SKRIPSI

PERANCANGAN DAN PEMBUATAN ANJUNGAN MANDIRI INFORMASI NILAI AKADEMIK DI ITN MALANG MENGGUNAKAN RFID BERBASIS AT89S8253

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

Disusun Oleh :

Herman Bagoes Christmanto

NIM : 04.12.211

JURUSAN TEKNIK ELEKTRO S-1
KOSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009

LEMBAR PERSETUJUAN

PERANCANGAN DAN PEMBUATAN ANJUNGAN MANDIRI INFORMASI NILAI AKADEMIK DI ITN MALANG MENGGUNAKAN RFID BERBASIS AT89S8253

SKRIPSI

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

Disusun Oleh :

Herman Bagoes Christmanto

NIM : 04.12.211

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Joseph Dedy Irawan, ST, MT.	**I Komang Somawirata, ST, MT .**
NIP. 132 315 178	**NIP.Y 103 010 0361**

Mengetahui,

Ketua Jurusan Teknik Elektro S-1

Ir. F. Yudi Limpraptono, MT

NIP.Y 103 950 0274

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTRONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2009

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo KM 2
MALANG

BERITA ACARA UJIAN SKRIPSI

FAKULTAS TEKNOLOGI INDUSTRI

Nama Mahasiswa : Herman Bagoes Christmanto

N I M : 04.12.211

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Elektronika

Judul Skripsi : Perancangan dan Pembuatan *Anjungan Mandiri* Informasi Nilai Akademik Di ITN Malang Menggunakan RFID Berbasis AT89S8253

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

Hari : Selasa

Tanggal : 17 Maret 2009

Dengan Nilai : 84.8 (A)

Panitia Ujian Skripsi

(Ir. H. Sidik Noertjahjono, MT)
Ketua

(Ir. F. Yudi Limpraptono, MT)
Sekretaris

Anggota Penguji

(Ir. Mimien Mustikawati, MT)
Penguji I

(M. Ibrahim Ashari, ST, MT)
Penguji II

KATA PENGANTAR

Puji syukur kehadirat Allah SWT dan Nabi Muhammad SAW yang telah memberikan rahmat dan hidayahnya sehingga saya dapat menyelesaikan Laporan Akhir dengan judul **"PERANCANGAN DAN PEMBUATAN ANJUNGAN MANDIRI INFORMASI NILAI AKADEMIK DI ITN MALANG MENGGUNAKAN RFID BERBASIS AT89S8253".**

Laporan Akhir ini disusun sebagai salah satu syarat penilaian mata kuliah tugas akhir di Program Studi Teknik Elektronika S-1.

Terselesaikannya Laporan Akhir ini tidak lepas dari bantuan banyak pihak. Untuk itu penulis menyampaikan ucapan terima kasih kepada :

1. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro.
2. Bapak Ir Widodo Pudji Muljanto, MT selaku Dosen Wali..
3. Bapak Joseph Dedy Irawan, ST, MT selaku dosen pembimbing 1 yang telah memberikan banyak pengarahan dalam menyusun Laporan Akhir ini.
4. Bapak I Komang Somawira, ST, MT selaku dosen pembimbing 2 yang telah memberikan banyak pengarahan dalam menyusun Laporan Akhir ini.
5. Seluruh staf pengajar di Program Studi Teknik Elektro.
6. Orang tua tercinta dan keluarga yang telah memberikan doa restu, kasih sayang serta semangat dan dukungan sehingga Laporan Akhir ini dapat terselesaikan.
7. Rekan-rekan seperjuangan angkatan 2004 yang telah memberi semangat dan dukungannya.
8. Semua pihak yang telah banyak membantu pembuatan Laporan Akhir ini.

Kami sangat mengharapkan kritik dan saran demi kemajuan dan pengembangan Laporan Akhir ini. Penulis berharap semoga Laporan Akhir ini bermanfaat bagi semua pihak.

Malang, Maret 2009

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAKSI

PERANCANGAN DAN PEMBUATAN ANJUNGAN MANDIRI INFORMASI NILAI AKADEMIK DI ITN MALANG MENGGUNAKAN RFID BERBASIS AT89S8253

Herman Bagoes Christmanto
Nim : 04.12.211
Dosen Pembimbing : I. Joseph Dedy Irawan, ST, MT.
II. I Komang Somawirata, ST, MT.
Jurusan Teknik Elektro S-1,Kosentrasi Teknik Elektronika
Fakultas Teknologi Industri, Institut Teknologi Nasional Malang
Jln. Raya Karanglo Km 2 Malang Tlp (0341)417634 Malang
Website : www.itnac.id Email : itn@itn.ac.id

Kata Kunci : *RFID*, Mikrokontroler AT89S8253, Keypad, Tombol Pilih, PC.

Dalam kemajuan teknologi yang berkembang pesat, terutama dibidang elektronika. Salah satu penerapannya adalah pemanfaatan teknologi RFID. Hal ini diakibatkan oleh beberapa hal yaitu kebutuhan yang besar dari aplikasi untuk konsumen dengan menggunakan teknologi ini. RFID menggunakan chip yang dapat dideteksi pada jarak beberapa meter oleh pembaca RFID. Dengan memanfaatkan tekonologi RFID (Radio Frequency Identification) yang digunakan pada pengambilan Kartu Hasil Studi Semester (KHS) dan Daftar Prestasi Akademik (DPA) di salah satu perguruan tinggi, yang diharapkan dapat mempermudah dan mempercepat pelayanan. Kemudian dengan bantuan RFID reader, keypad, tombol pilih beserta mikrokontroller AT89S8253 Tag tersebut dapat diidentifikasikan. Untuk menampilkan data dari mahasiswa tersebut berdasarkan kode dari Tag RFID yang telah teridentifikasi dilakukan dengan bantuan komputer. Dibutuhkan suatu program yang dapat menampilkan identitas mahasiswa yang berada didalam database, menggunakan bahasa pemrograman Delphi 7 sebagai tampilan visualnya.


INSTITUT TEKNOLOGI NASIONAL
ITN
MALANG

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 satuya penerapannya adalah di perguruan tinggi. Pada umumnya pengambilan informasi kartu hasil studi semester (KHS) dan daftar prestasi akademik (DPA) dilakukan oleh mahasiswa melalui recording pada waktu tertentu saja. Pada saat pengambilan informasi ini sering menimbulkan antrian sehingga terkadang terjadi kekecewaan bagi mahasiswa yang telah menunggu lama, sehingga diperlukan suatu inovasi yang menunjang kebutuhan tersebut, contohnya adalah memanfaatkan teknologi *Wireless* yaitu RFID *(Radio Frequency Identification)* sebagai pengambilan informasi (KHS) dan (DPA).

Sistem pengambilan informasi yaitu setiap mahasiswa yang melakukan pengambilan informasi harus memiliki *tag RFID*, kemudian dengan bantuan *RFID reader* beserta mikrokontroler AT89S8253, *tag* tersebut dapat

diidentifikasikan. Untuk menampilkan data dari mahasiswa tersebut, berdasarkan kode dari *Tag RFID* yang telah teridentifikasi dilakukan dengan bantuan komputer.

