of each of the eight UART signals can be individually inverted by setting bits in the internal EEPROM. Thus, CTS# (active low) can be changed to CTS (active high), or TXD can be inverted to TXD#.

FTDI Chip-ID™ - Each FT232R is assigned a unique number which is burnt into the device at manufacture. This ID number cannot be reprogrammed by product manufacturers or end-users. This allows the possibility of using FT232R based dongles for software licensing. Further to this, a renewable license scheme can be implemented based on the Chip-ID™ number when encrypted with other information. This encrypted number can be stored in the user area of the FT232R internal EEPROM, and can be decrypted, then compared with the protected FTDIChip-ID™ to verify if a license is valid. Web based applications can be used to maintain product licensing this way. An application note describing this feature is available separately from the [FTDI website](#).

Improved EMI Performance - The reduced operating current and improved on-chip VCC decoupling significantly reduces the ease of PCB design requirements in order to meet FCC, CE and other EMI related specifications.

Programmable Receive Buffer Timeout - The receive buffer timeout is used to flush remaining data from the receive buffer. This time defaults to 16ms, but is programmable over USB in 1ms increments from 1ms to 255ms, thus allowing the device to be optimised for protocols that require fast response times from short data packets.

Extended Operating Temperature Range - The FT232R operates over an extended temperature range of -40° to 85°C thus allowing the device to be used in automotive and industrial applications.

Package Options - The FT232R is available in two packages - a compact 28 pin SSOP (FT232RL) and an ultra compact 5mm x 5mm pinless QFN-32 package (FT232RQ). Both packages are lead (Pb) free, and use a halogen free compound. Both packages are fully compliant with European Union directive 2002/95/EC.

Block Diagram

Block Diagram (Simplified)

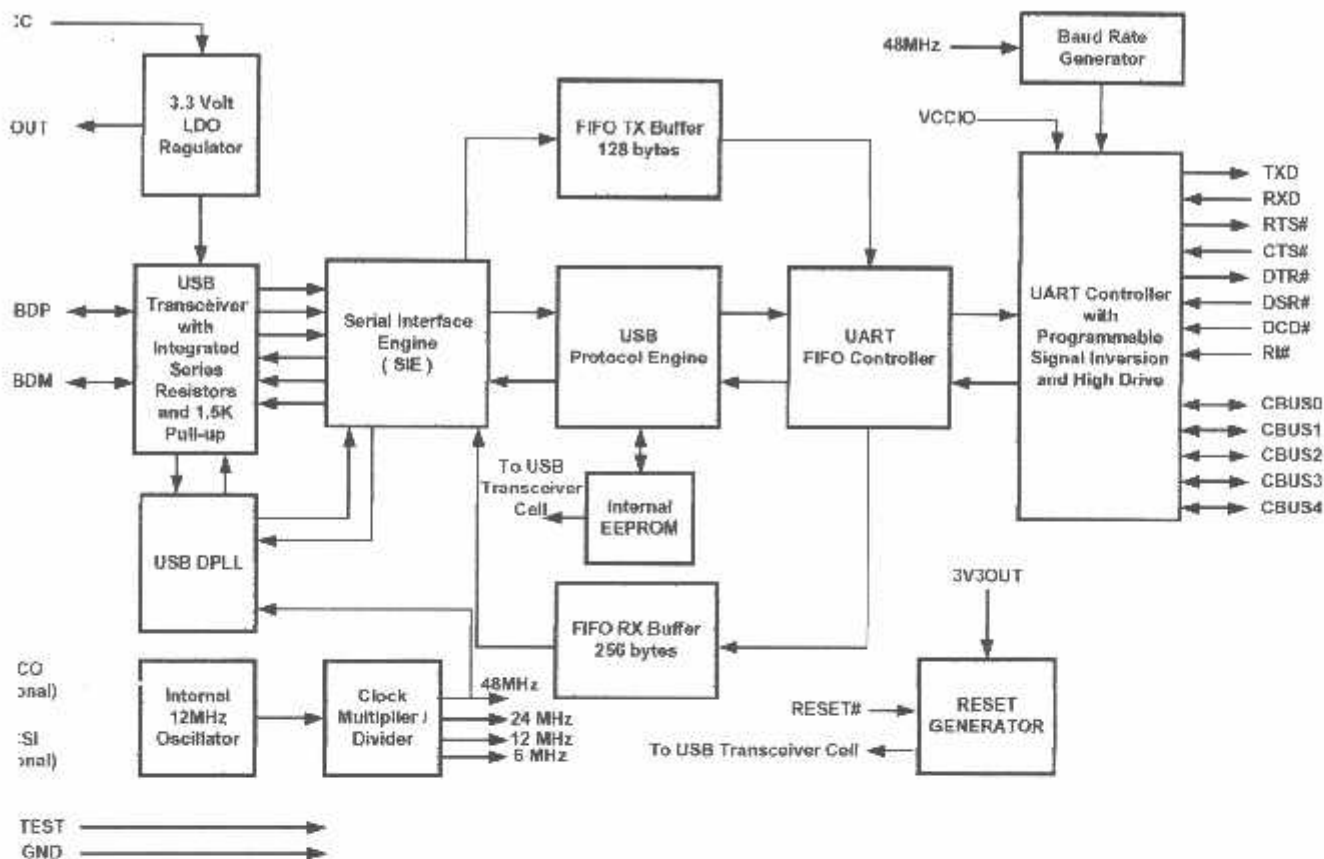


Figure 1 - FT232RL Block Diagram

3.2 Functional Block Descriptions

LDO Regulator - The 3.3V LDO Regulator generates the 3.3V reference voltage for driving the USB transceiver output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It provides 3.3V power to the 1.5k Ω internal pull up resistor on USBDP. The main function of this block is to power USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring a 3.3V nominal supply at a current of around than 50mA could also draw its power from the 3V3OUT pin, if needed.

Transceiver - The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB IC. The output drivers provide 3.3V level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SE0 and USB Reset condition detection. This Cell also incorporates internal series resistors on the USB data lines, and a 1.5k Ω pull up resistor on USBDP.

DPLL - The USB DPLL cell locks on to the incoming NRZI USB data and provides separate recovered clock and signals to the SIE block.

Internal 12MHz Oscillator - The Internal 12MHz Oscillator cell generates a 12MHz reference clock input to the x4 multiplier. The 12MHz Oscillator is also used as the reference clock for the SIE, USB Protocol Engine and FIFO controller blocks.

Multiplier / Divider - The Clock Multiplier / Divider takes the 12MHz input from the Oscillator Cell and produces the 48MHz, 24MHz, 12MHz, and 6MHz reference clock signals. The 48MHz clock reference is used for the PLL and the Baud Rate Generator blocks.

Serial Interface Engine (SIE) - The Serial Interface Engine (SIE) block performs the Parallel to Serial and Serial to Parallel conversion of the USB data. In accordance to the USB 2.0 specification, it performs bit stuffing / un-stuffing / CRC5 / CRC16 generation / checking on the USB data stream.

USB Protocol Engine - The USB Protocol Engine manages the data stream from the device USB control endpoint. It handles the low level USB protocol (Chapter 9) requests generated by the USB host controller and the commands for controlling the functional parameters of the UART.

TX Buffer (128 bytes) - Data from the USB data out endpoint is stored in the FIFO TX buffer and removed from the buffer to the UART transmit register under control of the UART FIFO controller.

RX Buffer (256 bytes) - Data from the UART receive register is stored in the FIFO RX buffer prior to being removed by the SIE on a USB request for data from the device data in endpoint.

UART FIFO Controller - The UART FIFO controller handles the transfer of data between the FIFO RX and TX buffers and the UART transmit and receive registers.

UART Controller with Programmable Signal Inversion and High Drive - Together with the UART FIFO Controller the UART Controller handles the transfer of data between the FIFO RX and FIFO TX buffers and the UART transmit and receive registers. It performs asynchronous 7 / 8 bit Parallel to Serial and Serial to Parallel conversion of the data on the RS232 (RS422 and RS485) interface. Control signals supported by UART mode include RTS, CTS, DTR, DCD and RI. The UART Controller also provides a transmitter enable control signal pin option (TXDEN) to assist with interfacing to RS485 transceivers. RTS / CTS, DSR / DTR and X-On / X-Off handshaking options are supported. Handshaking, where required, is handled in hardware to ensure fast response times. The UART also supports the RS232 BREAK setting and detection conditions. A new feature, programmable in the internal EEPROM, allows the UART signals to each be individually inverted. Another new EEPROM programmable feature allows a high current drive strength to be enabled on the UART interface and CBUS pins.

