

SKRIPSI

**PERANCANGAN DAN PEMBUATAN ABSENSI KARYAWAN
PEGAWAI NEGERI BERBASIS RFID YANG DIINTERFACE
KE PC MELALUI USB**



Disusun Oleh :

**HERI RAHARJO
NIM : 03.17.103**



**JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
MARET 2008**

SECRET

HAWAIIAN ISLANDS METEOROLOGICAL AND MARINE OBSERVATION
STATION, HONOLULU, HAWAII
OCEANOGRAPHIC DATA CENTER
NO. 1000 OF 1000

NOV 19 1966
HONOLULU, HAWAII

NOV 19 1966
OCEANOGRAPHIC DATA CENTER
NO. 1000 OF 1000

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANOGRAPHIC ADMINISTRATION
OCEANOGRAPHIC DATA CENTER
1000 OF 1000
NOV 19 1966

LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN ABSENSI KARYAWAN
PEGAWAI NEGERI BERBASIS RFID YANG DIINTERFACE
KE PC MELALUI USB**

SKRIPSI

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

Disusun Oleh :


**HERI RAHARJO
NIM : 03.17.103**

Diperiksa dan Disetujui

Dosen Pembimbing I


Dosen Pembimbing II


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


Sotyo Hadi, ST, MSc.
NIP.Y.1039700309

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**



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Heri Raharjo
NIM : 03.17.103
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : Perancangan Dan Pembuatan Absensi Karyawan Pegawai Negeri Berbasis RFID yang Diinterface Ke PC Melalui USB.

Dipertahankan di hadapan majelis penguji skripsi jenjang strata satu (S-1) pada :

Hari : Sabtu
Tanggal : 15 Maret 2008
Nilai : 79,5 (B+) *fy*



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, ST, MT)
NIP.Y. 1030100361

Handwritten text at the top of the page, possibly a header or title, which is mostly illegible due to fading.

Handwritten text in the upper middle section of the page.

Main body of handwritten text in the middle section of the page.

Handwritten text in the lower middle section of the page.

Handwritten text on the left side of the page.

Handwritten text on the left side of the page.

Handwritten text on the left side of the page.

Handwritten text on the left side of the page.

Handwritten text on the left side of the page.



Handwritten text on the right side of the page, partially overlapping the stamp.

Handwritten text on the right side of the page.

Handwritten text on the right side of the page.

**PERANCANGAN DAN PEMBUATAN ABSENSI KARYAWAN
PEGAWAI NEGERI BERBASIS RFID YANG
DIINTERFACE KE PC MALALUI USB**

Heri Raharjo

03.17.103

Jurusan Teknik Elektronika – Institut Teknologi Nasional Malang

Jln. Raya Karanglo Km 2 Malang

heri_raharjo@yahoo.com

Dosen Pembimbing : I. Ir.F.Yudi Limpraptono, MT.

II. Sotyohadi, ST, MSc

Kata Kunci : RFID, Mikrokontroler, Modul FT232BM.

Perkembangan pesat bidang komunikasi membuat komunikasi digital semakin populer. Di dalam segi penyampaian informasi tersebut ada banyak cara yang digunakan, contohnya pemanfaatan teknologi RFID. Dengan memanfaatkan teknologi tag RFID (Radio Frequency Identification) sebagai kartu anggota..

Dalam perancangan dan pembuatan alat ini didukung oleh sistem mikrokontroller sebagai pengendali RFID reader, dan LCD. Kemudian dengan bantuan mikrokontroler AT89S51 dan dibuat berbasis RFID yang diinterfacekan ke komputer melalui USB. Untuk menampilkan data dari anggota tersebut, berdasarkan kode dari tag RFID yang telah teridentifikasi dengan bantuan komputer. Untuk dapat melakukan pemantauan pada komputer maka menggunakan bahasa pemrograman Delphi sebagai tampilan visualnya.

Alat ini dibuat untuk meringkas dan meminimalisasikan pendaftaran yang diharapkan nantinya dapat memberikan kemudahan dalam pembuatan kartu anggota pada karyawan pegawai negeri.

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perancangan Dan Pembuatan Absensi Karyawan Pegawai Negeri Berbasis RFID Yang Diinterface Ke PC Melalui USB” 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 penyusun menyampaikan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1 dan selaku Dosen Pembimbing I.
2. Bapak Sotyohadi, ST, MSc selaku Dosen Pembimbing II dan Ka. Laboratorium Perancangan Elektronika.
3. Ayah dan Ibu serta saudara-saudara kami yang telah memberikan do'a restu, dorongan, semangat, dan biaya.
4. Rekan-rekan Instruktur di Laboratorium Elektronika dan Komponen.
5. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

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

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

Malang, Maret 2008

Penyusun

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

"In the Name of Allah, the Merciful, The Most Merciful"

this thesis is my dedicate as sign devote me to my father and beloved my mother which have given feel love and attention to me during the time finished of this study. Thank which the as sincerely as possible of innermost heart. I'm very realize these can be get without immolation sincere from my father and beloved mother, and these can be success without helped from the god too, because my inspire come from the God. To my old sister and my young sister thanks a lot of have to give me support and affection unlimited.

And for my guidance and examiner lecturer, thanks a lot of have to believed and guidance to me, so I can to get my future. And for Ir. Djojo Priatmono, MT, thanks a lot of give me opportunity to be instructor Laboratorium Elektronika and Komponen so I can get more science in the sector.

For my friends as Bang Andreas, Bang Irfan, Bang Erik, Bang imron, Bang Adi, Dian, Yuli and junior team it's can't I say the name, you are my good friends, I feel lucky have friends like your , it's can't be if the God to don't meet us.

Thanks a lot of the help everything ,pray and support at this time.

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR GAMBAR	ix
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	2
1.4. Tujuan.....	3
1.5. Metodologi Penelitian.....	3
1.5. Sistematika Penulisan.....	4
BAB II TEORI DASAR	5
2.1. Pendahuluan	5
2.2. RFID (<i>Radio Frequency Identification</i>)	5
2.2.1. Komponen <i>RFID Tag</i>	5
2.2.1.1. <i>Tag</i>	5
2.2.1.2. <i>Tag Reader</i>	6
2.2.1.3. <i>Server Database</i>	8
2.2.2. Mekanisme RFID	8

2.2.2.1. Prinsip Kerja <i>Reader</i> dan <i>Tag/Transpoder</i>	8
2.2.2.2. Pengiriman Data	10
2.2.2.3. Format Pembacaan ASCII	10
2.3. Mikrokontroler AT89S51	11
2.3.1. Konfigurasi Pin AT89S51	13
2.3.2. Struktur dan Operasi Port	16
2.3.3. Antarmuka Serial	17
2.3.4. Pengaturan Boud Rate Port Serial	19
2.4. Modul FT232BM - USB	22
2.4.1. Kabel USB	24
2.5. IC 74LS164	26
2.6. LCD (Liquid Crystal Display) M1632	28
2.7. Buzzer	29
2.8. Relay	30
2.9. Borland Delphi	32
2.9.1. IDE (<i>Integrated Development Enviroment</i>)	33
2.9.2. Menu Borland Delphi	34
BAB III PERANCANGAN DAN PEMBUATAN ALAT	36
3.1. Pendahuluan	36
3.2. Blok Diagram Keseluruhan Sistem	36
3.3. Prinsip Kerja Alat	37
3.4. Perancangan Perangkat Keras.....	38

3.4.1. RFID	38
3.5. Mikrokontroler AT89S51	40
3.5.1.1. Perancangan Minimum Sistem AT89S51 ...	40
3.5.1.2. Perancangan <i>Rangkaian Reset</i>	40
3.5.1.3. Perancangan Rangkaian <i>Clock</i>	42
3.5.1.4. Perancangan Pengaturan <i>Port</i>	42
3.6. Perancangan Rangkaian LCD	44
3.7. Modul FT232 BM	46
3.7.1. Penggunaan Pin-Pin Modul FT232BM	46
3.8. Rangkaian Switch Data.....	51
3.9. Perancangan Perangkat Lunak.....	52
3.9.1. Program Aplikasi Mikrokontroller	53
3.9.2. Program Aplikasi Komputer.....	53
3.9.3. Component Pallete Delphi7.....	54
3.9.3.1.Tab Standard.....	54
3.9.3.2.Tab Data Access	55
3.9.3.3.Tab Data Controls.....	55
3.9.4. Flowchart Program Keseluruhan	57
3.9.5. Flowchart Program dari Mikrokontroller AT89S51	58
3.9.6. Flowchart Program dari Komputer	59
3.9.7. Perencanaan Database.....	59

BAB IV PENGUJIAN ALAT	65
4.1. Pendahuluan	65
4.2. Pengujian RFID	65
4.2.1. Tujuan	65
4.2.2. Prosedur Pengujian	65
4.2.3. Hasil Pengujian	67
4.3. Pengujian FT232BM.....	68
4.3.1. Instalasi Driver FT232BM	69
4.3.2. Peralatan Yang Digunakan	73
4.3.3. Prosedur Pengujian	73
4.4. Pengujian LCD	75
4.4.1. Hasil Pengujian	77
4.5. Pengujian Sistem Keseluruhan.....	78
4.5.1. Tujuan	78
4.5.2. Prosedur Pengujian	78
4.5.3. Hasil Pengujian	79
4.6. Proses Identifikasi pada sistem anggota	79
4.7. Spesifikasi Alat	83
BAB V PENUTUP	85
5.1. Kesimpulan	85
5.2. Saran	85
DAFTAR PUSTAKA	86
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

2-1. Konfigurasi ID-12 (RFID Reader).....	7
2-2. <i>Transponder</i> dan <i>Transponder chip</i>	8
2-3. Komunikasi antara <i>Reader</i> dan <i>Transponder (Tag)</i>	10
2-4. Blok Diagram AT89S51	13
2-5. Konfigurasi Pin AT89S51	14
2-6. Modul FT232BM-USB.....	24
2-7. Konektor USB	25
2-8. Konektor Pin USB Tipe A	25
2-9. Diagram Detail Register Geser 8 Bit 74LS164	26
2-10. Konfigurasi Kaki-kaki 74LS164	27
2-11. Pin Out LCD M1632	29
2-12. Rangkaian Driver Buzzer.....	30
2-13. Relay	30
2-14 Relay SPST	31
2-15 Relay SPDT	31
2-16 Relay DPST	31
2-17 Relay DPDT.....	32
2-18. IDE (Integrated Development Enviroment	33
3-1. Diagram Blok Keseluruhan Sistem.....	36
3-2. Rangkaian RFID Reader	39
3-3. Perancangan Rangkaian <i>Reset</i>	41

3-4. Perancangan Rangkaian <i>clock</i>	42
3-5. Rangkaian MCU AT89S51	43
3-6. Perancangan Rangkaian LCD	44
3-7. Pemakaian PIN FT232	47
3-8. Kotak Dialog Fuond New Hardware Wizard.....	48
3-9. Kotak Dialog Memilih Instalasi	48
3-10. Kotak Dialog Lokasi Driver	49
3-11. Kotak Dialog Melanjutkan Instalasi.....	49
3-12. Kotak Dialog Proses Instalasi	49
3-13. Kotak Dialog Instalasi Driver FT232BM Selesai	50
3-14. Kotak Dialog Tampilan Device manager pada PC	50
3-15. Rangkaian Relay	51
3-16. Diagram Alir Keseluruhan.....	57
3-17. Diagram Alir Program Mikrokontroler	58
3-18. Diagram Alir Program Komputer	59
3-19. Awal data base delphi dekstop.....	60
3-20. Menentukan Tool Yang digunakan.....	60
3-21. Menentukan Nama Objek Data Base.....	61
3-22. Memilih Letak Data Base.....	61
3-23. Tampilan Setelah Penyetingan Nama dan Letak Data Base	62
3-24. Tampilan Menentukan Directory	62
3-25. Membuka Data Dari Directory.....	63
3-26. Jenis Data Base	63

3-27. Tampilan Data Base.....	64
3-28. Tampilan Delphi Anggota.....	64
3-29. Tampilan Catatan Absesnsi Pegawai	64
4-1. Kotak Dialog Connection Description	66
4-2. Kotak Dialog Connect To	66
4-3. Kotak Dialog COM 1 Properties	66
4-4. Identifikasi Reader Terhadap kartu	67
4-5. Blok Pengujian Modul USB.....	68
4-6. Kotak Dialog Fuond New Hardware Wizart.....	69
4-7. Kotak Dialog Memilih Instalasi.....	70
4-8. Kotak Dialog Lokasi Driver.....	70
4-9. Kotak Dialog Melanjutkan Instalasi.....	71
4-10. Kotak Dialog Proses Instalasi.....	71
4-11. Kotak Dialog Instalasi Driver FT232BM Selesai.....	72
4-12. Kotak Dialog Tampilan Device manager pada PC.....	72
4-13. Tampilan Delphi pada Saat Penyetingan Komponen Comport.....	74
4-14. Tampilan Delphi pada Saat Program diJalankan.....	74
4-15. Tampilan pada Modul USB.....	75
4-16. Blok Pengujian LCD	75
4-17. Hasil Pengujian LCD	77
4-18. Tampilan LCD Setelah Selulur Perangkat di hubungkan.....	80
4-19. Contoh tampilan Identitas Karyawan.....	80
4-20. Tampilan LCD Sukses Setelah READER Mendeteksi TAG	80

4-21. Tampilan LCD Gagal Setelah READER Mendeteksi TAG	81
4-22. Tampilan LCD Status Pulang Setelah TAG Terdeteksi READER .	81
4-23. Tampilan Layar Monitor Membuat Anggota Baru.....	82
4-24. Tampilan Layar Monitor Untuk Penyetingan Jam Masuk dan keluar	82
4-25. Tampilan settingan Port dan Baud Rate	83
4-26. Alat Absensi Karyawan.....	84

DAFTAR TABEL

2-1. Fungsi Pin & Format Data	8
2-2. Fungsi Alternatif Port	17
2-3. Rumus Penghitungan Baudrate pada Komunikasi Serial.....	22
2-4. Tabel Kebenaran IC 74LS164	27
2-5. Fungsi Pin – Pin LCD.....	29
4-1. Hasil Pengujian Pembacaan RFID.....	67

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sehubungan dengan perkembangan pola pikir manusia ilmu pengetahuan dan teknologi ternyata mengalami kemajuan terus menerus. Perbaikan terhadap teknologi yang sudah ada terus dilakukan agar menjadi lebih mudah. Salah satu bidang teknologi yang mengalami perkembangan lebih pesat adalah teknologi elektronika yang tidak terlepas dari tuntutan masyarakat yang terus – menerus berkembang sesuai dengan kondisi dan situasi yang dihadapi.

Seiring dengan perkembangan teknologi yang semakin berkembang, maka pemanfaatan teknologi dapat diterapkan dalam berbagai bidang, salah satunya adalah bidang perkantoran. Dalam sebuah perkantoran setiap karyawan pegawai negeri diharapkan untuk selalu dapat disiplin dalam tepat waktu dikala memasuki sebuah perkantoran, sehingga diperlukan suatu akses yang mudah dan cepat agar tidak saling tunggu-menunggu dengan karyawan pegawai negeri yang lainnya, sehingga diperlukan suatu inovasi yang menunjang kebutuhan tersebut, salah satunya adalah memanfaatkan teknologi *Wireless* yaitu RFID (*Radio Frequency Identification*) sebagai kartu absensi karyawan pegawai negeri.

Sistem absensinya yaitu setiap karyawan pegawai negeri yang melakukan absensi harus memiliki *tag RFID*, kemudian dengan bantuan *RFID reader* beserta mikrokontroler AT89S51, dan *LCD*, *tag* tersebut dapat diidentifikasi. Untuk menampilkan data dari karyawan pegawai negeri tersebut, berdasarkan kode dari

Tag RFID yang telah teridentifikasi dilakukan dengan bantuan komputer. Kemudian setelah proses absensi maka para karyawan pegawai negeri tersebut dapat melakukan tugasnya sebagaimana mestinya.

1.2. Rumusan Masalah

Mengacu pada permasalahan yang diuraikan pada latar belakang, maka rumusan masalah ditekankan pada:

1. Bagaimana menggunakan RFID sebagai masukan pada mikrokontroler.
2. Bagaimana membuat sistem absensi karyawan pegawai negeri agar dapat ditampilkan pada sebuah monitor.
3. Bagaimana merancang dan membuat perangkat keras (*hardware*) dan perangkat lunak (*software*) agar sistem bekerja dengan baik.

1.3. Batasan Masalah

Dengan mengacu pada permasalahan yang telah dirumuskan, maka hal-hal yang berkaitan dengan masalah tersebut dibatasi sebagai berikut :

1. Data berisi: Jabatan, ID Karyawan, Nama Karyawan, Foto Karyawan, Jenis Kelamin, Jam Masuk, Jam Keluar, Alamat, Berapakah Mendapatkan Sangsi Dan No Hp.
2. Tidak membahas catu daya dan frekuensi-frekuensi radio.
3. Mikrokontroler yang digunakan adalah AT89S51, *RFID reader* yang digunakan ID-12, *LCD* yang digunakan adalah tipe M163.
4. Tidak membahas InfraRed.

5. Yang diinformasikan hanya data absensi masuk dan pulang yang berisi nama dan ID karyawan pegawai negeri dan foto pegawai.

1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah untuk meningkatkan kedisiplinan karyawan pegawai negeri di perkantoran serta dengan mudah dan cepat sehingga tidak membutuhkan waktu yang lama untuk absensi, sehingga penulis mencoba merancang dan membuat Sistem Absensi Pada Karyawan Pegawai Negeri yang berbasis RFID dengan Tampilan *LCD* dan diinterface ke PC melalui USB.

1.5. Metodologi Penulisan

Metode yang digunakan dalam penulisan skripsi ini adalah

1. Studi Pustaka

Memperoleh data dengan cara membaca dan mempelajari buku *literature* yang berhubungan dengan penyusunan skripsi ini.

2. Studi Lapangan

Memperoleh data dengan cara praktek secara langsung untuk menunjang pembuatan alat.

3. Pengolahan Data

Mengolah data dengan jalan membuat analisa dan menarik kesimpulan dari hasil pengujian yang ada.

1.6. Sistematika Penulisan

Sistematika pembahasan dari skripsi ini terdiri dari pokok pembahasan yang saling berkaitan antara satu dengan lainnya, yaitu :

BAB I Pendahuluan

Pada bab ini dibahas tentang latar belakang permasalahan, rumusan masalah, batasan masalah, metodologi dan sistematika penulisan dari alat yang direncanakan.

BAB II Landasan Teori

Pada bab ini dibahas tentang teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Perencanaan Dan Pembuatan Alat

Pada bab ini dibahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*).

BAB IV Pengujian Alat

Pada bab ini dibahas tentang proses serta hasil dari pengujian alat, yang didasarkan oleh pengukuran-pengukuran yang diperlukan.

BAB V Penutup

Pada bab ini akan disampaikan kesimpulan dan saran dari perencanaan dan pembuatan sistem ini.

BAB II

TEORI DASAR

2.1. Pendahuluan

Pada bab ini akan dibahas mengenai teori dasar yang berkaitan dengan sistem. Teori dasar ini akan membahas tentang komponen dan peralatan pada alat yang dibuat.

2.2. RFID (*Radio Frequency Identification*)

RFID merupakan sebuah sistem yang mampu mengirimkan identitas secara otomatis dengan menggunakan gelombang radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah devais kecil yang disebut *tag / transponder (transmitter + responder)*. *Tag RFID* akan mengenali diri sendiri ketika mendeteksi sinyal dari devais yang kompatibel, yaitu pembaca RFID (*RFID Reader*) dengan *Range* kisaran pembacaan yang bekerja pada frekuensi 125 KHz.

2.2.1. Komponen *RFID Tag*

- ❖ *RFID Tag* atau *transponder*, yang menampung identifikasi data obyek.
- ❖ *RFID tag reader* atau *transceiver* yang berfungsi untuk membaca dan menulis data *tag*.
- ❖ *Server database* yang menyimpan kumpulan *record* isi dari *tag*.

2.2.1.1. *Tag*

Tag tersusun dari *microchip* yang berfungsi untuk menyimpan dan sebuah antena *chip* mikro itu sendiri yang ukurannya sekitar 0.4 mm. *Chip* tersebut menyimpan nomor seri yang unik atau informasi lainnya tergantung tipe memorinya yaitu *read-only* dan *read-write*.

Klasifikasi *tag* dibedakan menjadi tiga yaitu :

- ❖ *Tag* aktif : mempunyai sumber tenaga seperti baterai dan dapat dilakukan komunikasi untuk dibaca dan ditulis.
- ❖ *Tag semi-pasif* : mempunyai baterai tetapi hanya merespon transmisi yang datang (*incoming transmissions*).
- ❖ *Tag* pasif : menerima tenaga dari *reader*, antena yang akan menjadi sumber tenaga dengan memanfaatkan medan magnet yang ditimbulkan dari pembaca (*reader*).

Empat macam frekuensi yang digunakan RFID tag adalah: tag frekuensi rendah (125 atau 134.2 KHz), tag frekuensi tinggi (13.56 MHz), tag UHF (868 sampai 956 MHz) dan tag gelombang mikro (2.45 GHz).

2.2.1.2. *Tag Reader*

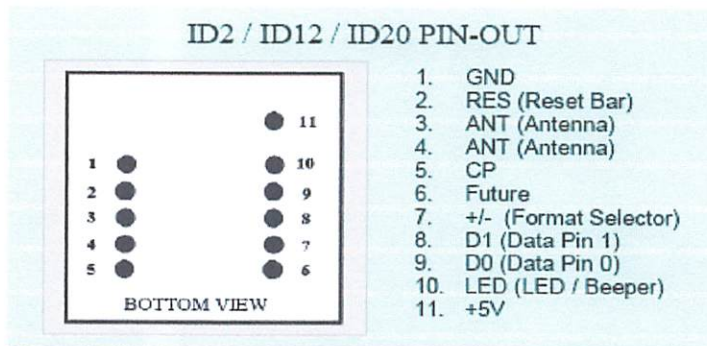
Tag reader digunakan untuk membaca data yang ada pada *tag* melewati RF *interface*. Untuk menambah fungsi *reader* dilengkapi dengan *internal storage*, dan aplikasi perangkat lunak untuk menyimpan data pada *server database*. Pada prakteknya *tag reader* dapat berupa perangkat keras yang terletak pada 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* yang keluar. Pada dasarnya *tag reader* merupakan suatu peralatan yang sederhana dan dapat digabungkan kedalam perlengkapan *mobile* seperti telepon.

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

Spesifikasi Reader ID-12 :

- Power Requirement : 5V@13mA nominal
- Card Format : Temec Q55555 or compatible
- Frequency : 125 KHZ
- Encoding : Manchester 62bit, modulus64
- I/O Output Current : 20mA sink/source
- Drive Current : 300 mA
- Antenna : 100 Volt PKPK



Gambar 2.1. Konfigurasi ID-12 (RFID Reader) ^[1]

Tabel 2-1. Fungsi Pin & Format Data^[1]

Pin No	Description	ASCII
Pin 1	Zero Volt and Tuning Capacitor Ground	GND 0 V
Pin 2	Strap to +5 Volt	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna
Pin 4	To External Antenna	Antenna
Pin 5	Card Present	No Fuction
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	+5 V

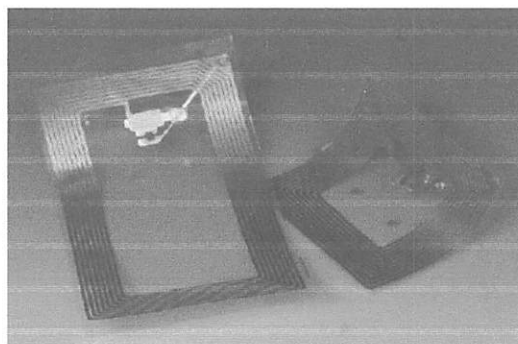
2.2.1.3. *Server Database*

Untuk menyimpan data yang ada pada *tag* digunakan *server database*.

2.2.2. Mekanisme RFID

2.2.2.1. Prinsip Kerja *Reader* dan *Tag/Transpoder*

Suatu *transponder* secara induktif yang tergabungkan terdiri atas suatu data elektronik di dalam suatu *michrochip* yang pada umumnya tunggal dan suatu *coil area* besar yang berfungsi sebagai suatu antena.



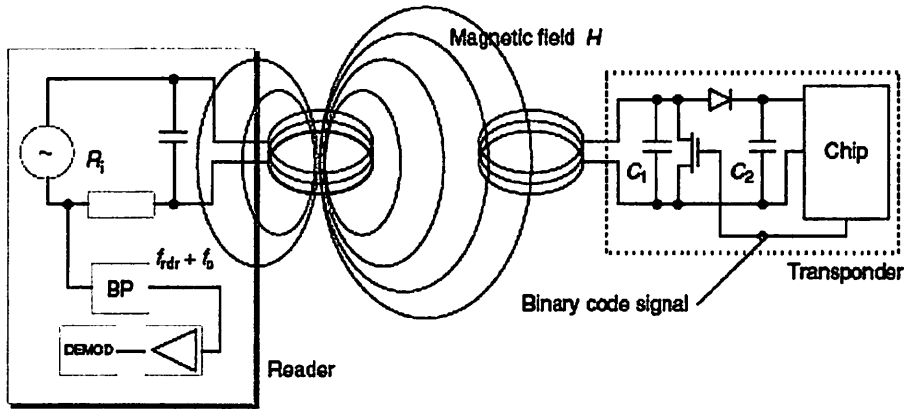
Gambar 2-2. *Transponder* dan *Transponder chip*^[2]

Secara induktif *transponders* dioperasikan dengan *pasif* yaitu semua energi yang diperlukan untuk operasi *microchip* harus disajikan oleh pembaca *reader*. Antena pembaca menghasilkan suatu bidang elektro magnet frekuensi, yang menembus penampang-lintang area *coil* dan area di sekitar *coil* itu. Sebab panjang gelombang cakupan frekuensi menggunakan *low frekuensi* (125 kHz – 135 kHz)

Suatu bidang elektro magnet yang dipancarkan menembus *coil* antena *transponder*, yang mana saat terinduksi, suatu tegangan dihasilkan *coil antena transponder*. Tegangan ini berfungsi sebagai power untuk pengaktifan data dalam *microchip*.

Suatu kapasitor C yang dihubungkan paralel dengan *coil* antena pembaca berkombinasi dengan induksi *coil* antena untuk membentuk suatu rangkaian resonansi paralel, dengan suatu frekuensi resonan yang sesuai dengan frekuensi transmisi pembaca yang dihasilkan di dalam *coil* antena pembaca akan meningkatkan rangkaian resonan yang paralel tersebut, yang dapat digunakan untuk menghasilkan kekuatan bidang elektro magnet.

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



Gambar 2-3. Komunikasi antara Reader dan Transponder (Tag) ^[2]

2.2.2.2. Pengiriman Data

Saat model alat identifikasi sangatlah bermacam-macam, ada yang berupa kartu dengan lubang, *barcode*, *RFID*, dll. *RFID* (*RF Identification*) merupakan suatu alat untuk identifikasi yang biasanya ditempelkan pada barang atau dibuat menjadi kartu. Pembacaan format data yang dikeluarkan oleh *RFID reader* dengan format *output ASCII*.

RFID reader mempunyai banyak sekali tipe, antara lain : ID-10, ID-12, EM-13, dll. Biasanya *RFID reader* ini memiliki dua bentuk *output serial* yaitu : ASCII dan *Wiegand 26-bit*. Yang sering digunakan adalah *output* dengan format ASCII, karena *output* ini sangat mudah untuk dihubungkan pada mikrokontroler atau *PC* menggunakan komunikasi serial UART.

2.2.2.3. Format Pembacaan ASCII

Output yang memiliki format *ASCII* memiliki struktur sebagai berikut :

02	10 Data Karakter ASCII	2 Karakter ASCII (Checksum)	CR	LF	03
----	------------------------	-----------------------------	----	----	----

Checksum merupakan hasil EXOR (*Exclusive OR*) dari 5 biner data *byte*, misalnya data *output serial* (dalam *hexadesimal*) yang kita tangkap adalah sebagai berikut :

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

Langkah pertama adalah merubah semua nilai data diatas menjadi karakter *ASCII*. Misalnya 30H menjadi karakter “0”, 34H menjadi karakter “4” dst. Langkah kedua adalah menyusun data - data tersebut kedalam format data *ASCII*. Dari contoh data *hexadesimal* diatas maka dapat dibuat tabel seperti tabel dibawah ini :

Data Hexsa	30	34	36	32	30	31	44	37	36	43
Data ASCII	0	4	6	2	0	1	D	7	6	C

Untuk data yang pertama yaitu angka “04” merupakan data untuk jenis – jenis kartu. Yang digunakan adalah data ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah “ 6201D76C “. Kemudian gabungkan data karakter *ASCII* menjadi bilangan *hexadesimal*, setelah itu konversikan bilangan *hexadesimal* tersebut ke desimal. Hasilnya adalah 6201D76C Heksa menjadi 1644287852, angka–angka ini merupakan no kartu sebenarnya yang tertera pada badan kartu yang disebut *tag RFID*.

2.3. Mikrokontroler AT89S51

AT89S51 adalah *microcontroller* keluaran Atmel dengan 4K *byte* Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89S51

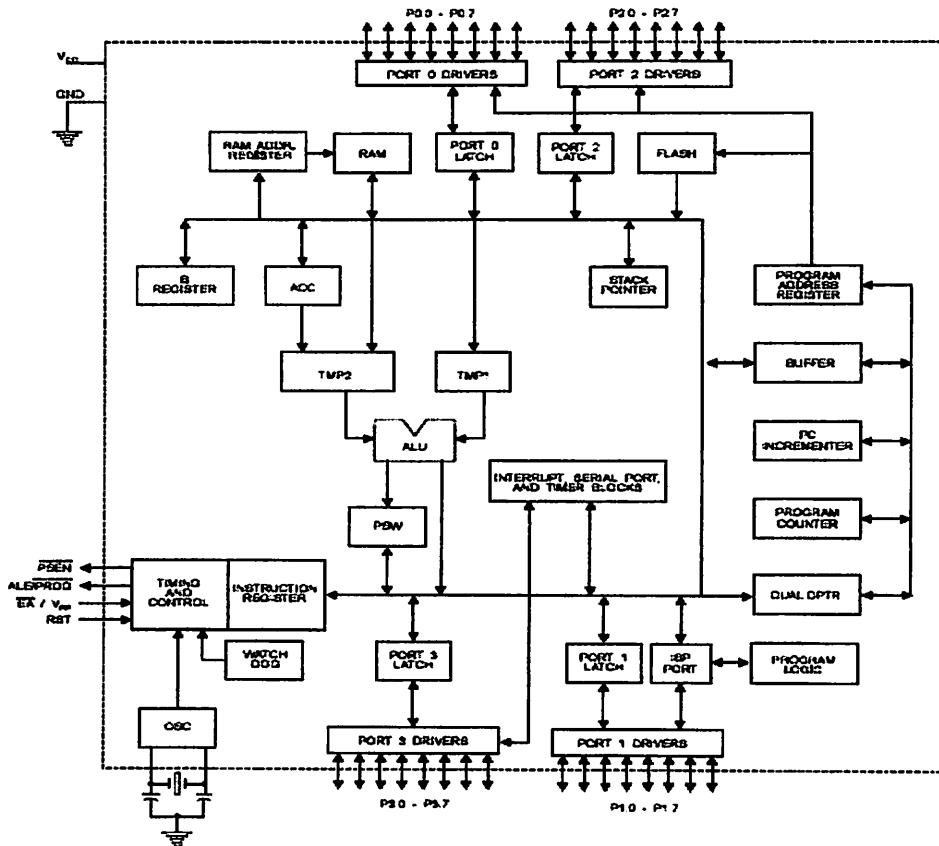
merupakan memori dengan teknologi *nonvolatile memory*, isi memori tersebut dapat diisi ulang ataupun dihapus berkali-kali.

Memori ini biasa digunakan untuk menyimpan instruksi (perintah) berstandar MCS-51 code sehingga memungkinkan *mikrokontroller* ini untuk bekerja dalam mode *single chip operation* (mode operasi keping tunggal) yang tidak memerlukan *external memory* (memori luar) untuk menyimpan source code tersebut.

Sebagai suatu sistem kontrol, mikrokontroler AT89S51 bila dibandingkan dengan mikroprosesor memiliki kemampuan dan segi ekonomis yang bisa diandalkan karena dalam mikrokontroler sudah terdapat RAM dan ROM sedangkan mikroprosesor didalamnya tidak terdapat keduanya. Secara umum konfigurasi yang dimiliki mikrokontroler AT89S51 adalah sebagai berikut :

- Sebuah CPU 8 bit dengan menggunakan teknologi dari Atmel.
- Memiliki memori baca-tulis (RAM) sebesar 128 *byte*.
- Empat buah *programmable port I/O*, masing-masing terdiri atas 8 buah jalur I/O.
- Sebuah *port serial* dengan kontrol *full duplex*.
- *Reprogrammable Flash memory* yang besarnya 4 kbyte untuk memori program.
- Mampu beroperasi sampai 33 MHz.

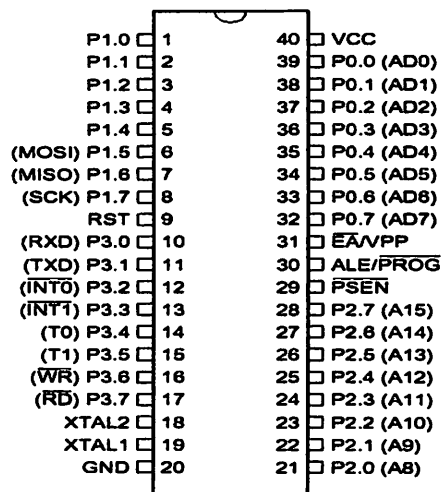
AT89S51 adalah mikrokontroler mempunyai kompatibilitas instruksi dan konfigurasi pin dengan mikrokontroler MCS-51. Blok diagram MCS-51 ditunjukkan pada Gambar 2.4.



Gambar 2.4. Blok Diagram AT89S51^[3]

2.3.1. Konfigurasi pin AT89S51

Masing-masing kaki dalam mikrokontroler AT89S51 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89S51, perancangan aplikasi mikrokontroler AT89S51 akan lebih mudah merencanakan dan membuat sistem yang dirancang. AT89S51 mempunyai 40 pin, susunan masing-masing pin dapat dilihat pada Gambar 2.5.



Gambar 2.5. Konfigurasi Pin AT89S51^[3]

Fungsi kaki-kaki AT89S51 adalah:

1. *Port 1* (Pin 1..8), merupakan saluran masukan/keluaran dua arah.
2. Pin 9 RST, merupakan saluran dua masukan untuk mereset mikrokontroler dengan cara memberi masukan logika tinggi.
3. *Port 3* (Pin 10..17), merupakan saluran masukan/keluaran dua arah dan mempunyai fungsi khusus. Fungsi khusus meliputi TXD (*Transmit Data*), RXD (*Receive Data*), $\overline{INT0}$ (*Interrupt 0*), $\overline{INT1}$ (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), \overline{WR} (*Write*), \overline{RD} (*Read*).
4. Pin 18 dan 19 (XTAL₁ dan XTAL₂), merupakan saluran untuk mengatur pewaktuan sistem. Untuk pewaktuan dapat menggunakan pewaktuan internal maupun eksternal.
5. Pin 20 V_{SS}, merupakan hubungan ke *ground* dari rangkaian.

6. *Port 2* (Pin 21..28), merupakan saluran masukan/keluaran dua arah.
7. Pin 29 $\overline{\text{PSEN}}$ (*Program Store Enable*), merupakan sinyal baca untuk mengaktifkan memori program eksternal.
8. Pin 30 $\overline{\text{ALE/PROG}}$ (*Address Latch Enable*), merupakan pulsa yang berfungsi untuk menahan alamat rendah (A0-A7) dalam *port 0*, selama proses baca/tulis memori eksternal. Frekuensi ALE adalah $\frac{1}{6}$ kali frekuensi osilator, dan dapat digunakan sebagai pewaktu. Pin ini juga berfungsi sebagai saluran program selama dilakukan pemrograman jika menggunakan memori program internal.
9. Pin 31 $\overline{\text{EA}}/\text{VPP}$ (*External Access Enable*), untuk mengatur penggunaan memori program eksternal dan internal. Pin ini harus dihubungkan dengan *ground* bila menggunakan memori program eksternal dan dihubungkan dengan VPP sebesar 5 volt jika menggunakan memori program internal.
10. *Port 0* (Pin 32..39), merupakan saluran masukan/keluaran *open drain*.
11. Pin 40 V_{CC} , merupakan saluran masukan untuk catu daya positif sebesar 5 volt DC dengan toleransi lebih kurang 10%.

2.3.2. Struktur dan Operasi Port

Mikrokontroler AT89S51 memiliki 4 buah port. Setiap port memiliki 8 buah jalur I/O yang bersifat *bidirectional*. Beberapa karakteristik port mikrokontroler AT89S51 dijelaskan secara singkat sebagai berikut:

1. Port 0 merupakan port I/O 8-bit yang tidak mempunyai *pull-up* internal. Sebagai sebuah keluaran, maka setiap pin juga dapat mengendalikan 8 beban TTL. Port 0 juga dapat digunakan untuk memultipleks address bus rendah dan data memori dengan menggunakan *pull-up* internal. Selain itu, port 0 juga menerima kode mesin (dalam byte) selama pemrograman EPROM dan mengeluarkan kode mesin selama program verifikasi dari EPROM. Selama program verifikasi dibutuhkan *pull-up* eksternal. Pada port ini berlaku ketentuan yang berbeda dengan port-port lain, yaitu bila digunakan sebagai keluaran harus diberikan tambahan resistor *pull-up*.
2. Port 1 merupakan sebuah port I/O *bidirectional* yang mempunyai *pull-up* internal. Buffer keluaran dari port 1 dapat mengendalikan 4 beban TTL. Pin-pin dari port 1 dapat juga digunakan sebagai masukan jika di *pull-up* tinggi oleh *pull-up* internal dan jika *pull-up low* internal. Port 1 juga menerima address bus rendah (dalam byte) selama pemrograman EPROM dan selama program verifikasi dari EPROM.
3. Port 2 dapat dipergunakan sebagai input atau output seperti pada port 1. Alternatif lain dari port 2 dapat dipergunakan sebagai address bus high byte pada saat mengakses memori eksternal.