1.2. Rumusan Masalah

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

1. Bagaimana menggunakan RFID sebagai masukkan pada mikrokontroler.
2. Bagaimana membuat sistem pengambilan informasi (KHS) dan (DPA) mahasiswa 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 nama mahasiswa, no induk mahasiswa, ID mahasiswa, foto mahasiswa, angkatan, jenis kelamin dll.
2. Tidak membahas catu daya dan frekuensi-frekuensi radio.
3. Mikrokontroler yang digunakan adalah AT89S8253, *RFID reader* yang digunakan ID-12, dan personal computer.
4. Tidak membahas InfraRed.
5. Yang diinformasikan hanya data (KHS) dan (DPA) mahasiswa.

1.4. Tujuan

Tujuan dari penulisan skripsi ini adalah untuk mempercepat proses pengambilan informasi kartu hasil studi semester (KHS) dan daftar prestasi akademik (DPA) mahasiswa di perguruan tinggi agar dengan mudah dan cepat, sehingga penulis mencoba merancang dan membuat Sistem Anjungan Mandiri Informasi Nilai Akademik di ITN Malang Menggunakan RFID Berbasis Mikrokontroler AT89S8253.

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 lansung 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 ukuranya 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*

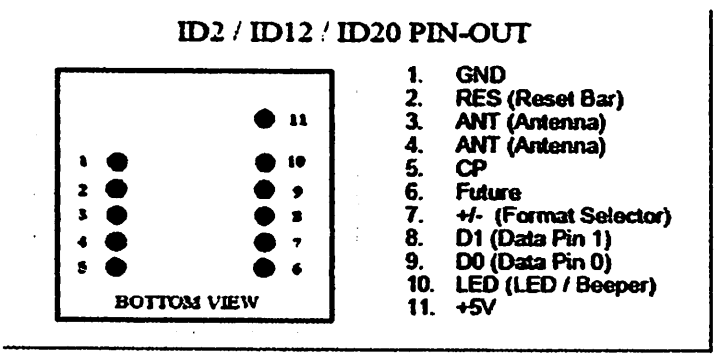
Tag reader digunakan untuk membaca data yang ada pada *tag* melewati RF *interface*. Untuk menambah fungsi *reader* dilengkapi dengan *internal storage*, dan aplikasi perangkat lunak untuk menyimpan data pada *server database*. Pada prakteknya *tag reader* dapat berupa perangkat keras yang terletak pada 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) [11]
(Sumber : *Data Sheet* RFID ID 12 www.datasheet rfid.com)

Tabel 2-1. Fungsi Pin & Format Data

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

(Sumber : *Data Sheet* RFID ID 12 www.alldatasheet rfid.com)

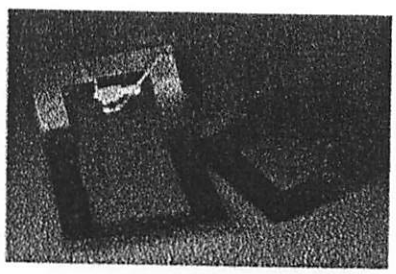
2.2.1.3. *Server Database*

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]

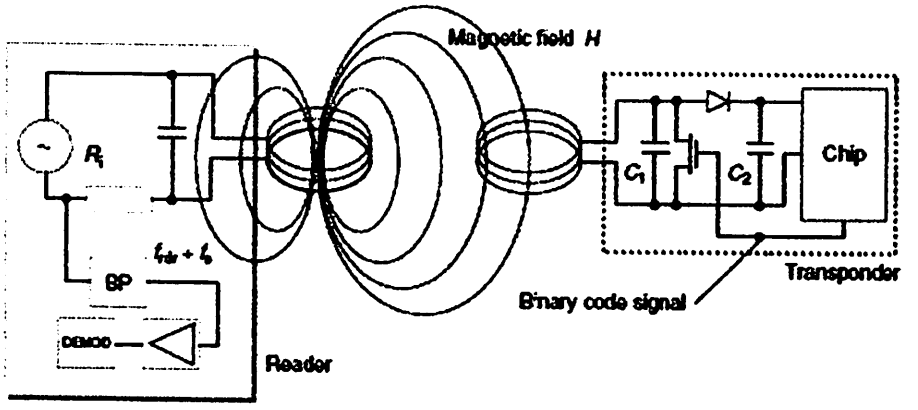
(Sumber : www.rfid.com)

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 elekro 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]
(Sumber : www.rfid.com)

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

(Sumber : ID Series Data Sheet www.datasheet rfid.com)

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

0	3	3	3	3	3	3	4	3	3	4	4	4	0	0	0
2	0	4	6	2	0	1	4	7	6	3	4	3	D	A	3

(Sumber : ID Series Data Sheet www.datasheet rfid.com)

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

(Sumber : ID Series Data Sheet www.datasheet rfid.com)

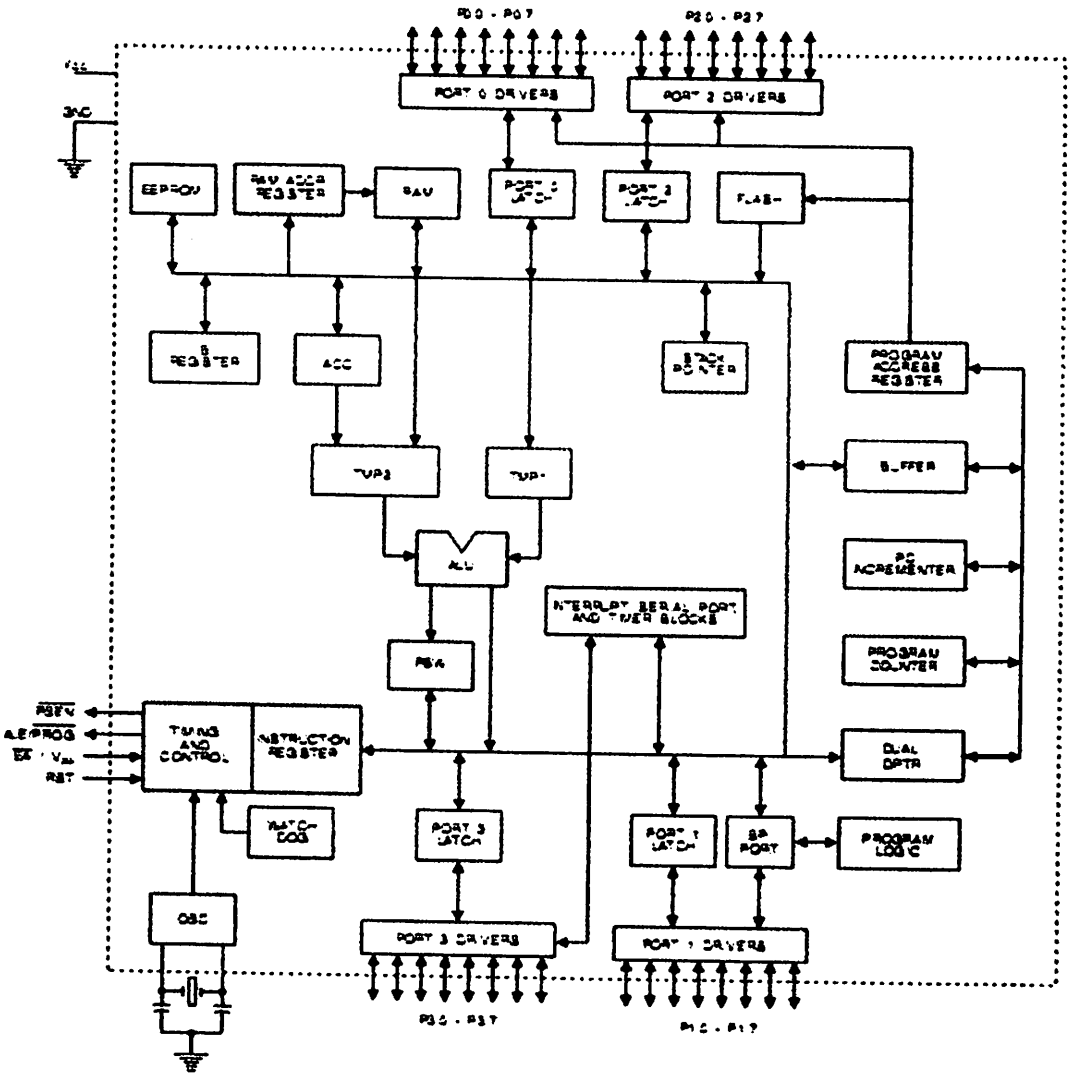
Untuk data yang pertama yaitu angka "04" merupakan data untuk jenis – jenis kartu. Yang digunakan adalah data ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah " 6201D76C ". Kemudian gabungkan data karakter *ASCII* menjadi bilangan *hexadesimal*, setelah itu konversikan bilangan *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 AT89S8253