Baud Rate Generator - The Baud Rate Generator provides a x16 clock input to the UART Controller from the 48MHz reference clock and consists of a 14 bit prescaler and 3 register bits which provide fine tuning of the baud rate (divided to divide by a number plus a fraction or "sub-integer"). This determines the Baud Rate of the UART, which is programmable from 183 baud to 3 million baud.

FT232R supports all standard baud rates and non-standard baud rates from 300 Baud up to 3 Megabaud. Available non-standard baud rates are calculated as follows -

$$\text{Baud Rate} = 3000000 / (n + x)$$

where n can be any integer between 2 and 16,384 ($= 2^{14}$) and x can be a sub-integer of the value 0, 0.125, 0.25, 0.5, 0.625, 0.75, or 0.875. When n = 1, x = 0, i.e. baud rate divisors with values between 1 and 2 are not available.

This gives achievable baud rates in the range 183.1 baud to 3,000,000 baud. When a non-standard baud rate is required simply pass the required baud rate value to the driver as normal, and the FTDI driver will calculate the required divisor, and set the baud rate. See FTDI application note AN232B-05 for more details.

Reset Generator - The integrated Reset Generator Cell provides a reliable power-on reset to the device internal logic on power up. A RESET# input pin is provided to allow other devices to reset the FT232R. RESET# can be connected to VCCIO or left unconnected, unless it is a requirement to reset the device from external logic or an external reset generator I.C.

Internal EEPROM - The internal EEPROM in the FT232R can be used to store USB Vendor ID (VID), Product ID (PID), device serial number, product description string, and various other USB configuration descriptors. The internal EEPROM is also used to configure the CBUS pin functions. The device is supplied with the internal EEPROM settings programmed as described in Section 10.

Device Pin Out and Signal Descriptions

4.1 28-LD SSOP Package

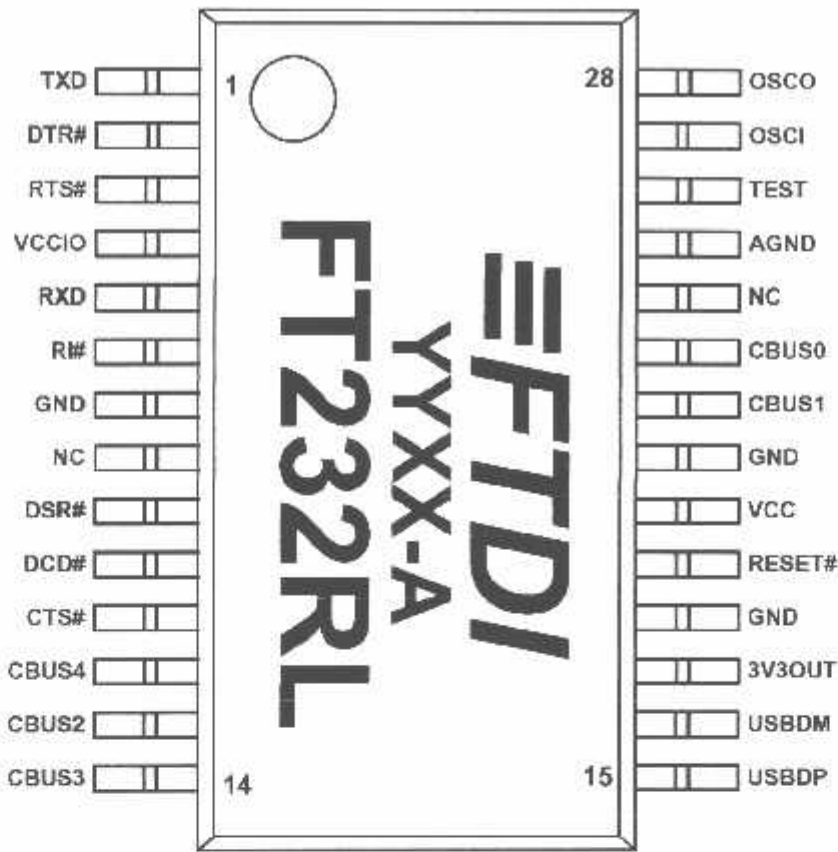


Figure 2 - 28 Pin SSOP Package Pin Out

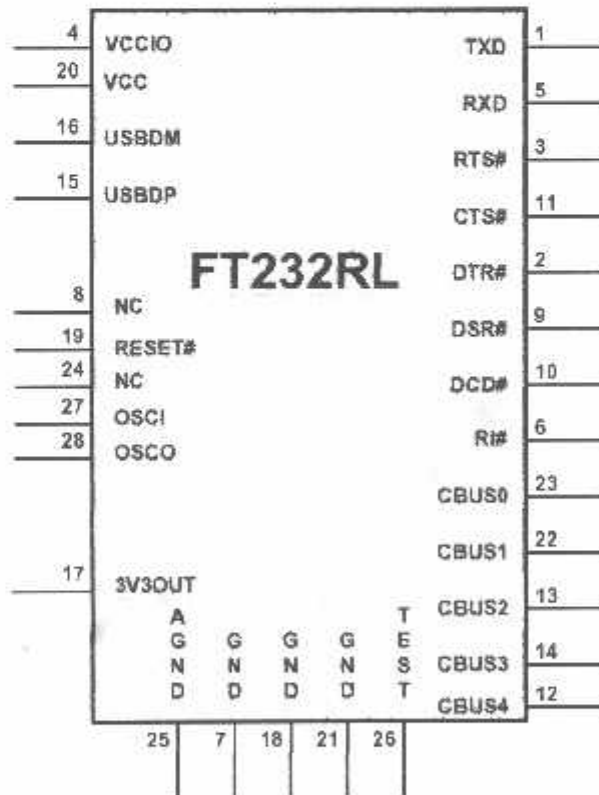


Figure 3 - 28 Pin SSOP Package Pin Out (Schematic Symbol)

4.2 SSOP-28 Package Signal Descriptions

Table 1 - SSOP Package Pin Out Description

Pin No.	Name	Type	Description
USB Interface Group			
	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5kΩ pull up resistor to 3.3V
	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor.
Power and Ground Group			
	VCCIO	PWR	+1.8V to +5.25V supply to the UART Interface and CBUS group pins (1...3, 5, 6, 9...14, 22, 23). In USB bus powered designs connect to 3V3OUT to drive out at 3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.
18, 21	GND	PWR	Device ground supply pins
	3V3OUT	Output	3.3V output from integrated L.D.O. regulator. This pin should be decoupled to ground using a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5kΩ pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT232R's VCCIO pin.
	VCC	PWR	3.3V to 5.25V supply to the device core.
	AGND	PWR	Device analog ground supply for internal clock multiplier
Miscellaneous Signal Group			
24	NC	NC	No internal connection.
	RESET#	Input	Can be used by an external device to reset the FT232R. If not required can be left unconnected, or pulled up to VCCIO.
	TEST	Input	Puts the device into I.C. test mode. Must be tied to GND for normal operation.
	OSCI	Input	Input to 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation. *
	OSCO	Output	Output from 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation if internal oscillator is used. *
UART Interface and CBUS Group **			
	TXD	Output	Transmit Asynchronous Data Output.
	DTR#	Output	Data Terminal Ready Control Output / Handshake signal.
	RTS#	Output	Request To Send Control Output / Handshake signal.
	RXD	Input	Receive Asynchronous Data Input.
	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low can be used to resume the PC USB host controller from suspend.
	DSR#	Input	Data Set Ready Control Input / Handshake signal.
	DCD#	Input	Data Carrier Detect Control input
	CTS#	Input	Clear to Send Control Input / Handshake signal.
	CBUS4	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is SLEEP#. See CBUS Signal Options, Table 3.
	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXDEN. See CBUS Signal Options, Table 3.
	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is PWREN#. See CBUS Signal Options, Table 3.
	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is RXLED#. See CBUS Signal Options, Table 3.
	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXLED#. See CBUS Signal Options, Table 3.