4. Port 3 merupakan sebuah port I/O 8-bit bidirectional yang mempunyai *pull-up* internal. Buffer keluaran dari port 3 dapat mengendalikan dan menghasilkan arus I_L karena adanya *pull-up* internal. Port 3 juga mempunyai fungsi yang lain seperti yang tertera dalam Tabel 2.2.

Tabel 2.2. Fungsi Alternatif Port 3^[3]

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

2.3.3. Antarmuka Serial

Port serial pada AT89S51 bersifat dupleks-penuh atau *full duplex*, artinya port serial bisa menerima dan mengirim secara bersamaan. Selain itu juga memiliki penyangga penerima, artinya port serial mulai bisa menerima *byte* yang kedua sebelum *byte* pertama dibaca oleh register penerima (jika *byte* yang kedua selesai diterima sedangkan *byte* pertama belum dibaca, maka salah satu *byte* akan hilang). Penerimaan dan pengiriman data port serial melalui register SBUF. Penulisan ke SBUF berarti mengisi register pengiriman SBUF sedangkan pembacaan dari SBUF berarti membaca register penerimaan SBUF yang memang terpisah secara fisik (secara perangkat lunak namanya menjadi satu yaitu SBUF).

Port serial pada AT89S51 bisa digunakan dalam 4 mode kerja yang berbeda. Dari 4 mode tersebut, 1 mode diantaranya bekerja secara sinkron dan 3 lainnya bekerja secara asinkron. Keempat mode kerja tersebut adalah :

1. Mode 0 : Mode ini bekerja secara sinkron, data serial dikirim dan diterima melalui kaki P3.0 (RxD), sedangkan kaki P3.1 (TxD) dipakai untuk menyalurkan detak pendorong data serial yang dibangkitkan AT89S51. Data dikirim/diterima 8 bit sekaligus, dimulai dari bit yang bobotnya paling kecil atau LSB (bit 0) dan diakhiri dengan bit yang bobotnya paling besar atau MSB (bit 7). Kecepatan pengiriman data (*baudrate*) adalah 1/12 frekuensi kristal yang digunakan.
2. Mode 1 : Pada mode ini tetap yaitu, data dikirim dan diterima melalui kaki P3.0 (RxD), secara asinkron (juga mode 2 dan 3). Pada Mode 1 data dikirim/diterima 10 bit sekaligus, diawali dengan 1 bit *start*, disusul dengan 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), diakhiri dengan 1 bit *stop*. Pada AT89S51 yang berfungsi sebagai penerima bit *stop* adalah RB8 dalam register SCON. Kecepatan pengiriman data (*baud rate*) bisa diatur sesuai dengan keperluan. Mode inilah (mode 2 dan 3) yang umum dikenal sebagai UART(*Universal Asynchronous Receiver/Transmitter*).
3. Mode 2 : Data dikirim/diterima 11 bit sekaligus, diawali dengan 1 bit *start*, disusul 8 bit data yang dimulai dari bit yang bobotnya paling kecil (bit 0), kemudian bit ke 9 yang bisa diatur lebih lanjut, diakhiri dengan 1 bit *stop*. Pada AT89S51 yang berfungsi sebagai penerima, bit 9 ditampung pada bit

RB8 dalam register SCON, sedangkan bit *stop* diabaikan tidak ditampung. Kecepatan pengiriman data (*baud rate*) bisa dipilih antara 1/32 atau 1/64 frekuensi kristal yang digunakan.

4. Mode 3 : mode ini sama dengan mode 2, hanya saja kecepatan pengiriman data (*baud rate*) bisa diatur sesuai dengan keperluan, seperti halnya Mode 1. Pada mode asinkron (mode 1, mode 2, mode 3), port AT89S51 bekerja secara *fullduplex*.

2.3.4. Pengaturan Baud Rate Port Serial

Baud rate ini merupakan *baud rate* pada komunikasi data yang digunakan pada port serial mikrokontroler AT89S51. *Baud rate* sebanding dengan frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data. Komunikasi asinkron tidak memerlukan sinyal *clock* sebagai sinkronisasi, namun pengiriman data ini harus diawali dengan start bit dan diakhiri dengan stop bit. Sinyal *clock* yang merupakan *baud rate* dari komunikasi data ini dibangkitkan oleh masing-masing baik penerima maupun pengirim data dengan frekuensi yang sama.

Satuan *baud rate* pada umumnya adalah bps (*bit per second*), yaitu jumlah bit yang dapat ditransmisikan per detik. *Baud rate* untuk mode 0 dan mode 2 bernilai tetap yaitu untuk mode 0 adalah 1/12 frekuensi osilator dan mode 2 adalah 1/64 frekuensi osilator. Dengan mengubah bit SMOD yang terletak pada register PCON menjadi set (kondisi awal saat sistem reset adalah clear), *baud rate* pada mode 1,2, dan 3 akan berubah menjadi dua kali lipat.

Baudrate untuk mode 0 nilainya tetap dan mengikuti persamaan berikut :

$$\text{Baudrate Mode 0} = \frac{\text{Frekuensi_osilator}}{12} \quad (\text{bps})$$

Dengan,

Baud Rate Mode 0 = Frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data dengan mode 0 (bps)

Frekuensi Kristal = Frekuensi sumber *clock* eksternal (MHz)

Baudrate untuk Mode 2 bergantung pada nilai bit SMOD pada register *Power Control Register* (PCON). Jika SMOD=0, *baudrate*-nya 1/64 frekuensi kristal, jika SMOD=1 maka *baudrate*-nya 1/32 frekuensi kristal, atau dengan kata lain, *baudrate* untuk mode 2 ini mengikuti persamaan:

$$\text{Baudrate Mode 2} = \frac{2^{\text{SMOD}}}{64} \times \text{Frekuensi_osilator} \quad (\text{bps})$$

Dengan,

Baud Rate Mode 2 = Frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data dengan mode 2 (bps)

SMOD = Serial port enable bit (0 atau 1)

Frekuensi Kristal = frekuensi sumber *clock* eksternal (MHz)

Pada saat *Timer 1* digunakan sebagai generator *baudrate*, maka *baudrate* Mode 1 dan 3 ditentukan berdasar laju timpahan *Timer 1* dan nilai SMOD dengan persamaan:

$$\text{Baudrate Mode 1 \& 3} = \frac{2^{\text{SMOD}}}{32} \times (\text{laju Timpahan Timer 1}) \quad (\text{bps})$$

Dengan,

Baud Rate = Frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data (bps)

SMOD = Serial port enable bit (0 atau 1)

Laju timpahan Timer 1 (kali/detik)

Interupsi *Timer 1* sebaiknya dimatikan untuk aplikasi ini (*Timer 1* digunakan sebagai generator *baudrate*). *Timer 1* ini sendiri dapat dikonfigurasi baik sebagai pewaktu atau pencacah. Pada umumnya *Timer 1* dikonfigurasi sebagai pewaktu dan *baudrate*-nya mengikuti persamaan :

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Frekuensi Osilator}}{12 \times (256 - (\text{TH1}))} \text{ (bps)}$$

Dengan,

Baud Rate = Frekuensi *clock* yang digunakan dalam pengiriman dan penerimaan data (bps)

SMOD = Serial port enable bit (0 atau 1)

Frekuensi Osilator = frekuensi sumber *clock* eksternal (MHz)

TH1 = Nilai 8-bit *reload* pada *Timer 1* (bit)

Satu hal yang harus diperhatikan dalam pengaturan *baud rate* adalah nilai *baud rate* dan nilai TH1 diusahakan harus tepat dan bukan merupakan pembulatan. Untuk komunikasi serial kecepatan tinggi, pembulatan terhadap nilai-nilai tersebut dapat mengakibatkan kekacauan dalam proses pengiriman atau penerimaan. Jika terdapat nilai pecahan, user disarankan untuk mengganti osilator dengan frekuensi yang sesuai. Untuk komunikasi dengan kecepatan rendah, toleransi terhadap kesalahan cukup besar sehingga pembulatan masih boleh dilakukan.

Tabel 2.3. Rumus Penghitungan Baudrate pada Komunikasi Serial^[3]

Mode	Baud Rate	
0	1/12 f osilator	
1	$\text{SMOD} = 0$ $\text{Baudrate} = \frac{f_{osc}}{12 \times [256 - TH1] \times 32}$	$\text{SMOD} = 1$ $\text{Baudrate} = \frac{f_{osc}}{12 \times [256 - TH1] \times 16}$
2	1/64 f	1/32 f
3	$\text{Baudrate} = \frac{f_{osc}}{12 \times [256 - TH1] \times 32}$	$\text{Baudrate} = \frac{f_{osc}}{12 \times [256 - TH1] \times 16}$

2.4. Modul FT232BM-USB

Perkembangan teknologi komputer dimana seiring dengan perkembangan ilmu pengetahuan yang semakin cepat pada PC atau Laptop – laptop untuk keluaran terbaru jumlah port serial RS232 semakin lama semakin berkurang, jika pada PC lama biasanya terdapat dua buah konektor RS232 maka sekarang hanya terdapat satu buah konektor saja sehingga keberadaan port serial RS232 sekarang telah digantikan oleh port USB yang mempunyai banyak kelebihan dibandingkan port serial RS232. adapun contoh pengiriman informasi secara serial melalui sebuah mikrokontroler yang dikirimkan ke PC melalui port USB seperti Modul FT232BM-USB yaitu modul interface yang digunakan untuk aplikasi dari mikrokontroler ke USB, hubungan ini dilakukan secara serial dengan kata lain sebuah modul yang dapat mengkonversikan data serial yang berasal dari mikrokontroler. secara garis besar modul ini berfungsi untuk mengubah data USB yang berasal dari port USB menjadi data serial dengan level tegangan TTL sehingga pengguna dapat melakukan komunikasi data serial (UART) melalui port USB. Keunggulan digunakannya modul ini adalah

kemampuannya untuk mengirim data lebih cepat dibandingkan dengan komunikasi serial dengan menggunakan Port RS232 serta untuk kecepatan transfer data serial yang dapat di pakai oleh IC ini yaitu sebesar 300bps sampai 9600 bps.

Modul FT232BM-USB mempunyai spesifikasi sebagai berikut:

1. Memiliki tegangan kerja 4,4- 5,25 volt DC.
2. Tersedia 2 LED untuk indicator Tx dan Rx data pada komunikasi serial.
3. Memiliki boudrate 3Mbps (TTL), 1Mbps (RS-232), 3Mbps (RS-422/RS-485).
4. Pin sinyal kontrol (arah) untuk komunikasi RS-485 yang bekerja secara otomatis.
5. Kompatibel dengan USB 1.1 dan USB 2.0.

Modul menggunakan konfigurasi daya Self Powered.

6. Memiliki output dengan level TTL 5 volt .
7. Memiliki EEPROM eksternal untuk menyimpan data PID,VID,nomor serial,dan deskripsi produk.pengisian datanya melalui USB.
8. Virtual COM port driver (VCP) dan D2xx (USB Direct Drivers + DLL S/W Interface) untuk windows 98,98SE,ME,2000 dan XP.
9. Mendukung format UART dengan 7/8 bit data , 1 / 2 stop bit dan Odd / Even / Mark / Space / No Parity.

Adapun gambar fisik dari modul FT232BM-USB dapat digambarkan sebagai berikut:

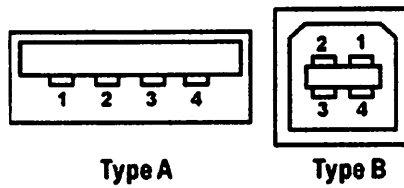


Gambar 2.6. Modul FT232BM-USB^[4]

Gambar diatas merupakan modul dari FT232BM-USB secara fisik yang terdiri dari konektor USB TIPE B dan chip FT232BM sebagai komponen utama dari modul COM to USB tersebut serta komponen-komponen pendukung lainnya.

2.4.1. Kabel USB

Di dalam menghubungkan modul FT232BM-USB dengan komputer maka dibutuhkan kabel konektor USB dimana hanya terdapat ada 2 macam konektor yang digunakan dalam menghubungkan modul tersebut, yaitu konektor type A dan konektor type B seperti terlihat dalam Gambar 1. Konektor type A dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada bagian computer sedangkan Konektor type B dipakai untuk menghubungkan kabel USB ke terminal USB yang ada pada modul FT232BM-USB sedangkan untuk peralatan USB yang sederhana, misalnya mouse, biasanya tidak pakai konektor B melainkan konektor tipe A hal ini dilakukan untuk menghemat biaya sehingga kabel langsung dihubungkan ke bagian dalam mouse.

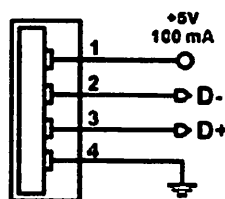


Gambar 2.7. Konektor USB^[4]

Kabel USB terdiri dari 4 kabel ditambah konduktor pembungkus kabel seperti pelindung yang biasanya dijumpai dalam kabel audio.

Kabel nomor 1 dipakai untuk menyalurkan sumber daya dengan tegangan 5 Volt, jika diperlukan peralatan USB boleh mengambil daya dari saluran ini tidak lebih dari 100 mA. Komputer yang dilengkapi dengan kemampuan USB, wajib menyediakan daya sebesar 500 mA untuk keperluan ini. Peralatan USB yang memerlukan daya lebih dari ketentuan tersebut di atas, harus menyediakan sendiri sumber daya untuk keperluan kerja peralatan tersebut.

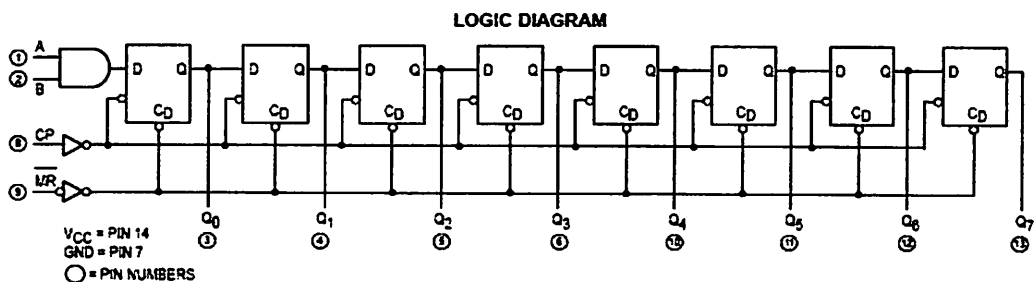
Kabel nomor 4 adalah ground sebagai saluran balik sumber tegangan 5 Volt. Kabel nomor 2 dan nomor 3 dipakai untuk pengiriman sinyal. Kabel nomor 2 bernama D- dan kabel nomor 3 bernama D+, tegangan pada dua saluran ini berubah antara 0 Volt dan 3,3 Volt seperti terlihat pada gambar 2.8 berikut:



Gambar 2.8 Konektor Pin USB Tipe A ^[4]

2.5. C 74LS164

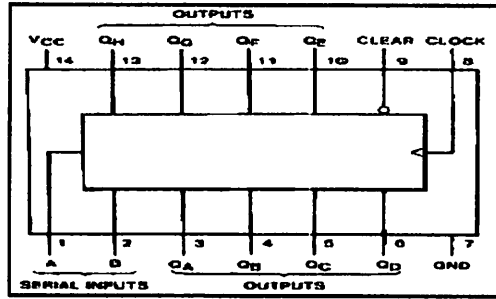
Bagian ini akan merinci salah satu dari sekian banyak register geser, yaitu register yang masukannya seri dan keluarannya paralel 8 bit. IC yang digunakan adalah IC 74LS164 yang merupakan register 8 bit yang terpicu sentuh dengan masukan serial. Semua keluaran paralel tersedia atas setiap *flip-flop D internal*. Diagram rinci tersedia dalam gambar 2.9, memperlihatkan penggunaan delapan *flip-flop D internal*, masing-masing dengan keluaran data paralelnya (QA sampai QB).



Gambar 2.9

Diagram Detail Register Geser 8 Bit 74LS164^[6]

IC 74LS164 terlihat memiliki sebuah masukan serial. Data dimasukkan secara serial melalui salah satu dari dua masukan (A dan B) di NAND-kan. Kedua masukan ini bisa digabung menjadi satu masukan atau bisa juga salah satu ditentukan sebagai logika tinggi, sedangkan yang satu lagi untuk pemasukan data. Masukan pengatur ulang utama *Clear* untuk IC 74LS164 merupakan input aktif rendah. Dalam tabel kebenaran (Tabel 2.4) memperlihatkan bahwa bila diaktifkan, masukan *Clear* akan mematikan semua input yang lainnya, dan mengembalikan semua *flip-flop* ke 0. IC 74LS164 menggeser data suatu tempat kekanan dalam setiap peralihan rendah ke tinggi dalam masukan *Clock*.



Gambar 2.10

Konfigurasi Kaki-Kaki 74LS164 ^[6]

Untuk tabel kebenaran dari IC jenis dapat dilihat pada tabel berikut

Tabel 2.4. Tabel Kebenaran IC 74LS164 ^[6]

Inputs				Outputs			
Clear	Clock	A	B	Q _A	Q _B	...	Q _H
L	X	X	X	L	L	...	L
H	L	X	X	Q _{A0}	Q _{B0}	...	Q _{H0}
H	↑	H	H	H	Q _{An}	...	Q _{Gn}
H	↑	L	X	L	Q _{An}	...	Q _{Gn}
H	↑	X	L	L	Q _{An}	...	Q _{Gn}

H = Logika Tinggi

L = Logika Rendah

X = Sembarang masukan termasuk transisi

↑ = Transisi dari rendah ke tinggi

Q_{A0}, Q_{B0}, Q_{H0} = masing-masing logika pada Q_A, Q_B atau Q_H sebelum terjadi kondisi-kondisi jalan masuk stasioner yang ditunjukkan.

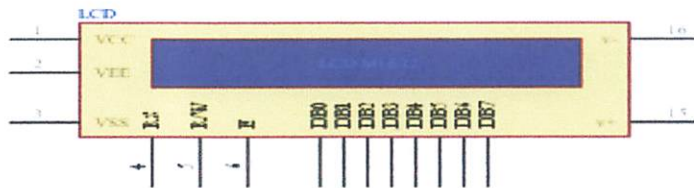
Q_{An}, Q_{Gn} = logika Q_A atau Q_G sebelum transisi pada *clock* yang paling akhir, menunjukkan penggeseran satu bit.

2.6. LCD (Liquid Crystal Display) M1632

Liquid Crystal Display adalah modul tampilan yang mempunyai konsumsi daya yang relatif rendah dan terdapat sebuah kontroller CMOS didalamnya. Kontroller tersebut sebagai pembangkit ROM/RAM dan display data RAM. Semua fungsi tampilan di kontrol oleh suatu instruksi modul LCD dapat dengan mudah diinterfacekan dengan MPU.

Spesifikasi dari LCD M1632:

- ❖ Terdiri dari 32 karakter yang dibagi menjadi 2 baris dengan display dot matrik 5 X 7 ditambah cursor
- ❖ Karakter generator ROM dengan 192 karakter
- ❖ Karakter generator RAM dengan 8 tipe karakter
- ❖ 80 X 8 bit display data RAM
- ❖ Dapat diinterfacekan dengan MPU 8 atau 4 bit
- ❖ Dilengkapi fungsi tambahan : Display clear, cursor home, display ON/OFF, cursor ON/ OFF, display character blink, cursor shift dan display shift
- ❖ Internal data
- ❖ Internal otomatis dan reset pada power ON
- ❖ +5 V power supply tunggal



Gambar 2.11

Pin Out LCD M1632 [5]

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

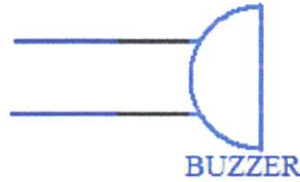
Tabel 2.5. Fungsi Pin – Pin LCD^[7]

Nama Pin	Jumlah	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Tri state bidirectional lower data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
DB4-DB7	4	I/O	MPU	Tri state bidirectional upper fourdata bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
E	1	Input	MPU	Sinyal operasi dimulai: sinyal aktif baca/tulis
R/W	1	Input	MPU	Sinyal pilih data dan tulis (0:tulis,1:baca)
RS	1	-	Power supply	Sinyal pilih register 0: Instruction register (write) Busy flag dan address counter (read) 1:Data register (write dan read)
VLC	1	-	Power supply	Penyetelan kontras pada tampilan LCD
VDD	1	-	Power supply	+ 5V
VSS	1	-	Power supply	Ground 0V

2.7. BUZZER

Perangkat *Buzzer* digunakan untuk menghasilkan bunyi, merupakan komponen resonator Riezoelectric yang digunakan untuk mengadakan isyarat

terdengar sebagai *indikator*. *Buzzer* akan aktif dengan cara mengeluarkan sinyal suara (berbunyi) dengan lama waktu sesuai dengan perencanaan nanti.

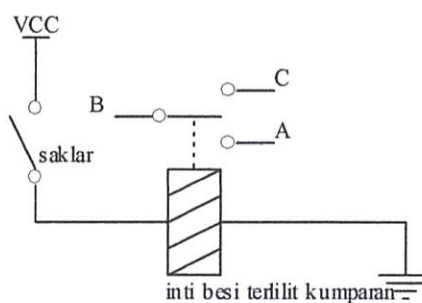


Gambar 2-12

Rangkaian *Driver Buzzer*^[8]

2.8. Relay

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



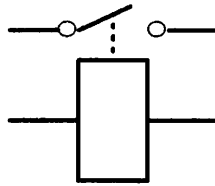
Gambar 2-13

Cara kerja relay^[9]

Relay merupakan suatu alat untuk menghubungkan kontak antar komponen, dan dalam memutus atau menghubungkan kontak digerakkan oleh fluksi yang ditimbulkan dari adanya medan magnet yang dihasilkan oleh kumparan yang melilit pada besi lunak yang terdapat di relay.

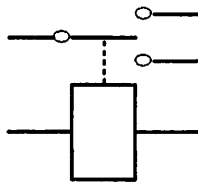
Ada beberapa macam relay, antara lain :

- SPST (*Single Pin Single Terminal*)



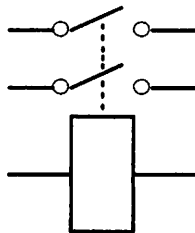
Gambar 2-14 Relay SPST^[9]

- SPDT (*Single Pin Dual Terminal*)



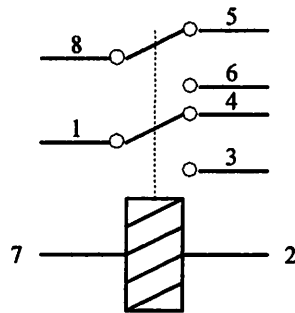
Gambar 2-15 Relay SPDT^[9]

- DPST (*Dual Pin Single Terminal*)



Gambar 2-216 Relay DPST^[9]

- DPDT (*Dual Pin Dual Terminal*)



Gambar 2-17 Relay DPDT^[9]

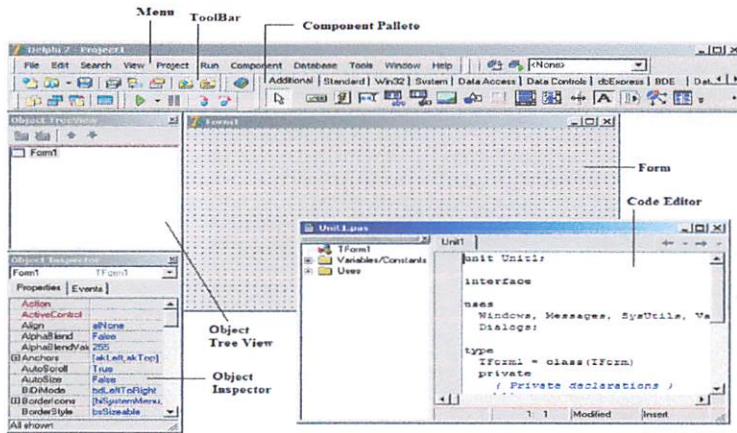
2.9. Borland Delphi

Secara umum, Borland Delphi adalah sebuah program untuk membuat aplikasi – aplikasi berbasis Windows. Bahasa pengembangan yang digunakan oleh Delphi adalah bahasa Pascal. Turbo Pascal dikenal dengan kelebihan dalam kecepatan eksekusi dan kompilasi, dibandingkan dengan bahasa pemrograman lain yang berkembang saat ini. *Integrated Development Enviroment* (IDE) yang diperkenalkan dan diterapkan oleh Turbo Pascal sangat memudahkan para programmer merealisasikan program aplikasi mereka. Dengan IDE seorang programmer dapat dengan cepat dan mudah menulis kode program, melakukan kompilasi, melihat kesalahan (*error*) program, serta langsung menuju letak kesalahan dan memperbaiki kesalahan tersebut. Kemudian Turbo Pascal dirubah menjadi yang berorientasi obyek (*Object Oriented Programming*) berbasis tampilan visual yang menarik, dan dilengkapi kemampuan akses ke basis data. Inilah yang kemudian dikenal sebagai Delphi.

Delphi dapat digolongkan ke dalam bahasa tingkat tinggi (*High Type Language*) karena segala kemudahan ditawarkan untuk perancangan sebuah aplikasi.

2.9.1. 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 2.18 IDE (*Integrated Development Enviroment*)^[10]

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.

2.9.2. 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 Pallette* 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.

11. *Menu Help*

Berisi fasilitas menerima bantuan atau keterangan tentang Delphi.

BAB III

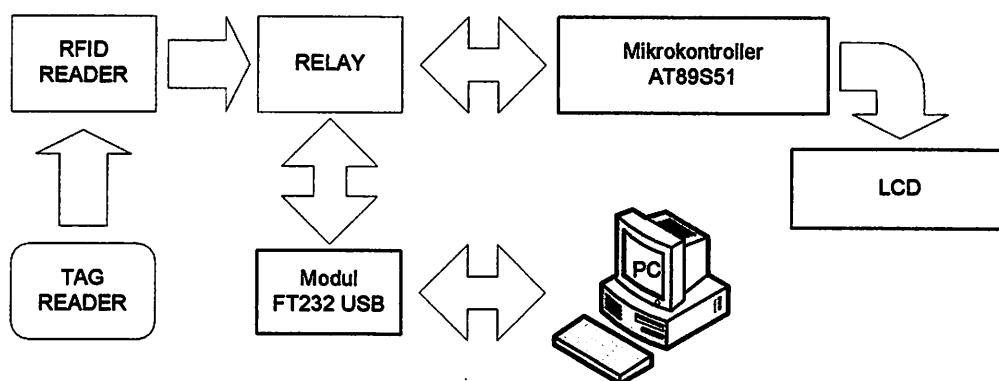
PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan membahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam sistem absensi pegawai negeri menggunakan kartu RFID, Mikrokontroler AT89S51 yang di antar muka (*interface*) ke PC menggunakan USB, pembahasan dilakukan pada setiap blok rangkaian yang terdiri atas: cara kerja masing-masing blok rangkaian, fungsi masing-masing blok rangkaian.

3.2. Blok Diagram Keseluruhan Sistem

Perancangan dan pembuatan alat ditunjukkan dengan gambar blok diagram dibawah ini :



Gambar 3.1. Diagram Blok Keseluruhan Sistem

Keterangan fungsi dari masing-masing blok diagram diatas sebagai berikut :

- **Tag RFID**

Devais yang menyimpan informasi untuk identifikasi objek.

- **RFID Reader**

Devais yang kompatibl dengan *tag* RFID yang akan berkomunikasi secara *wireless* dengan *tag*. Digunakan tipe ID-12 sebagai RFID *reader* pada perancangan ini.

- **LCD**

Digunakan sebagai penampil data yang diberikan oleh mikrokontroler.

- **Mikrokontroler AT89S51**

Berfungsi sebagai pengontrol masukan dan keluaran data.

- **PC Notebook (Laptop)**

Berfungsi sebagai database, menerima data dari mikrokontroller sekaligus mengirimkan perintah ke mikrokontroller.

- **Modul FT232 BM (PC Link USBer USB-UART Converter)**

Berfungsi sebagai *interface* antara mikrokontroler dengan *PC Notebook*

- **Relay**

Berfungsi sebagai saklar yang menghubungkan antara RFID Reader dengan Mikrokontroler dan sebaliknya antara PC dengan Mikrokontroler.

3.3. Prinsip Kerja Alat

Setiap karyawan pegawai negeri harus memiliki *Tag* RFID sebagai kartu absensi dengan kode yang berbeda-beda. *Tag* RFID akan menerima pancaran gelombang radio, yang berasal dari RFID *reader*, kemudian *Tag* akan mengirimkannya kembali ke dalam RFID *reader* berupa data code dari *Tag*.

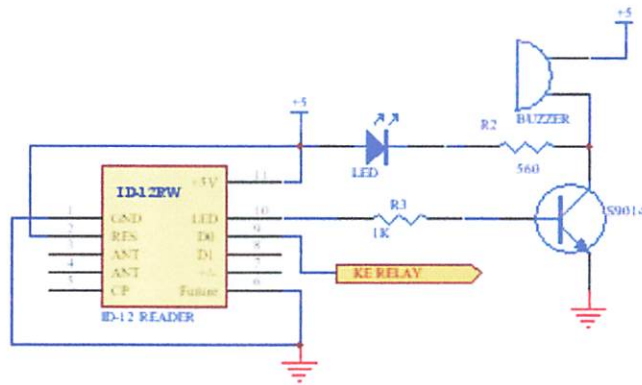
Gelombang radio tersebut membawa kode-kode yang akan diproses oleh mikrokontroler dan dikirim pada *PC* melalui komunikasi serial FT232, kemudian *PC* dicocokkan dengan database yang telah dirancang, jika data tersebut valid maka *PC* mengupdate database. Selain itu *PC* juga memerintah Mikrokontroler untuk mencetak teks di LCD.

3.4. Perancangan Perangkat Keras (*Hardware*)

3.4.1. RFID (*Radio Frequency Identification*)

Pada perencanaan sistem alat ini menggunakan *tag pasif*, yaitu *tag* yang tidak memiliki catu daya sendiri serta tidak dapat menginisialisasi komunikasi dengan *reader*. Ukuran *tag pasif* sekitar 0.4 mm x 0.4 mm. Daya jangkau RFID *tag pasif* agar dapat terbaca oleh RFID *transceiver* adalah mulai dari sekitar 1 cm sampai 6 meter. *Tag* yang digunakan adalah ISO Card GK-4001 dan RFID reader nya menggunakan ID-12 dengan frekuensi rendah 125 kHz, format data *output* menggunakan ASCII yang dihubungkan ke mikrokontroler menggunakan komunikasi serial UART. *Tag* ini yang digunakan sebagai kartu absensi yang pada saat kartu ini didekatkan ke pembaca RFID maka akan dikenali dan akan teridentifikasi nomor seri yang ada didalamnya, sehingga setiap karyawan mempunyai kartu dengan nomor seri yang berbeda. Bentuk asli *tag* masih polos maka untuk model kartu bisa dilapisi dengan berbagai macam variasi.

Rangkaian untuk RFID yang direncanakan sebagai berikut :



Gambar 3.2. Rangkaian *RFID Reader*

Keterangan fungsi dari masing-masing rangkaian diatas sebagai berikut :

- Buzzer digunakan sebagai indikator berupa suara, sehingga jika reader mendeteksi sinyal yang berasal dari tag maka buzzer akan berbunyi.
- Pada perancangan rangkaian RFID nilai I_b yang diinginkan 5mA, $V_{BE} = 0.65$ V dan $V_{CC} = 5V$ maka di dapatkan : $V_{CC} = R_B \cdot I_B + V_{BE}$, sehingga

$$\begin{aligned}
 V_{CC} &= R_B \cdot I_B + V_{BE} \\
 5 &= R_B \cdot 5 \cdot 10^{-3} + 0.65 \\
 R_B &= \frac{5 - 0.65}{5 \cdot 10^{-3}} \\
 &= \frac{4.35}{5 \cdot 10^{-3}} \\
 &= 0,87 \cdot 10^3 \Omega
 \end{aligned}$$

Jadi dengan nilai $R_B = 0,87k\Omega$ tidak terdapat dipasaran maka diganti dengan nilai resistor yang mendekati yaitu $R_B = 1 k\Omega$.

Transistor type S9014 dalam rangkaian ini mempunyai nilai I_B sesuai data sheet = 5mA dan dapat digunakan untuk Switch pada rangkaian RFID.

- LED berfungsi sebagai indikator cahaya saat terjadinya pendeteksian Tag terhadap Reader. Dengan nilai arus sebesar yang diinginkan 5mA dan tegangan maju LED (V_{LED}) = 2V, dalam tegangan +5 V maka:

$$R_{LED} = \frac{V_{CC} - V_{LED}}{I_{LED}} = \frac{(5-2) V}{5 mA} = 600 \Omega$$

Jadi nilai resistor yang digunakan minimal sebesar 600 Ω dan resistor yang digunakan pada rangkaian RFID Reader ini sebesar $R= 560\Omega$.

Dalam perencanaan R_{LED} yang digunakan sebesar 560 Ω sehingga dapat dihitung kembali :

$$I_{LED} = \frac{(5-2) V}{560 \Omega} = 5,35 mA$$

3.5. Mikrokontroler AT89S51

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

3.5.1.1. Perancangan minimum sistem AT89S51

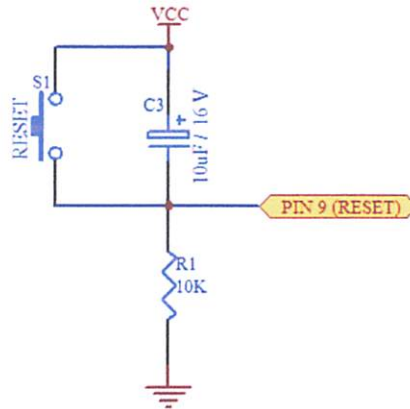
Pada bagian ini perancangan yang akan dibahas meliputi :

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

3.5.1.2. Perancangan Rangkaian Reset

Untuk *mereset* mikrokontroler AT89S51, maka pin RST diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan

sinyal *reset* kapasitor dihubungkan dengan Vcc dan sebuah resistor yang dihubungkan ke *ground*. Rangkaian *reset* ditunjukkan dalam gambar 3-4 sebagai berikut :



Gambar 3.3. Perancangan Rangkaian *Reset*

Karena kristal yang digunakan mempunyai frekuensi sebesar 11,0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11,0592 \text{ MHz}} = 9,042 \times 10^{-8} \text{ s}$$

Sehingga waktu minimal logika tinggi yang dibutuhkan untuk *mereset* mikrokontroler adalah :

$$\begin{aligned} \text{reset(min)} &= T \times \text{periode yang dibutuhkan} \\ &= 9,042 \times 10^{-8} \times 24 = 2,17 \mu\text{s} \end{aligned}$$

Jadi mikrokontroler membutuhkan waktu minimal 2,17 μs untuk *mereset*. Waktu minimal inilah yang dijadikan pedoman untuk menentukan nilai R dan C. Dengan menentukan nilai R = 8,2 k Ω dan C = 10 μF , maka :

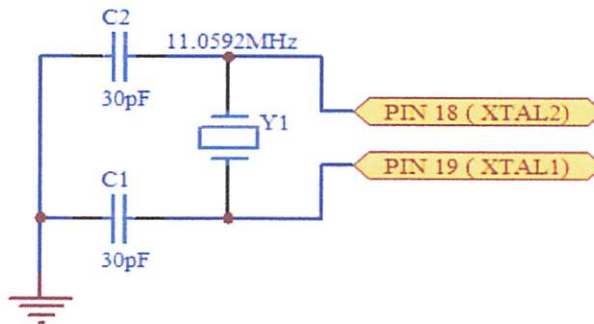
$$t = 0,357 R.C = 0,357 \times 8200\Omega \times 10.10^{-6} = 29,274 \text{ ms}$$

Jadi dengan nilai komponen $R = 8,2 \text{ k}\Omega$ dan $C = 10\mu\text{F}$ dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

3.5.1.3. Perancangan Rangkaian Clock

Kecepatan proses yang dilakukan oleh mikrokontroler ditentukan oleh sumber *clock* yang mengendalikan mikrokontroler tersebut. Sistem yang dirancang ini menggunakan osilator internal yang telah tersedia dalam *chip* AT89S51. Untuk menentukan frekuensi osilatornya cukup dengan menghubungkan kristal dalam pin 19 (X_1) dan pin 18 (X_2) serta dua buah kapasitor ke *ground*.

Besarnya kapasitansinya disesuaikan dengan spesifikasi dalam lembar data AT89S51 yaitu 30 pF. Kristal yang digunakan adalah 11,0592 MHz. Gambar 3.4 memperlihatkan rangkaian *clock* yang dirancang.



Gambar 3.4. Perancangan Rangkaian *Clock*

3.5.1.4. Perancangan Pengaturan Port pada Mikrokontroler AT89S51

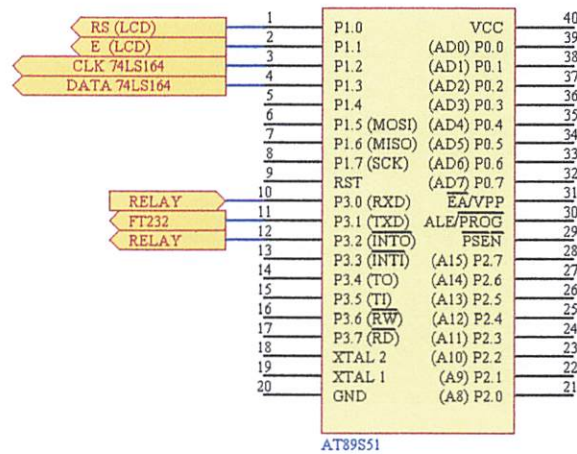
Pada perancangan port ini akan dipaparkan port apa saja yang akan digunakan pada Mikokontroler AT89S51 yaitu:

1) Port 1,

- a) P1.0 (pin 1) digunakan sebagai port untuk pin RST pada LCD
- b) P1.1 (pin 2) digunakan sebagai port untuk pin E pada LCD.
- c) P1.2 (pin 3) digunakan sebagai CLOCK 74LS164.
- d) P1.2 (pin 3) digunakan sebagai DATA 74LS164.