AT89S8253 merupakan salah satu mikrokontroler keluaran ATMEL Corp. yang memiliki berbagai fasilitas penting sebagai sebuah pengendali, antara lain adalah sebagai berikut:

- Memiliki 12 Kb *Flash* PEROM (*Programmable and Erasable Read Only Memory*) yang dapat diprogram ulang dengan fasilitas SPI (*Serial Programming Interface*). Memori *Flash* PEROM ini dapat bertahan untuk dihapus dan ditulis ulang kira-kira sebanyak 10.000 kali.
- Memiliki 2 Kb EEPROM (*Electrically Erasable Programmable Read Only Memory*) yang dapat berfungsi sebagai data memori. EEPROM ini dapat bertahan untuk dihapus dan ditulis ulang sekitar 100.000 kali.
- Beroperasi pada kisaran tegangan 2,7 V sampai dengan 5,5 V.
- Mendukung sistem pewaktuan (*Clock*) antara 0 HZ sampai dengan 24 MHz *Clock Speed.*
- Fasilitas pengamanan program memori sebanyak 3 level pengamanan data (*Program Memory Lock*).
- Memiliki RAM (*Random Access Memory*) internal sebanyak 256 byte.
- Memiliki 32 unit input dan output (*port pins*) yang dapat diprogram sebagai jalur masukan atau keluaran. Terbagi dalam 4 *port parallel* dan 1 *port serial* yang dapat berjalan dalam mode 2 arah (*Full Duplex*).
- Memiliki fasilitas *Timer* dan *Counter* sebanyak 3 buah, masing-masing dapat difungsikan dengan perhitungan 16 bit.

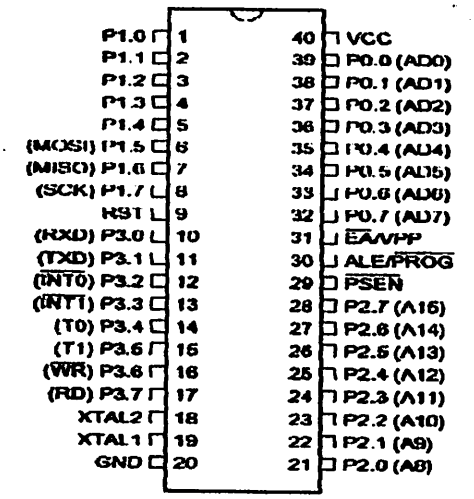
- Memiliki fasilitas *Watchdog Timer* yang dapat diprogram sesuai keinginan.
- Memiliki 2 buah DPTR (*Data Pointer*). *Data Pointer* berguna untuk pengalamatan memori dengan alamat 16 bit.



Gambar 2.4. Diagram Blok AT89S8253
(Sumber : *Datasheet* AT89S8253 www.atmel.com)

2.3.1. Konfigurasi pin AT89S8253

Masing-masing kaki dalam mikrokontroler AT89S8253 mempunyai fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki mikrokontroler AT89S8253, perancangan aplikasi mikrokontroler AT89S8253 akan lebih mudah merencanakan dan membuat sistem yang dirancang. AT89S8253 mempunyai 40 pin, susunan masing-masing pin dapat dilihat pada Gambar 2.5.



Gambar 2-5. Skema *Pinout* Mikrokontroler AT89S8253
(Sumber : *Datasheet* AT89S8253 www.atmel.com)

Mikrokontroler AT89S8253 dibangun dalam sebuah *chip* yang memiliki 40 buah pin. Penjelasan dari masing-masing pin adalah sebagai berikut:

- ***Power Pin*** : terdiri dari pin 40 yang berfungsi sebagai *Vcc*, dan pin 20 adalah sebagai hubungan *Ground*.
- ***Input/Output Pin*** : Mikrokontroler AT89S8253 memiliki 4 buah *port parallel* dan 1 buah *port serial*. Total dari pin *input/output* adalah sebanyak 32 jalur. *Port* 0 menempati 8 buah pin dari nomor pin 32 sampai

dengan pin 39. *Port* 1 menempati 8 buah pin dari nomor pin 1 sampai dengan pin 8. *Port 2* menempati 8 buah pin dari nomor pin 21 sampai dengan pin 28. *Port 3* menempati 8 buah pin dari nomor pin 10 sampai dengan pin 17. Perlu diketahui juga bahwa beberapa pin memiliki fungsi khusus, termasuk pin-pin *serial port* yang terdapat pada *port* 3.

- ***Reset Pin*** : Merupakan *input* dari tombol *reset* pada mikrokontroler. Untuk melakukan *reset* terhadap proses yang sedang dilakukan oleh mikrokontroler, maka diperlukan pemberian nilai 1 (*high*) pada pin ini selama 2 siklus mesin.
- ***Memory Addressing Pin*** : Pin-pin ini berfungsi apabila mikrokontroler melakukan pengaksesan terhadap alamat memori eksternal. *Port* yang dilibatkan dalam hal ini adalah *port* 0 dan *port* 2, sedangkan pin-pin fungsional yang terlibat adalah pin WR (pin nomor 36) dan RD (pin nomor 37).
- **ALE/PROG** : ALE (*Address Latch Enable*) adalah pin *output* untuk melakukan proses *Latching* terhadap pengaksesan *byte* rendah (*low byte*) memori eksternal. *Latching* adalah proses pergantian fungsi antara pengiriman dan penerimaan alamat dan data memori eksternal. Selain itu, pin ini juga memegang peranan dalam proses pengisian program pada *Flash* PEROM. ALE hanya bereaksi dengan perintah MOVX dan MOVC saja. ALE dapat diaktifkan dan di non-aktifkan dengan mengakses bit pertama (bit 0) dari SFR (*Special Function Register*). Pada saat aktif, ALE

akan mengeluarkan sinyal dengan frekuensi 1/6 dari frekuensi osilator mikrokontroler.