For detailed FTDI technical support for details on how to use an external crystal, ceramic resonator, or oscillator with the FT232R.

When used in Input Mode, these pins are pulled to VCCIO via internal 200kΩ resistors. These pins can be configured to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

4.3 QFN-32 Package

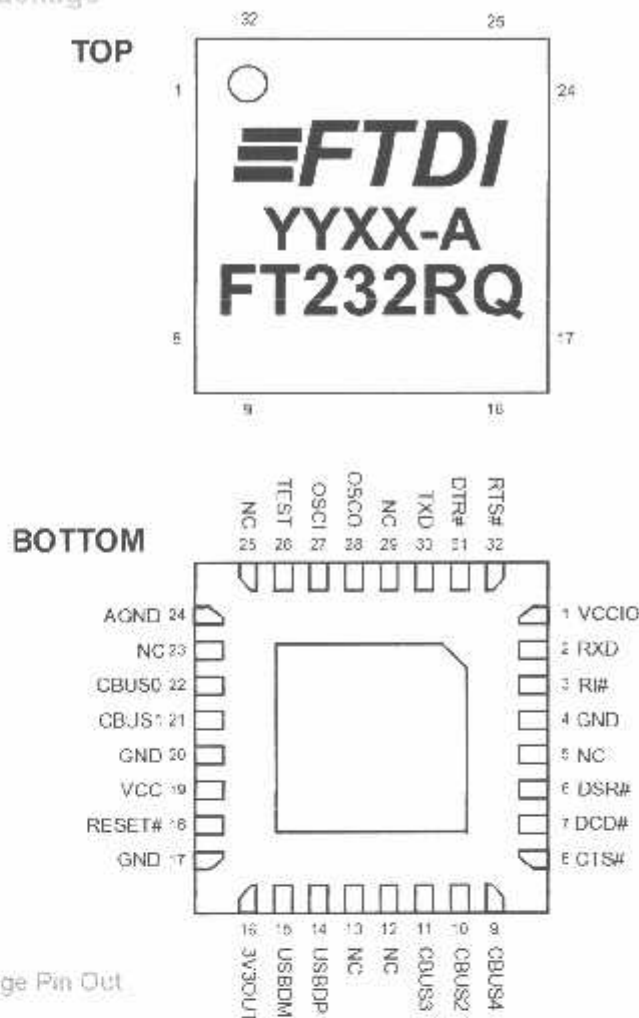


Fig 4 - QFN 32 Package Pin Out

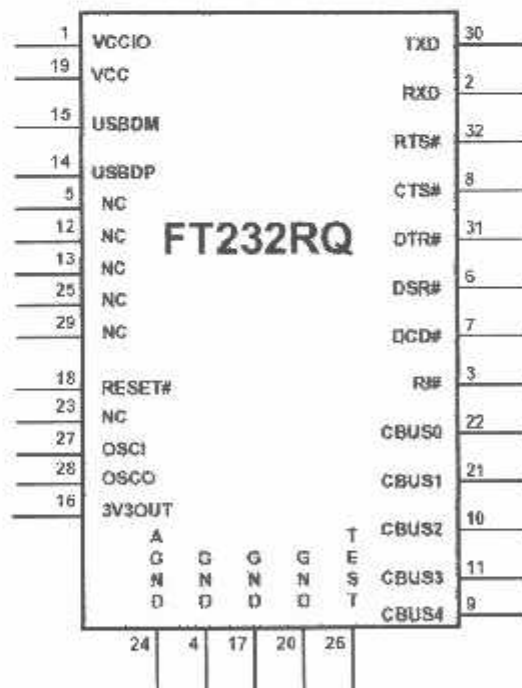


Fig 5 - QFN-32 Package Pin Out (Schematic Symbol)

4.4 QFN-32 Package Signal Descriptions

Table 2 QFN Package Pin Out Description

Pin No.	Name	Type	Description
USB Interface Group			
	USBDP	I/O	USB Data Signal Plus, incorporating internal series resistor and 1.5k Ω pull up resistor to 3.3V
	USBDM	I/O	USB Data Signal Minus, incorporating internal series resistor
Power and Ground Group			
	VCCIO	PWR	+1.8V to +5.25V supply to UART interface and CBUS group pins (2,3, 6, ...,11, 21, 22, 30...32). In USB bus powered designs connect to 3V3OUT to drive out at 3.3V levels, or connect to VCC to drive out at 5V CMOS level. This pin can also be supplied with an external 1.8V - 2.8V supply in order to drive out at lower levels. It should be noted that in this case this supply should originate from the same source as the supply to Vcc. This means that in bus powered designs a regulator which is supplied by the 5V on the USB bus should be used.
17, 20	GND	PWR	Device ground supply pins
	3V3OUT	Output	3.3V output from integrated L.D.O. regulator. This pin should be decoupled to ground using a 100nF capacitor. The prime purpose of this pin is to provide the internal 3.3V supply to the USB transceiver cell and the internal 1.5k Ω pull up resistor on USBDP. Up to 50mA can be drawn from this pin to power external logic if required. This pin can also be used to supply the FT232R's VCCIO pin.
	VCC	PWR	3.3V to 5.25V supply to the device core
	AGND	PWR	Device analog ground supply for internal clock multiplier
Miscellaneous Signal Group			
12, 13, 25, 29	NC	NC	No internal connection.
	RESET#	Input	Can be used by an external device to reset the FT232R. If not required can be left unconnected or pulled up to VCCIO.
	TEST	Input	Puts the device into I.C. test mode. Must be tied to GND for normal operation.
	OSCI	Input	Input to 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation. *
	OSCO	Output	Output from 12MHz Oscillator Cell. Optional - Can be left unconnected for normal operation if internal oscillator is used. *
RT Interface and CBUS Group **			
	TXD	Output	Transmit Asynchronous Data Output.
	DTR#	Output	Data Terminal Ready Control Output / Handshake signal.
	RTS#	Output	Request To Send Control Output / Handshake signal.
	RXD	Input	Receive Asynchronous Data Input.
	RI#	Input	Ring Indicator Control Input. When remote wake up is enabled in the internal EEPROM taking RI# low can be used to resume the PC USB host controller from suspend.
	DSR#	Input	Data Set Ready Control Input / Handshake signal.
	DCD#	Input	Data Carrier Detect Control input.
	CTS#	Input	Clear to Send Control input / Handshake signal.
	CBUS4	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is SLEEP#. See CBUS Signal Options, Table 3.
	CBUS2	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXDEN. See CBUS Signal Options, Table 3.
	CBUS3	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is PWREN#. See CBUS Signal Options, Table 3.
	CBUS1	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is RXLED#. See CBUS Signal Options, Table 3.
	CBUS0	I/O	Configurable CBUS I/O Pin. Function of this pin is configured in the device internal EEPROM. Factory Default function is TXLED#. See CBUS Signal Options, Table 3.

For more information, contact FTDI technical support for details on how to use an external crystal, ceramic resonator, or oscillator with the FT232RL.

When used in Input Mode, these pins are pulled to VCCIO via internal 200k Ω resistors. These pins can be configured to gently pull low during USB suspend (PWREN# = "1") by setting an option in the internal EEPROM.

4.5 CBUS Signal Options

The following options can be configured on the CBUS I/O pins. CBUS signal options are common to both package versions of the FT232R. These options are all configured in the internal EEPROM using the utility software MPROG, which can be downloaded from the FTDI website. The default configuration is described in Section 10.