2) Port 3,

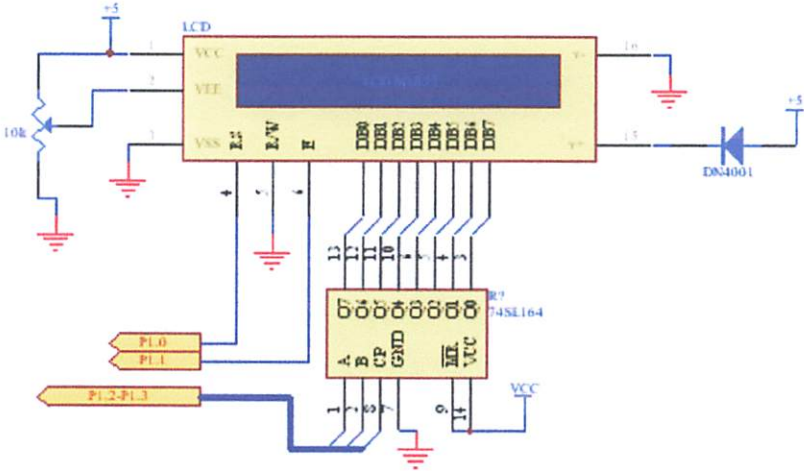
- a) P3.0 (pin 10) RXD serial input yang dihubungkan ke relay yang digunakan untuk menerima RFID reader dan PC.
- b) P3.1 (pin 11) TXD digunakan untuk port TX serial output, port ini digunakan mikrokontroler untuk mengirim data serial ke PC menggunakan IC FT232
- c) P3.2 (pin 12) yang dihubungkan ke relay yang digunakan sebagai selektor antara RFID reader dan PC



Gambar 3.5. Rangkaian MCU AT89S51

3.6. Perancangan Rangkaian LCD

Dalam aplikasi ini menggunakan sebuah layar LCD (*Liquid Crystal Display*) yaitu M1632 yang merupakan LCD dua baris dengan setiap barisnya terdiri 16 karakter dan menggunakan IC 74LS164 yang merupakan sebuah IC yang berfungsi sebagai *shift register* 8-bit yang memiliki 2 serial *input*, *input clock*, *input reset*, dan *output paralel*. Pada saat logika LOW, kedua input mengalami masukan data baru dan *reset* flip-flop pertama menjadi LOW pada saat pulsa *clock* berikutnya, hal ini mengendalikan data yang masuk. Sedangkan pada saat HIGH, kedua *input* memungkinkan adanya inputan yang lain, yang mana akan menentukan status flip-flop yang pertama, data pada serial input kemungkinan akan berubah pada saat *clock high* atau *low* akan tetepi informasinya saja yang akan masuk. Gambar hubungan antara LCD, IC74LS164 dan mikrokontroler dapat dilihat dalam gambar berikut :



Gambar 3.6. Perencanaan Rangkaian LCD

LCD dot matrik ini membutuhkan sepuluh buah pin masukan/keluaran dari mikrokontroler dan IC 74164 Adapun dua buah pin yakni port PC5 pada penyemat RS yang digunakan sebagai sinyal pemilih register dan port PC4 pada penyemat Enable

digunakan sebagai sinyal operasi awal, sinyal enable ini mengaktifkan data tulis atau baca oleh mikrokontroler, pin DB0-DB7 yang dihubungkan ke pin data IC74164 digunakan untuk menampilkan karakter yang dikehendaki oleh mikrokontroler. Ketika terdapat data pada jalur data, data tersebut akan ditahan dengan memberikan *clock* pin E pada LCD. Pin RS menentukan apakah data yang ditahan akan digunakan sebagai instruksi untuk mengatur *setting* tampilan pada LCD atau sebagai kode karakter yang diperlukan LCD untuk menampilkan suatu karakter. Sedangkan untuk pin R/W pada LCD dihubungkan ke *ground* karena dalam hal ini LCD hanya melakukan operasi write atau operasi menampilkan karakter.

Untuk pin Vcc pada LCD dihubungkan ke supply +Vcc dan Vss dihubungkan ke *ground*. Pin V_{EE} beserta pin Vcc dan Vss dihubungkan ke *trimmer potensio* atau kadang disebut dengan *trimpot*. *Trimpot* ini digunakan untuk mengatur kontras dari tampilan LCD dengan cara mengubah tegangan pada pin V_{EE}. Daftar tabel fungsi penyemat pada LCD dapat dilihat dalam Tabel 3.1.

Tabel 3.1. Fungsi penyemat LCD^[3]

Penyemat	Fungsi
DB0 – DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan di LCD.
Enable	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca.
R/W	Sinyal seleksi tulis atau baca 0: tulis 1: baca
RS	Sinyal pemilih <i>register</i> 0: masukan data 1: masukan instruksi

Seperti telah disebutkan sebelumnya bahwa data yang terdapat pada jalur data selain dianggap sebagai kode karakter dapat digunakan sebagai suatu perintah instruksi untuk mengatur setting dari tampilan LCD. Cara pemakaian data antara sebagai instruksi dengan kode karakter berbeda. Perbedaan hanyalah keadaan pin RS ketika data yang ada di jalur data ditahan oleh LCD dengan memberikan *clock* pada pin E.

Pin – pin yang digunakan adalah

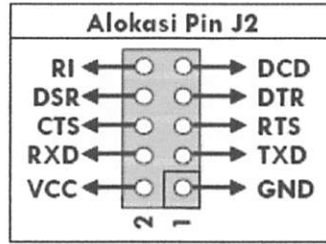
- Pin DB0-DB7 terhubung pada IC 74LS164 yang kemudian pada IC tersebut terhubung pada AT89S51 dengan pin 2 (Data) terhubung pada P 1.3 dan pin 8 (CLK) terhubung pada port 1.2 sedangkan pin 7 (Ground).
- Pin Enable pada LCD terhubung pada AT89S51 yaitu port 1.1
- Pin RS pada LCD terhubung pada AT89S51 yaitu port 1.0

3.7. Modul FT232BM

Pada skripsi ini Modul Modul FT232BM berfungsi sebagai Modul USB (Universal Serial Bus) yang menjembatani antara mikrokontroler AT89S51 dengan komputer. Pada modul USB ini ada beberapa pin yang mesti di pakai dengan seksama sesuai dengan kegunaannya yang terdiri pin J2 sebagai port serial sebagai I/O.

3.7.1. Penggunaan Pin-Pin Modul FT232BM

Gambar 3.7 menunjukkan pin-pin Modul FT232BM, namun dalam perancangan sistem ini beberapa pin saja yang dipakai.



Gambar 3.7. Pemakaian PIN pada port Serial dari Modul FT232BM^[4].

Pin-pin yang digunakan dalam perancangan ini:

- Pin 1, GND (Ground)

Pin 1 pada J2 adalah pin yang digunakan sebagai ground pada modul USB ini, sedangkan pada IC FT232BM ground terletak pada pin 9,17,dan 29.

- Pin 2, VCC

Modul USB ini di operasikan dengan tegangan supply 4,4V – 5,25 V DC, pada IC FT232BM Vcc berada pada pin 3,13,26,30.

- Pin 3, RxD

Pin 3 pada J2 merupakan pin yang berada pada pin 24 dari IC FT232BM.

Dalam perancangan pin digunakan sebagai port input serial.

- Pin 4, TxD

Pin 4 pada J2 merupakan pin 25 dari IC FT232BM dan dalam perencanaannya, pin ini digunakan sebagai output serial.

Agar modul dapat dikendalikan dengan PC maka diperlukan sebuah *driver* pendukung , tipe *driver* ini adalah virtual com port (VCP) driver. Dengan *driver* ini maka modul dapat dikendalikan oleh PC. Langkah-langkahnya sebagai berikut:

1. Sambungkan *device* ke port USB di PC. Ini secara otomatis akan dikenal sebagai hardware yang baru pada PC sehingga akan tampil layar (*screen*) sebagai berikut:



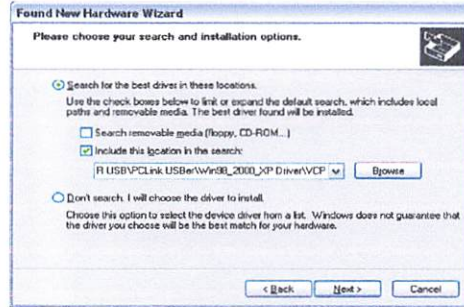
Gambar 3.8. Kotak Dialog Found New Hardware Wizard..

2. Pilih "Install from a list or specific location (Advanced)" seperti yang ditampilkan dibawah kemudian klik "Next"



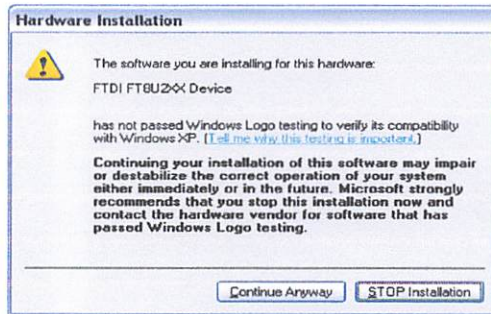
Gambar 3.9. Kotak Dialog Memilih Instalasi.

3. Pilih "Search for the best driver in these locations" kemudian pilih pada bagian combo-box ("C:\VCP Drivers" in the example below) atau browser dengan mengklik pada tombol browser. Salah satu bagian file yang telah terpilih didalam kotak (*box*), Klik next untuk memproses.



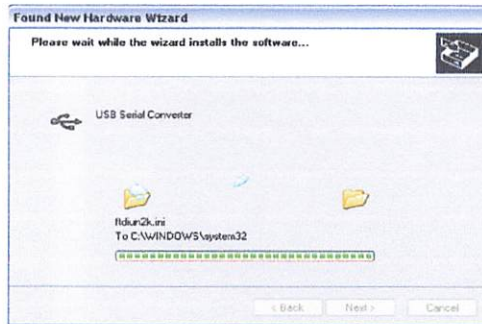
Gambar 3.10. Kotak Dialog Lokasi Driver.

4. Kemudian pada proses inatalasi akan menampilkan layar (*screen*) seperti yang dibawah, klik pada tombol “Continue Anyway”.



Gambar 3.11. Kotak Dialog Melanjutkan Instalasi.

5. Ikuti langkah tampilan pada layar, yang akan menampilkan proses pengopian file driver, yang dibutuhkan Window Xp.



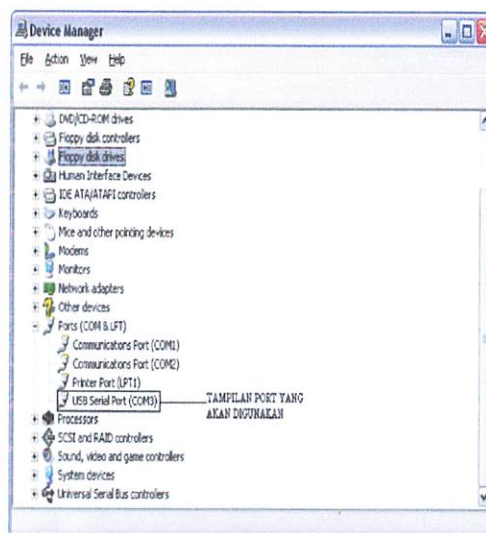
Gambar 3.12. Kotak Dialog Proses Instalasi.

Sehingga window akan menampilkan sebuah pesan bahwa instalasi telah sukses. Klik “Finish” untuk melankapi instalasi. instalasi ini merupakan serial converter.



Gambar 3.13. Kotak Dialog Instalasi Driver FT232BM Selesai

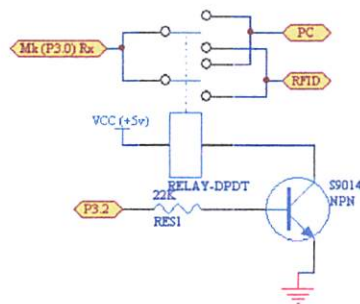
- Setelah klik finish dari pengenalan driver yang baru akan dilanjutkan dengan menginstal COM port driver. Prosedur sama seperti instalasi serial converter diatas. Buka *Device Manager* dilokasi system control panel kemudian pilih *Hardware* dan klik *Device Manager* pilih pada bagian jenis device, device akan terlihat menambah COM port dengan label “USB Serial port”.



Gambar 3.14. Kotak Dialog Tampilan Device manager pada PC.

3.8. Perancangan Rangkaian Relay

Dalam aplikasinya mikrokontroller jika digunakan untuk mengontrol rangkaian *eksternal* selalu menggunakan *driver relay*. Hal ini digunakan untuk melakukan pergantian dalam pengiriman data antara RFID dengan PC. Dengan hal ini kami menggunakan *driver relay* dengan gambar rangkaian sebagai berikut :



Gambar 3.15. Rangkaian Relay

1. Relay ini memiliki type DPTR yang memiliki 8 pin yang terdiri atas :
 - o 2 pin sebagai power supply (+/-) yang nantinya akan mengerakan 2 pin yang dihubungkan ke mikrokontroller.
 - o 2 pin dihubungkan ke mikrokontroler port P3.1 yang nantinya akan menghubungkan ke PC atau ke RFID,
 - o 2 pin di hubungkan ke PC.
 - o 2 pin lagi dihubungkan ke RFID (reader).

sehingga dalam seperti dalam gambar diatas relay tersebut akan bekerja sebagai transistor saklar, maka dengan data-data pengukuran Hfe Transistor didapat :

$$hfe \text{ Transistor S9014} = \beta_{dc} = 300 \text{ (hasil pengukuran)}$$

$$R_C = R_{\text{Relay}} = 62 \Omega$$

$$V_{CC} = +5V$$

$$V_B = +5V$$

$$V_{BE} = 0.6 V$$

$$\begin{aligned} I_C &= \frac{V_{CC}}{R_C} \\ &= \frac{5}{62} \\ &= 0.0806 A \end{aligned}$$

$$\begin{aligned} I_B &= \frac{I_C}{h_{fe}} \\ &= \frac{0.0806}{300} \\ &= 0.00026 A \end{aligned}$$

$$\begin{aligned} R_B &= \frac{V_B - V_{be}}{I_b} \\ &= \frac{5 - 0.6}{0.00026} \\ &= 16.9 K\Omega \end{aligned}$$

Jadi nilai arus yang melalui relay adalah 80,64 mA dan arus basis yang mengalir pada transistor adalah 0,26 mA. Resistor yang digunakan untuk R_B (tahanan basis) minimal sebesar 17 K Ω . Jadi dengan nilai $R_B = 17 k\Omega$ tidak terdapat dipasaran maka diganti dengan nilai resistor yang mendekati yaitu $R_b = 22 k\Omega$.

3.9. Perancangan Perangkat Lunak (*Software*)

Perangkat lunak adalah sebuah jembatan yang menghubungkan keseluruhan komponen yang ada pada sebuah komputer. Pada perancangan ini ada

dua macam, yaitu perancangan perangkat lunak untuk mikrokontroler menggunakan bahasa *Assembly* Mikrokontroler standard MCS-51 dan perancangan perangkat lunak untuk komputer menggunakan *Delphi7*

3.9.1. Program Aplikasi Mikrokontroller

Program mikrokontroler bertujuan untuk mengontrol masukan dan keluaran. Hal-hal yang dikontrol mikrokontroler adalah proses komunikasi serial antara RFID dengan Komputer *Laptop*.

Didalam proses komunikasi serial antara MCU dengan *PC* terlebih dahulu ditentukan *baud rate* yang digunakan. *Baud rate* yang dibangkitkan *Timer1* dengan *Timer2* (8 bit *auto reload*) yang hanya menggunakan register TH1. Pada sistem ini digunakan *baudrate* sebesar 9600 dengan SMOD = 0 dengan menggunakan Frekuensi_ osilator = 11.0592 MHz.

$$Baud\ rate = \frac{2^{SMOD}}{32} \times \frac{Frekuensi_osilator}{12 \times [256 - (TH1)]}$$

$$9600 = \frac{2^0}{32} \times \frac{11.059 \times 10^6}{12 \times [256 - (TH1)]}$$

$$[256 - (TH1)] = \frac{1}{32} \times \frac{11.059 \times 10^6}{12 \times 9600}$$

$$[256 - (TH1)] = 3$$

$$TH1 = 256 - 3 = 253_D = 0xFD_H$$

3.9.2. Program Aplikasi Komputer

Program aplikasi adalah *software* di dalam komputer yang berfungsi untuk melakukan pengendalian dengan dunia luar. Program komputer ini bertujuan

untuk mengorganisasi komunikasi antara komputer sebagai pengolah data dengan mikrokontroler sebagai saluran masukan dan keluaran, sehingga perangkat lunak dalam program komputer perlu diketahui terlebih dahulu aturan-aturan atau protokol komunikasi yang digunakan untuk mengatur jalannya komunikasi antara komputer dengan mikrokontroler. Dalam perencanaan ini juga menggunakan bahasa pemrograman *visual* yaitu *Delphi7*. Komponen-komponen *Delphi7* yang digunakan untuk menunjang perancangan sistem.

alat dapat bekerja dengan baik, maka ada beberapa komponen tambahan yang bukan standard bawaan *Delphi7* yang juga digunakan dalam perancangan sistem alat ini.

3.9.3. *Component Pallete Delphi7*

Komponen-komponen Delphi disusun di bagian *Component Pallete*, dan dikelompokkan ke dalam *Tab/page*. Komponen-komponen yang digunakan dalam perancangan sistem alat ini adalah sebagai berikut :

3.9.3.1. *Tab Standard* :

Sesuai dengan namanya *page standard* ini berisi komponen yang diperlukan untuk membangun aplikasi Windows yang standard. Bagian ini berisikan komponen visual maupun nonvisual yang terdiri dari empat belas komponen, namun dalam perencanaan ini hanya menggunakan beberapa komponen saja yaitu:

- ***Edit***

Komponen ini dapat digunakan sebagai input/output satu baris teks. Pemakai program dapat mengubah teks ini.

- ***Label***

Komponen ini digunakan untuk membuat teks di form atau obyek lain tanpa dapat diubah oleh pemakaian program (VC).

- ***Memo***

Komponen ini dipakai untuk menerima masukan/ menampilkan beberapa baris teks.

- ***Button***

Untuk membuat tombol dengan beberapa pilihan style.

3.9.3.2. Tab Data Access

Page ini semuanya berisi nonvisual yang mengakses sumber data misalnya database, seperti Paradox, MS SQL, MS Access juga dapat berhubungan lewat ODBC.

- ***Data Source***

Data Source bertindak sebagai penghubung antara komponen pengakses data dengan DataSet.

3.9.3.3. Tab Data Controls

Komponen Data Controls merupakan komponen visual dan merupakan komponen data-aware yang sering disebut dengan komponen visual.

- ***DBNavigator***

Untuk menampilkan nilai suatu field dalam suatu label.

- ***DBGrid***

Menampilkan dan mengedit sebuah dataSet.

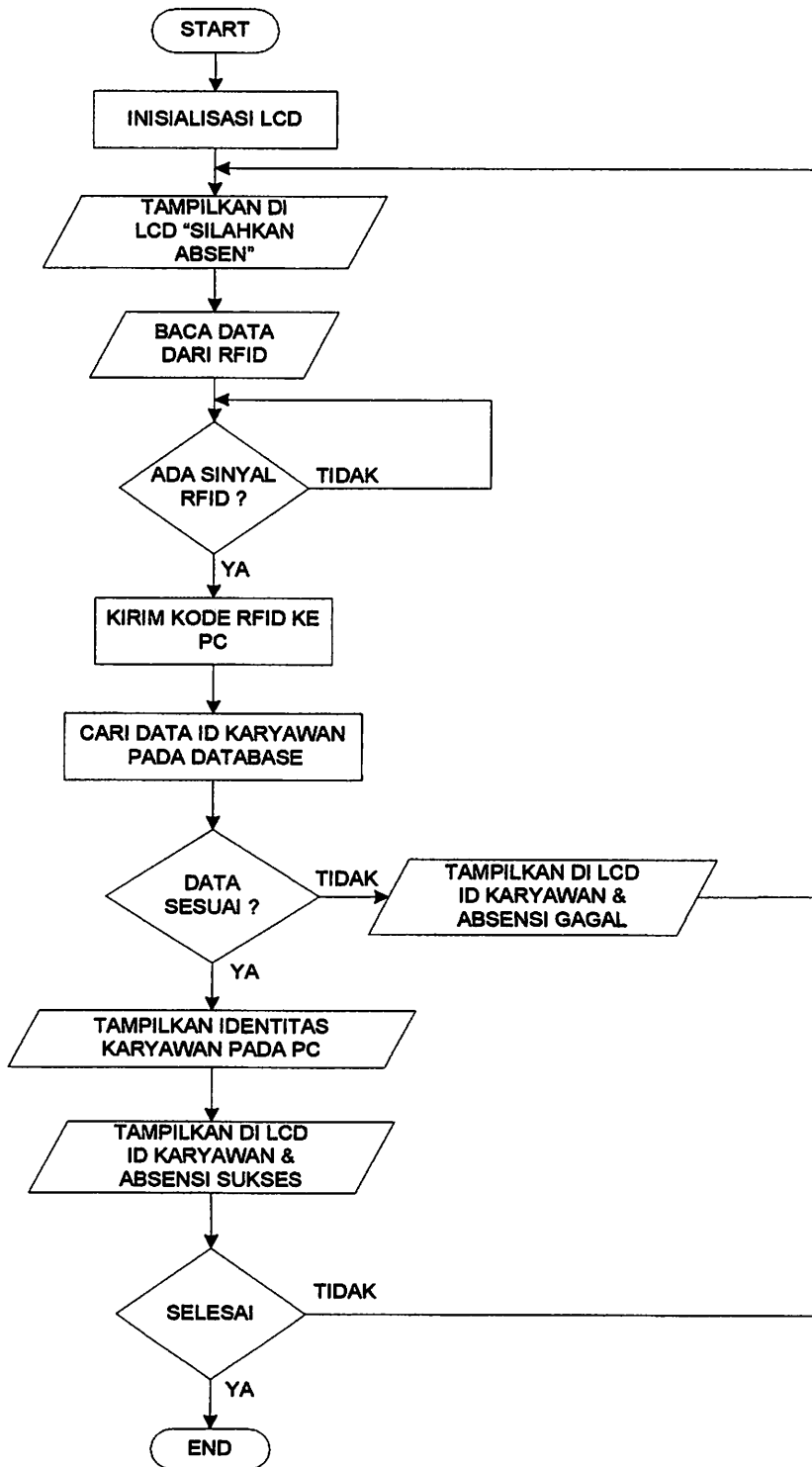
- ***DBNavigator***

Mengendalikan posisi record yang akan dipilih,yang terdiri dari tombol Previous Record,Next Record,First Record,Last Record dan Refresh.

- ***NOTEBOOK***

Komponen ini digunakan untuk membuat tumpukan lembar – lembar yang digunakan oleh komponen TabSet

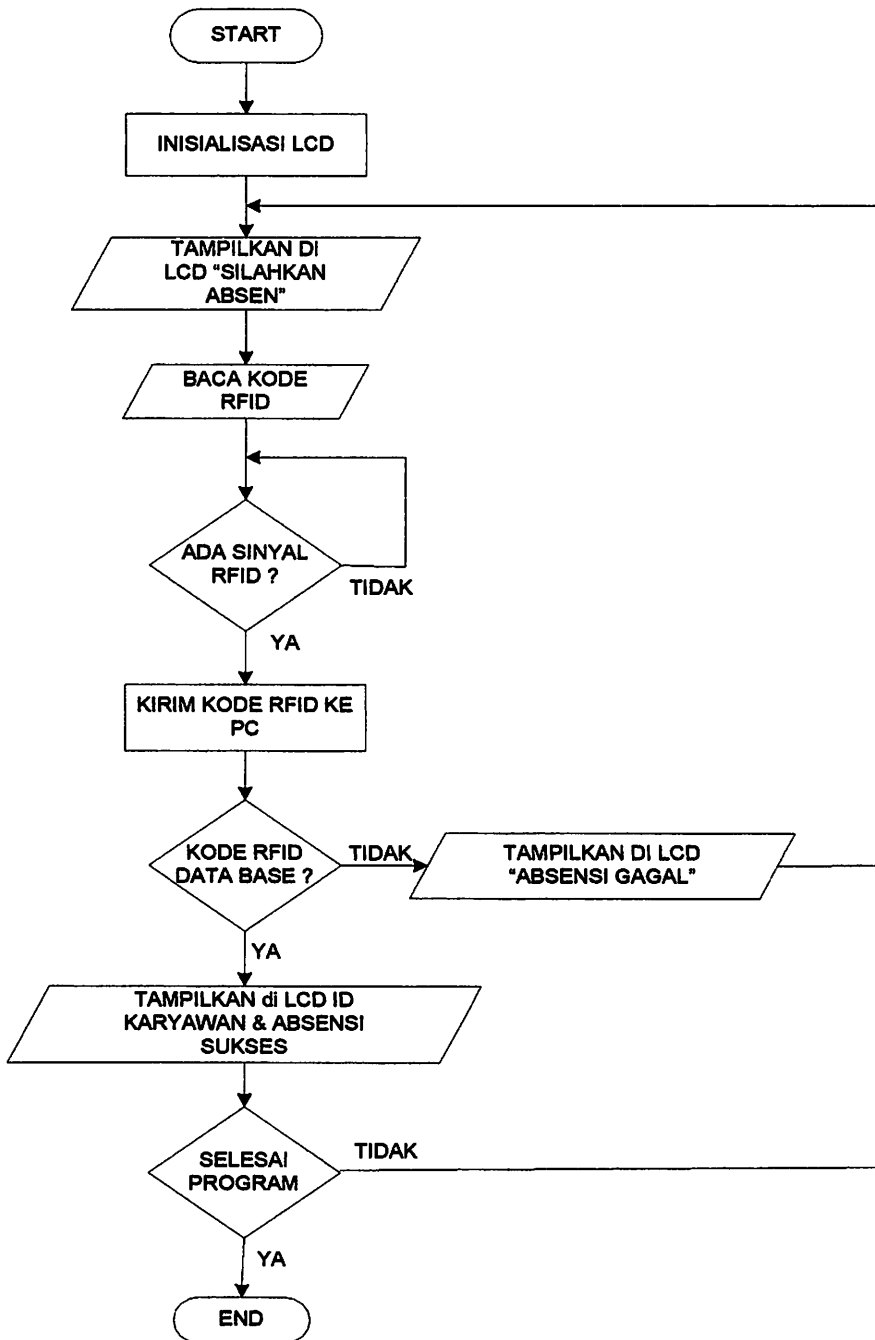
3.9.4. Flowchart Program Keseluruhan



Gambar 3.16.

Diagram Alir Keseluruhan

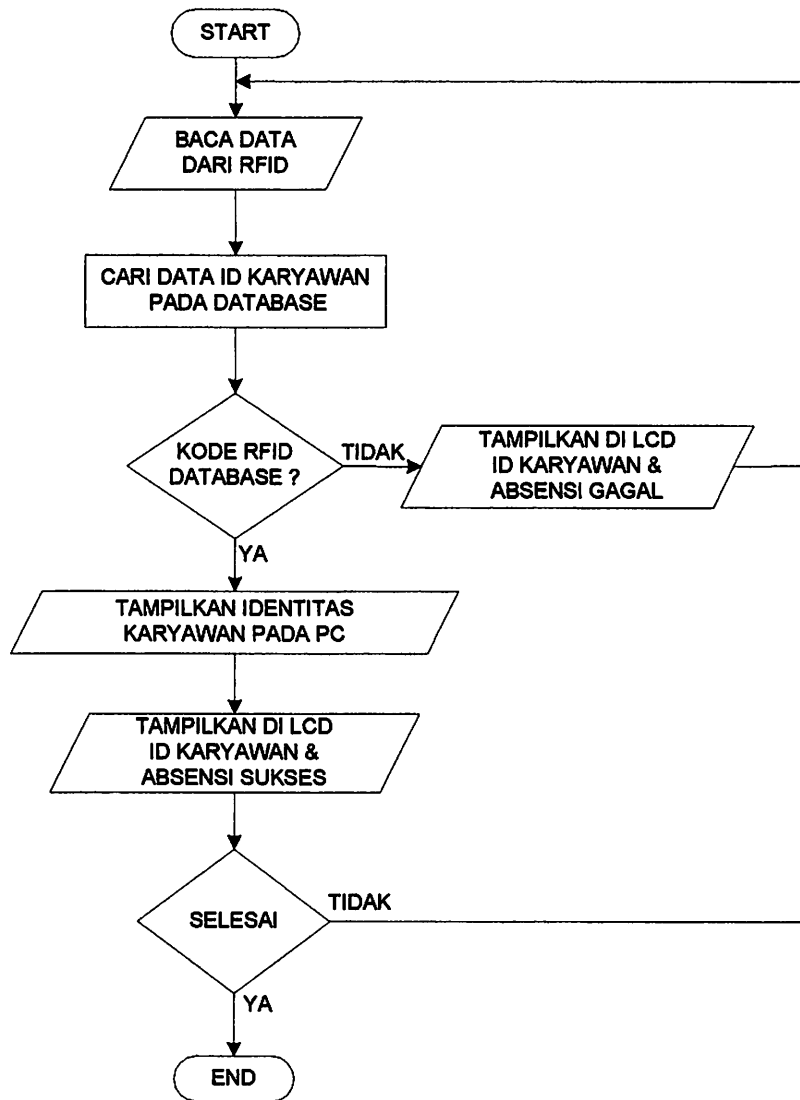
3.9.5. Flowchart Program dari Mikrokontroller AT89S51



Gambar 3.17

Diagram Alir Program Mikrokontroler

3.9.6. Flowchart Program dari Komputer



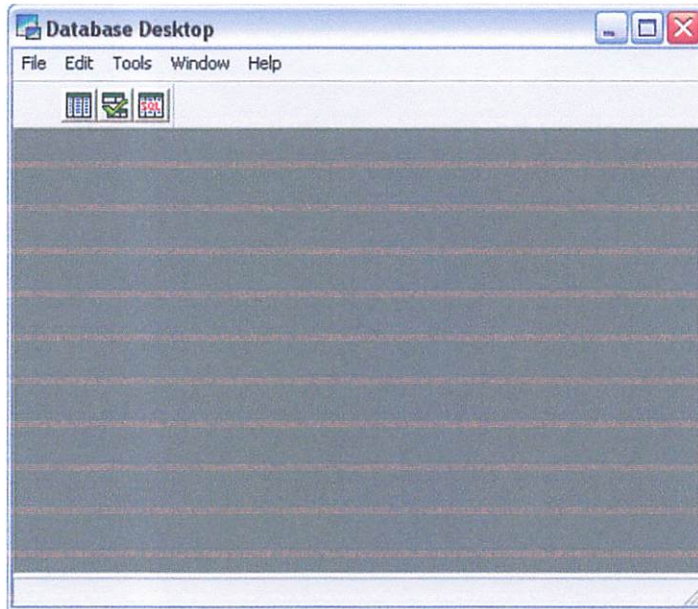
Gambar 3.18

Diagram Alir Program Komputer

3.9.7. Perencanaan Database

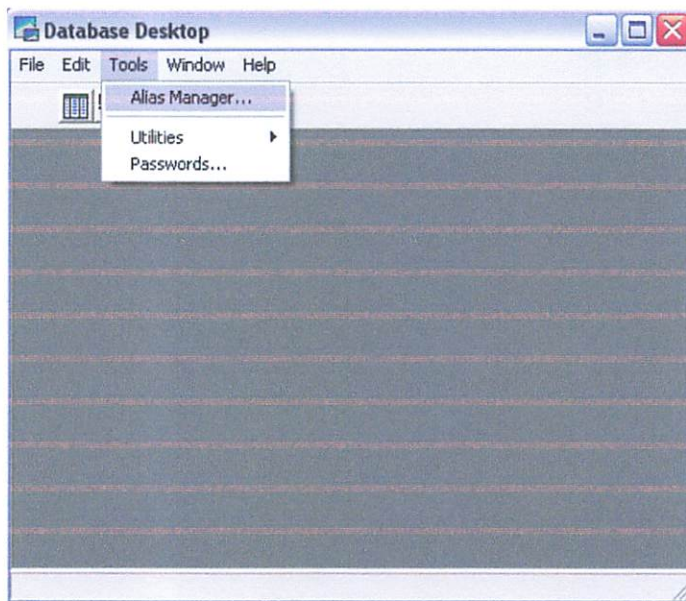
Program aplikasi yang digunakan dalam laporan skripsi ini menggunakan program delphi 7 dimana pada database menggunakan tabel dengan tipe tabel paradox 7. proses perencanaannya sebagai berikut:

- 1) Tampilkan lembar kerja Database Dekstop pada program delphi 7, seperti yang ditampilkan dibawah:



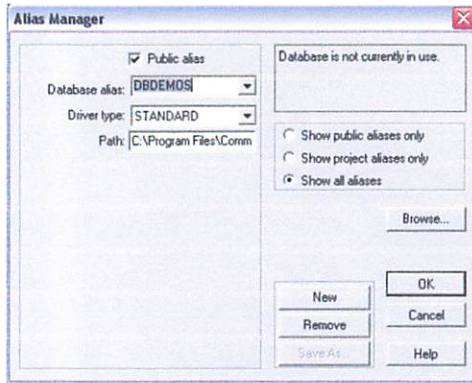
Gambar 3.19. Awal data base delphi dekstop

- 2) Pilih “Tool” kemudian Alias Manager, seperti gambar dibawah:



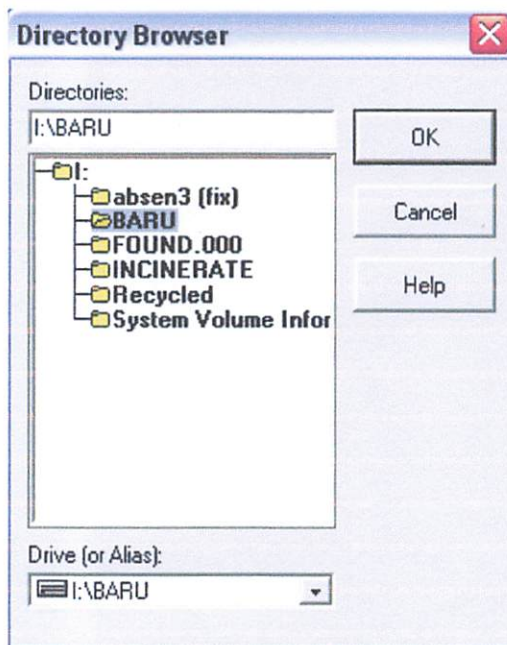
Gambar 3.20. Menentukan Tool Yang digunakan

- 3) Isikan nama Alias dari Database yang akan kita buat pada kolom Database alias, misalnya data seperti di bawah:



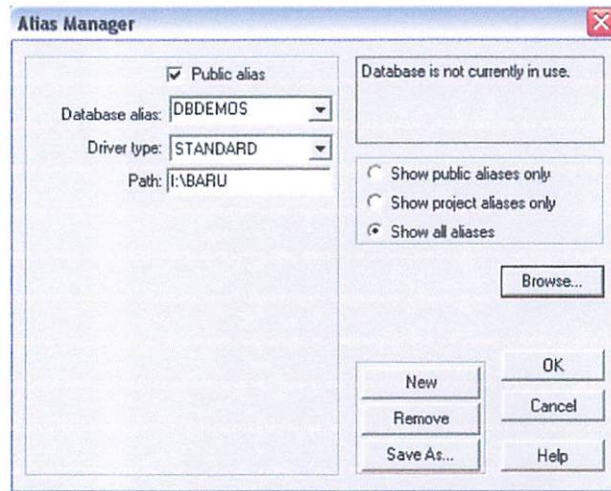
Gambar 3.21. Menentukan Nama Objek Data Base

- 4) Pilih Browser dan cari directory yang dipakai untuk lokasi penyimpanan data yang sudah kita buat pada langkah pertama.



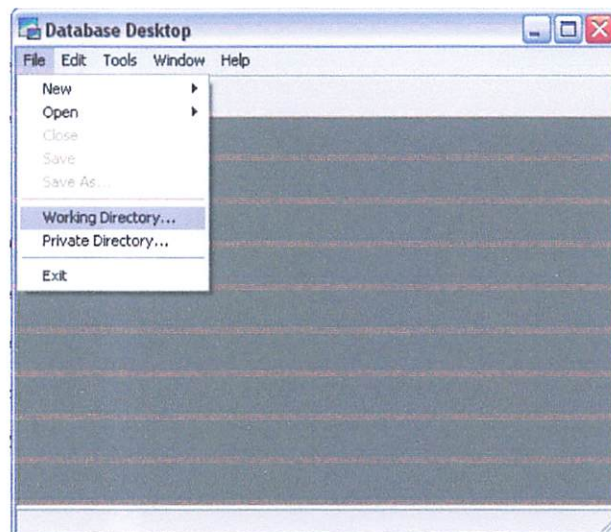
Gambar 3.22. Memilih Letak Data Base

- 5) Setelah directory dipilih maka akan berubah seperti di bawah kemudian OK.



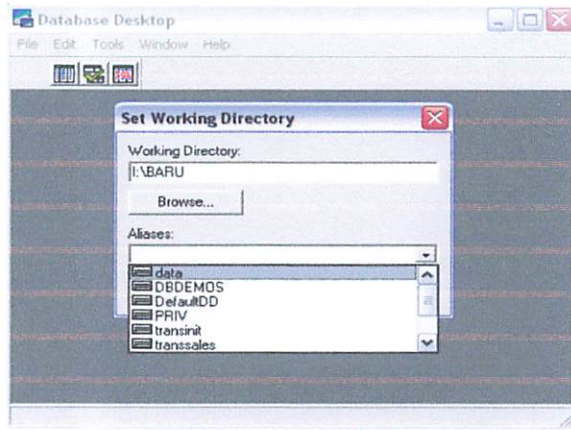
Gambar 3.23. Tampilan Setelah Penyetingan Nama dan Letak Data Base

- 6) Langkah selanjutnya menentukan Working Directory dengan cara: Pilih File kemudian Working Directory.



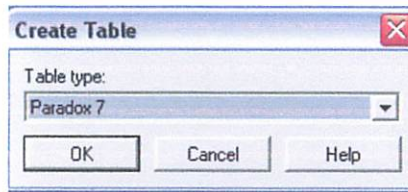
Gambar 3.24. Tampilan Menentukan Directory

7) Pilih aliansi yang telah kita buat kemudian tekan OK.