- **PSEN** : PSEN (*Program Store Enable*) berfungsi dalam pemberian sinyal *strobe* apabila mikrokontroler melakukan pengaksesan memori program yang berada pada memori eksternal. PSEN diaktifkan sebanyak 2 kali setiap siklus mesin (*machine cycle*) pada saat mikrokontroler mengakses program memori dan memori eksternal.
- **EA/VPP** : EA (*External Access Enable*) harus dihubungkan ke *ground* apabila ingin melakukan akses ke program memori pada memori eksternal. Namun apabila program memori ada di dalam memori eksternal mikrokontroler, dan tidak diperlukan akses ke memori eksternal maka EA harus diberi nilai 1 (*high*) dengan cara dihubungkan ke Vcc. Pin EA berfungsi ganda sebagai penerima tegangan 12 Volt DC (Vpp) dalam proses pengisian program pada *Flash* PEROM.
- ***CPU Clock Pin*** : berfungsi dalam pewaktuan prosesor dari mikrokontroler.
- **XTAL 1** : Masukan *clock* dari kaki osilator kristal ke untai penguat pembalik (*Inverting*) dan untai pewaktu (*clock*) internal mikrokontroler.
- **XTAL 2** : Keluaran dari untai penguat pembalik internal menuju osilator kristal.

2.3.2. Struktur dan Operasi Port

Mikrokontroler AT89S8253 memiliki 4 buah port. Setiap port memiliki 8 buah jalur I/O yang bersifat *bidirectional.* Beberapa karakteristik port mikrokontroler AT89S8253 dijelaskan secara singkat sebagai berikut:

- ***Port 0*** : *Port* 0 adalah salah satu jalur *input/output* 2 arah (*bidirectional*) yang dapat diprogram sesuai kebutuhan. Selain berfungsi sebagai *input/output, port* 0 juga memiliki fungsi khusus dalam pengalamatan 8 bit atau 16 bit alamat memori eksternal. *Port* 0 berfungsi memberikan 8 bit pertama (*low byte*). *Port* 0 juga memiliki *pull-up* internal. *Port* 0 juga berfungsi untuk menerima kode program pada saat *Flash* PEROM mikrokontroler diisi dan juga memberikan verifikasi terhadap kode program tersebut. Dalam pemberian proses verifikasi, port 0 memerlukan bantuan *pull-up* eksternal.
- ***Port 1*** : *Port* 1 juga merupakan 8 bit jalur *input/output* 2 arah (*bidirectional*) seperti halnya *port* 0. *Port* 1 juga memiliki *pull-up* internal. *Port* 1 juga berperan dalam proses pengisian program *Flash* PEROM dan proses verifikasi, yakni sebagai penerima *byte* alamat pertama (*low- order address byte*). Beberapa pin dari *port* 1 memiliki fungsi khusus. Fungsi Khusus pin-pin pada *port* 1 tersebut dapat dilihat pada tabel berikut:

Tabel 2-2. Fungsi Port 1

Nama Pin	Nomor Pin	Deskripsi
P1.0	1	T2 (*Input timer* atau *counter* 2 dari sumber *clock* eksternal)
P1.1	2	T2EX (*Timer/counter 2 capture /reload trigger* dan *direction control*)
P1.4	5	SS (Untuk memilih *slave port*)
P1.5	6	MOSI (*Master data output*, masukan *slave data input pin* untuk fasilitas SPI)
P1.6	7	MISO (*Master data input*, keluaran *slave data output pin* untuk fasilitas SPI)
P1.7	8	SCK (*Master clock output*, masukan untuk *pin slave clock* untuk fasilitas SPI *channel*)

(Sumber : *Datasheet* AT89S8253 www.atmel.com)

- ***Port 2*** : *Port* 2 adalah salah satu jalur *input/output* 2 arah (*bidirectional*) yang dapat diprogram sesuai kebutuhan. *Port* 2 juga memiliki *pull-up* internal. Dalam pengaksesan memori eksternal, *port* 2 berperan dalam pengalamatan *byte* terakhir terhadap memori eksternal (*high-order address byte*) yang memiliki alamat sampai dengan 16 bit. Sedangkan pada saat pengaksesan memori eksternal 8 bit, *port* 2 berfungsi untuk mengeluarkan isi dari SFR P2. Pada saat proses pengisian program *Flash* PEROM dan proses verifikasi, port 2 berfungsi menerima *byte* alamat kedua (*high - order address byte*) dan beberapa sinyal control lainnya.
- ***Port 3*** : Port 3 adalah salah satu jalur *input/output* 2 arah (*bidirectional*) yang dapat diprogram sesuai dengan kebutuhan. *Port* 3 memiliki internal *pull-up*. Dalam proses pemograman *Flash* PEROM dan verifikasi, *port* 3 berfungsi untuk menerima beberapa sinyal kontrol. *Port* 3 adalah *port*

paling spesial dari mikrokontroler AT89S8253 karena memiliki berbagai macam fungsi khusus seperti dijelaskan oleh tabel berikut:

Tabel 2-3. Fungsi Port3

Nama Pin	Nomor Pin	Deskripsi
P3.0	10	RXD (Port penerima komunikasi serial)
P3.1	11	TXD (Port pengirim komunikasi serial)
P3.2	12	INT0 (Masukan untuk fasilitas interupsi eksternal 0)
P3.3	13	INT1 (Masukan untuk fasilitas interupsi eksternal 1)
P3.4	14	T0 (Masukan untuk fasilitas timer eksternal 0)
P3.5	15	T1 (Masukan untuk fasilitas timer eksternal 1)
P3.6	16	WR (External data memory write strobe)
P3.7	17	RR (External data memory read strobe)

(Sumber : *Datasheet* AT89S8253 www.atmel.com)

2.3.3. Antarmuka Serial

AT89S8253 memiliki fasilitas *on-chip serial port* untuk berkomunikasi dengan modem atau terminal yang memiliki port serial untuk komunikasi dengan dunia luar. SBUF (alamat 99 H) dan SCON (alamat 9A H) memegang peranan terpenting dalam komunikasi *serial* mikrokontroler, masing – masing atas 1 byte *register*. *Register* SBUF berkaitan dengan data, sedangkan register SCON lebih ke arah control. Selain 2 buah *register* di atas, adapula 1 bit yang terletak pada *register* PCON yang berperan dalam komunikasi *serial*, bit tersebut disebut dengan SMOD.

Tabel 2-4. Peta *Bit Register* SCON

SCON. 7	SCON. 6	SCON. 5	SCON. 4	SCON. 3	SCON. 2	SCON. 1	SCON. 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

(Sumber: Teknik Antarmuka Mikrokontroler Dengan Komputer Berbasis Delphi)

Keterangan dan fungsi dari masing – masing bit adalah sebagai berikut:

SM0 dan SM1 berfungsi untuk menentukan mode dari komunikasi serial yang akan dipakai. Ringkasan dari penggunaan SM0 dan SM1 dapat dilihat pada tabel berikut:

Tabel 2-5. Mode Komunikasi *Serial* Mikrokontroler

SM0	SM1	Mode Komunikasi Serial
0	0	Mode 0 (8 bit Shift Register)
0	1	Mode 1 (8 bit UART)
1	0	Mode 2 (9 bit UART)
1	1	Mode 3 (9 bit UART)

(Sumber: Teknik Antarmuka Mikrokontroler Dengan Komputer Berbasis Delphi)

Ada 4 macam tipe operasi serial yang didukung oleh mikrokontroler. 3 diantaranya menganut sistem transmisi asinkron dan satu menganut sistem transmisi sinkron. Berikut adalah penjelasan dari masing – masing mode tersebut:

1. **Mode 0**

   Mode 0 juga dikenal dengan mode 8 bit *shift register*. Mode ini dipilih dengan cara menuliskan nilai 0 ke dalam bit SM0 dan SM1 pada *register* SCON. Data *serial* masuk dan keluar melalui pin RxD (port P3.0) dan pin TxD (port P3.1) melakukan *clock*. 8 buah bit data dikirimkan dan diterima dengan urutan LSB (*Least Significant Bit*) terlebih dahulu. *Baudrate* yang digunakan 1/12 dari frekuensi osilator.