Table 3 - CBUS Signal Options

CBUS Signal Option	Available On CBUS Pin	Description
TXDEN	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Enable transmit data for RS485
PWREN#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low after the device is configured by USB, then high during USB suspend. Can be used to control power to external logic P-Channel logic level MOSFET switch. Enable the interface pull-down option when using the PWREN# pin in this way
TXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Transmit data LED drive - pulses low when transmitting data via USB. See Section 9 for more details.
RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Receive data LED drive - pulses low when receiving data via USB. See Section 9 for more details.
TX&RXLED#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	LED drive - pulses low when transmitting or receiving data via USB. See Section 9 for more details.
EEP#	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	Goes low during USB suspend mode. Typically used to power down an external TTL to RS232 level converter I.C. in USB to RS232 converter designs.
K48	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	48MHz Clock output.
K24	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	24MHz Clock output.
K12	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	12MHz Clock output.
K6	CBUS0, CBUS1, CBUS2, CBUS3, CBUS4	6MHz Clock output.
BitBangI/O	CBUS0, CBUS1, CBUS2, CBUS3	CBUS bit bang mode option. Allows up to 4 of the CBUS pins to be used as general purpose I/O. Configured individually for CBUS0, CBUS1, CBUS2 and CBUS3 in the internal EEPROM. A separate application note will describe in more detail how to use CBUS bit bang mode.
BitBangWRn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode WR# strobe Output
BitBangRDn	CBUS0, CBUS1, CBUS2, CBUS3	Synchronous and asynchronous bit bang mode RD# strobe Output



FEATURES

Power dissipation

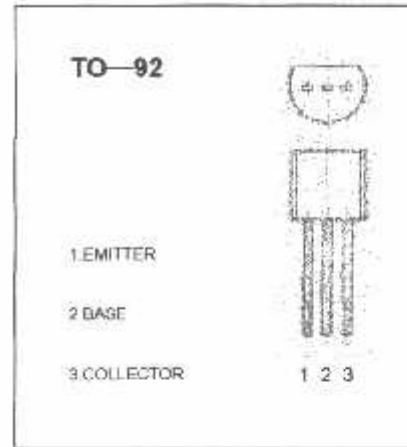
$$P_{CM} : 0.4 \text{ W (Tamb=25}^\circ\text{C)}$$

Collector current

$$I_{CM} : 0.1 \text{ A}$$

Collector-base voltage

$$V_{(BR)CBO} : 50 \text{ V}$$

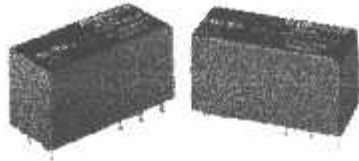


ELECTRICAL CHARACTERISTICS (Tamb=25°C unless otherwise specified)

Parameter	Symbol	Test conditions	MIN	TYP	MAX	UNIT
Collector-base breakdown voltage	$V_{(BR)CBO}$	$I_C = 100 \mu\text{A}, I_E = 0$	50			V
Collector-emitter breakdown voltage	$V_{(BR)CEO}$	$I_C = 0.1 \text{ mA}, I_B = 0$	45			V
Emitter-base breakdown voltage	$V_{(BR)EB0}$	$I_E = 100 \mu\text{A}, I_C = 0$	5			V
Collector cut-off current	I_{CBO}	$V_{CB} = 50 \text{ V}, I_E = 0$			0.1	μA
Collector cut-off current	I_{CEO}	$V_{CE} = 35 \text{ V}, I_B = 0$			0.1	μA
Emitter cut-off current	I_{EB0}	$V_{EB} = 3 \text{ V}, I_C = 0$			0.1	μA
DC current gain(note)	$H_{FE(1)}$	$V_{CE} = 5 \text{ V}, I_C = 1\text{mA}$	60		1000	
Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_C = 100\text{mA}, I_B = 5 \text{ mA}$			0.3	V
Base-emitter saturation voltage	$V_{BE(sat)}$	$I_C = 100 \text{ mA}, I_B = 5\text{mA}$			1	V
Transition frequency	f_T	$V_{CE} = 5 \text{ V}, I_C = 10\text{mA}$ $f = 30\text{MHz}$	150			MHz

CLASSIFICATION OF $H_{FE(1)}$

Rank	A	B	C	D
Range	60-150	100-300	200-600	400-1000



RT series (DC Coil) 16 Amp PC Board Miniature Relay

File E22575
File LR15734
NR 6108

Users should thoroughly review the technical data before selecting a product part number. It is recommended that users also seek out the pertinent approvals files of the agencies/laboratories and review them to ensure the product meets the requirements for a given application.

Features

PST through DPDT contact arrangements
Immersion cleanable and flux tight versions available
DE 10mm spacing, 5kV dielectric, coil to contacts,
UL Class F (155°C) coil insulation system,
conforms to UL 508, 1873, 353 and 1950.
Low profile: 15.7mm height
Resistive coil, 400mW
Withstand surge voltage of 10,000V
Enter X Brumfield or Schrack brand.

Contact Data

Arrangements: 1 Form A (SPST-NO) Wiring Diagram Code 1, 2, 3
2 Form A (DPST-NO) Wiring Diagram Code 5
1 Form C (SPDT) Wiring Diagram Code 1, 2, 3,
2 Form C (DPDT) Wiring Diagram Code 5

Material: Silver clad 90/10.

Minimum Load: 12W/100mA.

Expected Mechanical Life: 10 million operations

Initial Contact Resistance: 100 milliohms max @ 1A 12VDC.

Designed to meet UL/CSA/VDE ratings with relay properly vented. Remove vent nib after soldering and cleaning.

CSA/VDE Ratings @ 25°C

Code	NO/NC Load	Type	Operations
A	10A/10A @ 277VAC	Resistive/GP	100K
	10A/10A @ 30VDC	Resistive	100K
	12A/12A @ 250VAC	Resistive/EP	30K
	12A/12A @ 30VDC	Resistive	30K
	3/4 HP @ 480VAC*	Motor	6K
	1/2 HP @ 240VAC*	Motor	6K
	1/4 HP @ 120VAC*	Motor	6K
	4B LRA/10 FLA @ 240VAC*	Motor	30K
	TV-3 @ 120VAC*	Tungsten	25K
	A300, 720VA @ 240VAC*	Pilot Duty	30K
B	16A/16A @ 277VAC	Resistive/GP	50K
	20A/20A @ 277VAC	Resistive/GP	30K
	20A/20A @ 24VDC	Resistive	30K
	16A/16A @ 30VDC	Resistive	30K
	1 HP @ 480VAC*	Motor	6K
	1 HP @ 240VAC*	Motor	6K
	1/2 HP @ 120VAC*	Motor	6K
	60 LRA/10 FLA @ 250VAC*	Motor	30K
	TV-5 @ 120VAC*	Tungsten	25K
	A300, 720VA @ 240VAC*	Pilot Duty	30K
C	B300, 360VA @ 240VAC**	Pilot Duty	30K
	8A/8A @ 277VAC	Resistive/GP	100K
	8A/8A @ 30VDC	Resistive	100K
	10A/10A @ 250VAC	Resistive/GP	30K
	10A/10A @ 30VDC	Resistive	30K
	1/2 HP @ 240VAC*	Motor	6K
	1/4 HP @ 120VAC*	Motor	6K
	34.8 LRA/6 FLA @ 120VAC*	Motor	30K
	17.4 LRA/6 FLA @ 240VAC*	Motor	30K
	B300, 360VA @ 240VAC*	Pilot Duty	30K
TV-3 @ 120VAC*	Tungsten	25K	

* A only
** B only

Dielectric Strength

Open Contacts: >1,000VAC (1 minute)
Open Poles (code 5): >2,500VAC (1 minute)
Open Coil and Contacts: >5,000VAC (1 minute)
Withstand Voltage (DC): >10,000VAC x (1.2 x h) (used)

Dimensions are shown for
surchassis only.

Dimensions are in inches over
(millimeters) unless otherwise
specified.

Coil Data @ 25°C

Voltage: 5 to 11kVDC.

Nominal Power @ 25°C: 400mW.

Duty Cycle: Continuous.

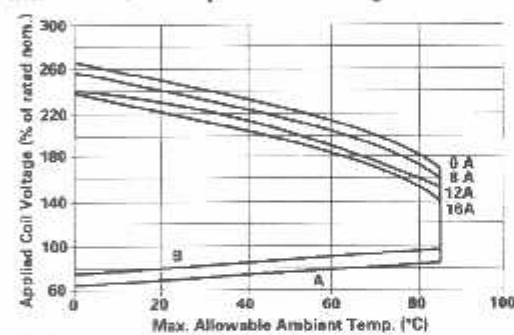
Initial Insulation Resistance: 10,000 megohms, min., at 20°C, 500VDC
and 50% rel. humidity.