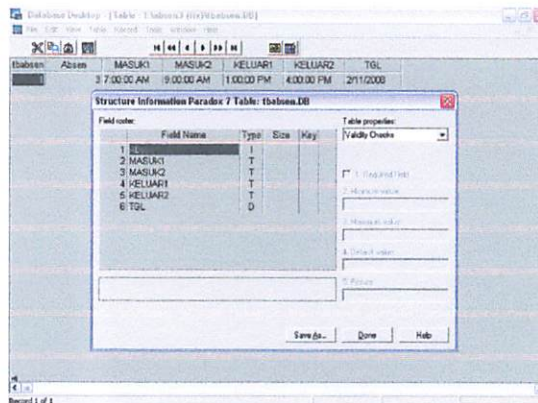
Gambar 3.25 Membuka Data Dari Directory

8) Pada Database Desktop pilihlah *File* menudian *New , Tabel*. Kemudian pilih tabel Paradox 7.



Gambar 3.26. Jenis Data Base

9) Berikut contoh tabel yang telah dibuat.



Gambar 3.27. Tampilan Data Base

10) Tampilan identitas karyawan.

The screenshot shows a web application window titled "Absensi Karyawan". It contains a form for employee information and a table of attendance records.

ID	0000593871
NAMA	HEFI
No. HP	089765453
Jenis Kelamin	LAKHLAKI
JABATAN	ADMINISTRASI

Buttons: Ambil Gambar, TAMBAH, SIMPAN, BATAL

Tgl	Masuk	Keluar	Lembur
2/25/2008	8:28:57 AM	1:37:13 PM	3
2/26/2008	8:37:47 AM	1:38:04 PM	4

Gambar 3.28. Tampilan Delphi Anggota

11) Tampilan laporan dalam sebulan.

The screenshot shows a "Report Preview" window displaying a monthly attendance report for employee DONY.

NIP : 0000593871 NAMA : DONY
JUMLAH ABSEN : 1 NO. HP : 0812455467

TABEL CATATAN PEGAWAI

Tgl	Jam Masuk	Jam Keluar	Lembur
2/24/2008	8:02:14 AM	1:02:49 PM	5
2/27/2008	8:18:10 AM		

Gambar 3.29. Tampilan Catatan Absesnsi Pegawai

BAB IV

ANALISIS DAN PENGUJIAN ALAT

4.1. Pendahuluan

Dalam bab ini membahas tentang pengujian dan pengukuran dari peralatan yang dibuat. Secara umum pengujian ini bertujuan untuk mengetahui apakah piranti yang telah direalisasikan dapat bekerja sesuai dengan perencanaan yang telah direncanakan.

4.2 Pengujian RFID

4.2.1. Tujuan

Tujuan dari pengujian ini adalah untuk mengetahui *tag* RFID bisa dibaca oleh *reader* RFID. Adapun cara pengujianya adalah dengan merangkai rangkaian RFID dan kemudian menghubungkan ke COM 1 PC. Untuk menguji *reader* bisa membaca kartu RFID dilakukan melalui *Hyper Terminal*.

4.2.2. Prosedur Pengujian

- a. Menghubungkan rangkaian RFID ke COM 1 PC.
- b. Membuka Hyper Terminal (Start → all program → accessories → Communication → hyperterminal).
- c. Memberi nama dan memilih *icon* pada *Connection Desert*.

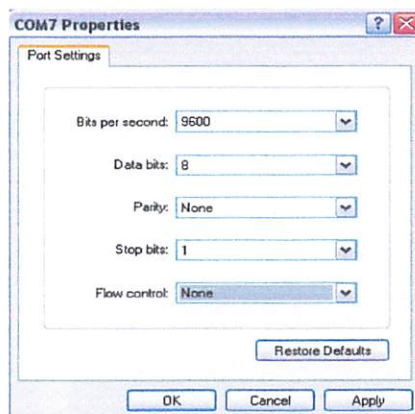


Gambar 4.1. Kotak Dialog Connection Description.

d. Memilih COM 1 pada kotak dialog *connect to*.

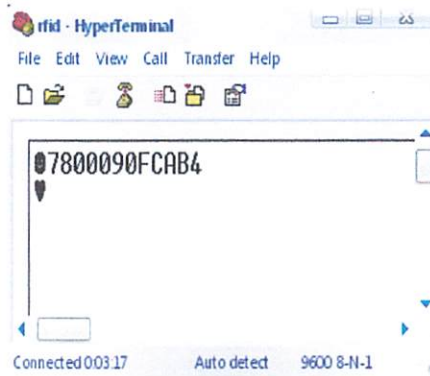


Gambar 4.2. Kotak Dialog Connect To COM.



Gambar 4.3. Kotak Dialog COM 1 Properties

- e. Menempatkan kartu pada jarak yang dijangkau *reader* sehingga menampilkan angka dari kartu tersebut.



Gambar 4.4. Identifikasi Reader Terhadap kartu.

4.2.3. Hasil Pengujian Pembacaan RFID

Tabel 4.1. Hasil Pengujian Pembacaan RFID

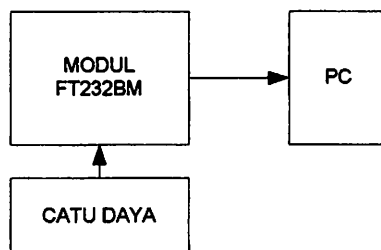
Jarak	Percobaan									
	1	2	3	4	5	6	7	8	9	10
1 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
5 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
6 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
7 cm	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
8 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
9 cm	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗

10cm	x	x	x	x	x	x	x	x	x	x
11 cm	x	x	x	x	x	x	x	x	x	x
12 cm	x	x	x	x	x	x	x	x	x	x

Tabel diatas merupakan hasil pengujian dimana kartu yang menghadap *reader* adalah bagian belakang. Jarak yang baik untuk bisa teridentifikasi adalah 6 cm. Untuk bagian depan menghasilkan data yang sama, tetapi untuk pengujian dimana kartu tegak lurus dengan *reader* hanya bisa saat kartu berjarak sangat dekat dengan *reader* (menempel).

4.3. Pengujian FT232BM

Tujuan dari pengujian ini adalah untuk mengetahui apakah Modul USB bisa digunakan untuk mengirim dan menerima data atau tidak. Adapun cara pengujianya dilakukan dengan cara memilih com yang dideteksi oleh PC kemudian mencoba melakukan pengiriman data melalui program sederhana yang telah dibuat, data akan dikirim melalui kabel USB menuju Modul USB FT232BM dengan menggunakan indikator apabila data di kirim dari PC ke USB maka led indikator pada modul (Rx) akan nyala dan sebaliknya pada saat data dikirim kembali dari USB ke PC maka indikator pada modul (Tx) akan menyala,dengan prosedur pengujian seperti dibawah ini :



Gambar 4.5. Blok Pengujian Modul USB

- Menghubungkan rangkaian Modul FT232BM ke PC menggunakan kabel USB.
- Menginstal driver modul USB.
- Membuka program pada delphi

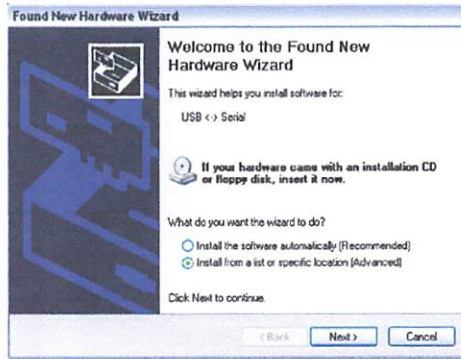
4.3.1. Instalasi Driver FT232BM

- Menghubungkan FT232BM dengan Port USB pada PC. Maka Windows pada PC akan secara otomatis menampilkan Found New Hardware Wizard yang ditunjukkan pada gambar 4.6.



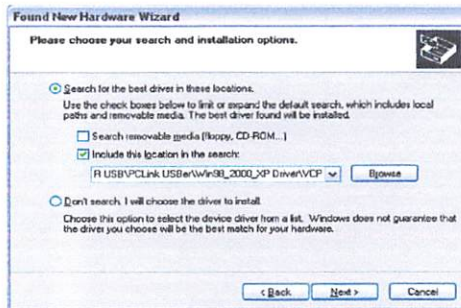
Gambar 4.6. Kotak Dialog Found New Hardware Wizard..

- Proses selanjutnya saat button *Next* ditekan maka pc Memilih keterangan a list specific location (advanced) pada wizard pada gambar 4.7. Perintah tersebut merupakan pemberitahuan apakah penginstallan dilakukan secara otomatis atau manual, tetapi pada saat menginstall dipilih yang manual:



Gambar 4.7. Kotak Dialog Memilih Instalasi.

- Proses selanjutnya saat button *Next* ditekan maka pc akan menampilkan perintah memilih lokasi driver yang tersimpan pada PC. Pada gambar 4.8 menunjukan letak dari lokasi driver yaitu (drive C: \VCP drivers) :



Gambar 4.8. Kotak Dialog Lokasi Driver.

- Proses selanjutnya saat button *Next* ditekan maka pc akan menampilkan perintah apakah proses instalasi dilanjutkan atau tidak , untuk melanjutkan instalasi maka memilih keterangan Continue anyway seperti ditunjukkan pada gambar 4.9.

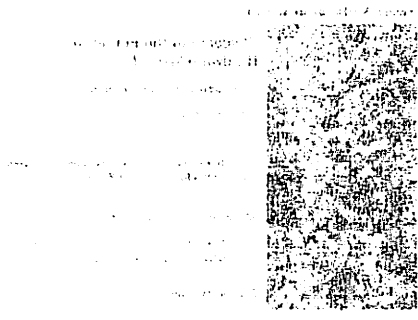


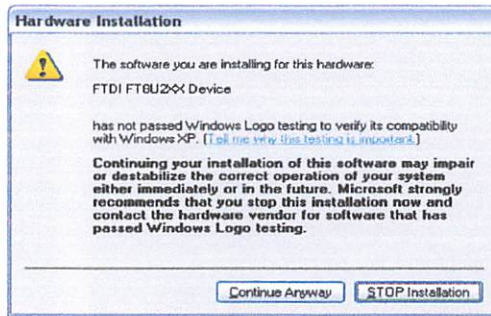
Diagram illustrating the structure of a column of text.

The diagram shows a vertical stack of rectangular blocks, representing a column of text. The blocks are arranged in a regular pattern, with some appearing slightly offset or overlapping. This structure is typical of a columnar layout in a document or a data table.



Diagram illustrating the structure of a table or structured data set.

The diagram shows a grid of rectangular blocks, representing a table or a structured data set. The blocks are arranged in a regular pattern, with some appearing slightly offset or overlapping. This structure is typical of a tabular layout in a document or a data table.



Gambar 4.9. Kotak Dialog Melanjutkan Instalasi.

- Proses selanjutnya saat button Continue anyway ditekan maka pc akan melakukan Proses Instalasi FT232BM pada PC seperti ditunjukkan pada gambar 4.10.



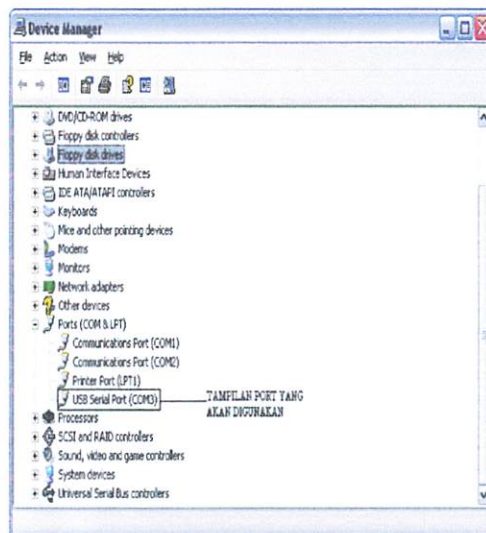
Gambar 4.10. Kotak Dialog Proses Instalasi.

- Proses Instalasi FT232BM pada PC telah berhasil, maka akan tampil bahwa driver baru telah selesai di install seperti gambar 4.11. Klik Finish Untuk mengakhiri instalasi.



Gambar 4.11. Kotak Dialog Instalasi Driver FT232BM Selesai.

- Tampilan Device manager jika FT232BM dihubungkan dengan USB Port pada PC.



Gambar 4.12. Kotak Dialog Tampilan Device manager pada PC.

Setelah Proses penginstallan selesai maka untuk melakukan pengujian terhadap modul USB. Pengujian modul FT232BM menggunakan sebuah program simpel delphi yang dibuat agar dapat digunakan untuk menguji modul USB tersebut.

4.3.2. Peralatan Yang Digunakan

- ❖ Komputer
- ❖ Modul FT232BM
- ❖ Program Uji Komunikasi Modul USB :

Cuplikan program uji komunikasi serial di komputer menggunakan program Delphi 7.0

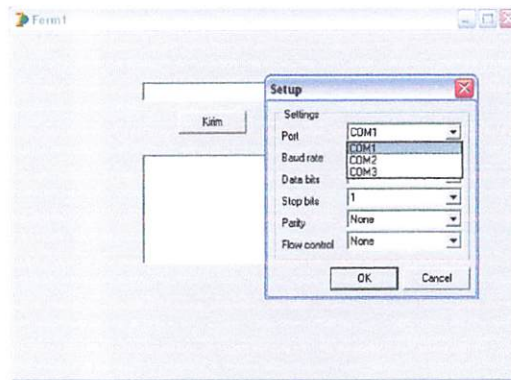
```
procedure TForm1.Button1Click(Sender: TObject);
begin
ComPort1.WriteStr('RFID')
end;

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

4.3.3. Prosedur Pengujian

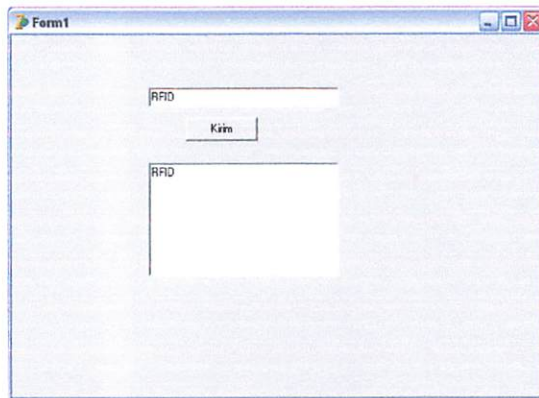
1. Menyusun rangkaian seperti gambar 4.5.
 2. Buat form baru Delphi sesuai dengan gambar4.13.
- Pada pembuatan form baru pada delphi, maka komponen yang dibutuhkan adalah 1 button, 1 edit, 1 memo, 1 Comport. Setelah itu Klik dua kali

pada comport, maka akan tampil setingan port yang ingin digunakan, seperti pada gambar 4.13.



Gambar 4.13. Tampilan Delphi pada Saat Penyetingan Komponen Comport.

- Setelah pembuatan proyek pada delphi selesai dibuat, maka dilanjutkan dengan membuat program sesuai dengan keterangan diatas, Setelah Program selesai dibuat, maka tekan F9 untuk melakukan esekusi, seperti dutunjukkan gambar 4.14.



Gambar 4.14 Tampilan Delphi pada Saat Program diJalankan.

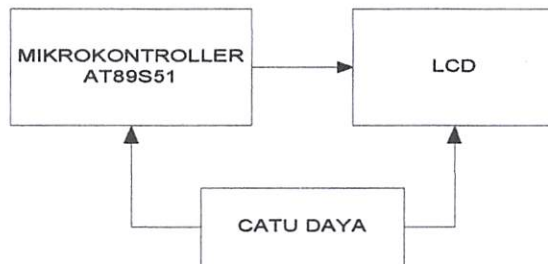
- Pada gambar 4.15 menunjukan proses pengiriman data sedang berlangsung, dengan melihat indikator LED RX dan TX menyala.



Gambar 4.15 Tampilan pada Modul USB.

4.4. Pengujian LCD

Tujuan dari pengujian ini adalah untuk mengetahui apakah LCD dapat berfungsi untuk menampilkan data yang diinginkan dengan baik atau tidak, adapun prosedur pengujianya yaitu



Gambar 4.16. Blok Pengujian LCD

1. Menyusun rangkaian seperti gambar 4.16.
2. Menjalankan program untuk menampilkan tulisan ke LCD.
3. Mengamati keluaran pada LCD.
4. Isi memori program seperti dibawah ini yang menampilkan tulisan
“ABSENSI KARYAWAN” DAN “SILAHKAN ABSEN”

❖ Program Pengujian LCD

```

                ORG 00H

MULAI  :   CALL INIS
          MOV DPTR,#BARIS_A
          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_B
          MOV R3,#16
          MOV A,#0C0H
          CALL TULIS_INST

TULIS1  :   CLR A
          MOVC A,@A+DPTR
          INC DPTR
          CALL TULIS_DATA
          DJNZ R3,TULIS1
          JMP $

INIS    :   MOV A,#03FH
          CALL TULIS_INST
          MOV A,#0DH
          CALL TULIS_INST
          MOV A,#01H
          CALL TULIS_INST
          TULIS_INST : CLR P2.6

```

```

MOV P0,A
SETB P2.7
CLR P2.7
CALL TUNDA
RET
TULIS_DATA : SETB P2.6
MOV P0,A
SETB P2.7
CLR P2.7
CALL TUNDA
RET
TUNDA      : MOV R0,#0FFH
TUNDA1     : MOV R1,#50H
            DJNZ R1,$
            DJNZ R0,TUNDA1
            RET
BARIS_A    : DB ' ABSENSI KARYAWAN '
BARIS_B    : DB ' SILAHKAN ABSEN '
END

```

4.4.1. Hasil Pengujian

Setelah data diolah mikrokontroler maka hasil tampilan LCD berupa tulisan “ABSENSI KARYAWAN SILAHKAN ABSEN “.



Gambar 4.17. Hasil Pengujian Tampilan LCD .

4.5. Pengujian Sistem Keseluruhan

4.5.1. Tujuan

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

4.5.2. Prosedur Pengujian

- Prosedur pertama dalam melakukan pengujian alat dengan menghubungkan keseluruhan rangkaian sesuai dengan diagram blok dan setelah terhubung semua maka aktifkan hardware tersebut.
- Proses kedua adalah dengan menjalankan program Database pada delphi, Dengan syarat driver dan port usb telah diseting.
- Setelah database dengan hardware siap digunakan maka Proses ketiga dilakukan pengujian keseluruhan sistem. Dengan melakukan proses identifikasi RFID dengan cara, mendekatkan tag RFID pada reader RFID, jika proses berhasil maka indikator buzzer dan LED akan menyala.
- Setelah Proses ketiga berhasil maka proses keempat adalah Proses pembacaan database apakah code yang dikirimkan sesuai dengan database. Jika kode yang dimasukkan sesuai maka identitas berupa Nama karyawan, Id Kartu, No Hp, Jenis Kelamin karyawan akan tampil di PC melalui USB dan mikrokontroller akan melanjutkan ke proses selanjutnya dengan menampilkan di LCD dengan text “ABSENSI SUKSES”, tapi jika code yang dimasukan salah maka identitas karyawan tidak tampil di PC melalui USB dan

mikrokontroller akan melanjutkan ke proses selanjutnya dengan menampilkan di LCD dengan text bahwa “ABSENSI GAGAL” begitu juga jika belum menjadi anggota akan menampilkan bahwa “ABSENSI GAGAL”.

4.5.3. Hasil Pengujian

1. Tampilan Informasi Pada Display LCD di Operator.
 - a. Menampilkan Tulisan “ABSENSI KARYAWAN” dan “SILAHKAN ABSEN”.
 - b. Menampilkan Tulisan “0000593870” dan “ABSENSI SUKSES”
 - c. Menampilkan Tulisan “0000593870” dan “ABSENSI GAGAL”.
 - d. Menampilkan Tulisan “0000593870” dan “PULANG”.
2. Tampilan Informasi Pada Program Delphi
 - a. Menampilkan Form DataBase Absensi Karyawan.
 - b. Menampilkan Form Pendaftaran Anggota Baru
 - c. Menampilkan Form Menambahkan Anggota Baru
 - d. Menampilkan Form Pengaturan Jam Absensi.
 - e. Menampilkan Form Pengaturan Port dan Baud Rate

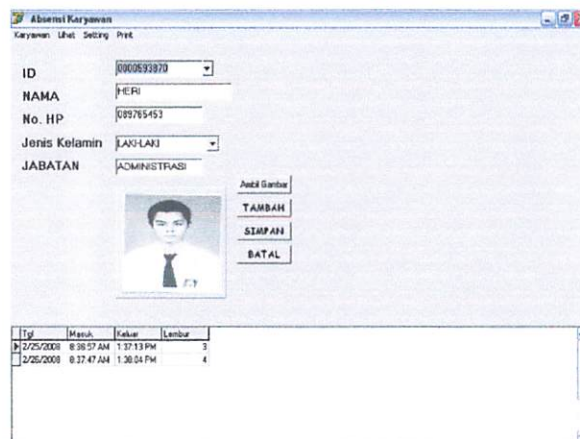
4.6. Proses Identifikasi dan Sistem Absensi Karyawan Pegawai Negeri Berbasis RFID

- a. Setelah seluruh perangkat di hubungkan maka display LCD di operator akan menampilkan tulisan seperti gambar 4.18.



Gambar 4.18. Tampilan LCD Setelah Selulur Perangkat di hubungkan.

- b. Mikrokontroller akan menunggu ID dari kartu untuk dibandingkan dengan PC melalui USB apakah ID tersebut terdaftar atau tidak, jika ID tersebut terdaftar maka PC melalui FT232BM dan mencocokkan dengan database delphi. PC akan menampilkan Database. Pada layar monitor akan menampilkan ID, Nama, No telpon, Jenis Kelamin seperti gambar 4.19.

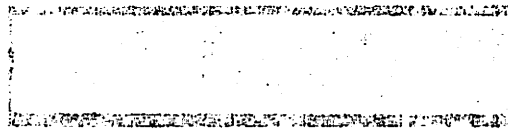


Gambar 4.19. Contoh tampilan Identitas Karyawan.

- c. Kemudian mengirimkan pesan bahwa ABSENSI SUKSES pada LCD seperti yang ditunjukkan gambar 4.20.



Gambar 4.20. Tampilan LCD Sukses Setelah READER Mendeteksi TAG.



Gambar 4.18. Tampilan LCD Setelah selesai melakukan program

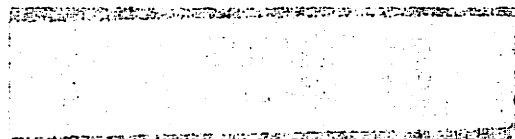
ditambahkan akan muncul ID dan akan muncul dengan PC setelah itu akan muncul dengan ID tersebut maka akan muncul F333B4 dan akan muncul dengan ID yang akan muncul di Database. Pada bagian monitor akan muncul ID yang akan muncul pada bagian seperti gambar 4.19



Gambar 4.19. Tampilan LCD Setelah selesai

dan akan muncul dengan ID yang akan muncul pada LCD

seperti gambar 4.20



Gambar 4.20. Tampilan LCD Setelah selesai melakukan program

Namun jika ID tersebut tidak terdaftar maka PC akan mengirimkan pesan bahwa ABSENSI GAGAL dengan tampilan di LCD seperti yang ditunjukkan pada gambar 4.21.



Gambar 4.21. Tampilan LCD Gagal Setelah READER Mendeteksi TAG.

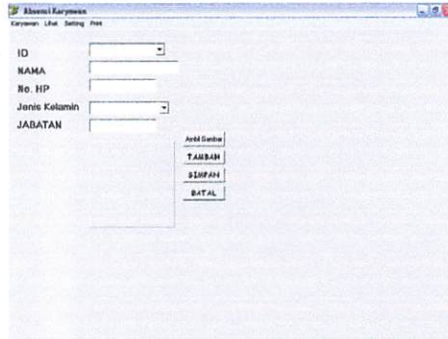
- d. Pada saat melakukan proses pada jam pulang maka PC akan mengirimkan pesan berupa status dari kartu tersebut dengan tampilan seperti gambar 4.22.



Gambar 4.22. Tampilan LCD Status Pulang Setelah TAG Terdeteksi READER.

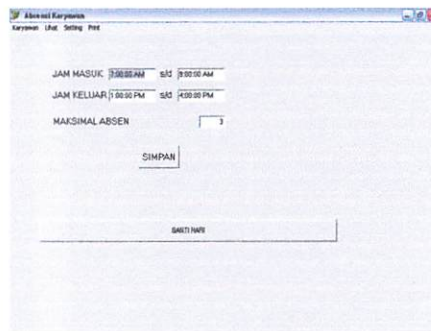
Jika tidak terjadi kesalahan dalam memasukan ID maka PC akan mengirimkan pesan ke mikrokontroller bahwa ABSENSI GAGAL.

- e. Pada gambar 4.23 yaitu tahap untuk proses anggota baru, anggota baru harus mengisi kolom yang tersedia pada database delphi yaitu berupa ID, Nama, Jabatan, No telpon, account secara lengkap. Setelah data terisi semua maka data disimpan pada database.



Gambar 4.23. Tampilan Layar Monitor Membuat Anggota Baru.

- f. Pada gambar 4.29. menunjukkan Jika ingin mennubah jam masuk atau pulang.



Gambar 4.24.

Tampilan Layar Monitor Untuk Penyetingan Jam Masuk dan keluar

- g. Pada gambar 4.25 menunjukkan settingan Port dan Baud Rate yang digunakan pada PC yang harus disesuaikan.

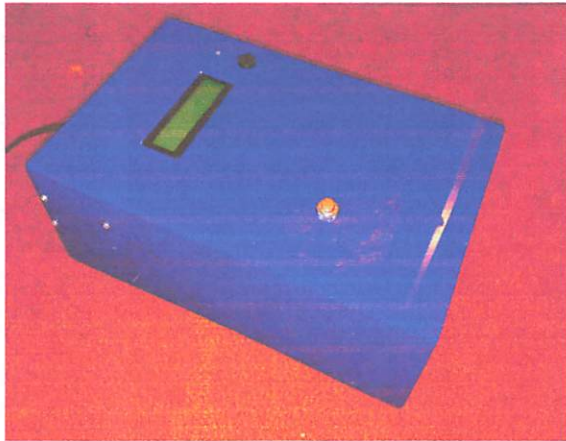


Gambar 4.25. Tampilan settingan Port dan Baud Rate

4.7. Spesifikasi Alat

➤ **Alat Absensi Anggota Pegawai Negeri**

- Dimensi : 21 x 15,5 x 9 cm
- Catu Daya : 5 volt
- Mikrokontroller AT89S51
- LCD M163
- Modul Reader RFID ID 12
- Tag RFID
- IC 74HC164
- Modul FT232BM
- BUZZER dan LED



Gambar 4.26. Alat Absensi Karyawan

BAB V

PENUTUP

5.1. Kesimpulan

1. Pada saat komunikasi dengan computer, *USB* membutuhkan driver agar dapat mengenal port berapa yang akan digunakan.
2. Pada saat konektor *USB* dipindahkan dari satu konektor ke konektor *USB* yang lain pada PC maka *COM* yang terdeteksi pada PC akan berubah-ubah.
3. Pada saat pengujian, pembacaan *RFID tag* oleh Reader dapat mencapai jarak 7 cm, sedangkan jarak yang bagus dalam pembacaan adalah 6 cm

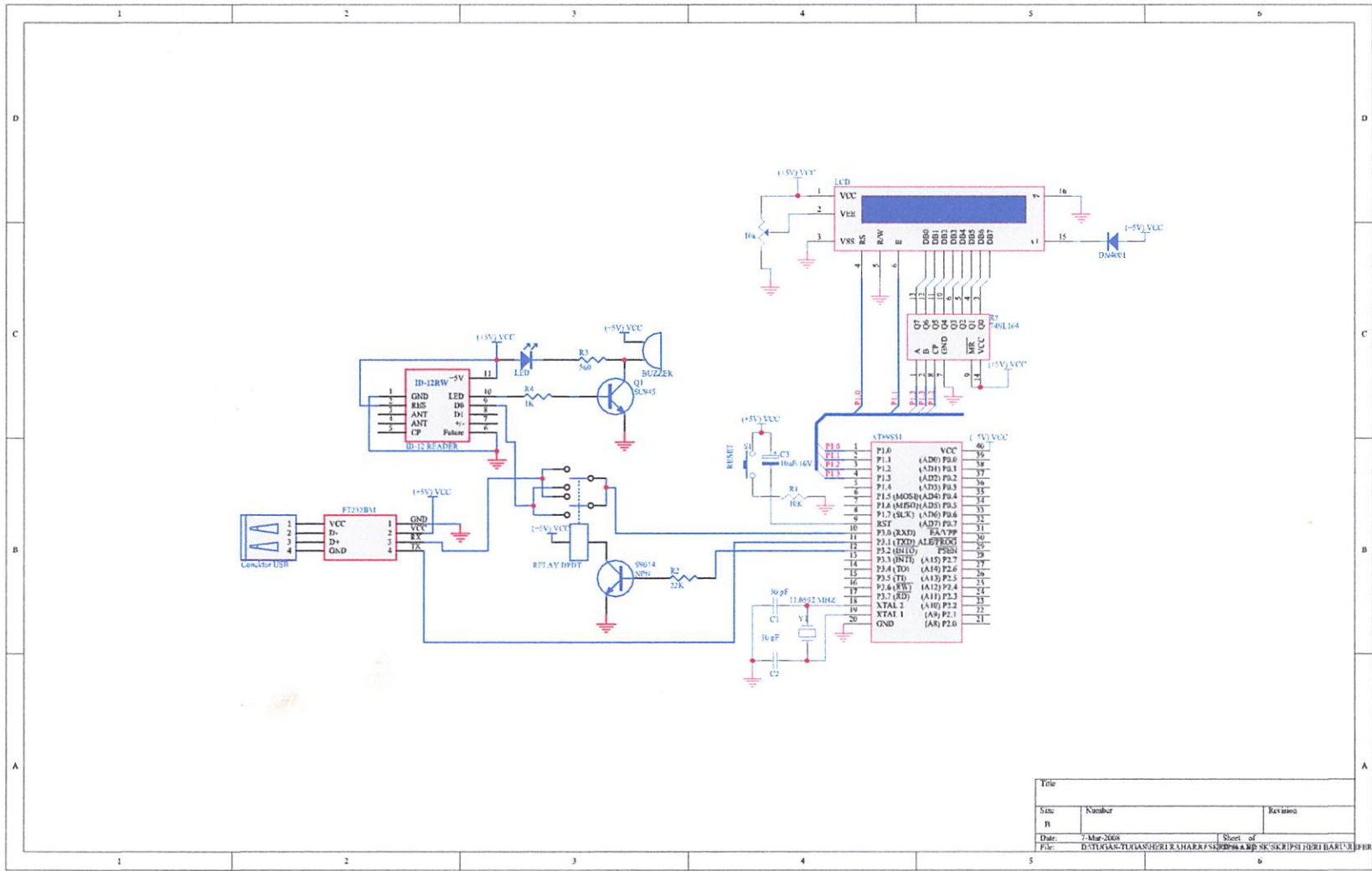
5.2. Saran

Untuk saat ini alat tersebut masih memiliki kekurangan pada system administrasi absensi karena penjaga dapat saja lengah setiap saat sehingga disarankan dengan memberi sebuah kamera pada alat absensi agar dapat berjalan sesuai yang diinginkan.

DAFTAR PUSTAKA

- [1] www.alldatasheet.com, Data Sheet RFID ID 12.
- [2] John Wiley & Son – RFID Handbook
- [3] www.atmel.com, Data Sheet AT89S51.
- [4] support@innovativeelectronics.com USBer USB - UART Converter.
- [5] *Serial*. <http://www.delta-electronics.com>
- [6] www.mot.com , Data Sheet 74LS164.
- [7] LCD Module User Manual.
- [8] www.cui.com Data Sheet Buzzer
- [9] www.tycoelectronics.com, Technical support: Refer to inside back cover.
- [10] Nurchasanah, Teknik Antarmuka Mikrokontroler dengan menggunakan Komputer Berbasis Delphi, Salemba Infotek, Jakarta, 2006.

Lampiran





FORMULIR BIMBINGAN SKRIPSI

Nama : Heri Raharjo
Nim : 03.17.103
Masa Bimbingan : 07-November-2007 s/d 07-Mei-2008
Judul Skripsi : Perancangan dan Pembuatan Absensi Karyawan Pegawai Negeri Berbasis RFID Yang di interface Ke PC Melalui USB.

No	Tanggal	Uraian	Paraf Pembimbing
1	29/08	Bab I & Bab II perbaiki Bab II	
2	5/08 12	Perbaiki Bab III	
3	6/08 12	Perbaiki Flow chart	
4	11/08 12	Bab III (tambah data base)	
5	12/08 12	Bab IV Perbaiki	
6	16/08 12	Bab VI & VII	
7			
8			
9			
10			

Malang,

Dosen pembimbing I,

Ir. F. Yudi Lampraptono, MT

NIP. Y. 1039500274



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

FORMULIR BIMBINGAN SKRIPSI

Nama : Heri Raharjo
Nim : 03.17.103
Masa Bimbingan : 07-November-2007 s/d 07-Mei-2008
Judul Skripsi : Perancangan dan Pembuatan Absensi Karyawan Pegawai Negeri Berbasis RFID Yang di interface Ke PC Melalui USB.

No	Tanggal	Uraian	Paraf Pembimbing
1	23/2 08	Bimbingan demo alat	
2	26/2 08	Bimbingan matakalah seminar hasil	
3	4/3 08	Bimbingan BAB I, II, III, IV, V	
4			
5			
6			
7			
8			
9			
10			

Malang,

Dosen pembimbing II,

Sothyohadi, ST

Form S-4b



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA : *HERI RAHARJO*
N I M : *03.17.103.*
Perbaikan meliputi :

Malang, *15-03-2018*
[Signature]
[Signature]



INSTITUT TEKNOLOGI NASIONAL
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

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

NAMA :
N I M :
Perbaikan meliputi :

Malang,

(_____)


```
unit Unit1;
```

```
interface
```

```
uses
```

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

```
type
```

```
TForm1 = class(TForm)  
    TbNIP: TTable;  
    DSNIP: TDataSource;  
    Notebook1: TNotebook;  
    Label1: TLabel;  
    Label2: TLabel;  
    DBEdit1: TDBEdit;  
    DBEdit2: TDBEdit;  
    Button1: TButton;  
    Button2: TButton;  
    Button3: TButton;  
    DBGrid1: TDBGrid;  
    DBNavigator1: TDBNavigator;  
    ComPort1: TComPort;  
    Button4: TButton;  
    Label3: TLabel;  
    Edit1: TEdit;  
    MainMenu1: TMainMenu;  
    Karyawan1: TMenuItem;  
    Baru1: TMenuItem;  
    Baru2: TMenuItem;  
    Lihat1: TMenuItem;  
    DataKeseluruhan1: TMenuItem;  
    FormKaryawan1: TMenuItem;  
    DBLookupComboBox1: TDBLookupComboBox;  
    Table2: TTable;  
    DataSource2: TDataSource;  
    FormUtama1: TMenuItem;  
    Label9: TLabel;  
    Label10: TLabel;  
    DBEdit8: TDBEdit;
```

Panel1: TPanel;
Image1: TImage;
Button8: TButton;
OpenPictureDialog1: TOpenPictureDialog;
SavePictureDialog1: TSavePictureDialog;
Setting1: TMenuItem;
Jamdanabsensi1: TMenuItem;
Port1: TMenuItem;
Label5: TLabel;
Label6: TLabel;
Label8: TLabel;
Label7: TLabel;
Label4: TLabel;
Label11: TLabel;
DBEdit4: TDBEdit;
Label12: TLabel;
DBEdit5: TDBEdit;
DBEdit7: TDBEdit;
Label13: TLabel;
DBEdit6: TDBEdit;
Label14: TLabel;
Label15: TLabel;
DBEdit3: TDBEdit;
Button6: TButton;
Button7: TButton;
BitBtn1: TBitBtn;
Timer1: TTimer;
Tbslave: TTable;
DSSlave: TDataSource;
Tbpresensi: TTable;
DSPresensi: TDataSource;
Print1: TMenuItem;
CatatanPegawai1: TMenuItem;
RvDataSetConnection1: TRvDataSetConnection;
RvProject1: TRvProject;
RvDataSetConnection2: TRvDataSetConnection;
DBGrid2: TDBGrid;
DBComboBox1: TDBComboBox;
DBEdit9: TDBEdit;
Label16: TLabel;
DBLookupComboBox2: TDBLookupComboBox;
DBEdit10: TDBEdit;

```

Label17: TLabel;
Label18: TLabel;
DBEdit11: TDBEdit;
Label19: TLabel;
Lembur1: TMenuItem;
Button5: TButton;
Button9: TButton;
Edit2: TEdit;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Baru2Click(Sender: TObject);
procedure Baru1Click(Sender: TObject);
procedure DataKeseluruhan1Click(Sender: TObject);
procedure FormKaryawan1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure kirimdata(data:string);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure FormUtama1Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Notebook1PageChanged(Sender: TObject);
procedure Port1Click(Sender: TObject);
procedure Jamdanabsensi1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure CatatanPegawai1Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Lembur1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  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.Button2Click(Sender: TObject);  
begin  
TbNIP.Edit;  
Image1.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+Tb  
NIP.fieldbyname('nip').AsString+'.jpg');  
if gambar<>" then TbNIP.FieldName('foto').AsString:='ada';  
TbNIP.Post;  
Notebook1.ActivePage:='lihat';  
gambar:="";  
Image1.Visible:=false;  
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
TbNIP.Edit;  
Image1.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+Tb  
NIP.fieldbyname('nip').AsString+'.jpg');  
if gambar<>" then TbNIP.FieldName('foto').AsString:='ada';  
TbNIP.Post;  
TbNIP.Append;  
gambar:="";  
Image1.Visible:=false;  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```
begin
TbNIP.Cancel;
Notebook1.ActivePage:='lihat';
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);
begin
ComPort1.ShowSetupDialog;
ComPort1.Open;
Notebook1.ActivePage:='lihat';
Edit1.Text:='';
RvProject1.ProjectFile:=ExtractFilePath(Application.ExeName)+'Project1.rav';
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 TbNIP.FindKey([Edit1.Text]) then
begin
Notebook1.ActivePage:='baru';
BitBtn1.Click;
kirimdata(Edit1.Text+'%');
if TbNIP.fieldbyname('masuk').IsNull then
begin
// if (Time<Table2.fieldbyname('masuk2').AsDateTime) and
// (Time>Table2.fieldbyname('masuk1').AsDateTime) then
// begin
TbNIP.Edit;
TbNIP.fieldbyname('tgl').AsDateTime:=Date;
TbNIP.fieldbyname('masuk').AsDateTime:=Time;
TbNIP.Post;