Dengan menggunakan mode ini, pengiriman data dimulai dengan instruksi yang menuliskan data ke SBUF. Kemudian secara bergeser (*shift*), satu demi satu bit data tersebut akan dikirimkan melalui pin RxD sesuai *clock* selama 1 siklus mesin yang diberikan oleh pin TxD. Proses penerimaan data dimulai dengan membuat nilai REN pada register SCON menjadi 1 dan nilai RI menjadi 0. Ketika RI bernilai 0, maka pin TxD akan mengeluarkan 1 *clock*, sehingga data akan masuk ke pin RxD. Saat semua data diterima, maka nilai RI akan kembali menjadi 1.

2. **Mode 1**

Mode 1 lebih dikenal dengan mode 8 bit UART dengan *baudrate* yang dapat diatur. UART (*Universal Asynchronous Receiver/Transmitter*) adalah sebuah piranti yang menerima dan mengirim data *serial* dimana setiap blok data dimulai dengan pengiriman sebuah *start bit* dan diakhiri dengan pengiriman *stop bit*. Kadang juga sering disertakan *parity bit* diantara bit data terakhir dan *stop bit*.

Dengan mode 1, pada proses pengiriman 10 bit akan dikirimkan dari pin TxD dan diterima melalui pin RxD. 10 bit data ini terdiri dari satu buah *start bit* (selalu bernilai 0), delapan buah data bit dan satu buah *stop bit* (selalu bernilai 1). Pada proses penerimaan data, *stop bit* akan menempati bit RB8 dari register SCON. Pada mode ini, *baudrate* yang digunakan bisa bervariasi dan ditentukan dengan fasilitas *timer*. Pengiriman data dimulai dengan penulisan data pada SBUF. Seperti halnya mode 0, proses penerimaan diakhiri dengan membuat RI bernilai 1.

3. **Mode 2**

Mode 2 dikenal dengan nama UART 9 bit dengan *baudrate* tetap. Mode ini diset dengan membuat SM1 bernilai 1 dan SM0 bernilai 0. Pada mode ini, yang dikirimkan dan diterima oleh TxD dan RxD adalah sejumlah 11 bit. Susunannya terdiri dari sebuah *start bit*, delapan buah data bit, bit ke 9 yang dapat diprogram dan sebuah *stop bit*. Dalam proses pengiriman data, bit ke 9 ini memiliki alokasi tersendiri yakni pada bit TB8 pada register SCON. Sedangkan dalam proses penerimaan, bit tersebut menempati bit RB8 pada SCON. *Baudrate* yang dapat digunakan ditetapkan 1/32 dari frekuensi osilator atau 1/64 dari frekuensi osilator.

4. **Mode 3**

Sistem yang diterapkan mode 3 menyerupai mode 2 dalam hal metode pengiriman dan jumlah bit data dalam tiap paket data yang dikirim ataupun yang diterima. *Register* yang berperan di dalamnya juga sama. Hanya saja pada mode 3 *baudrate* yang digunakan bervariasi. *Baudrate* dibangkitkan dengan fasilitas *timer* seperti halnya pada pembangkitan *baudrate* mode 1.

2.3.4. Pengaturan Baud Rate Port Serial

Baud rate ini merupakan *baud rate* pada komunikasi data yang digunakan pada port serial mikrokontroller AT89S8253. *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 :

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

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)

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 2.6. Baudrate Komunikasi Serial Mikrokontroller

Baud Rate	Resonator Kristal	Nilai Bit SMOD	Nilai isi Ulang (reload) TH1
9600	12.000 MHz	1	-7(F9H)
2400	12.000 MHz	0	-13(F3H)
1200	12.000 MHz	0	-26(E6H)
19200	11.0592 MHz	1	-3(FDH)
9600	11.0592 MHz	0	-3(FDH)
2400	11.0592 MHz	0	-12(F4H)
1200	11.0592 MHz	0	-24(E8H)

(Sumber: Teknik antar muka Mikrokontroller dengan komputer)

Baudrate untuk Mode 2 bergantung pada nilai bit SMOD pada regiater *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:

$$Baudrate\,Mode\,2 = \frac{2^{SMOD}}{64} \; x \; Frekuensi\_osilator \quad \text{(bps)}$$

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:

$$BaudrateModel\&3 = \frac{2^{SMOD}}{32} x\,(lajuTimpahanTimer1) \qquad \text{(bps)}$$

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 :

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

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.

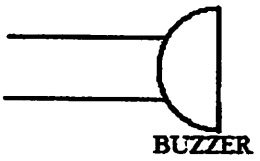
Tabel 2.7. Rumus Penghitungan Baudrate pada Komunikasi Serial[3]

Mode	Baud Rate	
0	1/12 f osilator	
1	SMOD = 0 $Baudrate = \frac{f_{osc}}{12x[256 - TH1]x32}$	SMOD = 1 $Baudrate = \frac{f_{osc}}{12x[256 - TH1]x16}$
2	1/64 f	1/32 f
3	$Baudrate = \frac{f_{osc}}{12x[256 - TH1]x32}$	$Baudrate = \frac{f_{osc}}{12x[256 - TH1]x16}$

(Sumber : Teknik antar muka Mikrokontroller dengan komputer)

2.4. 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 2-6. Rangkaian *Driver Buzzer*[8]

2.5. RESISTOR

Resistor atau sering disebut werstan, tahanan atau penghambat, adalah suatu komponen elektronik yang dapat menghambat gerak lajunya arus gelombang listrik.

Resistor biasa disingkat dengan huruf "R"(huruf R besar). Satuan resistor adalah Ohm. Kemampuan resistor untuk menghambat disebut juga resistansi atau hambatan listrik. Besarnya hambatan ditulis dalam satuan Ohm. Suatu resistor dikatakan memiliki hambatan 1 Ohm apbila resistor tersebut menjembatani beda tegangan sebesar 1 Volt dan arus listrik yang timbul akibat tegangan tersebut adalah sebesar ampere.

Hubungan antara hambatan, tegangan, dan arus, dapat disimpulkan melalui hukum berikut ini , yang terkenal sebagai hukum Ohm:

$$R = \frac{V}{I}.$$

Dimana V adalah beda potensial antara kedua ujung benda penghambat, adalah beda arus yang melalui benda penghambat, dan R adalah besarnya hambatan benda penghambat tersebut.

Berdasarkan pemakaiannya, resistor adalah:

- Resistor statis (tetap nilainya), adalah sebuah resistor penghambat gerak arus, yang nilaimya tidak adpat berubah, jadi selalu tetap (konstan). Resistor ini biasanya dibuat dari nikelin atau karbon.
- Resistor Variable (berubah-ubah nilainya), ialah sebuah resistor yang nilainya dapat berubah-ubah dengan jalan menggeser atau memutar toggle pada alat tersebut. Sehingga nilai resistor dapat kita bagi menjadi 2, potensiometer, rheosrat dan Trimpot (Trimmer Potensiometer) yang biasanya menempel pada papn rangkaian (Printed Circuit Board, PCB).