Coil Construction: UL Class F (155°C)

Coil Data @ 25°C

Nominal Voltage VDC	DC Resistance in Ohms ±10%	Must Operate Voltage VDC	Nominal Coil Current (mA) - 50/60Hz.
005	62	3.5	80
006	90	4.2	66.7
009	202	5.3	44.4
012	360	6.4	33.3
018	610	12.6	22.2
024	1,440	16.8	16.7
048	5,760	33.6	8.3
060	9,000	42.0	8.0
110	30,250	77.0	4.3

Max. Ambient Temp. vs. Coil Voltage



A: Coil temperature = Ambient temperature.
B: 110% of nominal coil voltage at rated contact load.

Operate Data @ 25°C

Must Operate Voltage (DC): 70% of nominal

Must Release Voltage (DC): 10% of nominal

Operate Time (Excluding Bounce):
7 ms, typ., 15ms max. at nom. voltage

Release Time (Excluding Bounce):
3 ms, typ., 6ms max. at nom. voltage

Environmental Data

Temperature Range:

Storage: -40°C to +105°C.

Operating: -40°C to +85°C at rated current.

Vibration, Operational

N.O.: 0.065" (1.65mm) max. excursions from 10 - 55 Hz

N.C.: 0.032" (0.82mm) max. excursions from 10 - 55 Hz
with no contact opening >10µ.

Mechanical Data

Termination: Printed circuit terminals

Enclosures: RT 1, 2, 3, 4: Flux-tight, top vented, plastic case.

RT B, C, D, E: Immersion cleanable, plastic case.

Weight: 0.35 oz. (10g) approximately.

Specifications and availability
subject to change

www.tycoelectronics.com
Technical support:
Refer to inside back cover

Ordering Information (DC Coil Models)

Typical Part Number →

RT

B

3

4

012

F

1. Basic Series:

RT = Miniature, printed circuit board relay.

2. Enclosure:

- 1 = 1 pole 12A, Pinning 3.5mm, flux-tight (Code 1)
- 2 = 1 pole 12A, Pinning 5mm, flux-tight (Code 2)
- 3 = 1 pole 16A, Pinning 5mm, flux-tight (Code 3)
- 4 = 2 pole 8A, Pinning 5mm, flux-tight (Code 5)
- B = 1 pole 12A, Pinning 3.5mm, sealed (Code 1)
- C = 1 pole 12A, Pinning 5mm, sealed (Code 2)
- D = 1 pole 16A, Pinning 5mm, sealed (Code 3)
- E = 2 pole 8A, Pinning 5mm, sealed (Code 5)

3. Contact Arrangement:

- 1 = 1 Form C (SPDT) (Requires wiring diagram codes 1, 2 or 3)
- 2 = 2 Form C (DPDT) (Requires wiring diagram code 5)
- 3 = 1 Form A (SPST-NO) (Requires wiring diagram codes 1, 2 or 3)
- 4 = 2 Form A (DPST-NO) (Requires wiring diagram code 5)

4. Contact Material:

4 = Silver-nickel 90/10 (standard stock)

5. Coil Voltage:

- 005 = 5VDC 009 = 9VDC 013 = 10VDC 048 = 48VDC 110 = 110VDC
- 006 = 6VDC 012 = 12VDC 024 = 24VDC 060 = 60VDC

5. Coil Insulation Classification, Brand and Case Color

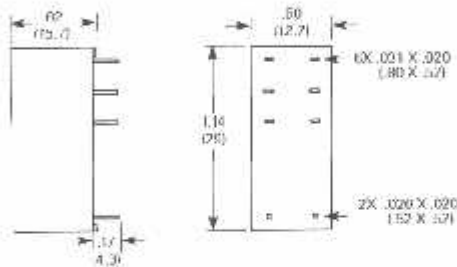
F = UL Class F, Potter & Brumfield Brand, Black Case

Leave Blank = UL Class F, Schrack Brand, Orange Case

Our authorized distributors are more likely to stock the following items for immediate delivery.

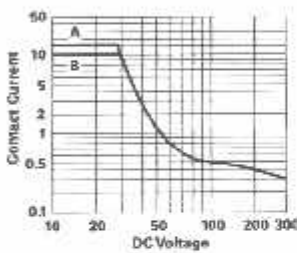
- RT114012F RTB14012F RTB34024F RTD14005F RTD34012F RTE24005F RTE44012F
- RT114024F RTB14024F RT314012F RTD14012F RT424012F RTE24012F RTE44024F
- RTB14005F RTB34012F RT314024F RTD14024F RT424024F RTE24024F

Outline Dimensions

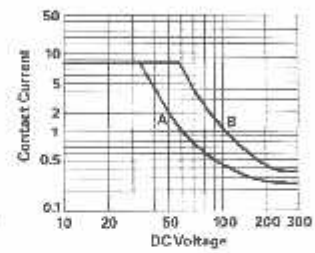


Breaking Capacity

1 Pole



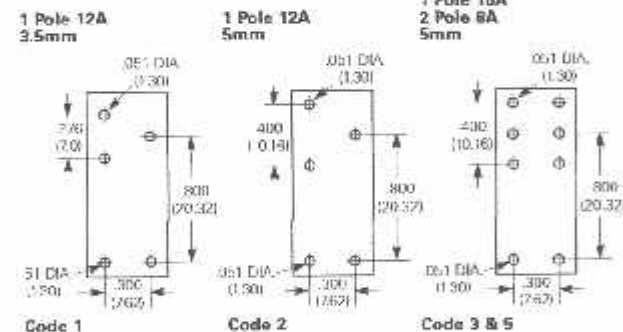
2 Pole



A: 16A Version
B: 12A Version

A: 1 Contact
B: 2 Contacts in series

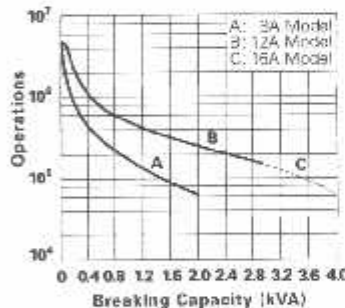
PC Board Layouts (Bottom View)



- Code 1
- Code 2
- Code 3 & 5

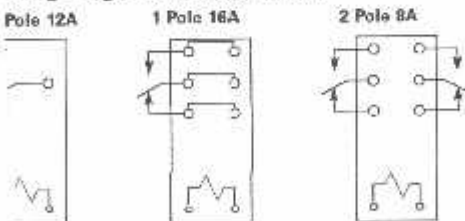
Note: 1. On single throw models, only necessary terminals are present.
2. With the recommended PCB hole sizes, a gull with a pattern from 0.0891 to 0.11 in (2.5 - 2.54 mm) can be used.

Contact Life for Resistive AC Load (Typical)



Note: Data from 250°C @ 70°C

Wiring Diagrams (Bottom View)



Notes 1 & 2: On single throw models, only necessary terminals are present.

Dimensions are shown for wire crosses only

Dimensions are in inches over (millimetres) unless otherwise specified.

Specifications and availability subject to change.

www.tycoelectronics.com
Technical support: refer to inside back cover

Features

- Compatible with MCS-51[®] Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 1.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

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

AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a sector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. 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 content but freezes the oscillator, disabling all other chip functions until the next interrupt or software reset.