Tbpresensi.Append;
Tbpresensi.fieldbyname('NO').AsInteger:=Tbpresensi.RecordCount+1;
Tbpresensi.fieldbyname('Nip').AsString:=Edit1.Text;
Tbpresensi.fieldbyname('tgl').AsDateTime:=Date;
```

```

Tbpresensi.fieldbyname('masuk').AsDateTime:=Time;
Tbpresensi.Post;
Application.ProcessMessages;
Tbslave.Refresh;
Application.ProcessMessages;

    if TbNIP.fieldbyname('absen').AsInteger>=
        Table2.fieldbyname('ABSEN').AsInteger then
DATA_KIRIM:='ABSENSI GAGAL%'
        ELSE DATA_KIRIM:='ABSENSI SUKSES%';
//      end
//      else DATA_KIRIM:='ABSENSI SUKSES%';
        end
    else if (Time<Table2.fieldbyname('keluar1').AsDateTime) and
        (Time>Table2.fieldbyname('masuk1').AsDateTime) then
        DATA_KIRIM:='ABSENSI SUKSES%'
    else begin
        if (Time<Table2.fieldbyname('keluar2').AsDateTime) and
            (Time>Table2.fieldbyname('keluar1').AsDateTime) AND
            TbNIP.fieldbyname('keluar').IsNull then
            begin
                TbNIP.Edit;
                TbNIP.fieldbyname('keluar').AsDateTime:=Time;
                TbNIP.Post;

                Tbslave.Last;
                Tbslave.Edit;
                Tbslave.fieldbyname('keluar').AsDateTime:=Time;
                Tbslave.Post;
                Application.ProcessMessages;
                Tbpresensi.Refresh;
                Application.ProcessMessages;
                DATA_KIRIM:='PULANG%';
            end
        else DATA_KIRIM:='ABSENSI GAGAL%';

    end;
        kirimdata(DATA_KIRIM);
    end
else
    begin
        kirimdata(Edit1.Text+'%');
    end

```

```
        kirimdata('ABSENSI GAGAL%');
    end;
    Edit1.Text:="";
end
else if length(Edit1.Text)>10 then Edit1.Text:="";
end;
```

```
procedure TForm1.Baru2Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=false;
    TbNIP.Append;
    Image1.Visible:=false;
end;
```

```
procedure TForm1.Baru1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=true;
    BitBtn1.Click;
end;
```

```
procedure TForm1.DataKeseluruhan1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='lihat';
end;
```

```
procedure TForm1.FormKaryawan1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=true;
    BitBtn1.Click;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Notebook1.ActivePage:='tampilan';
    Image1.Visible:=false;
    Table2.Edit;
    Table2.FieldName('TGL').AsDateTime:=date;
    Table2.Post;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  Tbslave.Last;
  Tbslave.Edit;
  Tbslave.FieldName('lembur').AsString:=Edit2.Text;
  Tbslave.Post;
  Notebook1.ActivePage:='lihat';
  Edit2.Text:='';
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  Table2.Edit;
  Table2.Post;
end;
```

```
procedure TForm1.FormUtama1Click(Sender: TObject);
begin
  Notebook1.ActivePage:='tampilan';
end;
```

```
procedure TForm1.Button7Click(Sender: TObject);
begin
  TbNIP.First;
  repeat
    if TbNIP.FieldName('keluar').IsNull then
      begin
        TbNIP.Edit;
```

```
TbNIP.FieldName('absen').AsInteger:=TbNIP.FieldName('absen').AsInteger+
1;
    TbNIP.Post;
  end;
  TbNIP.Edit;
  TbNIP.FieldName('tgl').AsString:='';
  TbNIP.FieldName('masuk').AsString:='';
  TbNIP.FieldName('keluar').AsString:='';
  TbNIP.Post;
  TbNIP.Next;
  until TbNIP.Eof;
end;
```



```

procedure TForm1.Button8Click(Sender: TObject);
begin
  if OpenPictureDialog1.Execute
  then begin
    Image1.Visible:=true;
    Image1.Picture.LoadFromFile (OpenPictureDialog1.FileName);
  //  ShowMessage(OpenPictureDialog1.FileName);
    gambar:='ada';
  end;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
IF TbNIP.FieldByName('foto').IsNull then
  begin
    ShowMessage('foto tidak ada');
    Image1.Visible:=false;
  end
else begin
  Image1.Visible:=true;

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\'+
TbNIP.fieldbyname('nip').AsString+'.jpg');
  end;
end;

procedure TForm1.Notebook1PageChanged(Sender: TObject);
begin
  Image1.Visible:=false;
end;

procedure TForm1.Port1Click(Sender: TObject);
begin
ComPort1.Close;
ComPort1.ShowSetupDialog;
ComPort1.Open;
end;

procedure TForm1.Jamdanabsensi1Click(Sender: TObject);
begin
Notebook1.ActivePage:='set';

```

end;

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
IF Table2.FieldName('TGL').AsDateTime<>DATE THEN
begin
Button7.Click;
Table2.Edit;
Table2.FieldName('TGL').AsDateTime:=date;
Table2.Post;
end;
end;
```

```
procedure TForm1.CatatanPegawai1Click(Sender: TObject);
begin
RvProject1.Execute;
end;
```

```
procedure TForm1.Button9Click(Sender: TObject);
begin
Notebook1.ActivePage:='lihat';
Edit2.Text:='';
end;
```

```
procedure TForm1.Lembur1Click(Sender: TObject);
begin
Notebook1.ActivePage:='lembur';
end;
```

end.

unit Unit1;

interface

uses

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

type

TForm1 = class(TForm)
 TbNIP: TTable;
 DSNIP: TDataSource;
 Notebook1: TNotebook;
 Label1: TLabel;
 Label2: TLabel;
 DBEdit1: TDBEdit;
 DBEdit2: TDBEdit;
 Button1: TButton;
 Button2: TButton;
 Button3: TButton;
 DBGrid1: TDBGrid;
 DBNavigator1: TDBNavigator;
 ComPort1: TComPort;
 Button4: TButton;
 Label3: TLabel;
 Edit1: TEdit;
 MainMenu1: TMainMenu;
 Karyawan1: TMenuItem;
 Baru1: TMenuItem;
 Baru2: TMenuItem;
 Lihat1: TMenuItem;
 DataKeseluruhan1: TMenuItem;
 FormKaryawan1: TMenuItem;
 DBLookupComboBox1: TDBLookupComboBox;
 Table2: TTable;
 DataSource2: TDataSource;
 FormUtama1: TMenuItem;
 Label9: TLabel;
 Label10: TLabel;
 DBEdit8: TDBEdit;

Panel1: TPanel;
Image1: TImage;
Button8: TButton;
OpenPictureDialog1: TOpenPictureDialog;
SavePictureDialog1: TSavePictureDialog;
Setting1: TMenuItem;
Jamdanabsensi1: TMenuItem;
Port1: TMenuItem;
Label5: TLabel;
Label6: TLabel;
Label8: TLabel;
Label7: TLabel;
Label4: TLabel;
Label11: TLabel;
DBEdit4: TDBEdit;
Label12: TLabel;
DBEdit5: TDBEdit;
DBEdit7: TDBEdit;
Label13: TLabel;
DBEdit6: TDBEdit;
Label14: TLabel;
Label15: TLabel;
DBEdit3: TDBEdit;
Button6: TButton;
Button7: TButton;
BitBtn1: TBitBtn;
Timer1: TTimer;
Tbslave: TTable;
DSSlave: TDataSource;
Tbpresensi: TTable;
DSPresensi: TDataSource;
Print1: TMenuItem;
CatatanPegawai1: TMenuItem;
RvDataSetConnection1: TRvDataSetConnection;
RvProject1: TRvProject;
RvDataSetConnection2: TRvDataSetConnection;
DBGrid2: TDBGrid;
DBComboBox1: TDBComboBox;
DBEdit9: TDBEdit;
Label16: TLabel;
DBLookupComboBox2: TDBLookupComboBox;
DBEdit10: TDBEdit;

```

Label17: TLabel;
Label18: TLabel;
DBEdit11: TDBEdit;
Label19: TLabel;
Lembur1: TMenuItem;
Button5: TButton;
Button9: TButton;
Edit2: TEdit;
procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure ComPort1RxChar(Sender: TObject; Count: Integer);
procedure Baru2Click(Sender: TObject);
procedure Baru1Click(Sender: TObject);
procedure DataKeseluruhan1Click(Sender: TObject);
procedure FormKaryawan1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure kirimdata(data:string);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure FormUtama1Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure Notebook1PageChanged(Sender: TObject);
procedure Port1Click(Sender: TObject);
procedure Jamdanabsensi1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure CatatanPegawai1Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Lembur1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Form1: TForm1;
  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.Button2Click(Sender: TObject);  
begin  
TbNIP.Edit;  
Image1.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+Tb  
NIP.fieldbyname('nip').AsString+'.jpg');  
if gambar<>" then TbNIP.FieldName('foto').AsString:='ada';  
TbNIP.Post;  
Notebook1.ActivePage:='lihat';  
gambar:="";  
Image1.Visible:=false;  
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
TbNIP.Edit;  
Image1.Picture.SaveToFile(ExtractFilePath(Application.ExeName)+'\gambar\'+Tb  
NIP.fieldbyname('nip').AsString+'.jpg');  
if gambar<>" then TbNIP.FieldName('foto').AsString:='ada';  
TbNIP.Post;  
TbNIP.Append;  
gambar:="";  
Image1.Visible:=false;  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
```

```

begin
TbNIP.Cancel;
Notebook1.ActivePage:='lihat';
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
ComPort1.ShowSetupDialog;
ComPort1.Open;
Notebook1.ActivePage:='lihat';
Edit1.Text:='';
RvProject1.ProjectFile:=ExtractFilePath(Application.ExeName)+'Project1.rav';
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 TbNIP.FindKey([Edit1.Text]) then
begin
Notebook1.ActivePage:='baru';
BitBtn1.Click;
kirimdata(Edit1.Text+'%');
if TbNIP.fieldbyname('masuk').IsNull then
begin
// if (Time<Table2.fieldbyname('masuk2').AsDateTime) and
// (Time>Table2.fieldbyname('masuk1').AsDateTime) then
// begin
TbNIP.Edit;
TbNIP.fieldbyname('tgl').AsDateTime:=Date;
TbNIP.fieldbyname('masuk').AsDateTime:=Time;
TbNIP.Post;

Tbpresensi.Append;
Tbpresensi.fieldbyname('NO').AsInteger:=Tbpresensi.RecordCount+1;
Tbpresensi.fieldbyname('Nip').AsString:=Edit1.Text;
Tbpresensi.fieldbyname('tgl').AsDateTime:=Date;

```

```

Tbpresensi.fieldbyname('masuk').AsDateTime:=Time;
Tbpresensi.Post;
Application.ProcessMessages;
Tbslave.Refresh;
Application.ProcessMessages;

    if TbNIP.fieldbyname('absen').AsInteger>=
        Table2.fieldbyname('ABSEN').AsInteger then
DATA_KIRIM:='ABSENSI GAGAL%'
        ELSE DATA_KIRIM:='ABSENSI SUKSES%';
//      end
//      else DATA_KIRIM:='ABSENSI SUKSES%';
        end
    else if (Time<Table2.fieldbyname('keluar1').AsDateTime) and
        (Time>Table2.fieldbyname('masuk1').AsDateTime) then
        DATA_KIRIM:='ABSENSI SUKSES%'
    else begin
        if (Time<Table2.fieldbyname('keluar2').AsDateTime) and
            (Time>Table2.fieldbyname('keluar1').AsDateTime) AND
            TbNIP.fieldbyname('keluar').IsNull then
            begin
                TbNIP.Edit;
                TbNIP.fieldbyname('keluar').AsDateTime:=Time;
                TbNIP.Post;

                Tbslave.Last;
                Tbslave.Edit;
                Tbslave.fieldbyname('keluar').AsDateTime:=Time;
                Tbslave.Post;
                Application.ProcessMessages;
                Tbpresensi.Refresh;
                Application.ProcessMessages;
                DATA_KIRIM:='PULANG%';
                end
            else DATA_KIRIM:='ABSENSI GAGAL%';

        end;
        kirimdata(DATA_KIRIM);
    end
else
    begin
        kirimdata(Edit1.Text+'%');

```



```
        kirimdata('ABSENSI GAGAL%');
    end;
    Edit1.Text:="";
end
else if length(Edit1.Text)>10 then Edit1.Text:="";
end;
```

```
procedure TForm1.Baru2Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=false;
    TbNIP.Append;
    Image1.Visible:=false;
end;
```

```
procedure TForm1.Baru1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=true;
    BitBtn1.Click;
end;
```

```
procedure TForm1.DataKeseluruhan1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='lihat';
end;
```

```
procedure TForm1.FormKaryawan1Click(Sender: TObject);
begin
    Notebook1.ActivePage:='baru';
    DBGrid2.Visible:=true;
    BitBtn1.Click;
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Notebook1.ActivePage:='tampilan';
    Image1.Visible:=false;
    Table2.Edit;
    Table2.FieldName('TGL').AsDateTime:=date;
    Table2.Post;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  Tbslave.Last;
  Tbslave.Edit;
  Tbslave.FieldName('lembur').AsString:=Edit2.Text;
  Tbslave.Post;
  Notebook1.ActivePage:='lihat';
  Edit2.Text:='';
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  Table2.Edit;
  Table2.Post;
end;
```

```
procedure TForm1.FormUtama1Click(Sender: TObject);
begin
  Notebook1.ActivePage:='tampilan';
end;
```

```
procedure TForm1.Button7Click(Sender: TObject);
begin
  TbNIP.First;
  repeat
    if TbNIP.FieldName('keluar').IsNull then
      begin
        TbNIP.Edit;
```

```
TbNIP.FieldName('absen').AsInteger:=TbNIP.FieldName('absen').AsInteger+
1;
    TbNIP.Post;
  end;
  TbNIP.Edit;
  TbNIP.FieldName('tgl').AsString:='';
  TbNIP.FieldName('masuk').AsString:='';
  TbNIP.FieldName('keluar').AsString:='';
  TbNIP.Post;
  TbNIP.Next;
  until TbNIP.Eof;
end;
```

```

procedure TForm1.Button8Click(Sender: TObject);
begin
    if OpenPictureDialog1.Execute
    then begin
        Image1.Visible:=true;
        Image1.Picture.LoadFromFile (OpenPictureDialog1.FileName);
// ShowMessage(OpenPictureDialog1.FileName);
        gambar:='ada';
        end;
end;

procedure TForm1.BitBtn1Click(Sender: TObject);
begin
IF TbNIP.FieldByName('foto').IsNull then
    begin
        ShowMessage('foto tidak ada');
        Image1.Visible:=false;
        end
else begin
    Image1.Visible:=true;

Image1.Picture.LoadFromFile(ExtractFilePath(Application.ExeName)+'\gambar\''+
TbNIP.fieldbyname('nip').AsString+'.jpg');
    end;
end;

procedure TForm1.Notebook1PageChanged(Sender: TObject);
begin
    Image1.Visible:=false;
end;

procedure TForm1.Port1Click(Sender: TObject);
begin
ComPort1.Close;
ComPort1.ShowSetupDialog;
ComPort1.Open;
end;

procedure TForm1.Jamdanabsensi1Click(Sender: TObject);
begin
Notebook1.ActivePage:='set';

```

end;

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
IF Table2.FieldByName('TGL').AsDateTime<>DATE THEN
  begin
  Button7.Click;
  Table2.Edit;
  Table2.FieldByName('TGL').AsDateTime:=date;
  Table2.Post;
  end;
end;
```

```
procedure TForm1.CatatanPegawai1Click(Sender: TObject);
begin
RvProject1.Execute;
end;
```

```
procedure TForm1.Button9Click(Sender: TObject);
begin
Notebook1.ActivePage:='lihat';
Edit2.Text:='';
end;
```

```
procedure TForm1.Lembur1Click(Sender: TObject);
begin
Notebook1.ActivePage:='lembur';
end;
```

end.

```

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

#define cs    P3_2//1_4

unsigned char j[16],i,trfid,k[20],a,b1,b2,b3,b4,b5,b6,b7,b8,tandap;
char mn,jm,dt;
//unsigned int atarget,r;
unsigned long Ndata;
bit t;

void konrfid()
{

//Nurjannah=0x10000000*kon(j[3])|0x1000000*kon(j[4])|0x100000*kon(j[5])|0x1
0000*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]);b6=kon(j[8]);
  b7=kon(j[9]);b8=kon(j[10]);

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

//Nurjannah=2657107;
cinta=Ndata/1000000000+0x30;k[0]=cinta;
dataout();
cinta=(Ndata/100000000)%10+0x30;k[1]=cinta;
dataout();
cinta=(Ndata/10000000)%10+0x30;k[2]=cinta;
dataout();
cinta=(Ndata/1000000)%10+0x30;k[3]=cinta;
dataout();
cinta=(Ndata/100000)%10+0x30;k[4]=cinta;
dataout();
cinta=(Ndata/10000)%10+0x30;k[5]=cinta;
dataout();
cinta=(Ndata/1000)%10+0x30;k[6]=cinta;
dataout();
cinta=(Ndata/100)%10+0x30;k[7]=cinta;
dataout();
cinta=(Ndata/10)%10+0x30;k[8]=cinta;
dataout();
cinta=Ndata%10+0x30;k[9]=cinta;
dataout();

```

```

t=1;
//pos(2,1);
}

void cetakpass()
{
a=0;
while(a!='E')
{
key();
a=cinta;
k[i]=cinta;i++;
cinta='*';
dataout();
}
}
void kirimkom()
{
i=0;
while(i<10)
{
cinta=k[i];
PutChar(cinta);
i++;
}
//i=0;
}

void tampil()
{
cinta=0;a=0;
while(cinta!='%')
{cinta=GetChar();
k[a]=cinta;a++;
}
a=0;cinta=0;
while(cinta!='%')
{cinta=k[a];
if(cinta=='%');
else dataout();
a++;
}
}
//-----
// Program Utama

```

```

// -----
void main ()
{
  /* Begin of Main
  //TMOD=0x10;
  initser(0xfd);
  initlcd();
  //initlcd2();
  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");
  //t=1000;
  while(1)/* Pengulangan Loop tanpa henti
  {
    /* Begin of While
    //busek();

    cetak(1,1,"ABSENSI KARYAWAN");
    cetak(2,1," MASUKKAN ID ");

    if (t==0)
    {
      hapus();cetak(1,1,"ID: ");konrfid();
      //cetakpass();
      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

```

```
if(i==1)
    heri=0x80+n-1;
else
    heri=0xc0+n-1;
    callout();
}
void cetak(int i,int n,unsigned char *text)
{
    //a=0;
    //n=0;
    pos(i,n);
while(*text)
    {
        heri>(*text++);
        dataout();
        //n=n+1;
        //a=a+1;
    }
}
void busek()
{
    cetak(1,1,"");
    cetak(2,1,"");
    pos(1,1);
}

#endif
```



```

    rrc a
    mov p1.3,c
    nop
    nop
    nop
    nop
    setb p1.2
    nop
    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()
{
    heri=0x1;
    callout();
    heri=0x38;
    callout();
    heri=0x6;
    callout();
    heri=0xc;
    callout();
}

void pos(int i,int n)
{

```

```

#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 heri;

```

```

void tunda1ms()
{
    _asm
        mov     R6,#2
Liip:  mov     R7,#230
Loop:  djnz   R7,Loop
        djnz   R6,Liip
    _endasm;
}

```

```

void time1ms()
{
    int i;
    for (i=0; i<150; i++)
        {;}
}

```

```

void delay(int n)
{
    int i;
    for (i=0; i<n; i++)
        tunda1ms();//time1ms();
}

```

```

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

```

```

void geser()
{
    _asm
        mov a,_heri
        clr p1.2
        mov b,#8
al:

```

```

#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)

```

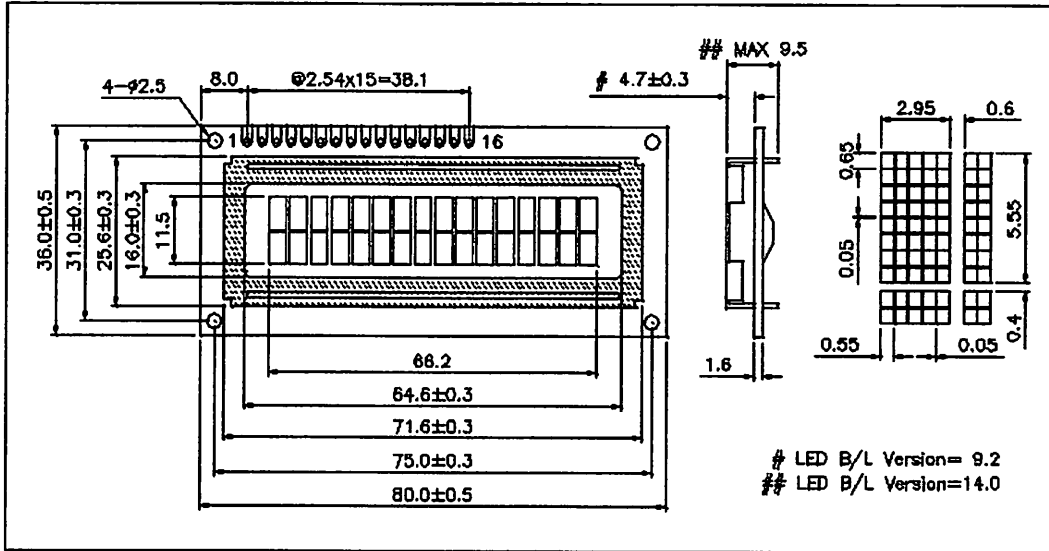
```
{
    while( *text )    // while not end of text
    {
        PutChar(*text++);    // Write character and increment position
    }
return;
}
```