2.6. TRANSISTOR

Prinsip kerja transistor adalah arus bias-emiter yang kecil mengatur besar arus colector-emitter. Bagian penting berikutnya adalah bagaimana caranya memberi arus bias yang tepat sehingga transistor dapat bekerja optimal.

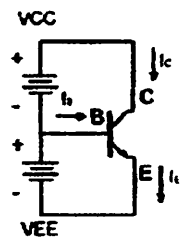
2.6.1. Arus Bias

Ada tiga cara yang umum untuk memberi arus bias pada transistor, yaitu rangkaian CE *(Common Emitter)*, CC *(Common Colector)*, CB *(Common Base)* . namun saat ini akan lebih detil jika dijelaskan bias transistor rangkaian CE. Dengan menganalisa rangkaian CE akan dapat diketahui beberapa parameter penting dan berguna, terutama untuk memilih transistor yang tepat untuk aplikasi tertentu. Misalnya untuk aplikasi pengolahan sinyal audio tentu saja tdak menggunakan transistor power.

2.6.2. Arus Emitor

Dari huum kirchoff diketahui bahwa jumlah arus yang masuk ke satu titik akan sama dengan arus yang keluar. Jika teorema tersebut diaplikasikan pada transistor, maka hukum itu menjelaskan hubungan :

$$I_E = I_C + I_B$$



Gambar 2-7. Arus Emitor
(Sumber: www.electroniclab.com.rubrik elka analog)

Persamaan di atas mengatakan arus emitor I_E adalah jumlah dari arus kolektor I_C dengan arus base I_B. Karena arus I_B sangat kecil atau disebutkan $I_B << I_C$, maka dapat dinyatakan:

$$I_E = I_C$$

2.6.3. Alpha (a)

Pada tabel data transistor *(data book)* sering dijumpai spesifikasi a_{dc} (alpha dc) yang tidak lain adalah:

$$a_{dc} = I_C / I_E$$

definisinya adalah perbandingan arus kolektor terhadap arus emitor. Karena besar arus kolektor umumnya hampir sama dengan besar arus emitor,

maka idealnya besarnya a_{dc} adalah = 1 (satu). Namun pada umunya transitor yang ada memiliki a_{dc} kurang lebih antara 0,95 sampai 0,99.

2.6.4. Bheta (b)

Bheta didefinisikan sebagai besra perbandingan antara arus kolektor dengan arus base.

$$b = I_C / I_B$$

Dengan kata lain, b adalah parameter yang menunjukkan kemampuan penguatan arus *(current gain)* dari suatu trasistor. Parameter ini tertera di *databook* transistor dan sangat membantu para perancang rankaian elektronika dalam merencanakan rangkaiannya.

Misalnya jika suatu transistor diketahui besar b= 260 dan diinginkan arus kolektor sebesar 10mA, maka berapakah arus bias base yang diperlukan. Tentu saja jawabannya sangat mudah yaitu:

$$I_B = I_C / b = 10mA / 250 = 40 \mu A$$

arus yang terjadi pada kolektor transistor yang memiliki b =200 jika diberi arus bias base sebesar 0,1 mA adalah :

$$I_C = b.I_B = 200.0{,}1\ mA = 20\ mA$$

Dari rumusan diatas lebih terlihat definisi penguatan arus transistor, yang sekali lagi, arus base yang lebih kecil menjadi arus kolektor yang lebih besar.

2.6.5. Common Emitter (CE)

Rangkaian CE adalah rangkaian yang paling sering digunakan untuk berbagai aplikasi yang menggunakan transistor. Dinamakan rangkaian CE sebab titik *ground* atau titik teganagan 0 Volt dihubungkan pada titik emitter.

Sekilas tentang notasi, ada beberapa notasi yang sering digunakan untuk menunjukkan besar teganagan pada suatu titik maupun antar titik. Notasi dengan satu *subscript* adalah untuk menunjukkan besar tegangan pada satu titik, misalnya V_C = tegangan kolektor, V_B =tegangan base, dan V_E =tegangan emitter.

Ada juga notasi dengan dua subcript yang dipakai untuk menunjukkan besar tegangan antara dua titik, yang disebut juga dengan tegangan jepit. Diantaranya adalah:

V_{CE} = tegangan jepit kolektor-emitor

V_{BE} =tegangan jepit base-emitor

V_{CB} =tegangan jepit kolektor base

Notasi seperti V_{BB}, V_{CC}, V_{EE} berturut-turut adalah besar sumber tegangan yang masuk ke titik base,kolektor, dan emitor.

2.6.6. Kurva Base

Hubungan antara IB dan VBE tentu saja akan berupa kurva dioda. Karena memang telah diketahui bahwa junction base-emitor tidak lain adalah sebuah dioda. Jika hukum Ohm diterapkan pada loop base diketahui adalah:

$$I_B = \frac{(V_{BB} - V_{BE})}{R_B}$$

V_{BE} adalah tegangan jepit dioda junction base-emitor. Arus hanya akan mengalir jika teganagn antara base-emitor lebih besar dari V_{BE}. Sehingga arus I_B mulai aktif mengalir pada saat nilai V_{BE} tertentu.

Besar V_{BE} umumnya tercantum dalam databook, tetapi untuk penyerhanaan umunya diketahui V_{BE}=0,7 Volt untuk transistor silikon dan V_{BE}= 0,3 Volt untuk transistor germanium. Nilai ideal V_{BE}=0 Volt.

Sampai disini akan sangat mudah mengetahui arus I_B dan arus I_C dari rangkaian berikut ini, jika diketahui besar b = 200. misalnya yang digunakan adalah transistor yang dibuat dari bahan silikon.

$$I_B = \frac{(V_{BB} - V_{BE})}{R_B}$$

$$= \frac{(2V - 0{,}7V)}{100K\Omega}$$

$$= 13\,\mu A$$

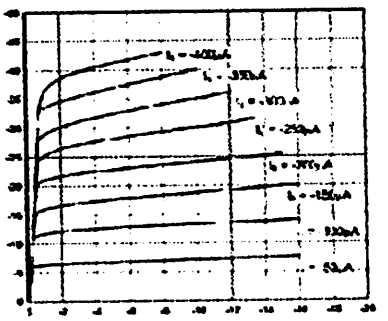
Dengan b = 200, maka arus kolektor adalah :

$$I_C = b.I_B$$

$$= 200 \times 13\,\mu A \quad = 2{,}6mA$$

2.6.7. Kurva Kolektor

Sekarang sudah diketahui konsep arus base dan arus kolektor, satu hal lain yang menarik adalah bagaimana hubungan arus base I_B, arus kolektor I_C, dan arus teganagn kolektor-emitter V_{CE}. Dengan menggunakan rangkaian, tegangan V_{BB} dan V_{CC} dapat diatur untuk memperoleh plot garis-garis kurva kolektor. Pada gambar berikut telah diplot beberapa kurva kolektor arus I_C terhadap V_{CE} dimana arus I_B dibuat konstan.



Gambar 2-8. Arus Emitor
(Sumber: www.electroniclab.com rubrik elka analog)

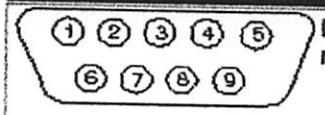
Dari kurva ini dapat terlihat ada beberapa region yang menunjukkan daerah kerja transistor. Pertama adalah daerah saturasi, lalu daerah cut-off, kemudian daerah aktif, dan seterusnya adalah daerah breakdown.