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

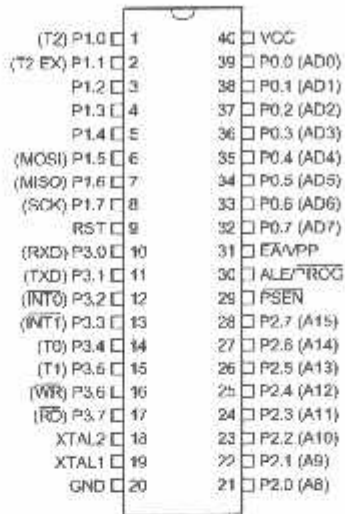
AT89S52

1919B-MICRO-11/03

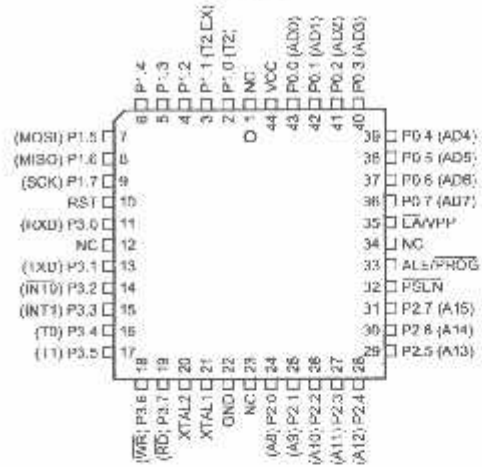


Pin Configurations

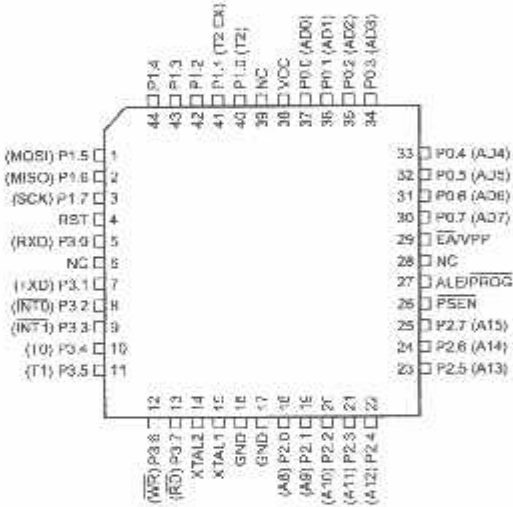
PDIP



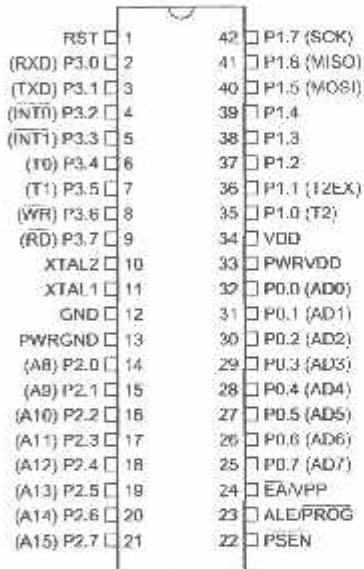
PLCC



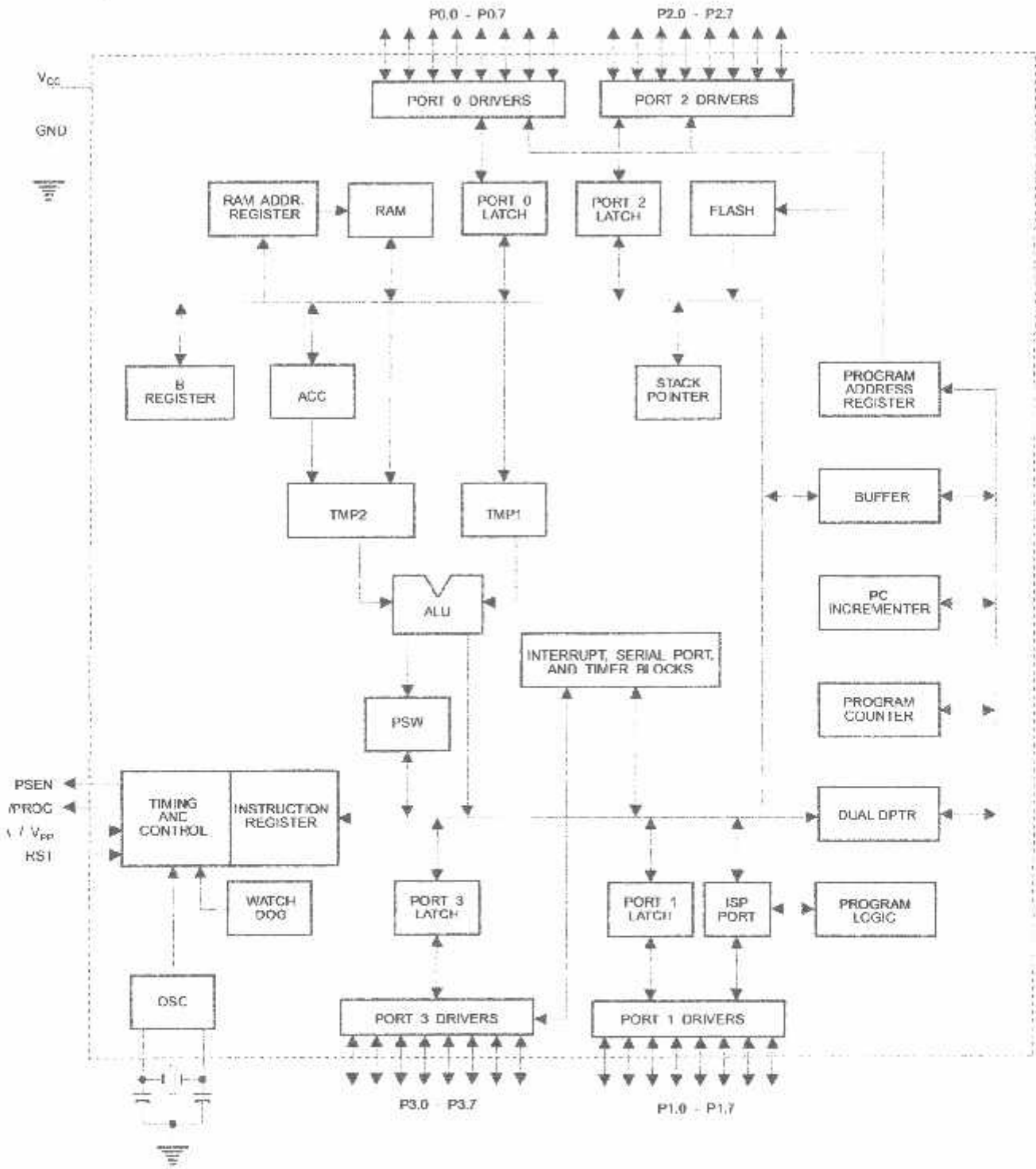
TQFP



PDIP



Block Diagram



Pin Description

- V_{CC}** Supply voltage.
- V_{DD}** Ground.
- PORT0** Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.
- Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal 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.**
- PORT1** Port 1 is an 8-bit bidirectional 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.
- In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.
- Port 1 also receives the low-order address bytes during Flash programming and verification.
- | Port Pin | Alternate Functions |
|----------|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.5 | MOSI (used for In-System Programming) |
| P1.6 | MISO (used for In-System Programming) |
| P1.7 | SCK (used for In-System Programming) |
- PORT2** Port 2 is an 8-bit bidirectional 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 @ R1), 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.
- PORT3** Port 3 is an 8-bit bidirectional 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 AT89S52, 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	\overline{WR} (external data memory write strobe)
P3.7	\overline{RD} (external data memory read strobe)

T Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

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

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

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

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

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

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

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

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

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

2 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 6) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

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

Table 1. AT89S52 SFR Map and Reset Values

3FH									0FFH
3EH	B 00000000								0FH
3DH									0EH
3CH	ACC 00000000								0E7H
3BH									0DFH
3AH	PSW 00000000								0D7H
39H	T2CON 00000000	T2MOD XXXXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
38H									0C7H
37H	IP XX000000								0B7H
36H	P3 11111111								0B7H
35H	IE 0X000000								0AFH
34H	P2 11111111		AUXR1 XXXXXXXX00				WDTRST XXXXXXXXXX		0A7H
33H	SCON 00000000	SBUF XXXXXXXXXX							9FH
32H	P1 11111111								97H
31H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXXXXXXX00		8FH
30H	P0 11111111	SP 00001111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XXX0000	87H

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C6H					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 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/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.

Table 3. AUXR: Auxiliary Register

AUXR	Address = 8EH	Reset Value = 0X000X0B						
	Not Bit Addressable							
Bit	7	6	5	4	3	2	1	0
	-	-	-	WDIDLE	DISRTO	-	-	DISALE
	Reserved for future expansion							
DISALE	Disable/Enable ALE							
DISALE	Operating Mode							
0	ALE is emitted at a constant rate of 1/6 the oscillator frequency							
1	ALE is active only during a MOVX or MOVC instruction							
DISRTO	Disable/Enable Reset out							
DISRTO								
0	Reset pin is driven High after WDT times out							
1	Reset pin is input only							
WDIDLE	Disable/Enable WDT in IDLE mode							
WDIDLE								
0	WDT continues to count in IDLE mode							
1	WDT halts counting in IDLE mode							

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

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

Table 4. AUXR1: Auxiliary Register 1

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

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

Program Memory If the \overline{EA} pin is connected to GND, all program fetches are directed to external memory. On the AT89S52, if \overline{EA} is connected to V_{CC} , program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

Data Memory The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

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

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

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

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

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

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

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

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Enabling the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is $98 \times T_{OSC}$, where $T_{OSC} = 1/F_{OSC}$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

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

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

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

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

UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select "Products", then "8051-Architecture Flash Microcontroller", then "Product Overview".