```
void kirimgps(unsigned char *text)
{
    while( *text!='\0' )    // 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
```



ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Min.	Max.	Unit
Supply Voltage(L logic)	V _{DD} -V _{SS}	-0.3	7.0	V
Supply Voltage(LCD)	V _{DD} -V _b	-0.3	13.0	V
Input Voltage	V _i	-0.3	V _{DD} +0.3	V
Operating Temp.	T _{op}	-20	70	°C
Storage Temp.	T _{stg}	-30	80	°C

MECHANICAL DATA

Item	Nominal Dimensions	Unit
Module Size(W x H x T)	80.0 x 36.0 x 9.5	mm
Viewing Area(W x H)	64.6 x 16.0	mm
Character Size(W x H)	2.95 x 5.55	mm
Dot Size(W x H)	0.55 x 0.65	mm
Weight		g

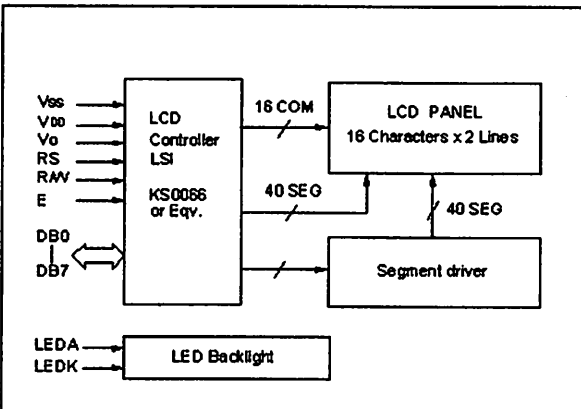
ELECTRICAL CHARACTERISTICS (V_{DD}=5V±0.25V)

Item	Symbol	Test Condition	Min.	Typ.	Max.	Unit
Input High Voltage	V _{ih}	--	2.2	--	V _{DD}	V
Input Low Voltage	V _{il}	--	-0.3	--	0.6	V
Output High Voltage	V _{oh}	I _{OH} = -0.2mA	2.4	--	V _{DD}	V
Output Low Voltage	V _{ol}	I _{OL} = 1.2mA	0	--	0.4	V
Supply Current	I _{DD}	V _{DD} = 5.0V	--	1.5	3.0	mA
LCD Driving Voltage	V _{DD} -V _b	T _a =25°C	--	4.6	--	V

PIN CONNECTIONS

Pin	Symbol	Level	Function
1	V _{SS}	--	GND (0V)
2	V _{DD}	--	Supply Voltage for Logic(+5V)
3	V _b	--	LCD Driving Voltage
4	RS	HAL	H : Data L : Instruction Code
5	RAW	HAL	H : Read L : Write
6	E	H,H=L	Enable Signal
7	DB0	HAL	Data Bus Line
8	DB1	HAL	
9	DB2	HAL	
10	DB3	HAL	
11	DB4	HAL	
12	DB5	HAL	
13	DB6	HAL	
14	DB7	HAL	
15	LEDA	--	LED Power Supply
16	LEDK	--	

BLOCK DIAGRAM



LED BACKLIGHT SPECIFICATIONS (T_a=25°C)

Item	Symbol	Typ.	Max.	Unit
Forward Voltage	V _f	4.1	4.3	V
Forward Current	I _f	158	--	mA
Emission Wave Length	λ _p	568	--	nm



TO-92



1.EMITTER

2.BASE

3.COLLECTOR



1 2 3

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	$V_{(BR)EBO}$	$I_E = 100 \mu\text{A}, I_C = 0$	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_{EBO}	$V_{EB} = 3 \text{ V}, I_C = 0$			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

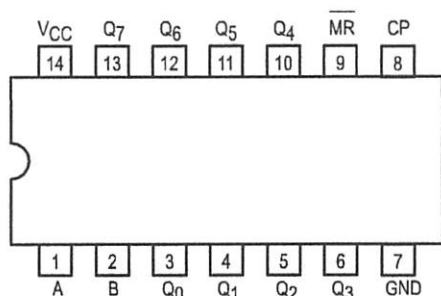


SERIAL-IN PARALLEL-OUT SHIFT REGISTER

The SN54/74LS164 is a high speed 8-Bit Serial-In Parallel-Out Shift Register. Serial data is entered through a 2-Input AND gate synchronous with the LOW to HIGH transition of the clock. The device features an asynchronous Master Reset which clears the register setting all outputs LOW independent of the clock. It utilizes the Schottky diode clamped process to achieve high speeds and is fully compatible with all Motorola TTL products.

- Typical Shift Frequency of 35 MHz
- Asynchronous Master Reset
- Gated Serial Data Input
- Fully Synchronous Data Transfers
- Input Clamp Diodes Limit High Speed Termination Effects
- ESD > 3500 Volts

CONNECTION DIAGRAM DIP (TOP VIEW)



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

PIN NAMES

A, B	Data Inputs
CP	Clock (Active HIGH Going Edge) Input
MR	Master Reset (Active LOW) Input
Q ₀ -Q ₇	Outputs (Note b)

LOADING (Note a)

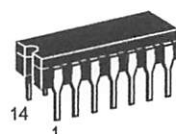
	HIGH	LOW
A, B	0.5 U.L.	0.25 U.L.
CP	0.5 U.L.	0.25 U.L.
MR	0.5 U.L.	0.25 U.L.
Q ₀ -Q ₇	10 U.L.	5 (2.5) U.L.

NOTES:

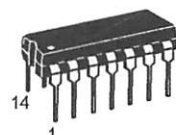
- a) 1 TTL Unit Load (U.L.) = 40 μ A HIGH/1.6 mA LOW.
b) The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

SN54/74LS164

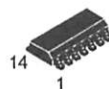
SERIAL-IN PARALLEL-OUT SHIFT REGISTER LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08



N SUFFIX
PLASTIC
CASE 646-06

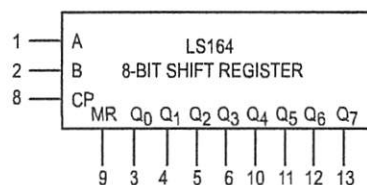


D SUFFIX
SOIC
CASE 751A-02

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXD	SOIC

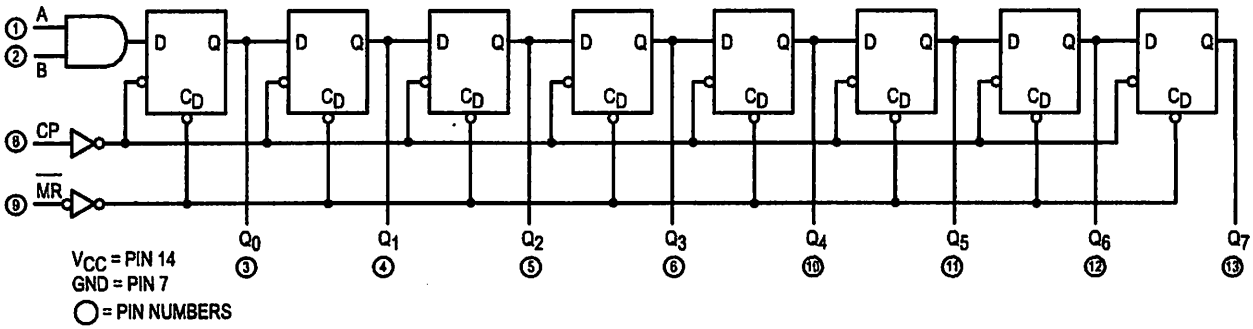
LOGIC SYMBOL



V_{CC} = PIN 14
GND = PIN 7

SN54/74LS164

LOGIC DIAGRAM



FUNCTIONAL DESCRIPTION

The LS164 is an edge-triggered 8-bit shift register with serial data entry and an output from each of the eight stages. Data is entered serially through one of two inputs (A or B); either of these inputs can be used as an active HIGH Enable for data entry through the other input. An unused input must be tied HIGH, or both inputs connected together.

Each LOW-to-HIGH transition on the Clock (CP) input shifts data one place to the right and enters into Q₀ the logical AND of the two data inputs (A•B) that existed before the rising clock edge. A LOW level on the Master Reset (MR) input overrides all other inputs and clears the register asynchronously, forcing all Q outputs LOW.

MODE SELECT — TRUTH TABLE

OPERATING MODE	INPUTS			OUTPUTS	
	MR	A	B	Q ₀	Q ₁ -Q ₇
Reset (Clear)	L	X	X	L	L - L
Shift	H	l	l	L	q ₀ - q ₆
	H	l	h	L	q ₀ - q ₆
	H	h	l	L	q ₀ - q ₆
	H	h	h	H	q ₀ - q ₆

L (l) = LOW Voltage Levels
H (h) = HIGH Voltage Levels
X = Don't Care
q_n = Lower case letters indicate the state of the referenced input or output one set-up time prior to the LOW to HIGH clock transition.

GUARANTEED OPERATING RANGES

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

SN54/74LS164

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
V _{IH}	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V _{IL}	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
V _{IK}	Input Clamp Diode Voltage		-0.65	-1.5	V	V _{CC} = MIN, I _{IN} = -18 mA	
V _{OH}	Output HIGH Voltage	54	2.5	3.5	V	V _{CC} = MIN, I _{OH} = MAX, V _{IN} = V _{IH} or V _{IL} per Truth Table	
		74	2.7	3.5			
V _{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I _{OL} = 4.0 mA V _{CC} = V _{CC} MIN, V _{IN} = V _{IH} or V _{IL} per Truth Table
		74		0.35	0.5	V	
I _{IH}	Input HIGH Current				20	μA	V _{CC} = MAX, V _{IN} = 2.7 V
					0.1	mA	V _{CC} = MAX, V _{IN} = 7.0 V
I _{IL}	Input LOW Current				-0.4	mA	V _{CC} = MAX, V _{IN} = 0.4 V
I _{OS}	Short Circuit Current (Note 1)	-20			-100	mA	V _{CC} = MAX
I _{CC}	Power Supply Current				27	mA	V _{CC} = MAX

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

AC CHARACTERISTICS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
f _{MAX}	Maximum Clock Frequency	25	36		MHz	V _{CC} = 5.0 V C _L = 15 pF
t _{PHL}	Propagation Delay MR to Output Q		24	36	ns	
t _{PLH} t _{PHL}	Propagation Delay Clock to Output Q		17 21	27 32	ns	

AC SETUP REQUIREMENTS (T_A = 25°C)

Symbol	Parameter	Limits			Unit	Test Conditions
		Min	Typ	Max		
t _W	CP, MR Pulse Width	20			ns	V _{CC} = 5.0 V
t _s	Data Setup Time	15			ns	
t _h	Data Hold Time	5.0			ns	
t _{rec}	MR to Clock Recovery Time	20			ns	

SN54/74LS164

AC WAVEFORMS

*The shaded areas indicate when the input is permitted to change for predictable output performance.

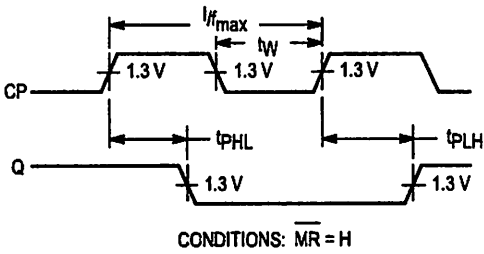


Figure 1. Clock to Output Delays and Clock Pulse Width

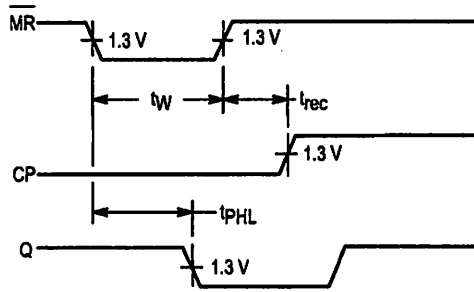


Figure 2. Master Reset Pulse Width, Master Reset to Output Delay and Master Reset to Clock Recovery Time

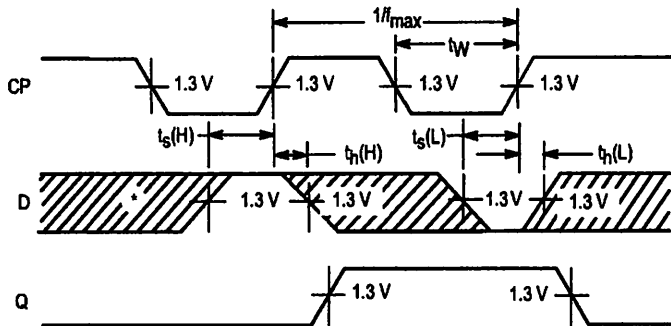


Figure 3. Data Setup and Hold Times

Features

- Compatible with MCS-51[®] Products
- 4K 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
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six 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 AT89S51 is a low-power, high-performance CMOS 8-bit microcontroller with 4K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S51 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.



8-bit Microcontroller with 4K Bytes In-System Programmable Flash

AT89S51

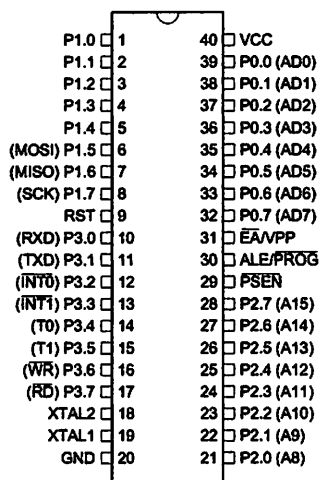
Rev. 2487A-10/01



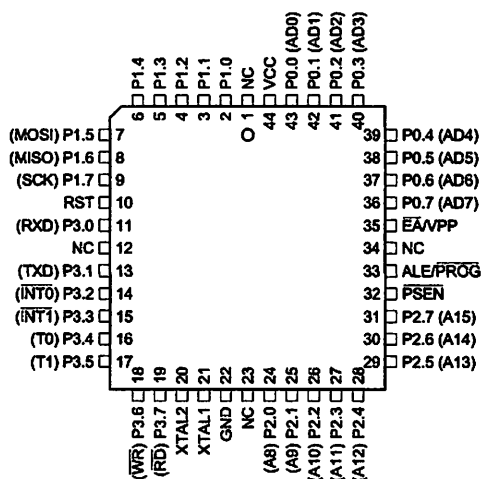


in Configurations

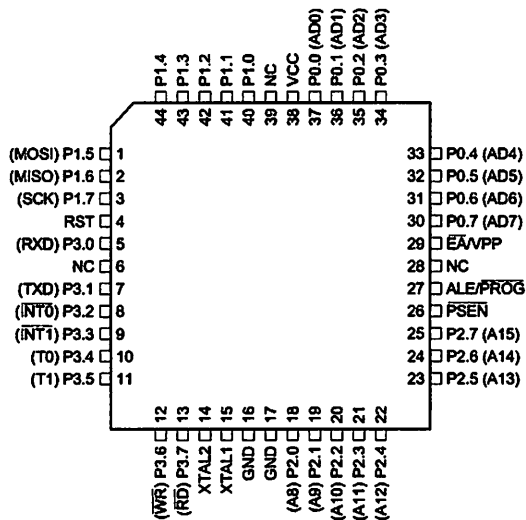
PDIP



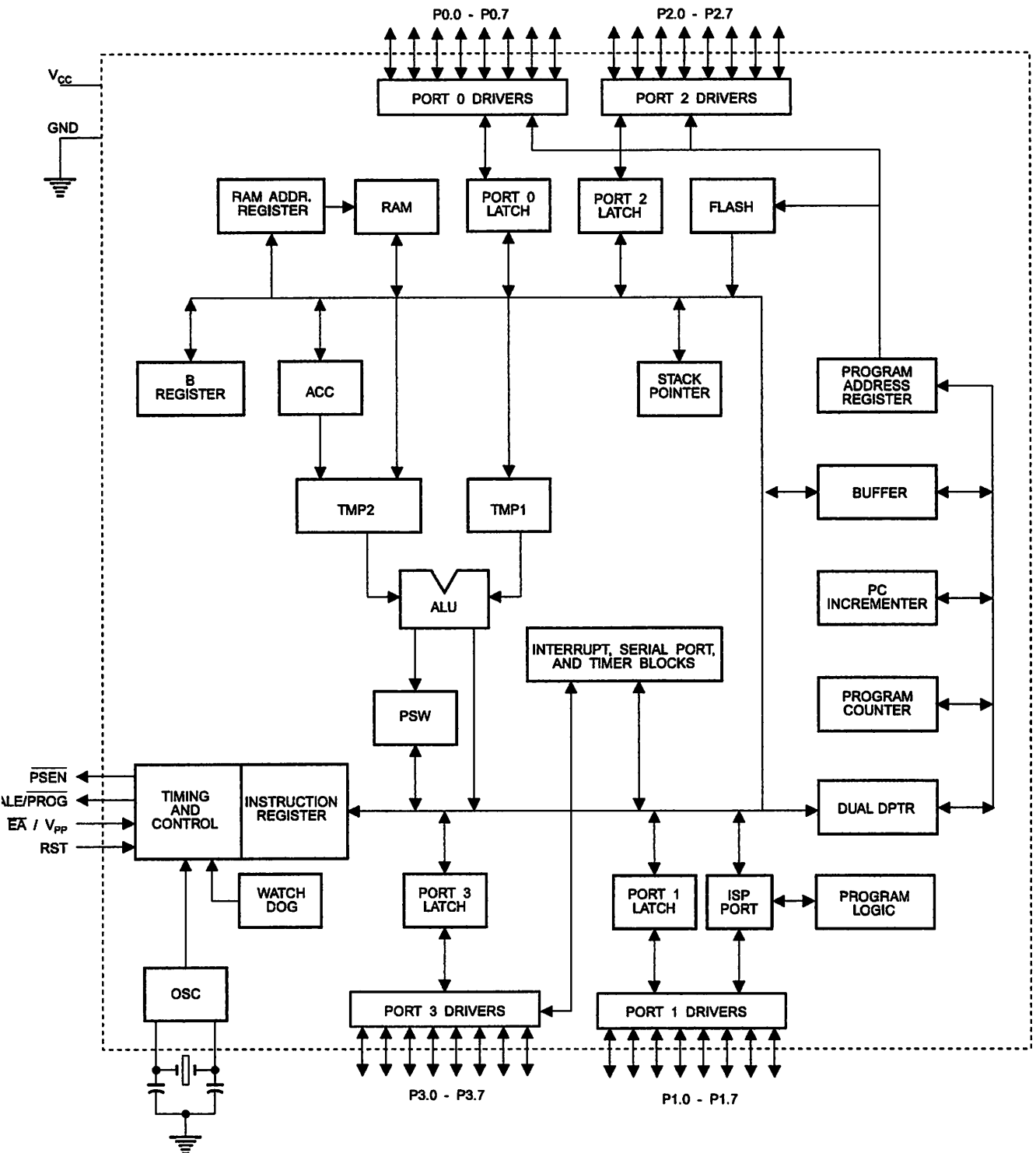
PLCC



TQFP



Block Diagram





Pin Description

CC Supply voltage.

ND Ground.

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

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

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

Port 1 Port 1 is an 8-bit 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.

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

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

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

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

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

Port 3 Port 3 is an 8-bit 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 AT89S51, as shown in the following table.

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

ST

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.

$\overline{\text{ALE/PROG}}$

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

$\overline{\text{PSEN}}$

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

When the AT89S51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA/VPP}}$

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

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

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

TAL1

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

TAL2

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.

Table 1. AT89S51 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111		AUXR1 XXXXXX0				WDRST XXXXXXX		0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XX00XX0		8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000		PCON 0XX0000	87H

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.

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

Table 2. AUXR: Auxiliary Register

AUXR		Address = 8EH					Reset Value = XXX00XX0B		
Not Bit Addressable									
		-	-	-	WDIDLE	DISRTO	-	-	DISALE
Bit		7	6	5	4	3	2	1	0
-		Reserved for future expansion							
DISALE		Disable/Enable ALE							
		DISALE							
		Operating Mode							
	0	ALE is emitted at a constant rate of 1/6 the oscillator frequency							
	1	ALE is active only during a MOVX or MOVC instruction							
DISRTO		Disable/Enable Reset out							
		DISRTO							
	0	Reset pin is driven High after WDT times out							
	1	Reset pin is input only							
WDIDLE		Disable/Enable WDT in IDLE mode							
		WDIDLE							
	0	WDT continues to count in IDLE mode							
	1	WDT halts counting in IDLE mode							

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





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

Table 3. AUXR1: Auxiliary Register 1

AUXR1							
Address = A2H							
							Reset Value = XXXXXXX0B
Not Bit Addressable							
	-	-	-	-	-	-	DPS
Bit	7	6	5	4	3	2	1
-	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 AT89S51, if \overline{EA} is connected to V_{CC} , program fetches to addresses 0000H through FFFH are directed to internal memory and fetches to addresses 1000H through FFFFH are directed to external memory.

Data Memory

The AT89S51 implements 128 bytes of on-chip RAM. The 128 bytes are accessible via direct and indirect addressing modes. Stack operations are examples of indirect addressing, so the 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.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 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 $98 \times TOSC$, where $TOSC = 1/FOSC$. To make the best use of the WDT, it

should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt, which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S51 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 AT89S51 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 AT89S51 operates the same way as the UART in the AT89C51. 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 AT89S51 operate the same way as Timer 0 and Timer 1 in the AT89C51. 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'.

Interrupts

The AT89S51 has a total of five interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), two timer interrupts (Timers 0 and 1), and the serial port interrupt. These interrupts are all shown in Figure 1.

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 4 shows that bit position IE.6 is unimplemented. In the AT89S51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

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





Table 4. Interrupt Enable (IE) Register

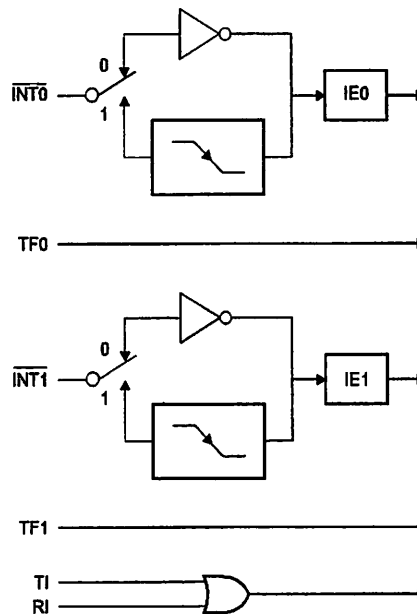
(MSB)				(LSB)			
EA	-	-	ES	ET1	EX1	ET0	EX0

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

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

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

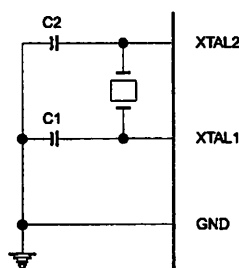
Figure 1. Interrupt Sources



Oscillator Characteristics

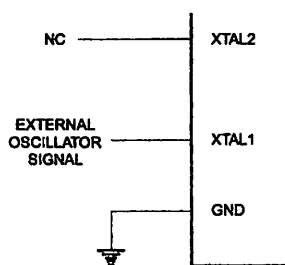
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 3. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 2. Oscillator Connections



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

Figure 3. External Clock Drive Configuration



Idle Mode

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

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

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by activation of an enabled external interrupt into $\overline{INT0}$ or $\overline{INT1}$. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.





Table 5. Status of External Pins During Idle and Power-down Modes

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

Program Memory Lock Bits

The AT89S51 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 6. Lock Bit Protection Modes

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

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

Programming the Flash – Parallel Mode

The AT89S51 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S51 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S51, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S51, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V.
5. Pulse ALE/PROG once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 μ s. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89S51 features \overline{Data} Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. \overline{Data} Polling may begin any time after a write cycle has been initiated.

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

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

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

(000H) = 1EH indicates manufactured by Atmel
 (100H) = 51H indicates 89S51
 (200H) = 06H

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

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

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

Programming the Flash – Serial Mode

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

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

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S51 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
Apply power between VCC and GND pins.
Set RST pin to "H".
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction that returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.





Power-off sequence (if needed):

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

Set RST to "L".

Turn V_{CC} power off.

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

Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 8 on page 18.

Programming Interface – Parallel Mode

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

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

Table 7. Flash Programming Modes

Mode	V_{CC}	RST	\overline{PSEN}	ALE/ PROG	\overline{EA}/V_{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.3-0	P1.7-0
												Address	
Write Code Data	5V	H	L	(2)	12V	L	H	H	H	H	D_{IN}	A11-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A11-8	A7-0
Write Lock Bit 1	5V	H	L	(3)	12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L	(3)	12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L	(3)	12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L	(1)	12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	51H	0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	0010	00H

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

Figure 4. Programming the Flash Memory (Parallel Mode)

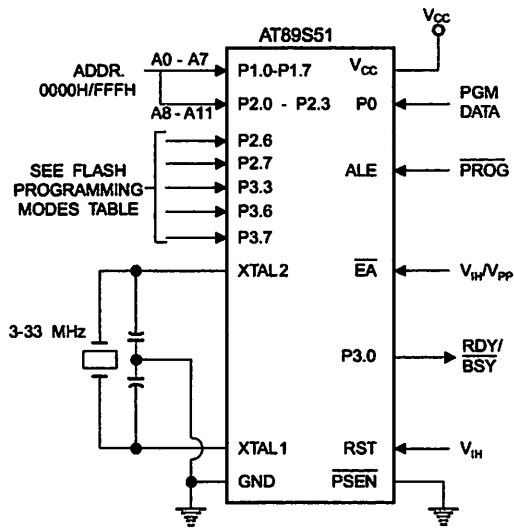
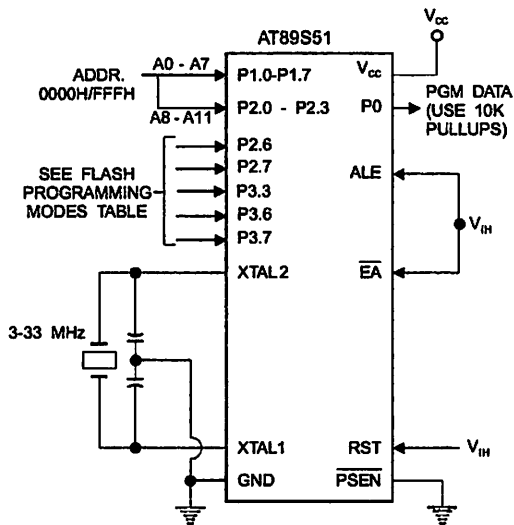


Figure 5. Verifying the Flash Memory (Parallel Mode)





Flash Programming and Verification Characteristics (Parallel Mode)

$T = 20^{\circ}\text{C}$ to 30°C , $V_{CC} = 4.5$ to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
f_{CLCL}	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{HSH}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{HSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	μs
t_{VQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{HBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		50	μs

Figure 6. Flash Programming and Verification Waveforms – Parallel Mode

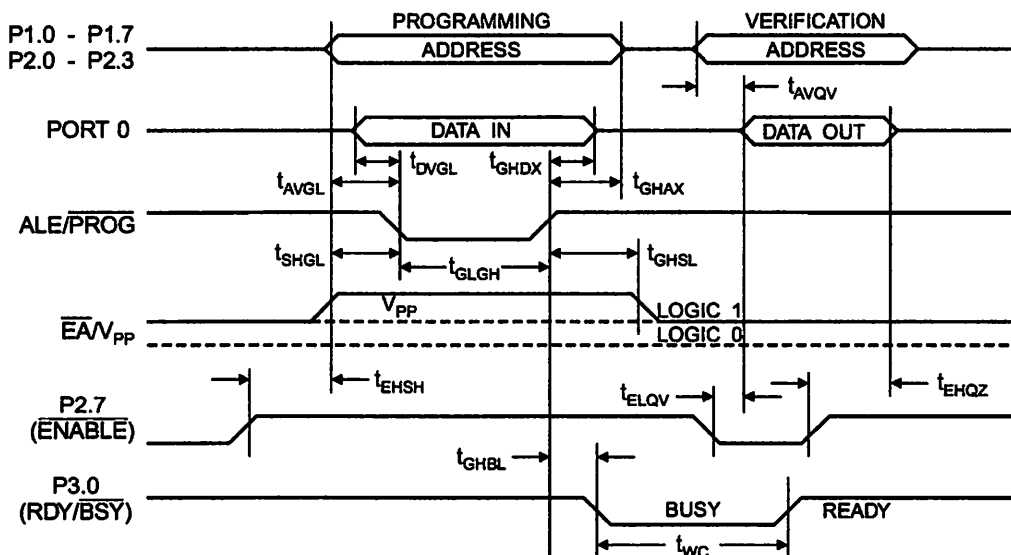
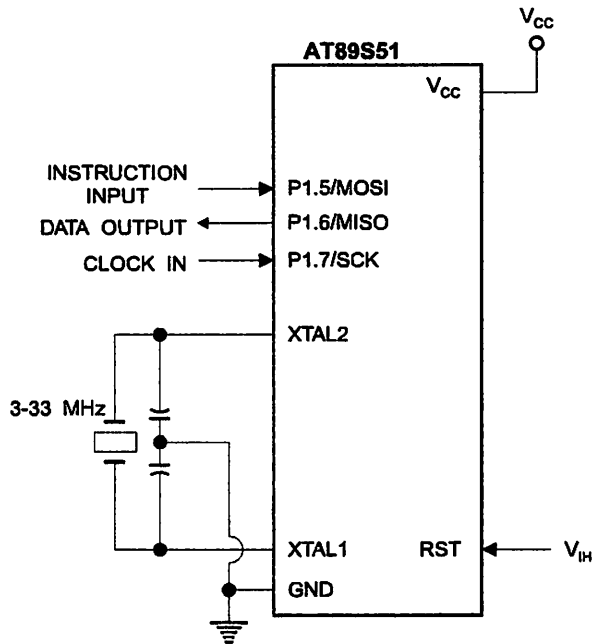


Figure 7. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 8. Serial Programming Waveforms

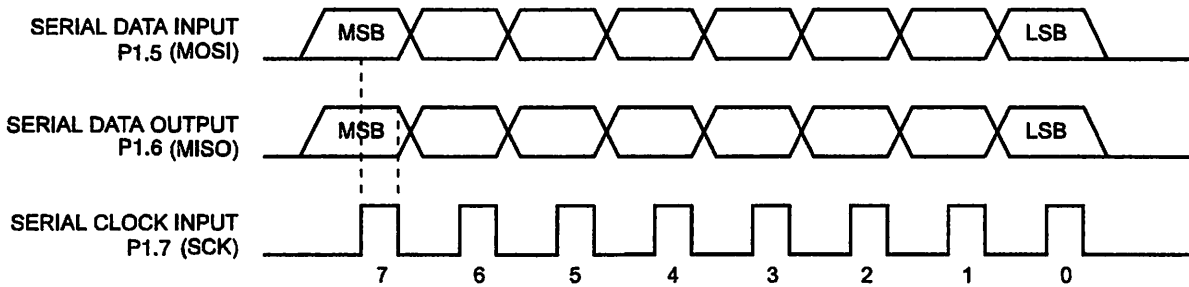




Table 8. Serial Programming Instruction Set

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)	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	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxxx A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits ⁽²⁾	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (2).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xx LP3 LP2 LP1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes ⁽¹⁾	0010 1000	xxx A5 A4 A3 A2 A1	A0 xxx xxxx	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxxx 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	xxxx A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

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

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

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

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

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

Serial Programming Characteristics

Figure 9. Serial Programming Timing

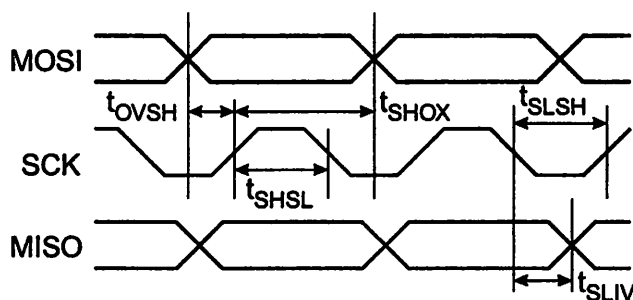


Table 9. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	$8 t_{CLCL}$			ns
t_{SLSH}	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_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	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
IO 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, \overline{PSEN})	$I_{OL} = 3.2 \text{ mA}$		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, \overline{PSEN})	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_L	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{IL}	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
R_{RST}	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽²⁾	$V_{CC} = 5.5\text{V}$		50	μA

- Notes: 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.

AT89S51

C Characteristics

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

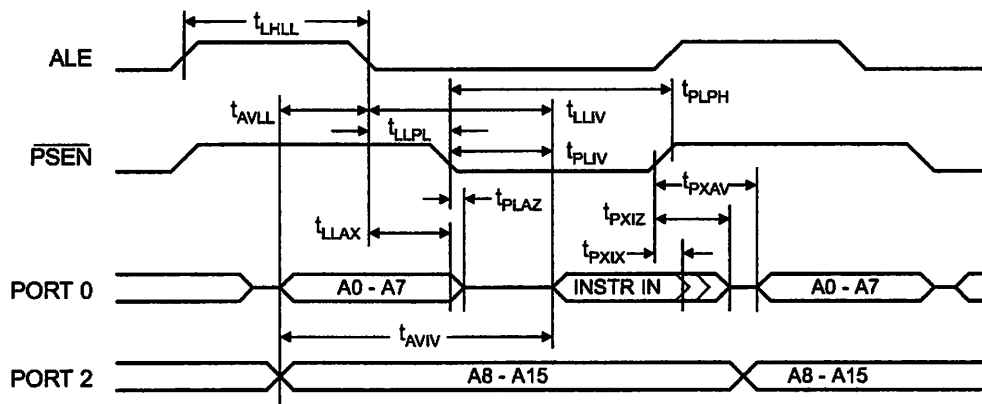
External Program and Data Memory Characteristics

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

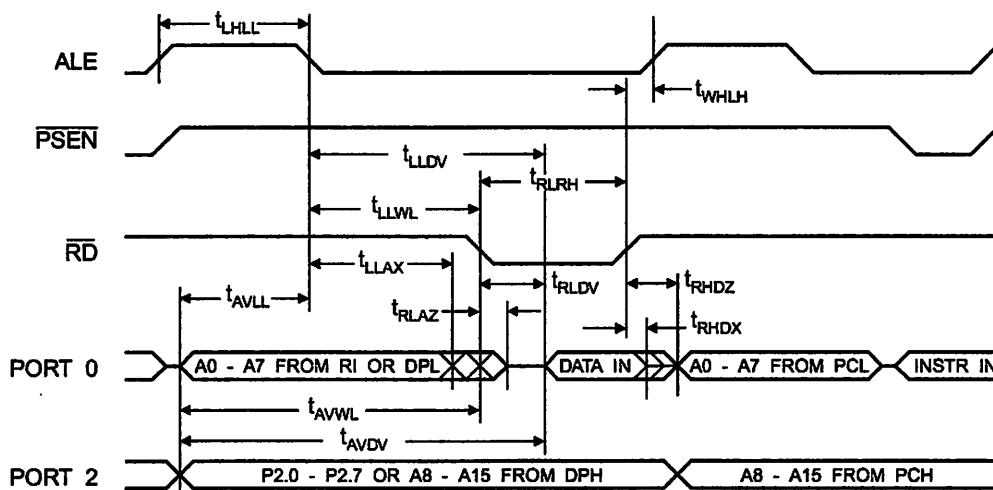




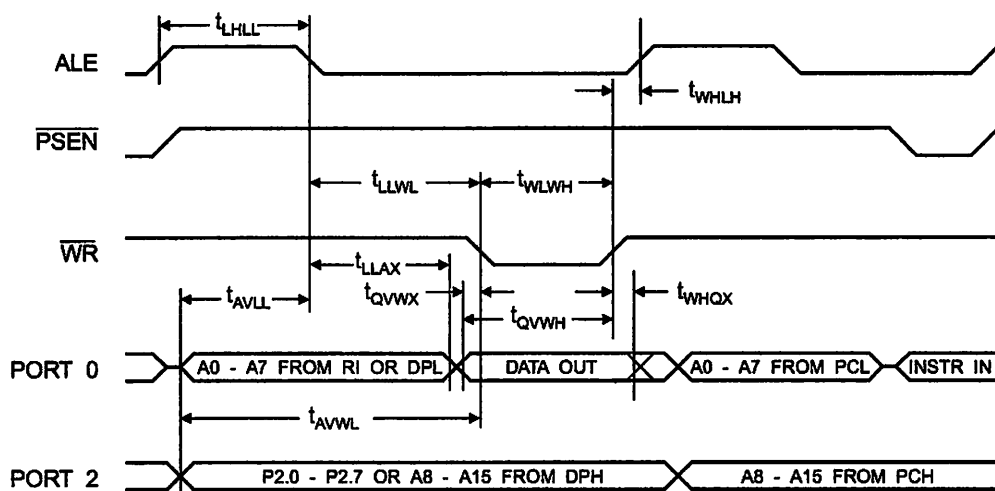
External Program Memory Read Cycle



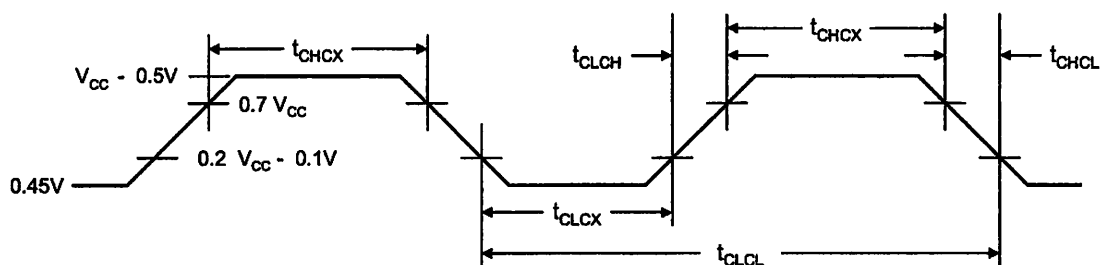
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

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

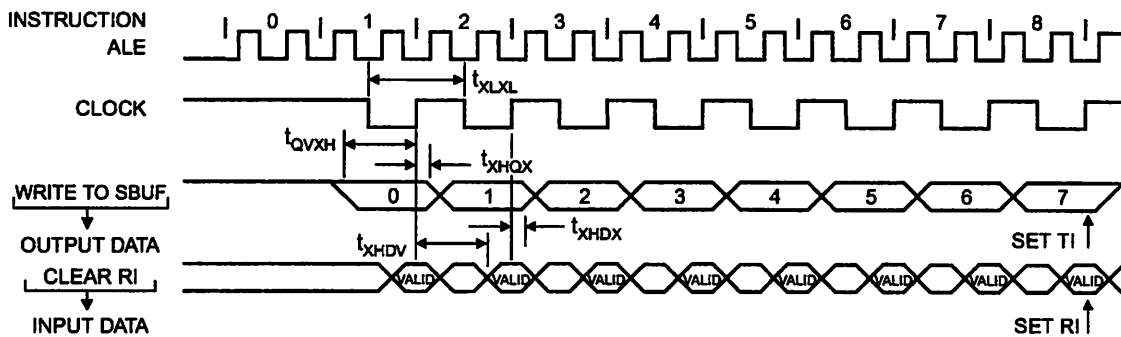


Serial Port Timing: Shift Register Mode Test Conditions

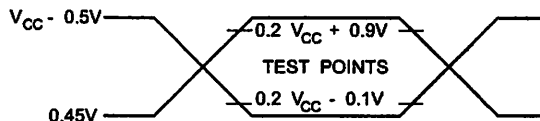
Values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = 80 pF .

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{CLCL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHGX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-80$		ns
t_{HDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms

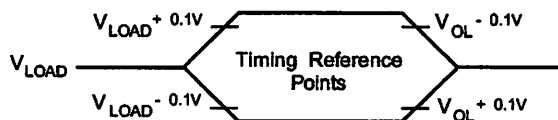


AC Testing Input/Output Waveforms⁽¹⁾



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


Load Waveforms⁽¹⁾



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

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S51-24AC	44A	Commercial (0° C to 70° C)
		AT89S51-24JC	44J	
		AT89S51-24PC	40P6	
		AT89S51-24AI	44A	Industrial (-40° C to 85° C)
		AT89S51-24JI	44J	
		AT89S51-24PI	40P6	
33	4.5V to 5.5V	AT89S51-33AC	44A	Commercial (0° C to 70° C)
		AT89S51-33JC	44J	
		AT89S51-33PC	40P6	

 = Preliminary Availability

Package Type

4A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
4J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
0P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

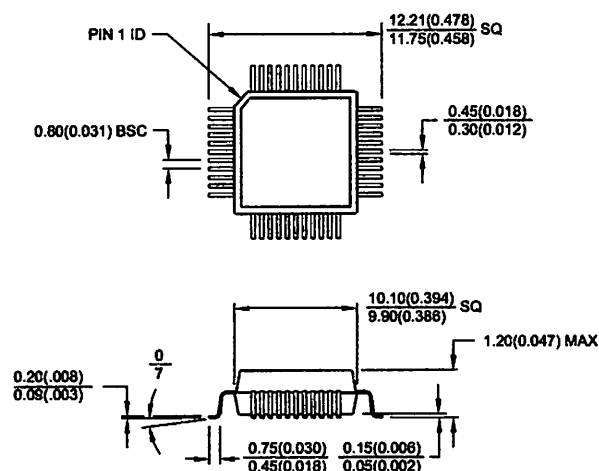




Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)

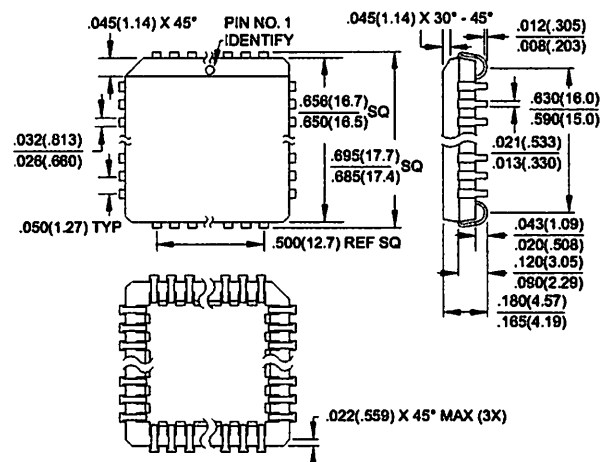
Dimensions in Millimeters and (Inches)*



*Controlling dimension: millimeters

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)

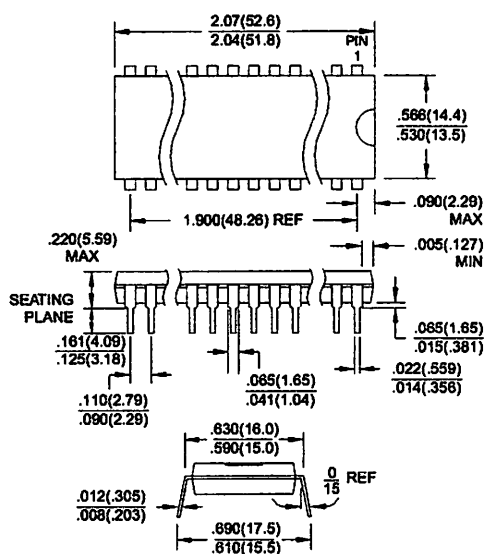
Dimensions in Inches and (Millimeters)



40P6, 40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)

Dimensions in Inches and (Millimeters)

JEDEC STANDARD MS-011 AC





Atmel Headquarters

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

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

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

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

Atmel Product Operations

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

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

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

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

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

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

e-mail
literature@atmel.com

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

Atmel Corporation 2001.

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

ATMEL® is the registered trademark of Atmel.

8051S-51® is the registered trademark of Intel Corporation. Terms and product names in this document may be trademarks of others.



Printed on recycled paper.

2487A-10/01/xM