2.7. DB 9

Proses tranfer secara serial dengan menggunakan EIA RS 232-C antara 2 buah terminal biasanya memerlukan sebuah DTE (Data Terminal Equipment)untuk masing-masing terminal. Kadang diperlukan seperangkat peralatan ntuk kebutuhan koumikasi yang lebih kompleks misalkaya dengan memanfaatlkan modem. Perangkat tersebut sering dsebut dengan DCE (Data Communication Equipment).

Pada prinsipnya proses transfer data dengan menggunakan sebuah serial interface ini sangat sederhana. Data yang ditransfer dari satu terminal akan diterima oleh terminal lainnya.berikut penjelasan tentang DB9 :

RS232 DB9 INTERFACE



PIN#	SIGNAL	PIN#	SIGNAL
1	N/A	6	DSR
2	RXD	7	RTS
3	TXD	8	CTS
4	DTR	9	N/A
5	GND		

Gambar 2-9. DB9 dan fungsi masing-masing pin

Jenis data yang akan ditransfer adalah dalam bentuk data biner (bit per bit transfer) dengan satuan baud untuk kecepatan proses transfernya (bit per detik).

Dalam proses transfer ini harus terdapt suatu peralatan yang berfungsi sebagai hand-shake (jabat tangan) yang berfungsi sebagai pematau status yang diterima/ada untuk memberikan suatu respon yang sesuai.

Dalam merancang software komunikasi serial, hand-shake disempurnakan dengan menambahjkan karakter pengendali dala deretan/jumlah bit data yang ditransfer yang biasa disebut sebagai start bit dan stop bit.

Secara sederhana dapat dijelaskan bagaimana konsep interface antara DTE dan DCE dilakuakan adalah sebagai berikut:

- ketika DTE ingin mengirimkan data, sebuah protokol yaitu RTS dikirimkan untuk memberitahu DCE.
- Pada saat itu input RTS pada DCE menjadi aktif.
- Jika DCE mampu menerima data, maka ia akan membalasnya dengan mengirimkan CTS.
- Begitu DTE menerima balasan, input CTS-nya diaktifkan.

- Pengiriman data dilakukan melalui TxD.
- Penerimaan data dilakukan melalui RxD.

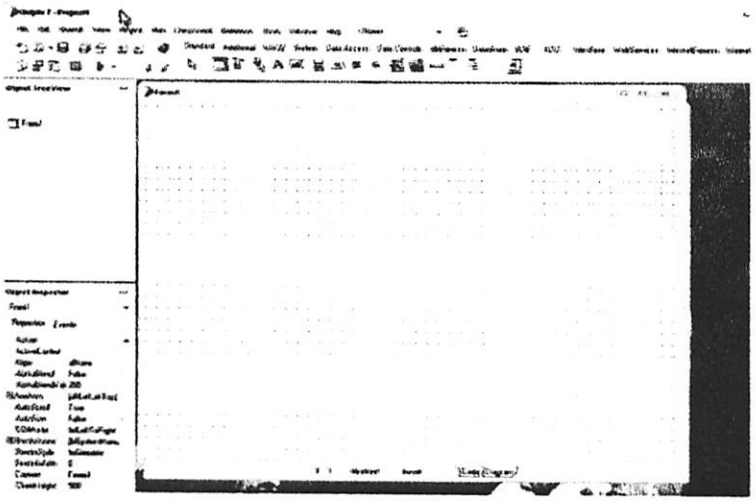
2.8. Borland Delphi

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 programer merealisasikan program aplikasi mereka. Dengan IDE seorang programer 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.

2.8.1. 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 2-10. IDE (*Integrated Development Enviroment*)[10]

(**Sumber:** Teknik Antarmuka Mikrokontroler Dengan Komputer Berbasis Delphi)

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.

2.8.2. 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 Pallete* 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

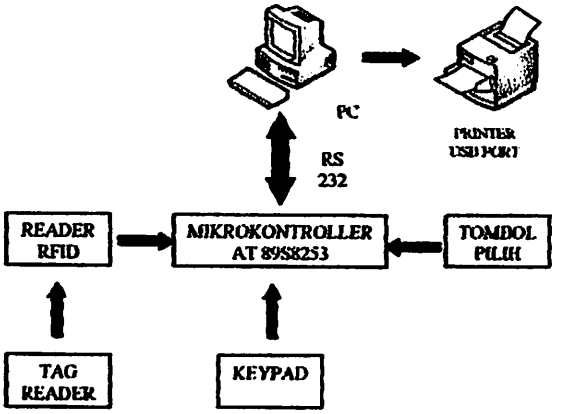
PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan membahas tentang perencanaan dan pembuatan keseluruhan sistem perangkat keras (*hardware*) dan perangkat lunak (*software*) yang digunakan dalam anjungan mandiri informasi nilai akademik di ITN Malang menggunakan RFID, Mikrokontroller AT89S8253 yang di antar muka *(interface)* ke PC menggunakan kabel serial, 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***

  Pada perancangan *RFID Reader* ini digunakan type ID-12 yang kompatibel dengan Tag RFID dan komunikasinya secara wireless.

- **Mikrokontroler AT89S8253**

  Sebagai penerjemah dari kode tag RFID untuk selanjutnya dikirim menuju personal komputer dan untuk membaca pasword dari keypad.

- **KEYPAD**

  Berfungsi untuk memasukkan pasword mahasiswa.

- **TOMBOL PILIH**

  Berfungsi untuk memilih menu yang di tampilkan pada PC.

- **RS 232**

  Berfungsi sebagai *interface* antara mikrokontroller dengan komputer.

- **PC**

  Berfungsi untuk mengolah data base mahasiswa dan menampilkan nama mahasiswa, no induk, foto, jenis kelamin, informasi (KHS) dan (DPA) mahasiswa.

3.3. Prinsip Kerja Alat

Setiap mahasiswa / mahasiswi harus memiliki *Tag* RFID untuk pengambilan informasi nilai akademik. *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 melalui komunikasi serial RS232, kemudian PC dicocokkan dengan database yang telah dirancang, jika data tersebut valid maka PC mengupdate database.

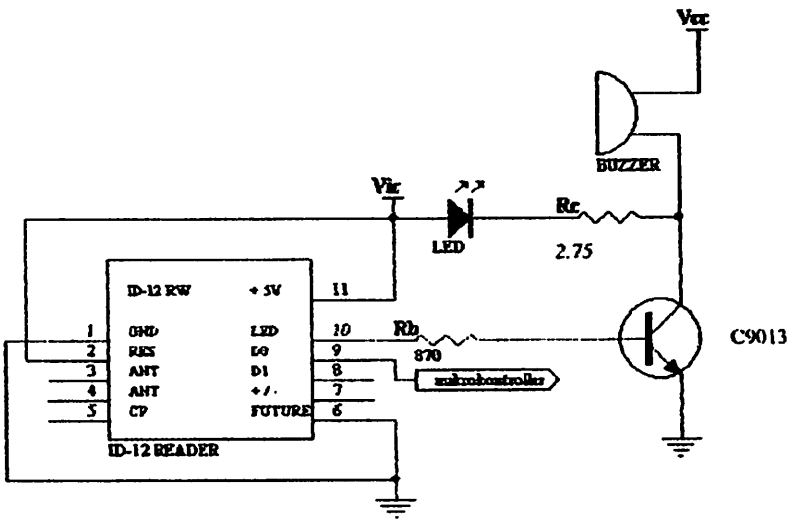
3.4. Perancangan Perangkat Keras (*Hardware*)