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select "Products", then "8051-Architecture Flash Microcontroller", then "Product Overview".

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T₂ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 5. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 5. Timer 2 Operating Modes

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

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

apture Mode

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

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 6). 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 1. Timer in Capture Mode

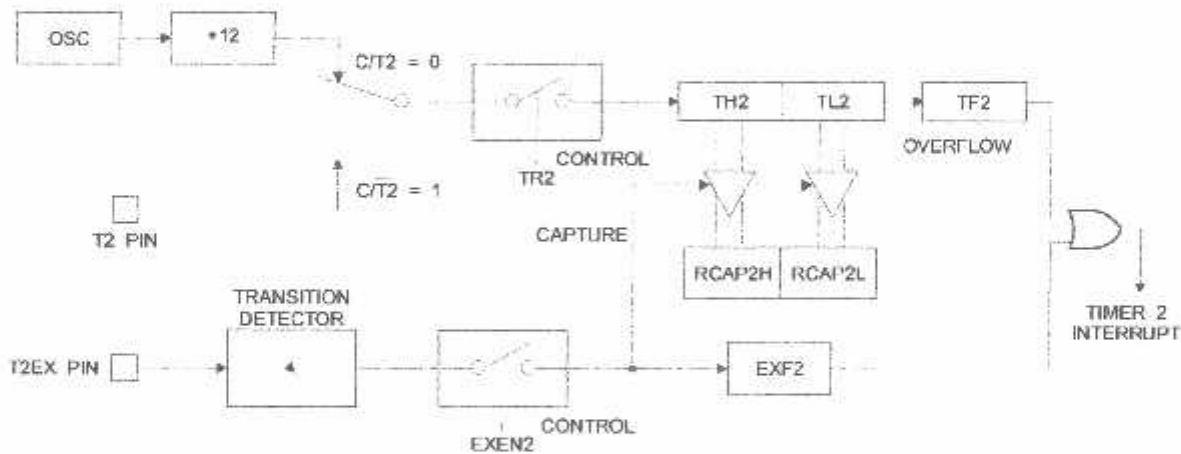


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H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

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 2. 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 Auto Reload Mode (DCEN = 0)

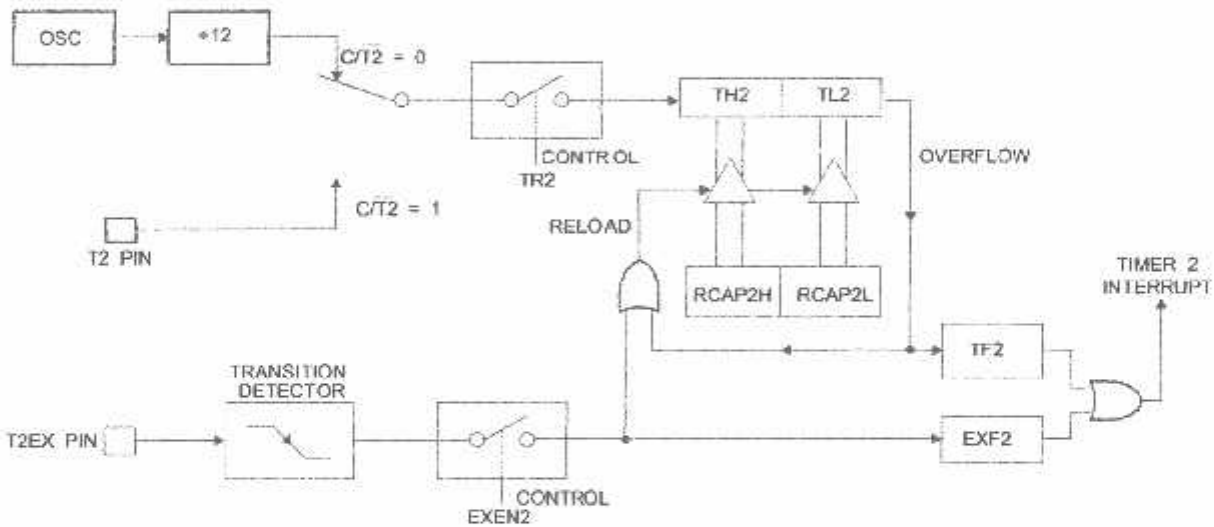
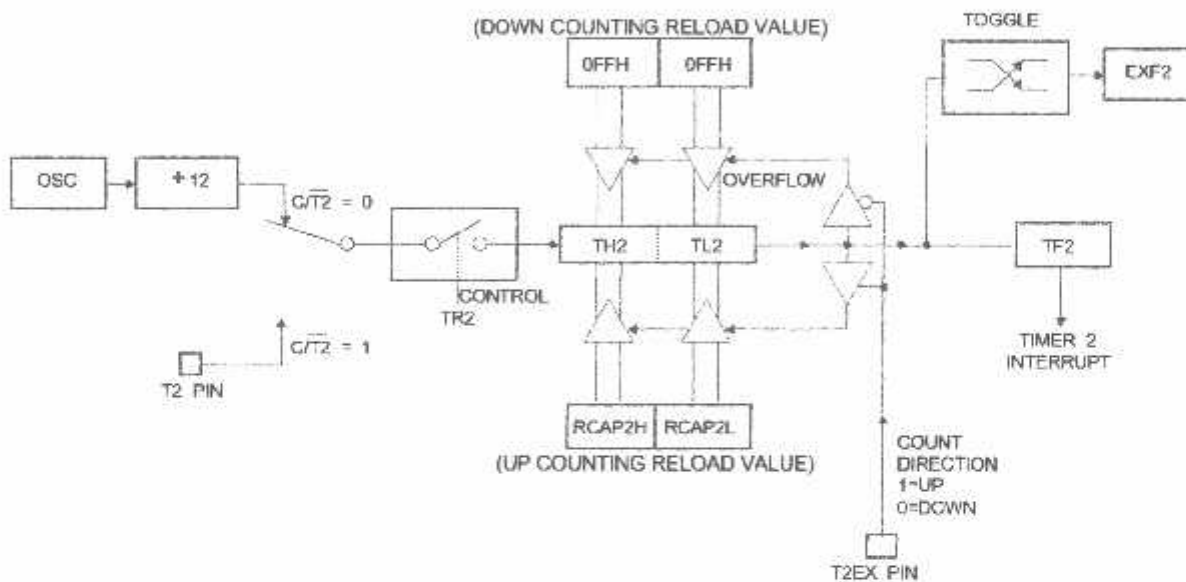