The FT232BM is the 2nd generation of FTDI's popular USB UART i.c. This device not only adds extra functionality to its FT8U232AM predecessor and reduces external component count, but also maintains a high degree of pin compatibility with the original, making it easy to upgrade or cost reduce existing designs as well as increasing the potential for using the device in new application areas.

0 Features

HARDWARE FEATURES

Single Chip USB ↔ Asynchronous Serial Data Transfer
Full Handshaking & Modem Interface Signals
UART I/F Supports 7 / 8 Bit Data, 1 / 2 Stop Bits and Odd/Even/Mark/Space/No Parity
Data rate 300 => 3M Baud (TLL)
Data rate 300 => 1M Baud (RS232)
Data rate 300 => 3M Baud (RS422/RS485)
384 Byte Receive Buffer / 128 Byte Transmit Buffer for high data throughput
Adjustable RX buffer timeout
Full hardware assisted hardware or X-On / X-Off handshaking
In-built support for event characters and line break condition
Auto Transmit Buffer control for RS485
Support for USB Suspend / Resume through SLEEP# and RI# pins
Support for high power USB Bus powered devices through PWREN# pin
Integrated level converter on UART and control signals for interfacing to 5v and 3.3v logic
Integrated 3.3v regulator for USB IO
Integrated Power-On-Reset circuit
Integrated 6MHz – 48Mhz clock multiplier PLL
USB Bulk or Isocronous data transfer modes
4.4v to 5.25v single supply operation
UHCI / OHCI / EHCI host controller compatible
USB 1.1 and USB 2.0 compatible
USB VID, PID , Serial Number and Product Description strings in external EEPROM
EEPROM programmable on-board via USB

VIRTUAL COM PORT (VCP) DRIVERS for

- Windows 98 and Windows 98 SE
- Windows 2000 / ME / XP
- Windows CE **
- MAC OS-8 and OS-9
- MAC OS-X **
- Linux 2.40 and greater

D2XX (USB Direct Drivers + DLL SW Interface)

- Windows 98 and Windows 98 SE
- Windows 2000 / ME / XP

APPLICATION AREAS

- USB ↔ RS232 Converters
- USB ↔ RS422 / RS485 Converters
- Upgrading RS232 Legacy Peripherals to USB
- Cellular and Cordless Phone USB data transfer cables and interfaces
- Interfacing MCU based designs to USB
- USB Audio and Low Bandwidth Video data transfer
- PDA ↔ USB data transfer
- USB Smart Card Readers
- Set Top Box (S.T.B.) PC - USB interface
- USB Hardware Modems
- USB Wireless Modems
- USB Instrumentation
- USB Bar Code Readers

[** = In planning or under development]

0 Enhancements

This section summarises the enhancements of the 2nd generation device compared to its FT8U232AM predecessor. For further details, consult the device pin-out description and functional descriptions.

Integrated Power-On-Reset (POR) Circuit

The device now incorporates an internal POR function. The existing RESET# pin is maintained in order to allow external logic to reset the device where required, however for many applications this pin can now be either left N/C or hard wired to VCC. In addition, a new reset output pin (RSTO#) is provided in order to allow the new POR circuit to provide a stable reset to external MCU and other devices. RSTO# was the TEST pin on the previous generation of devices.

Integrated RCCLK Circuit

In the previous devices, an external RC circuit was required to ensure that the oscillator and clock multiplier PLL frequency was stable prior to enabling the clock internal to the device. This circuit is now embedded on-chip – the pin assigned to this function is now designated as the TEST pin and should be tied to GND for normal operation.

Integrated Level Converter on UART interface and control signals

The previous devices would drive the UART and control signals at 5v CMOS logic levels. The new device has a separate VCC-IO pin allowing the device to directly interface to 3.3v and other logic families without the need for external level converter i.c.'s

- Improved Power Management control for USB Bus Powered, high current devices**
 The previous devices had a USBEN pin, which became active when the device was enumerated by USB. To provide power control, this signal had to be externally gated with SLEEP# and RESET#. This gating is now done on-chip - USBEN has now been replaced with the new PWREN# signal which can be used to directly drive a transistor or P-Channel MOSFET in applications where power switching of external circuitry is required. A new EEPROM based option makes the device pull gently down its UART interface lines when the power is shut 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.
- Lower Suspend Current**
 Integration of RCCLK within the device and internal design improvements reduce the suspend current of the FT232BM to under 200uA (excluding the 1.5k pull-up on USB DP) in USB suspend mode. This allows greater margin for peripherals to meet the USB Suspend current limit of 500uA.
- Support for USB Isocronous Transfers**
 Whilst USB Bulk transfer is usually the best choice for data transfer, the scheduling time of the data is not guaranteed. For applications where scheduling latency takes priority over data integrity such as transferring audio and low bandwidth video data, the new device now offers an option of USB Isocronous transfer via an option bit in the EEPROM.

Programmable Receive Buffer Timeout

In the previous device, the receive buffer timeout used to flush remaining data from the receive buffer was fixed at 16ms timeout. This timeout is now programmable over USB in 1ms increments from 1ms to 255ms, thus allowing the device to be better optimised for protocols requiring faster response times from short data packets.

TXDEN Timing fix

TXDEN timing has now been fixed to remove the external delay that was previously required for RS485 applications at high baud rates. TXDEN now works correctly during a transmit send-break condition.

Relaxed VCC Decoupling

The 2nd generation devices now incorporate a level of on-chip VCC decoupling. Though this does not eliminate the need for external decoupling capacitors, it significantly improves the ease of pcb design requirements to meet FCC,CE and other EMI related specifications.

Improved PreScaler Granularity

The previous version of the Prescaler supported division by $(n + 0)$, $(n + 0.125)$, $(n + 0.25)$ and $(n + 0.5)$ where n is an integer between 2 and 16,384 (2^{14}). To this we have added $(n + 0.375)$, $(n + 0.625)$, $(n + 0.75)$ and $(n + 0.875)$ which can be used to improve the accuracy of some baud rates and generate new baud rates which were previously impossible (especially with higher baud rates).

- **Bit Bang Mode**

The 2nd generation device has a new option referred to as "Bit Bang" mode. In Bit Bang mode, the eight UART interface control lines can be switched between UART interface mode and an 8-bit Parallel IO port. Data packets can be sent to the device and they will be sequentially sent to the interface at a rate controlled by the prescaler setting. As well as allowing the device to be used stand-alone as a general purpose IO controller for example controlling lights, relays and switches, some other interesting possibilities exist. For instance, it may be possible to connect the device to an SRAM configurable FPGA as supplied by vendors such as Altera and Xilinx. The FPGA device would normally be un-configured (i.e. have no defined function) at power-up. Application software on the PC could use Bit Bang Mode to download configuration data to the FPGA which would define it's hardware function, then after the FPGA device is configured the FT232BM can switch back into UART interface mode to allow the programmed FPGA device to communicate with the PC over USB. This approach allows a customer to create a "generic" USB peripheral who's hardware function can be defined under control of the application software. The FPGA based hardware can be easily upgraded or totally changed simply by changing the FPGA configuration data file. Application notes, software and development modules for this application area will be available from FTDI and other 3rd parties.

FT232BM USB UART (USB - Serial) I.C.

PreScaler Divide By 1 Fix

The previous device had a problem when the integer part of the divisor was set to 1. In the 2nd generation device setting the prescaler value to 1 gives a baud rate of 2 million baud and setting it to zero gives a baud rate of 3 million baud. Non-integer division is not supported with divisor values of 0 and 1.

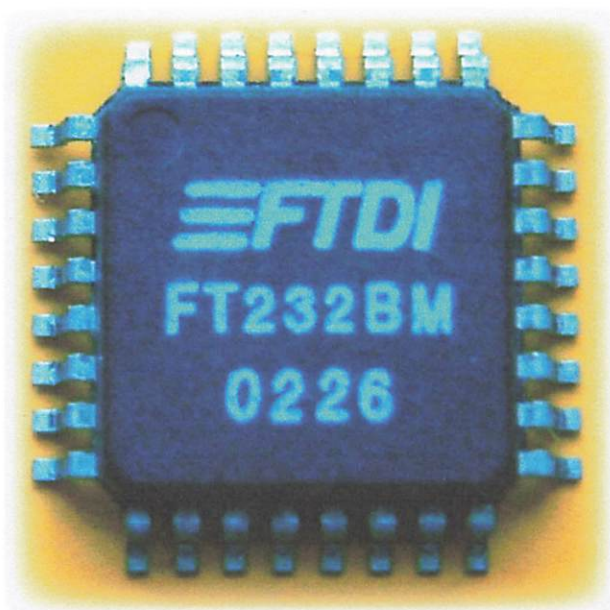
Less External Support Components

As well as eliminating the RCCLK RC network, and for most applications the need for an external reset circuit, we have also eliminated the requirement for a 100k pull-up on EECS to select 6MHz operation. When the FT232BM is being used without the configuration EEPROM, EECS, EESK and EEDATA can now be left n/c. For circuits requiring a long reset time (where the device is reset externally using a reset generator i.c., or reset is controlled by the IO port of a MCU, FPGA or ASIC device) an external transistor circuit is no longer required as the 1k5 pull-up resistor on USB DP can be wired to the RESETO# pin instead of to 3.3v. Note : RESETO# drives out at 3.3v level, not at 5v VCC level. This is the preferred configuration for new designs. In some other configurations, RSTO# can be used to reset external logic / MCU circuitry.

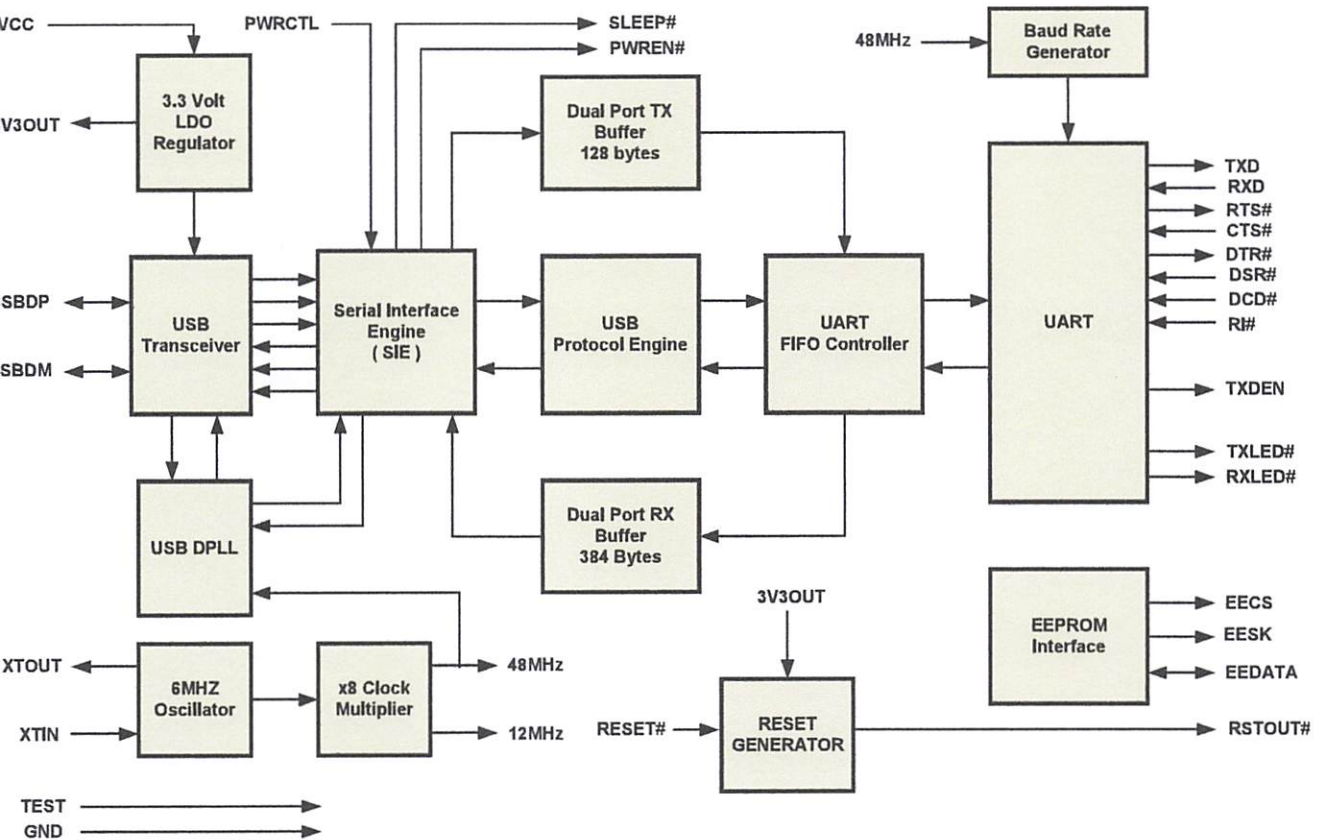
Extended EEROM Support

The previous generation of devices only supported EEPROM of type 93C46 (128 x 16 bit). The new devices will also work with EEPROM type 93C56 (256 x 16 bit) and 93C66 (512 x 16 bit). The extra space is not used by the device, however it is available for use by other external MCU / logic whilst the FT232BM is being held in reset.

- **USB 2.0 (full speed option)**
A new EEPROM based option allows the FT232BM to return a USB 2.0 device descriptor as opposed to USB 1.1. Note : The device would be a USB 2.0 Full Speed device (12Mb/s) as opposed to a USB 2.0 High Speed device (480Mb/s).
- **Multiple Device Support without EEPROM**
When no EEPROM (or a blank or invalid EEPROM) is attached to the device, the FT232BM no longer gives a serial number as part of it's USB descriptor. This allows multiple devices to be simultaneously connected to the same PC. However, we still highly recommend that EEPROM is used, as without serial numbers a device can only be identified by which hub port in the USB tree it is connected to which can change if the end user re-plugs the device into a different port.



0 Block Diagram (simplified)



1 Functional Block Descriptions

3.3V LDO Regulator

The 3.3V LDO Regulator generates the 3.3 volt reference voltage for driving the USB transceiver cell output buffers. It requires an external decoupling capacitor to be attached to the 3V3OUT regulator output pin. It also provides 3.3v power to the RSTOUT# pin. The main function of this block is to power the USB Transceiver and the Reset Generator Cells rather than to power external logic. However, external circuitry requiring 3.3v nominal at a current of not greater than 5mA could also draw it's power from the 3V3OUT pin if required.

USB Transceiver

The USB Transceiver Cell provides the USB 1.1 / USB 2.0 full-speed physical interface to the USB cable. The output drivers provide 3.3 volt level slew rate control signalling, whilst a differential receiver and two single ended receivers provide USB data in, SEO and USB Reset condition detection.

USB DPLL

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

6MHz Oscillator

The 6MHz Oscillator cell generates a 6MHz reference clock input to the X8 Clock multiplier from an external 6MHz crystal or ceramic resonator.

x8 Clock Multiplier

The x8 Clock Multiplier takes the 6MHz input from the Oscillator cell and generates a 12MHz reference clock for the SIE, USB Protocol Engine and UART FIFO controller blocks. It also generates a 48MHz reference clock for the USB DPPL 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 1.1 specification, it performs bit stuffing / un-stuffing and 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.

Dual Port TX Buffer (128 bytes)

Data from the USB data out endpoint is stored in the Dual Port TX buffer and removed from the buffer to the UART transmit register under control of the UART FIFO controller.

Dual Port RX Buffer (384 bytes)

Data from the UART receive register is stored in the Dual Port 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 Dual Port RX and TX buffers and the UART transmit and receive registers.

UART

The UART 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 the UART include RTS, CTS, DSR , DTR, DCD and RI. The UART provides a transmitter enable control signal (TXDEN) to assist with interfacing to RS485 transceivers. The UART supports RTS/CTS, DSR/DTR and X-On/X-Off handshaking options. Handshaking, where required, is handled in hardware to ensure fast response times. The UART also supports the RS232 BREAK setting and detection conditions.

Baud Rate Generator

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

RESET Generator

The Reset Generator Cell provides a reliable power-on reset to the device internal circuitry on power up. An additional RESET# input and RSTOUT# output are provided to allow other devices to reset the FT232BM or the FT232BM to reset other devices respectively. During reset, RSTOUT# is high-impedance otherwise it drives out at the 3.3v provided by the onboard regulator. RSTOUT# can be used to control the 1k5 pull-up on USB DP directly where delayed USB enumeration is required. It can also be used to reset other devices. RSTOUT# will stay high-impedance for approximately 5ms after VCC has risen above 3.5v AND the device oscillator is running AND RESET# is high. RESET# should be tied to VCC unless it is a requirement to reset the device from external logic or an external reset generator i.c.

EEPROM Interface

Though the FT232BM will work without the optional EEPROM, an external 93C46 (93C56 or 93C66) EEPROM can be used to customise the USB VID, PID, Serial Number, Product Description Strings and Power Descriptor value of the FT232BM for OEM applications. The EEPROM is also required for applications where multiple FT232BM's are connected to a single PC as the drivers rely on a unique serial number for each device to bind a unique virtual COM port to each individual device. Other parameters controlled by the EEPROM include Remote Wake Up, Isochronous Transfer Mode, Soft Pull Down on Power-Off and USB 2.0 descriptor modes. The EEPROM should be a 16 bit wide configuration such as a MicroChip 93LC46B or equivalent capable of a 1Mb/s clock rate at VCC = 4.4v to 5.25v. The EEPROM is programmable-on board over USB using a utility available from FTDI's web site (<http://www.ftdichip.com>). This allows a blank part to be soldered onto the PCB and programmed as part of the manufacturing and test process.

If no EEPROM is connected (or the EEPROM is blank), the FT232BM will use it's built-in default VID, PID Product Description and Power Descriptor Value. In this case, the device will not have a serial number as part of the USB descriptor.

0 Device Pin-Out

Figure 1

Pin-Out (LQFP-32 Package)

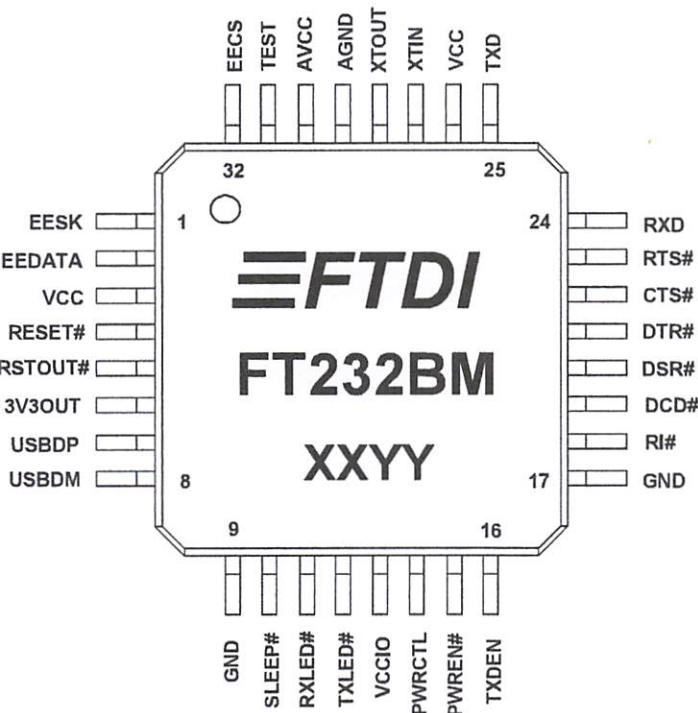
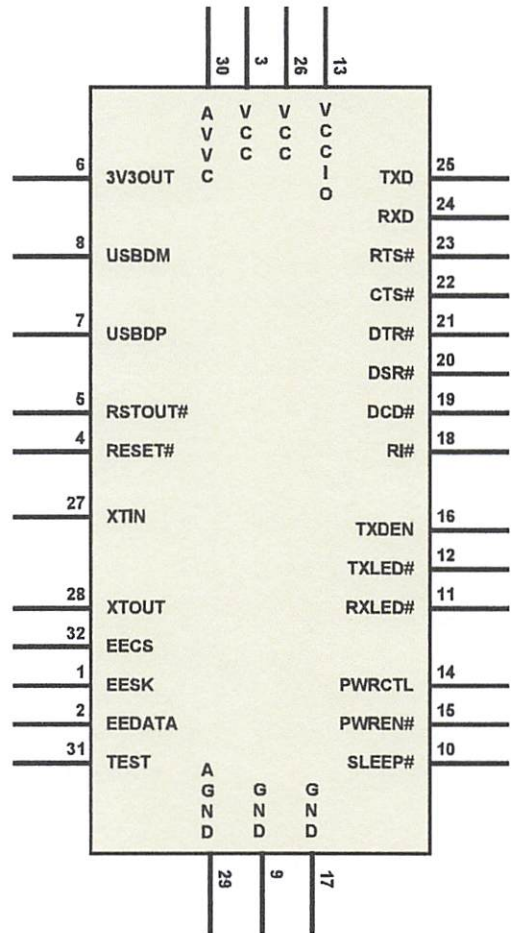


Figure 2

Pin-Out (Schematic Symbol)



1 Signal Descriptions

Table 1 - FT232BM - PINOUT DESCRIPTION

UART INTERFACE GROUP

Pin#	Signal	Type	Description
5	TXD	OUT	Transmit Asynchronous Data Output
4	RXD	IN	Receive Asynchronous Data Input
3	RTS#	OUT	Request To Send Control Output / Handshake signal
2	CTS#	IN	Clear To Send Control Input / Handshake signal
1	DTR#	OUT	Data Terminal Ready Control Output / Handshake signal
0	DSR#	IN	Data Set Ready Control Input / Handshake signal
9	DCD#	IN	Data Carrier Detect Control Input
8	RI#	IN	Ring Indicator Control Input. When the Remote Wakeup option is enabled in the EEPROM, taking RI# low can be used to resume the PC USB Host controller from suspend.
6	TXDEN	OUT	Enable Transmit Data for RS485

USB INTERFACE GROUP

Pin#	Signal	Type	Description
	USBDP	I/O	USB Data Signal Plus (Requires 1.5k pull-up to 3V3OUT or RSTOUT#)
	USBDM	I/O	USB Data Signal Minus

EEPROM INTERFACE GROUP

Pin#	Signal	Type	Description
2	EECS	I/O	EEPROM – Chip Select. For 48MHz operation pull EECS to GND using a 10k resistor. For 6MHz operation no resistor is required. Tri-State during device reset.
7	EESK	OUT	Clock signal to EEPROM. Tri-State during device reset, else drives out.
2	EEDATA	I/O	EEPROM – Data I/O Connect directly to Data-In of the EEPROM and to Data-Out of the EEPROM via a 2k2 resistor. Also pull Data-Out of the EEPROM to VCC via a 10k resistor for correct operation. Tri-State during device reset.

POWER CONTROL GROUP

Pin#	Signal	Type	Description
0	SLEEP#	OUT	Goes Low during USB Suspend Mode. Typically used to power-down an external TTL to RS232 level converter i.c. in USB -> RS232 converter designs.
5	PWREN#	OUT	Goes Low after the device is configured via USB, then high during USB suspend. Can be used to control power to external logic using a P-Channel Logic Level MOSFET switch. Enable the Interface Pull-Down Option in EEPROM when using the PWREN# pin in this way.
4	PWRCTL	IN	Bus Powered – Tie Low / Self Powered – Tie High

MISCELLANEOUS SIGNAL GROUP

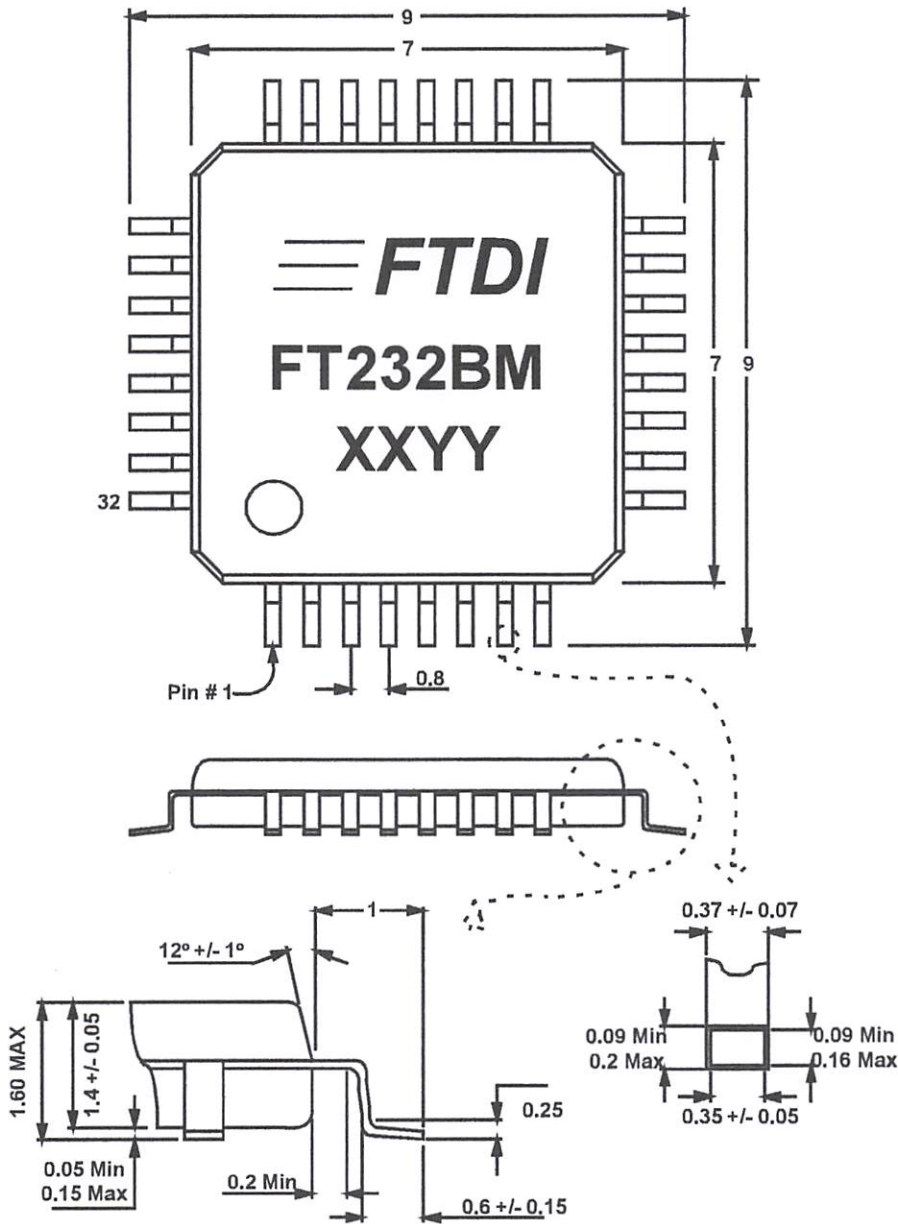
Pin#	Signal	Type	Description
4	RESET#	IN	Can be used by an external device to reset the FT232BM. If not required, tie to VCC.
5	RSTOUT#	OUT	Output of the internal Reset Generator. Stays high impedance for ~ 2ms after VCC > 3.5v and the internal clock starts up, then clamps it's output to the 3.3v output of the internal regulator. Taking RESET# low will also force RSTOUT# to go high impedance. RSTOUT# is NOT affected by a USB Bus Reset.
12	TXLED#	O.C.	LED Drive - Pulses Low when Transmitting Data via USB
11	RXLED#	O.C.	LED Drive - Pulses Low when Receiving Data via USB
27	XTIN	IN	Input to 6MHz Crystal Oscillator Cell. This pin can also be driven by an external 6MHz clock if required. Note : Switching threshold of this pin is VCC/2, so if driving from an external source, the source must be driving at 5V CMOS level or a.c. coupled to centre around VCC/2.
28	XTOUT	OUT	Output from 6MHz Crystal Oscillator Cell. XTOUT stops oscillating during USB suspend, so take care if using this signal to clock external logic.
31	TEST	IN	Puts device in i.c. test mode – must be tied to GND for normal operation.

POWER AND GND GROUP

Pin#	Signal	Type	Description
6	3V3OUT	OUT	3.3 volt Output from the integrated L.D.O. regulator This pin should be decoupled to GND using a 33nF ceramic capacitor in close proximity to the device pin. It's prime purpose is to provide the internal 3.3v supply to the USB transceiver cell and the RSTOUT# pin. A small amount of current (<= 5mA) can be drawn from this pin to power external 3.3v logic if required.
3,26	VCC	PWR	+4.4 volt to +5.25 volt VCC to the device core, LDO and and none-UART interface pins.
13	VCCIO	PWR	+3.0 volt to +5.25 volt VCC to the UART interface pins 10..12, 14..16 and 18..25. When interfacing with 3.3v external logic connect VCCIO to the 3.3v supply of the external logic, otherwise connect to VCC to drive out at 5v CMOS level.
9,17	GND	PWR	Device– Ground Supply Pins
30	AVCC	PWR	Device - Analog Power Supply for the internal x8 clock multiplier
29	AGND	PWR	Device - Analog Ground Supply for the internal x8 clock multiplier

0 Package Outline

Figure 3 – 32 LD LQFP Package Dimensions



The FT232BM is supplied in a 32 LD LQFP package as standard. This package has a 7mm x 7mm body (9mm x 9mm including leads) with leads on a 0.8mm pitch. An alternative 5mm x 5mm leadless chip scale package is available on special request for projects where package area is critical.

The above drawing shows the LQFP-32 package – all dimensions are in millimetres.

XXYY = Date Code (XX = 2 digit year number, YY = 2 digit week number.

0 Absolute Maximum Ratings

These are the absolute maximum ratings for the FT232BM device in accordance with the Absolute Maximum Rating System (JEDEC 60134). Exceeding these may cause permanent damage to the device.

Storage Temperature	-65°C to + 150°C
Ambient Temperature (Power Applied).....	0°C to + 70°C
VCC Supply Voltage	-0.5v to +6.00v
DC Input Voltage - Inputs	-0.5v to VCC + 0.5v
DC Input Voltage - High Impedance Bidirectionals	-0.5v to VCC + 0.5v
DC Output Current – Outputs	24mA
DC Output Current – Low Impedance Bidirectionals	24mA
Power Dissipation (VCC = 5.25v)	500mW
Electrostatic Discharge Voltage (I < 1uA)	+/- 2000v
Latch Up Current (Vi < 0 or Vi > Vcc)	100mA

1 D.C. Characteristics

D.C. Characteristics (Ambient Temperature = 0 .. 70°C)

Operating Voltage and Current

Parameter	Description	Min	Typ	Max	Units	Conditions
Vcc1	VCC Operating Supply Voltage	4.4	5.0	5.25	V	
Vcc2	VCCIO Operating Supply Voltage	3.0	-	5.25	V	
Icc1	Operating Supply Current	-	25	-	mA	Normal Operation
Icc2	Operating Supply Current	-	180	200	uA	USB Suspend ** Note 1

Note 1 – Supply current excludes the 200uA nominal drawn by the external pull-up resistor on USB DP.

UART IO Pin Characteristics (VCCIO = 5.0v)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	4.4	-	4.9	V	I source = 2mA
Vol	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
Vin	Input Switching Threshold	1.1	1.5	1.9	V	Note 2
VHys	Input Switching Hysteresis		200		mV	

UART IO Pin Characteristics (VCCIO = 3.3v)

Parameter	Description	Min	Typ	Max	Units	Conditions
Voh	Output Voltage High	2.7	-	3.2	V	I source = 2mA
Vol	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
Vin	Input Switching Threshold	1.0	1.4	1.8	V	Note 2
VHys	Input Switching Hysteresis		200		mV	

Note 2 – Inputs have an internal 200k pull-up resistor to VCCIO.

TIN / XTOU Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
<i>V_{oh}</i>	Output Voltage High	4.0	-	5.0	V	Fosc = 6MHz
<i>V_{ol}</i>	Output Voltage Low	0.1	-	1.0	V	Fosc = 6MHz
<i>V_{in}</i>	Input Switching Threshold	1.8	2.5	3.2	V	

RESET#, TEST, EECS, EESK, EEDATA, IO Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
<i>V_{oh}</i>	Output Voltage High	4.4	-	4.9	V	I source = 2mA
<i>V_{ol}</i>	Output Voltage Low	0.1	-	0.7	V	I sink = 4 mA
<i>V_{in}</i>	Input Switching Threshold	1.1	1.5	1.9	V	Note 3
<i>V_{Hys}</i>	Input Switching Hysteresis		200		mV	

Note 3 – EECS and EEDATA pins have an internal 200k pull-up resistor to VCC

STOUT Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
<i>V_{oh}</i>	Output Voltage High	3.0	-	3.6	V	I source = 2mA
<i>I_{ol}</i>	Leakage Current Tri-State	-	-	5	uA	

USB IO Pin Characteristics

Parameter	Description	Min	Typ	Max	Units	Conditions
<i>V_{oh}</i>	IO Pins Static Output (High)	2.8		3.6v	V	RI = 1k5 to 3V3Out (D+) RI = 15k to GND (D-)
<i>V_{ol}</i>	IO Pins Static Output (Low)	0		0.3	V	RI = 1k5 to 3V3Out (D+) RI = 15k to GND (D-)
<i>V_{se}</i>	Single Ended Rx Threshold	0.8		2.0	V	
<i>V_{com}</i>	Differential Common Mode	0.8		2.5	V	
<i>V_{dif}</i>	Differential Input Sensitivity	0.2			V	
<i>Z_{drv}</i>	Driver Output Impedance	29		44	ohm	Note 4

Note 4 – Driver Output Impedance includes the external 27R series resistors on USBDP and USBDM pins.

0 Device Configuration Examples

1 Oscillator Configurations

Figure 4
3-Pin Ceramic Resonator Configuration

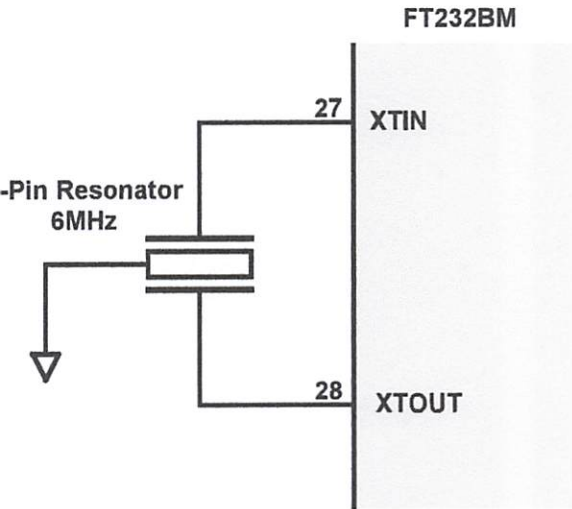


Figure 5
Crystal or 2-Pin Ceramic Resonator Configuration

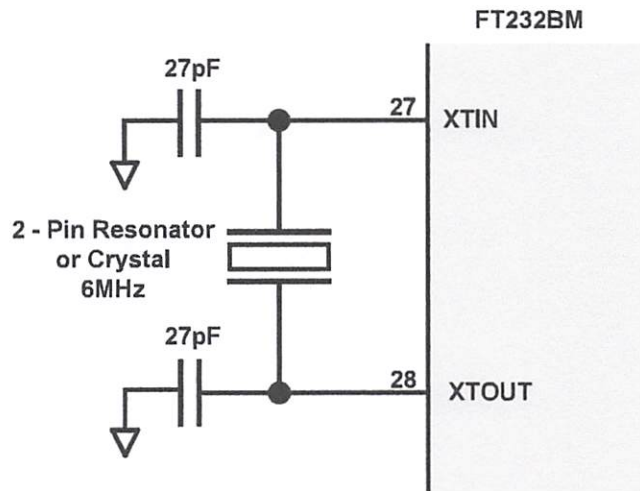


Figure 4 illustrates how to use the FT232BM with a 3-Pin Ceramic Resonator such as Murata Part # STLS6M00G53 or equivalent. 3-Pin resonators have the load capacitors built into the resonator so no external loading capacitors are required. This makes for an economical configuration. Though the typical accuracy of such a resonator is +/- 0.5% and is technically out-with the USB specification, it has been calculated that using such a device will work satisfactorily in practice with the FT232BM design.

Figure 5 illustrates how to use the FT232BM with a 6MHz Crystal or 2-Pin Ceramic Resonator. In this case, these devices do not have in-built loading capacitors so these have to be added between XTIN, XTOUT and GND as shown. A value of 27pF is shown as the capacitor in the example – this will be good for many crystals and some resonators but do select the value based on the manufacturers recommendations wherever possible. If using a crystal, use a parallel cut type. If using a resonator, see the previous note on frequency accuracy.

2 EEPROM Configuration

Figure 6
EEProm Configuration

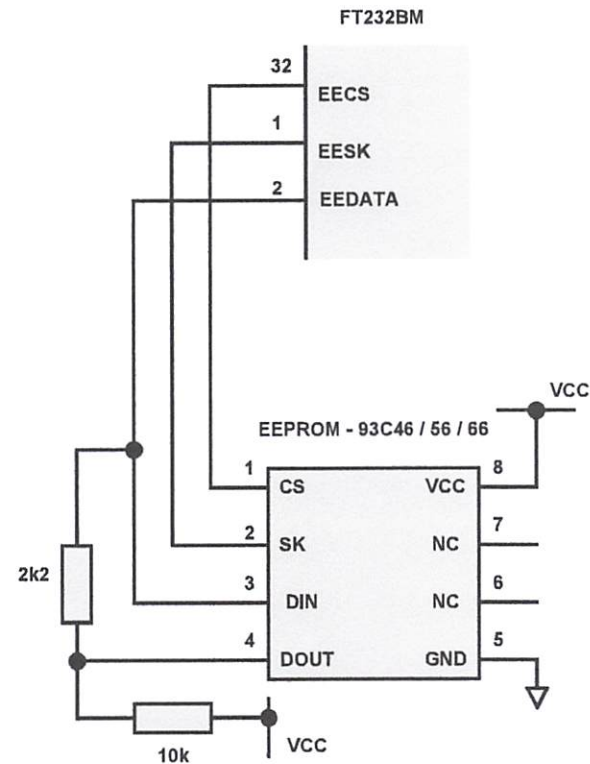


Figure 6 illustrates how to connect the FT232BM to the 93C46 (93C56 or 93C66) EEPROM. EECS (pin 32) is directly connected to the chip select (CS) pin of the EEPROM. EESK (pin 1) is directly connected to the clock (SK) pin of the EEPROM. EEDATA (pin 2) is directly connected to the Data In (Din) pin of the EEPROM. There is a potential condition whereby both the Data Output (Dout) of the EEPROM can drive out at the same time as the EEDATA pin of the FT232BM. To prevent potential data clash in this situation, the Dout of the EEPROM is connected to EEDATA of the FT232BM via a 2k2 resistor.

Following a power-on reset or a USB reset, the FT232BM will scan the EEPROM to find out a) if an EEPROM is attached to the Device and b) if the data in the device is valid. If both of these are the case, then the FT232BM will use the data in the EEPROM, otherwise it will use it's built-in default values. When a valid command is issued to the EEPROM from the FT232BM, the EEPROM will acknowledge the command by pulling it's Dout pin low. In order to check for this condition, it

necessary to pull Dout high using a 10k resistor. If the command acknowledge doesn't happen then EEDATA will be pulled high by the 10k resistor during this part of the cycle and the device will detect an invalid command or no EEPROM present.

There are two varieties of these EEPROMs on the market – one is configured as being 16 bits wide, the other configured as being 8 bits wide. These are available from many sources such as Microchip, ST, SIS etc. The FT232BM requires EEPROMs with a 16-bit wide configuration such as the Microchip 93LC46B device. The EEPROM must be capable of reading data at a 1Mb clock rate at a supply voltage of 4.4v to 5.25v. Most available parts are capable of this.

Check the manufacturers data sheet to find out how to connect pins 6 and 7 of the EEPROM. Some devices specify these as no-connect, others use them for selecting 8 / 16 bit mode or for test functions. Some other parts have their pinout rotated by 90° so please select the required part and it's options carefully.

It is possible to "share" the EEPROM between the FT232BM and another external device such as an MCU. However, this can only be done when the FT232BM is in it's reset condition as it tri-states it's EEPROM interface at that time. A typical configuration would use four bit's of an MCU IO Port. One bit would be used to hold the FT232BM reset (using RESET#) on power-up, the other three would connect to the EECS, EESK and EEDATA pins of the FT232BM in order to read / write data to the EEPROM at this time. Once the MCU has read / written the EEPROM, it would take RESET# high to allow the FT232BM to configure itself and enumerate over USB.

3 USB Bus Powered and Self Powered Configuration

Figure 7
USB Bus Powered Configuration

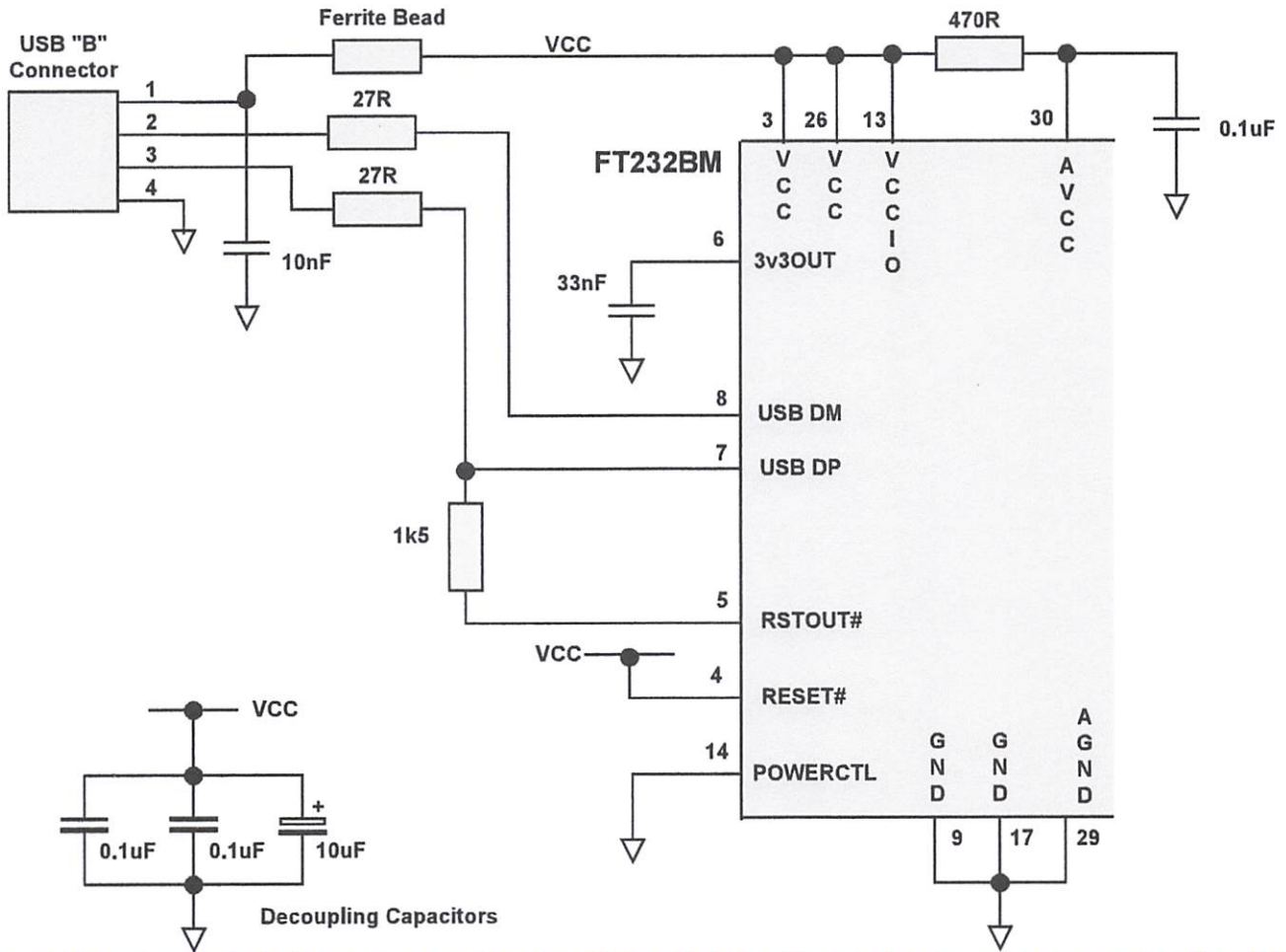


Figure 7 illustrates a typical USB bus powered configuration. A USB Bus Powered device gets its power from the USB bus. Basic rules for USB Bus power devices are as follows –

- a) On plug-in, the device must draw no more than 100mA
- b) On USB Suspend the device must draw no more than 500uA.
- c) A Bus Powered High Power Device (one that draws more than 100mA) should use the PWREN# pin to keep the current below 100mA on plug-in and 500uA on USB suspend.
- d) A device that consumes more than 100mA can not be plugged into a USB Bus Powered Hub
- e) No device can draw more that 500mA from the USB Bus.

POWERCTL (pin 14) is pulled low to tell the device to use a USB Bus Power descriptor. The power descriptor in the EPROM should be programmed to match the current draw of the device.

Ferrite Bead is connected in series with USB power to prevent noise from the device and associated circuitry (EMI being radiated down the USB cable to the Host). The value of the Ferrite Bead depends on the total current required by the circuit – a suitable range of Ferrite Beads is available from Steward (www.steward.com) for example Steward part # **MI0805K400R-00** also available as **DigiKey Part # 240-1035-1**.

Figure 8
USB Self Powered Configuration (1)

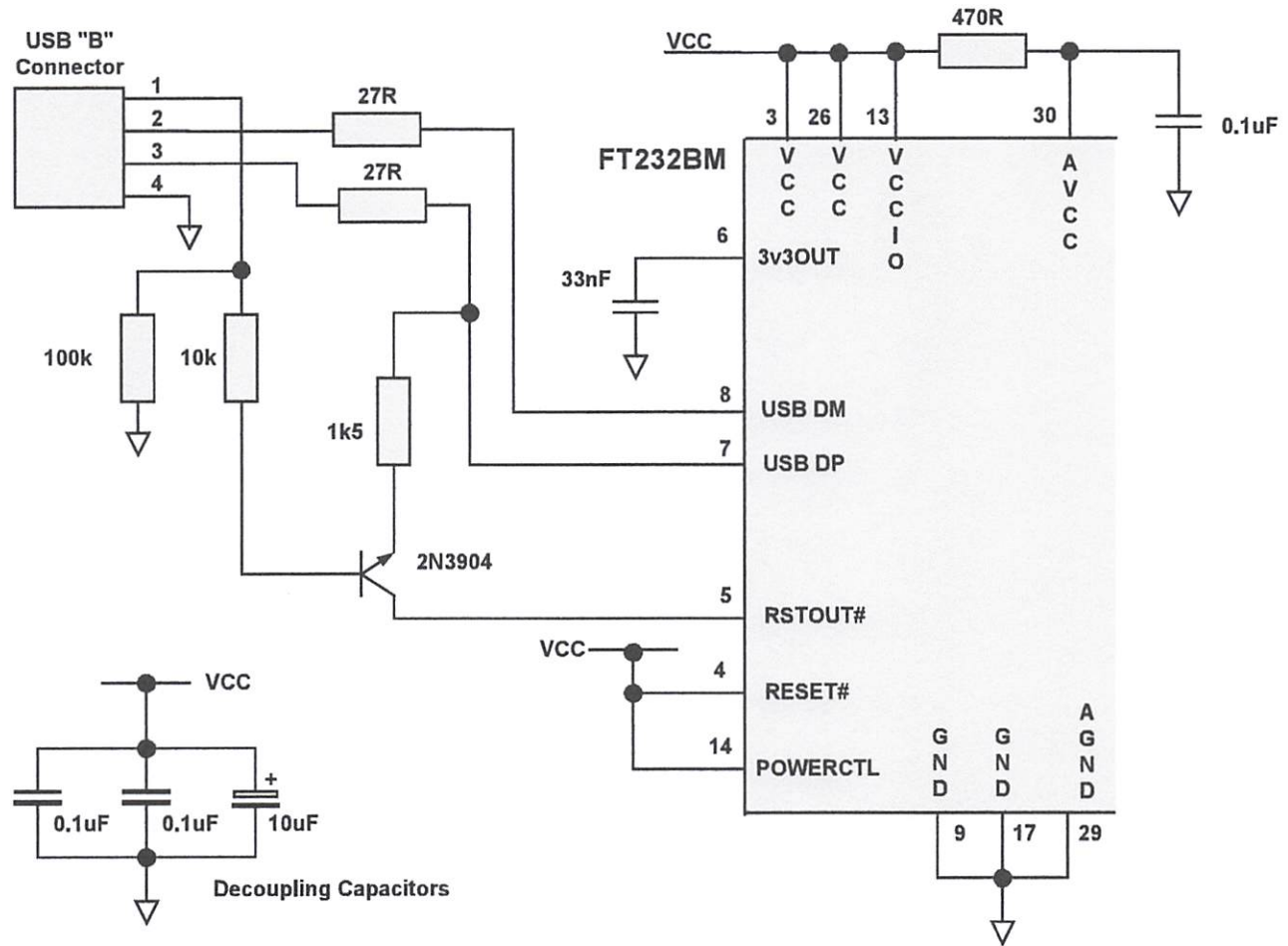


Figure 8 illustrates a typical USB self powered configuration. A USB Self Powered device gets its power from its own POWER SUPPLY and does not draw current from the USB bus. Basic rules for USB Self power devices are as follows

- A Self-Powered device should not force current down the USB bus when the USB Host or Hub Controller is powered down.
- A Self Powered Device can take as much current as it likes during normal operation and USB suspend as it has its own POWER SUPPLY.
- A Self Powered Device can be used with any USB Host and both Bus and Self Powered USB Hubs
- POWERCTL (pin 14) is pulled high to tell the device to use a USB Bus Power descriptor. The power descriptor in the EPROM should be programmed to a value of zero.
- To meet requirement a), the 1k5 pull-up circuit on USB DP has to be modified to prevent the device forcing current into the USB DP line via the 1k5 pull-up when the host or hub is powered down. Failure to do this may cause some USB host or hub controllers to power up erratically. A NPN small signal transistor (2N3906) is used to sense the power on the USB bus. It is connected as an emitter-follower circuit so that when there is power on the USB bus the transistor will saturate and pull the 1k5 resistor to the voltage of RSTOUT#. When the USB power is off, the transistor will turn off thus preventing current flow into the USB DP line.

Figure 9
USB Self Powered Configuration (2)

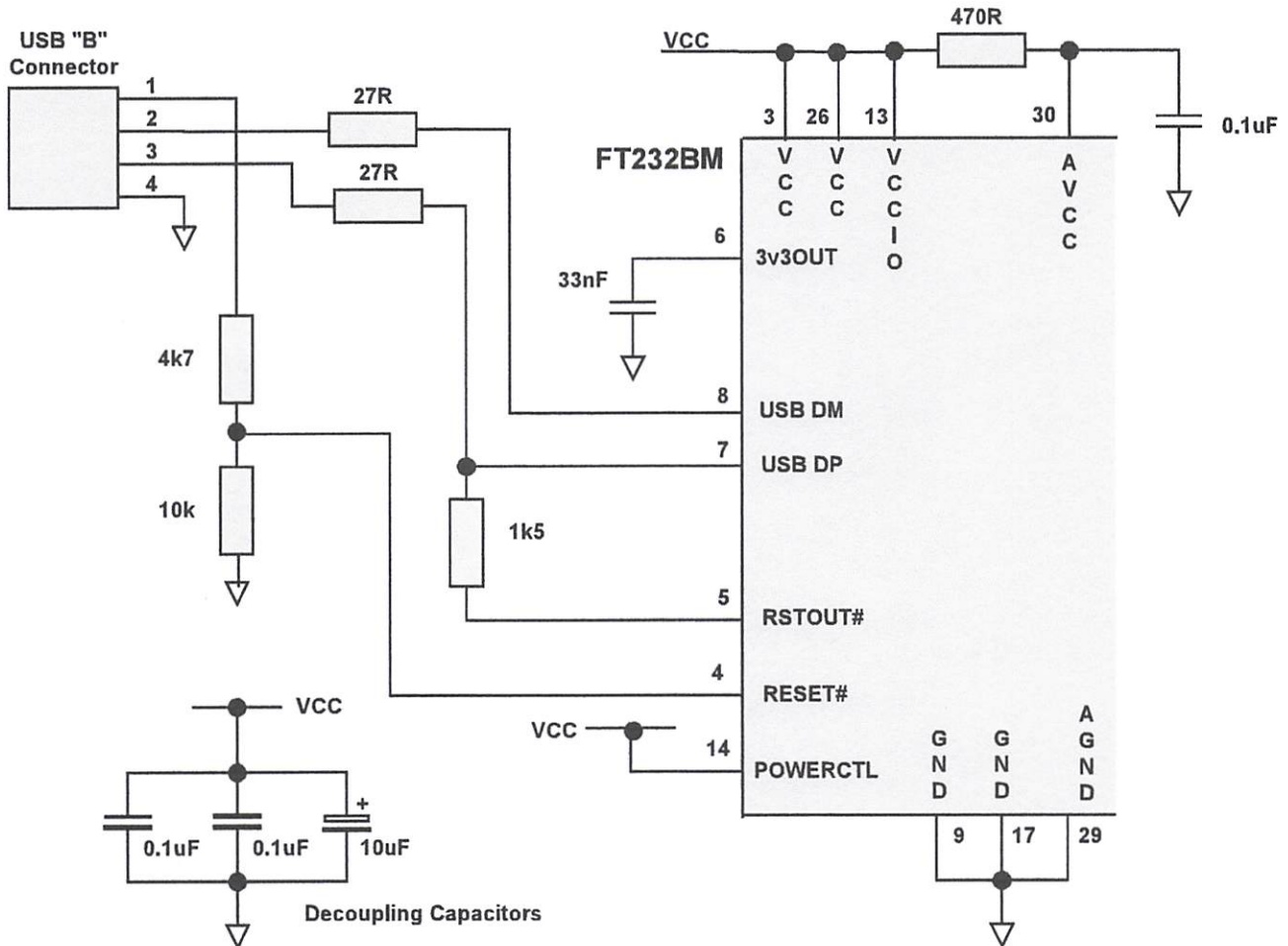


Figure 9 illustrates a variant of the circuit shown in Figure 8. This time, the 1k5 pull-up resistor on USB DP is connected to RSTOUT# as per the bus-power circuit. However, the USB Bus Power is used to control the RESET# pin of the FT232BM device. When the USB Host or Hub power is off, RESET# will go low and the device will be held in reset. As RESET# is low, RSTOUT# will also be low, so no current will be forced down USB DP via the 1k5 pull-up resistor.

Note : When the FT232B is in reset, the UART interface pins all go tri-state. These pins have internal 200k pull-up resistors to VCC-IO so they will gently pull high unless driven by some external logic.

Which of the two configurations to use depends on the nature of the peripheral design. With the configuration of Figure 8, the FT232BM is "live" – when power to the USB port is shut off, there will be no activity on the USB bus and the device will enter low power sleep mode after a few milliseconds. In this configuration, the RESET# pin is still available if required.

In the Figure 9 configuration, the FT232BM is held in reset when the USB power is off. In reset, the FT232BM 6MHz oscillator will still be running and the device will not be in low power mode.

4 UART Interface Configuration

Figure 10
USB => RS232 Converter Configuration

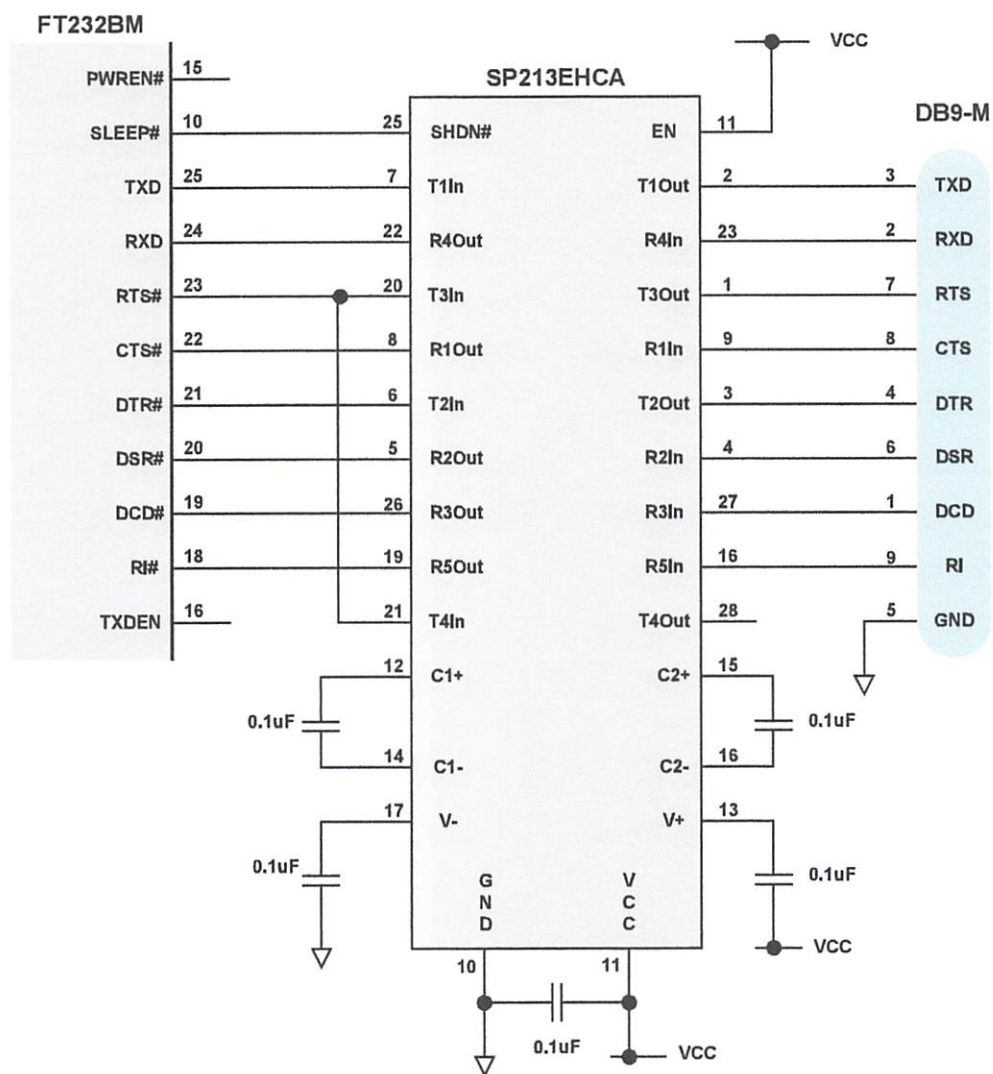


Figure 10 illustrates how to connect the UART interface of the FT232BM to a TTL – RS232 Level Converter I.C. to make a USB -> RS232 converter using the popular “213” series of TTL to RS232 level converters. These devices have 4 transmitters and 5 receivers in a 28 LD SSOP package and feature an in-built voltage converter to convert the 5v (nominal) VCC to the +/- 9volts required by RS232. An important feature of these devices is the SHDN# pin which can power down the device to a low quiescent current during USB suspend mode.

The device used in the example is a Sipex SP213EHCA which is capable of RS232 communication at up to 500k baud. If a lower baud rate is acceptable, then several pin compatible alternatives are available such as Sipex SP213ECA , Maxim MAX213CAI and Analog Devices ADM213E which are good for communication at up to 115,200 baud. If a higher baud rate is desired, use a Maxim MAX3245CAI part which is capable of RS232 communication at rates of up to 1M baud. The MAX3245 is not pin compatible with the 213 series devices, also it’s SHDN pin is active high so connect this to PWREN# instead of SLEEP#.

Figure 11
USB => RS422 Converter Configuration

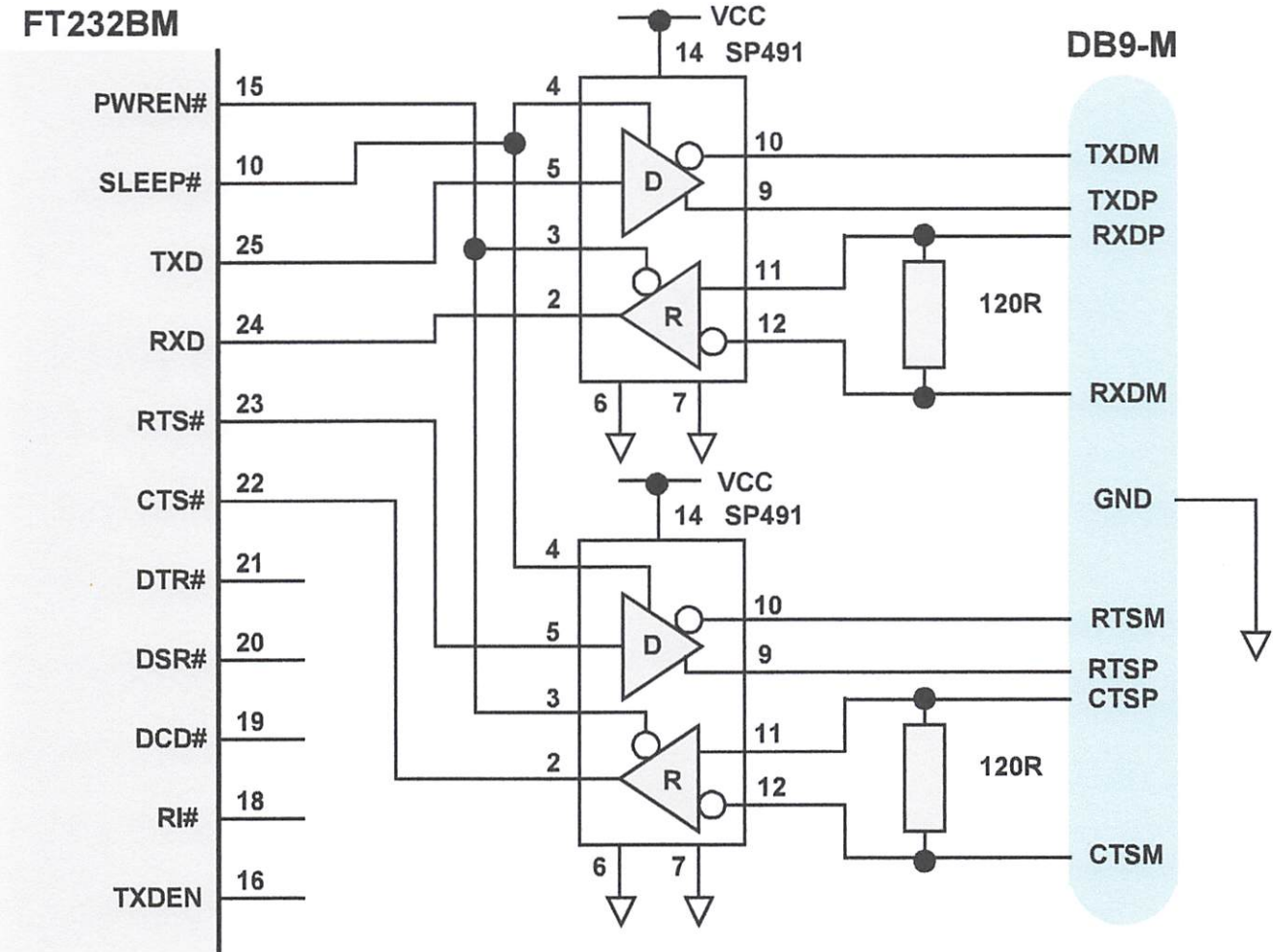


Figure 11 illustrates how to connect the UART interface of the FT232BM to a TTL – RS422 Level Converter I.C. to make a USB -> RS422 converter. There are many such level converter devices available – this example uses Sipex SP491 devices which have enables on both the transmitter and receiver. Because the transmitter enable is active high, it is connected to the SLEEP# pin. The receiver enable is active low and is connected to the PWREN# pin. This ensures that both the transmitters and receivers are enabled when the device is active, and disabled when the device is in USB suspend mode. If the design is USB BUS powered, it may be necessary to use a P-Channel logic level MOSFET (controlled by PWREN#) in the VCC line of the SP491 devices to ensure that the USB standby current of 100uA is met.

The SP491 is good for sending and receiving data at a rate of up to 5M Baud – in this case the maximum rate is limited to 3M Baud by the FT232BM.

Figure 12
USB => RS485 Converter Configuration

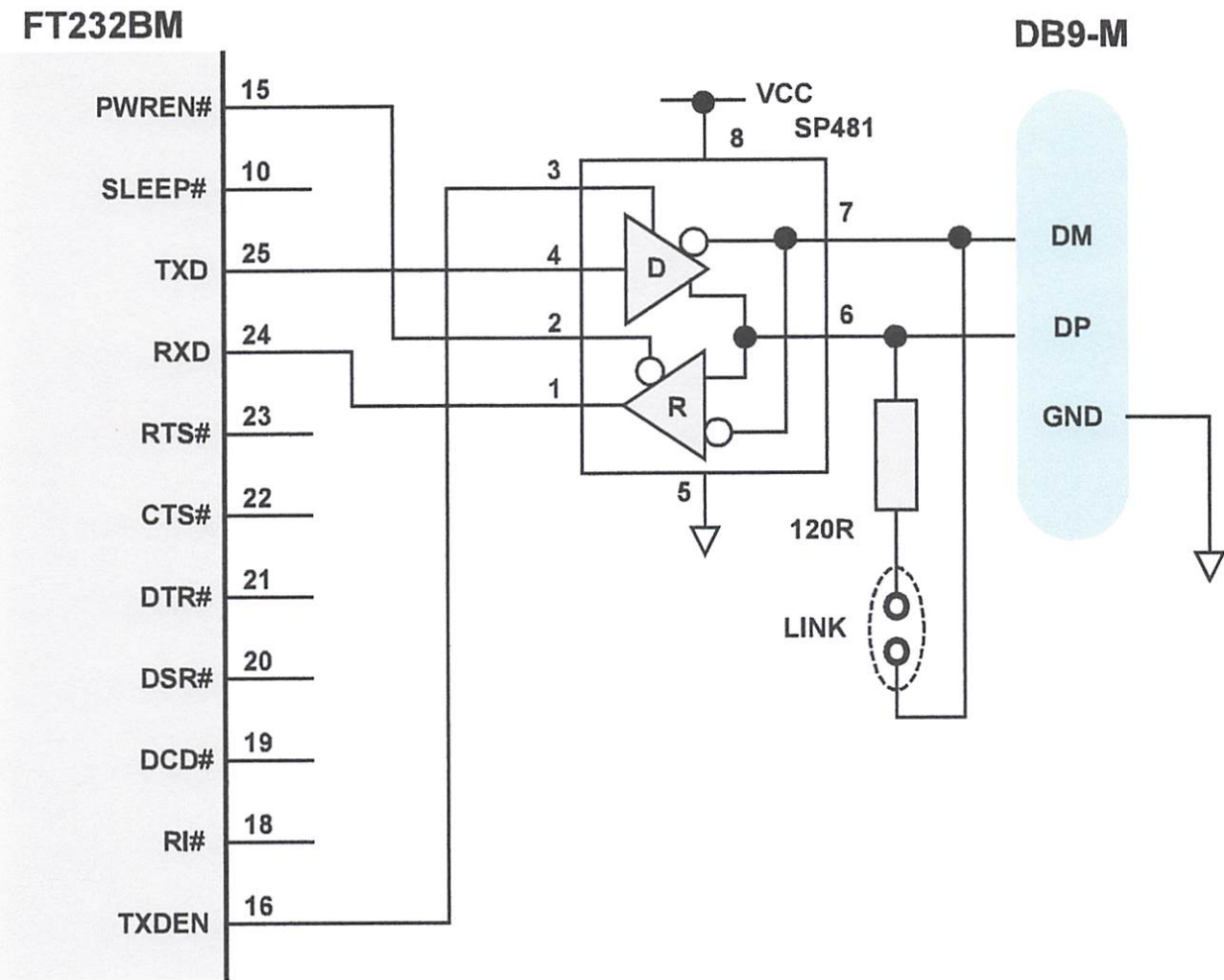


Figure 12 illustrates how to connect the UART interface of the FT232BM to a TTL – RS485 Level Converter I.C. to make a USB => RS485 converter. This example uses the Sipex SP491 device but there are similar parts available from Maxim and Analog Devices amongst others. The SP491 is a RS485 device in a compact 8 pin SOP package. It has separate enables on both the transmitter and receiver. With RS485, the transmitter is only enabled when a character is being transmitted from the UART. The TXDEN pin on the FT232BM is provided for exactly that purpose and so the transmitter enable is wired to TXDEN. The receiver enable is active low, so it is wired to the PWREN# pin to disable the receiver when in USB suspend mode.

RS485 is a multi-drop network – i.e. many devices can communicate with each other over a single two wire cable connection. The RS485 cable requires to be terminated at each end of the cable. A link is provided to allow the cable to be terminated if the device is physically positioned at either end of the cable.

In this example the data transmitted by the FT232BM is also received by the device that is transmitting. This is a common feature of RS485 and requires the application software to remove the transmitted data from the received data stream. With the FT232BM it is possible to do this entirely in hardware – simply modify the schematic so that RXD of the FT232BM is the logical OR of the SP481 receiver output with TXDEN using an HC32 or similar logic gate.

5 LED Interface

Figure 13
Dual LED Configuration

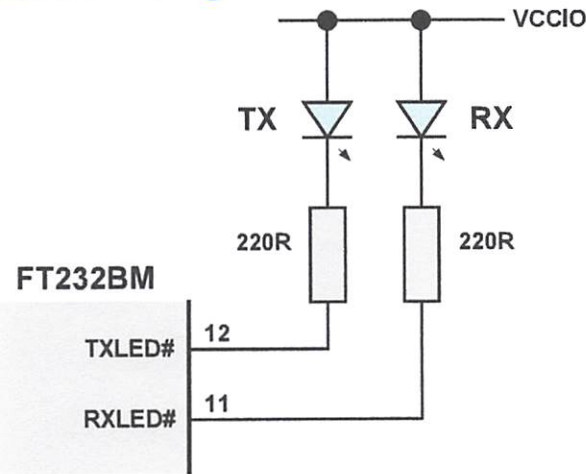
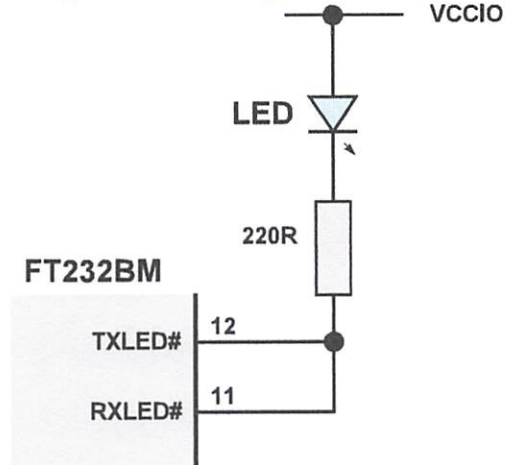


Figure 14
Single LED Configuration



The FT232BM has two IO pins dedicated to controlling LED status indicators, one for transmitted data the other for received data. When data is being transmitted / received the respective pins drive from tri-state to low in order to provide indication on the LEDs of data transfer. A digital one-shot timer is used so that even a small percentage of data transfer is visible to the end user. Figure 13 shows a configuration using two individual LED's – one for transmitted data the other for received data. In Figure 14, the transmit and receive LED indicators are wire-or'd together to give a single LED indicator which indicates any transmit or receive data activity. Another possibility (not shown here) is to use a 3 pin common anode tri-color LED based on the circuit in Figure 13 to have a single LED that can display activity in a variety of colors depending on the ratio of transmit activity compared to receive activity.

6 Interfacing to 3.3v Logic

Figure 15
USB Powered Circuit with 3.3v logic drive / supply voltage

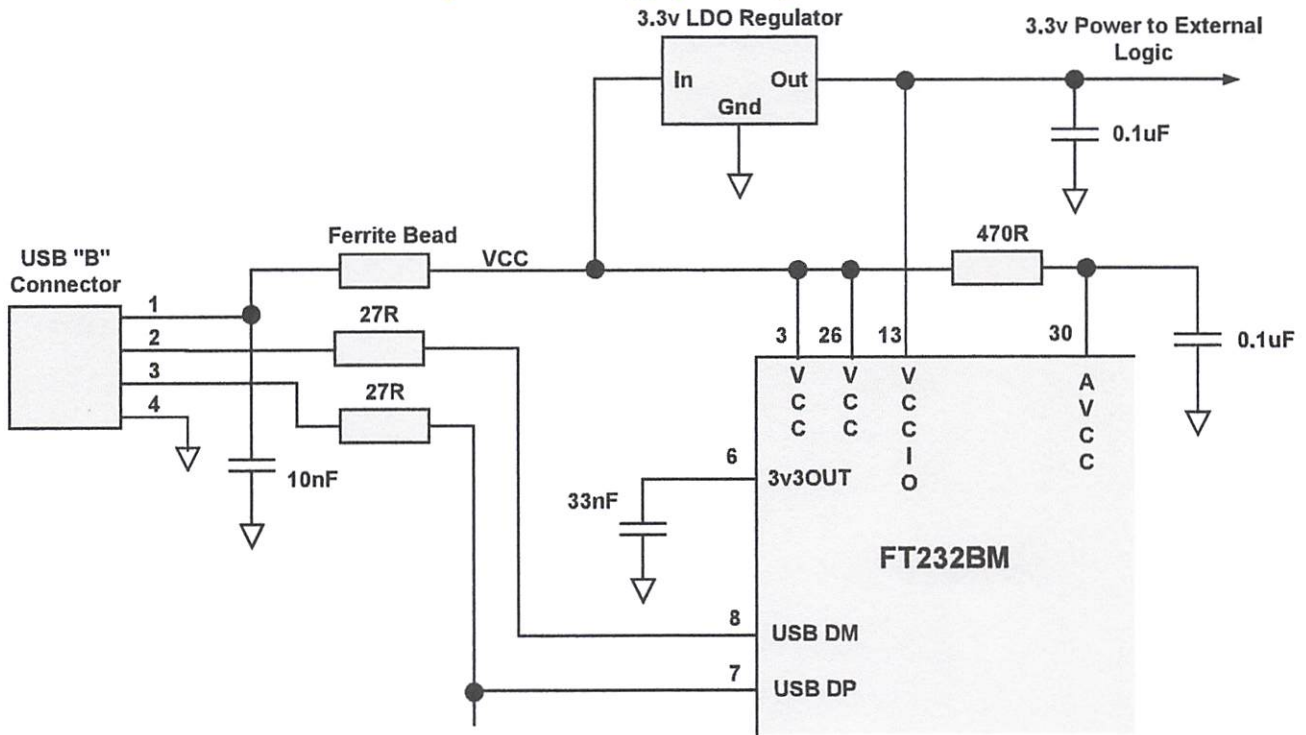


Figure 15 shows how to configure the FT232BM to interface with a 3.3v logic device. In this example, a discrete 3.3v regulator is used to supply the 3.3v logic from the USB supply. VCCIO is connected to the output of the 3.3v regulator, which in turn will cause the UART interface IO pins to drive out at 3.3v level. For USB bus powered circuits some considerations have to be taken into account when selecting the regulator –

- a) The regulator must be capable of sustaining its output voltage with an input voltage of 4.4 volts. A Low Drop Out (LDO) regulator must be selected.
- b) The quiescent current of the regulator must be low in order to meet the USB suspend total current requirement of $\leq 500\mu\text{A}$ during USB suspend.

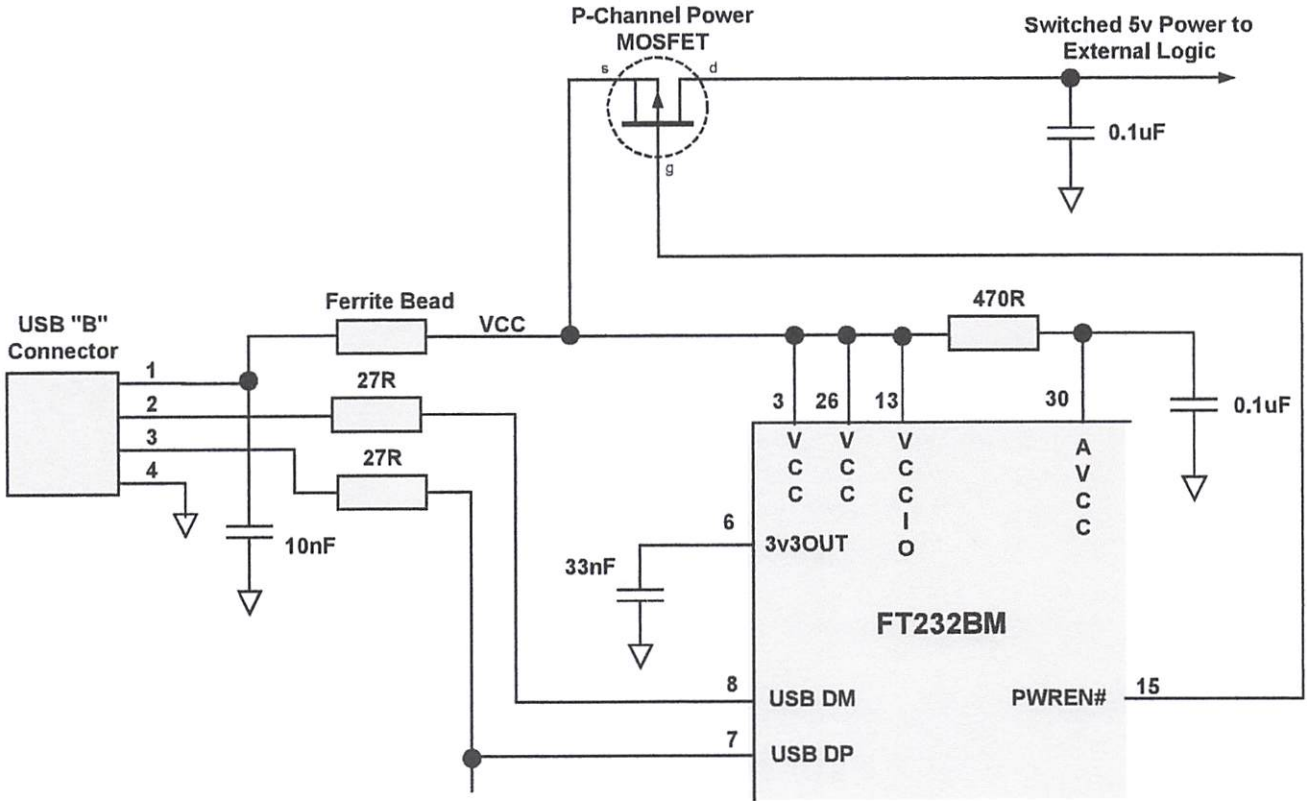
An example of a regulator family that meets these requirements is the MicroChip (Telcom) TC55 Series. These devices can supply up to 250mA current and have a quiescent current of under 1uA.

When using the FT232BM in a self powered USB design, simply connect VCCIO to the 3.3v supply rail of the external 3.3v logic. Suspend current is not a consideration for self powered designs.

In some cases, where only a small amount of current is required ($< 5\text{mA}$), it may be possible to use the in-built regulator of the FT232BM to supply the 3.3v without any other components being required. In this case, connect VCCIO to the 3v3OUT pin of the FT232BM.

7 Power Switching

Figure 16
USB Bus Powered Circuit (<= 100mA) with Power Control



USB Bus powered circuits need to be able to power down in USB suspend mode in order to meet the <= 500uA total suspend current requirement (including external logic). Some external logic can power itself down into a low current state by monitoring the POWEREN# pin. For external logic that cannot power itself down in that way, the FT232BM provides a simple but effective way of turning off power to external circuitry during USB suspend.

Figure 16 shows how to use a discrete P-Channel Logic Level MOSFET to control the power to external logic circuits. A suitable device could be a Fairchild NDT456P or equivalent. This configuration is suitable for powering external logic where the normal supply current is <= 100mA and the logic to be controlled does not generate an appreciable current surge at power-up. For power switching external logic that takes over 100mA or generates a current surge on power-up we recommend that a dedicated power switch i.c. with inbuilt "soft-start" is used instead of a MOSFET. A suitable power switch i.c. for such an application would be a Micrel (www.micrel.com) MIC2025-2BM or equivalent.

Please note the following points in connection with power controlled designs –

-) The logic to be controlled must have it's own reset circuitry so that it will automatically reset itself when power is re-applied on coming out of suspend.
-) Set the soft pull-down option bit in the FT232BM EEPROM.
-) For 3.3v power controlled circuits VCCIO must not be powered down with the external circuitry (PWREN# gets it's VCC supply from VCCIO). Either connect the power switch between the output of the 3.3v regulator and the external 3.3v logic OR if appropriate power VCCIO from the 3v3OUT pin of the FT232BM.

0 Document Revision History

DS232B Version 1.0 – Initial document created 30 April 2002.

DS232B Version 1.1 – Updated 04 August 2002

RESET# Pin description corrected (RESET# does not have an internal 200k pull-up to VCC as previously stated).

Figure 2 pin-out corrected (EECS = Pin 32).

0 Disclaimer

Future Technology Devices International Limited , 2002

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.

Future Technology Devices International Ltd. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

0.0 Contact Information

Future Technology Devices Intl. Limited

1st, George's Studios

13/97 St. George's Road,

Glasgow G3 6JA,

United Kingdom.

Telephone : +44 (0)141 353 2565

Facsimile : +44 (0)141 353 2656

E-Mail (Sales) : sales@ftdichip.com

E-Mail (Support) : support@ftdichip.com

E-Mail (General Enquiries) : admin@ftdichip.com

Web Site URL : <http://www.ftdichip.com>

Agents and Sales Representatives

At the time of writing our Sales Network covers over 40 different countries world-wide. Please visit the Sales Network page of our Web Site site for the contact details our distributor(s) in your country.

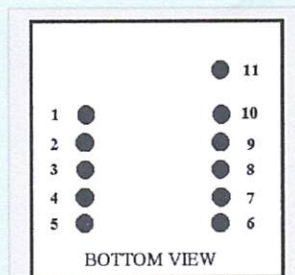
ID SERIES DATASHEET Feb 10 , 2004

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

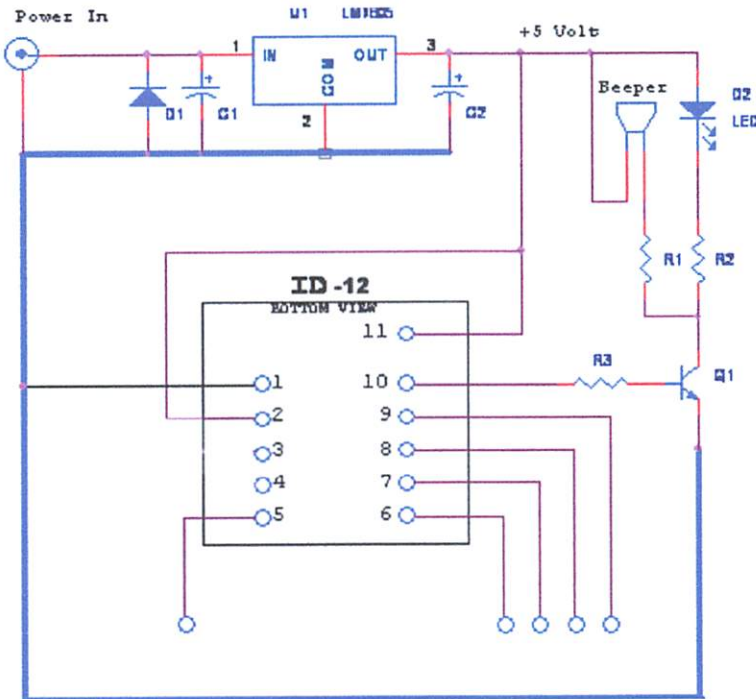
Pin No.	Description	ASCII	Magnet Emulation	Wiegand26
Pin 1	Zero Volts and Tuning Capacitor Ground	GND 0V	GND 0V	GND 0V
Pin 2	Strap to +5V	Reset Bar	Reset Bar	Reset Bar
Pin 3	To External Antenna and Tuning Capacitor	Antenna	Antenna	Antenna
Pin 4	To External Antenna	Antenna	Antenna	Antenna
Pin 5	Card Present	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

ID Innovations

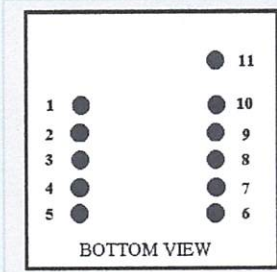
Advanced Digital Reader Technology

---Better by Design

ID-2RW, ID-12RW Brief Data

The ID2-RW, ID12-RW and ID15-RW are a new series of Read/Write modules for the Temec Q5 tag. It has full functionality including password. They contain built-in algorithms to assist customers programming the popular Sokymat Unique type tag. Password protection is allowed. Control is via a host computer using a simple terminal program such as hyper terminal or Qmodem.

ID2 / ID12 / ID20 PIN-OUT



- 1 GND
- 2 RES (Reset Bar)
- 3 ANT (Antenna)
- 4 ANT (Antenna)
- 5 Future
- 6 Program LED
- 7 ASCII in
- 8 Future
- 9 ASCII Out
- 10 Read (LED / Beeper)
- 11 +5V



Operational and Physical Characteristics

Parameters	ID-2RW	ID-12RW	ID-20RW
Read Range	N/A (no internal antenna)	12+ cm (Unique Format)	15+ cm (Unique Format)
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	Temec Q5555	Temec Q5555	Temec Q5555
Read Encoding	Manchester modulus 64	Manchester modulus 64	Manchester modulus 64
Power Requirement	5 VDC @ 13mA nominal	5 VDC @ 30mA nominal	5 VDC @ 50mA 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
Coil Detail	L = 0.6mH - 1.5mH, Q = 15-30	-	-

Description

A host computer is required to send the commands to the module. A simple terminal program such as Qmodem or Hyper-terminal can be used to send commands to the module. The blocks are individually programmable. If you have ever found that the Q5 can be a bit 'Twitchy' to program this programmer module is your solution. The command interface is simple to use and easily understood. The programmer also has two types of internal reader. One of these is provided to read Sokymat 'Unique' type tag configuration.

1944-1945
1946-1947
1948-1949

1950-1951

1952-1953
1954-1955
1956-1957
1958-1959
1960-1961



1962-1963

1964-1965	1966-1967
1968-1969	1970-1971
1972-1973	1974-1975
1976-1977	1978-1979
1980-1981	1982-1983
1984-1985	1986-1987
1988-1989	1990-1991
1992-1993	1994-1995
1996-1997	1998-1999
2000-2001	2002-2003

Year	Month	Day	Event	Location
1944	Jan	15
1944	Feb	20
1944	Mar	25
1944	Apr	30
1944	May	5
1944	Jun	10
1944	Jul	15
1944	Aug	20
1944	Sep	25
1944	Oct	30
1944	Nov	5
1944	Dec	10

...