Table 6. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN

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

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



Baud Rate Generator

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

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

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

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

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

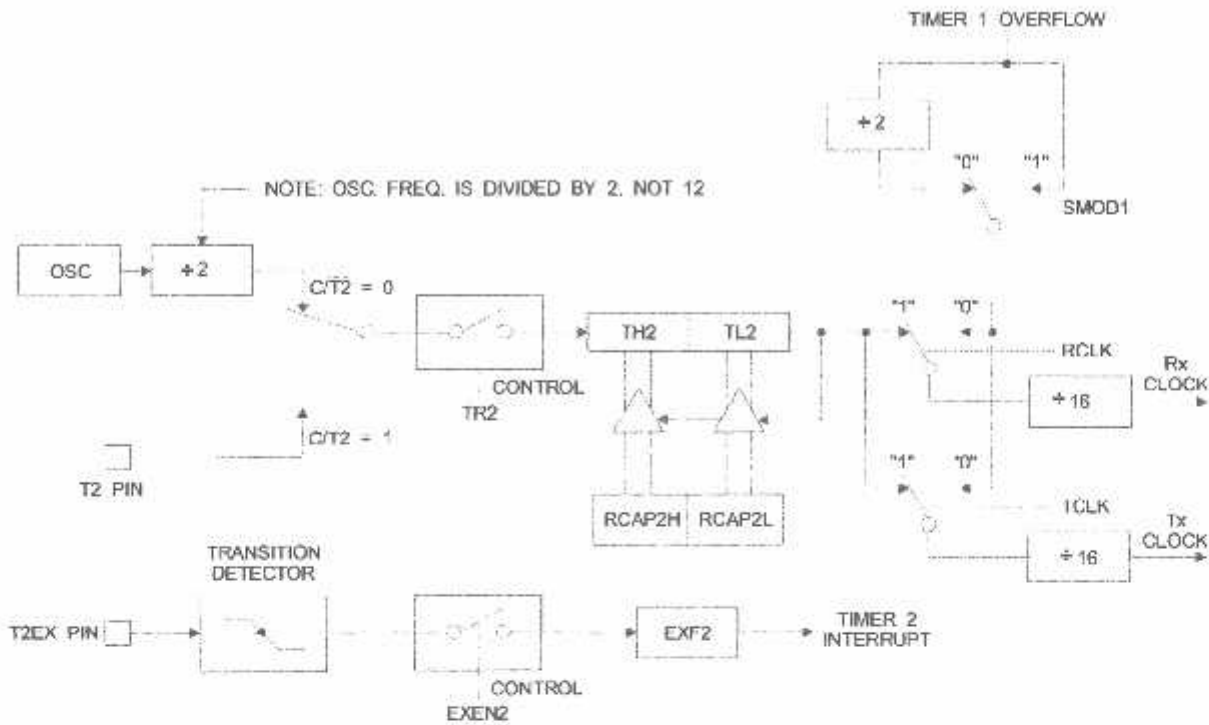
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - \text{RCAP2H}, \text{RCAP2L}]}$$

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.

Figure 4. Timer 2 in Baud Rate Generator Mode



**Programmable
Clock Out**

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 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).

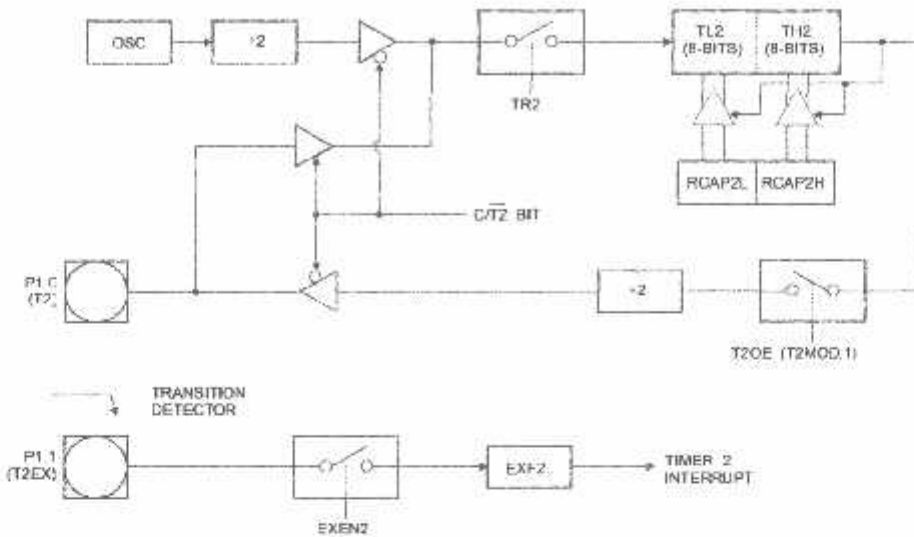
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

Figure 5. Timer 2 in Clock-Out Mode



Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

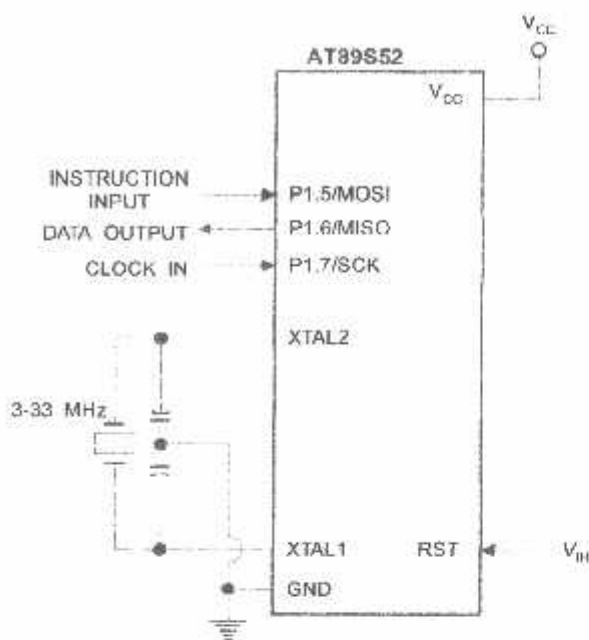
Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it 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.

Figure 12. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 13. Serial Programming Waveforms

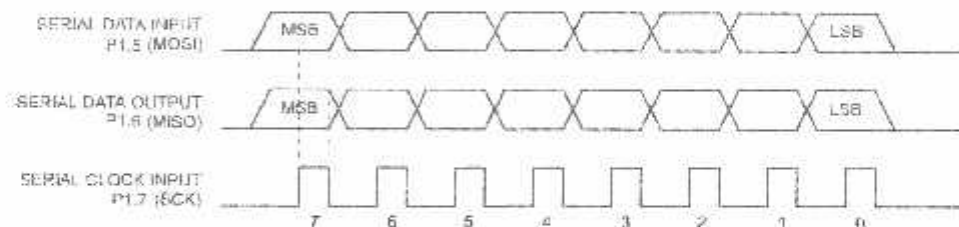


Table 11. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	7 DDDD DDDD DDDD DDDD	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	7 DDDD DDDD DDDD DDDD	Write data to Program memory in the byte mode
Write Lock Bits ⁽¹⁾	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxx A12 A11 A10 A9 A8	A7 xxx xxx0	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

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



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

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

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

Serial Programming Characteristics

Figure 14. Serial Programming Timing

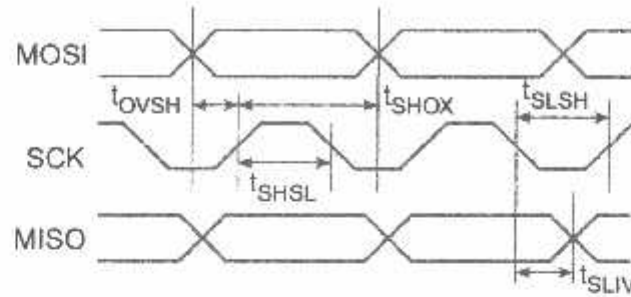


Table 12. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
f_{CLCL}	Oscillator Frequency	3		33	MHz
T_{CLCL}	Oscillator Period	30			ns
t_{HSL}	SCK Pulse Width High	$8 t_{CLCL}$			ns
t_{LSH}	SCK Pulse Width Low	$8 t_{CLCL}$			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	$2 t_{CLCL}$			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{BASE}	Chip Erase Instruction Cycle Time			500	ms
t_{WC}	Serial Byte Write Cycle Time			$64 t_{CLCL} + 400$	μs

Absolute Maximum Ratings*

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

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

C Characteristics

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

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage (\overline{EA})		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
H	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -80 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
H1	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	μA
	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
ST	Reset Pulldown Resistor		50	300	$\text{k}\Omega$
	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽¹⁾	$V_{CC} = 5.5\text{V}$		50	μA

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

AC Characteristics

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

External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
f_{CLCL}	Oscillator Frequency			0	33	MHz
t_{HEL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{AVL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-25$		ns
t_{LAX}	Address Hold After ALE Low	48		$t_{\text{CLCL}}-25$		ns
t_{LIV}	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
t_{LPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-25$		ns
t_{LPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-45$		ns
t_{LIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-60$	ns
t_{IX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
t_{IX}	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-25$	ns
t_{XAV}	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
t_{IV}	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-80$	ns
t_{AZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RI}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{WH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{IV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
t_{DX}	Data Hold After $\overline{\text{RD}}$	0		0		ns
t_{DZ}	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
t_{IV}	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
t_{IV}	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
t_{VL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
t_{VL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
t_{VX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-30$		ns
t_{VH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-130$		ns
t_{DX}	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-25$		ns
t_{Z}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{H}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-25$	$t_{\text{CLCL}}+25$	ns