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*. Ukuaran *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 pengambilan informasi nilai akademik yang pada saat kartu ini didekatkan ke pembaca / reader RFID maka akan dikenali dan akan teridentifikasi nomor seri yang ada

didalamnya, sehingga setiap mahasiswa / mahasiswi 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 diatas digunakan transistor type S9014 sebagai switch pada rangkaian RFID, dengan nilai I_b pada data sheet transistor yang diinginkan 5mA, V_{BE} = 0.65 V dan $V_{RFID} = V_{B\ pada\ transistor}$ = 5V maka di dapatkan :$V_{CC} = I_B.R_B + V_{BE}$, sehingga

$$Vic = I_B.R_B + V_{BE}$$
$$5 = 5.10^{-3}.R_B + 0.65$$
$$R_B = \frac{5-0.65}{5.10^{-3}}$$
$$= \frac{4.35}{5.10^{-3}}$$
$$= 0{,}87.10^{3}\Omega$$

Jadi dengan nilai R_B = 0,87kΩ tidak terdapat dipasaran maka diganti dengan nilai resistor yang mendekati yaitu R_B = 1 kΩ.

Transistor type S9013 dalam rangkaian ini mempunyai nilai I_B sesuai data sheet = 10mA dan dapat digunakan untuk Switch pada rangkaian RFID.

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

$$Rc = \frac{V_{RFID} - V_{LED} - V_{CE}}{Ic} = \frac{(5-2-0.3)V}{100mA} = 2{,}75\Omega$$

3.5. Mikrokontroller AT89S8253

Disini mikrokontroller berfungsi sebagai perantara untuk data yang dikirim dari *Tag Reader*, keypad, dan tombol pilih selanjutnya diolah oleh personal computer. Mikrokontroller juga sebagai pengendali dari alat, agar dapat melakukan prosesnya harus didukung oleh beberapa komponen tambahan, yakni berupa rangkaian clock dan rangkaian reset.

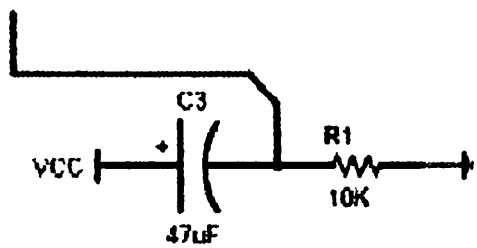
3.5.1 Perancangan minimum sistem AT89S8253

Pada bagian ini perancangan yang akan dibahas meliputi :

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

3.5.2 Perancangan Rangkaian Reset

Untuk *mereset* mikrokontroler AT89S8253, 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 3-3 sebagai berikut :



Gambar 3-3. Perancangan Rangkaian *Reset*

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

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

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

reset(min) = T × periode yang dibutuhkan

= 9,042 × 10 -8 × 24 = 2,17 μs

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 kΩ dan C = 10 μF, maka :

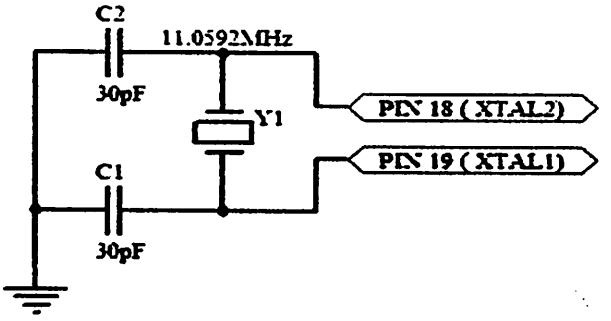
$$t = 0{,}357\ R.C = 0{,}357 \times 8200\Omega \times 10.10^{-6} = 29{,}274\ ms$$

Jadi dengan nilai komponen R = 8,2 kΩ dan C = 10μF dapat memenuhi syarat minimal untuk waktu yang dibutuhkan oleh mikrokontroler.

3.5.3 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* AT89S8253. 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.

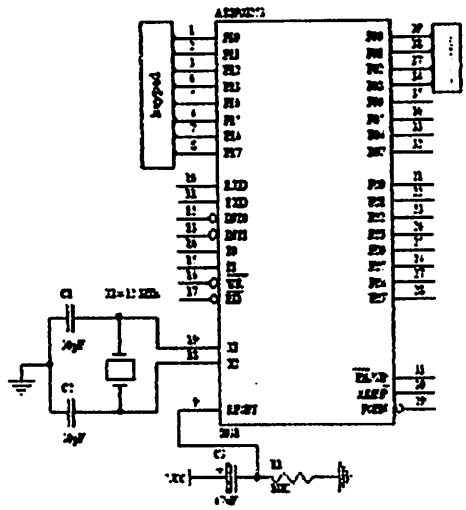


Gambar 3-4. Perancangan Rangkaian *Clock*

3.5.4 Perancangan Pengaturan Port pada Mikrokontroler AT89S8253

Pada perancangan port ini akan dipaparkan port apa saja yang akan digunakan pada Mikokontroller AT89S8253 yaitu:

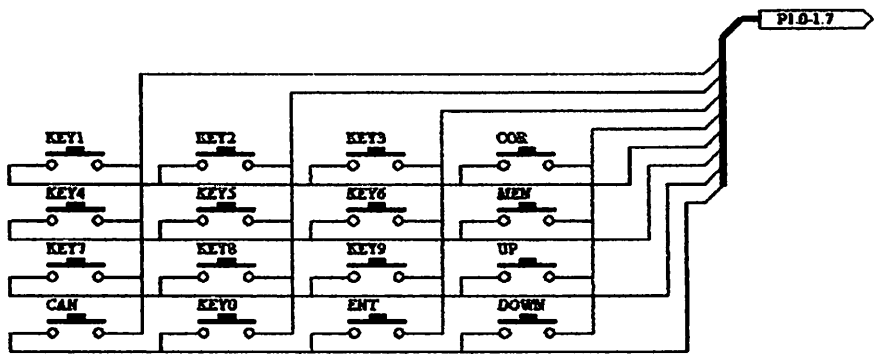
1) *Port* 1,
   a) P1.0 – P1.7 (pin 1 – pin 7) digunakan sebagai port untuk memasukkan data 8 bit pada *KEYPAD.*
2) *Port* 3,
   a) P3.0 (pin 10) RXD serial input yang dihubungkan ke D0 ID-12 *reader* yang digunakan untuk menerima data dari tag RFID *redaer.*
   b) P3.1 (pin 11) TXD serial *output, port* ini digunakan oleh mikrokontroller 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 AT89S8253

3.6 Perancangan Rangkaian *KEYPAD*

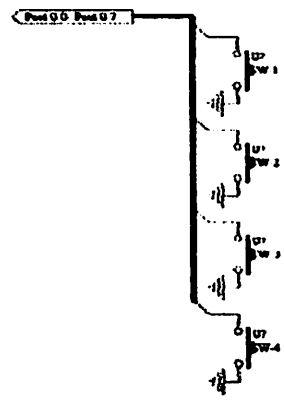
Keypad (papan tombol) merupakan rangkaian yang disusun secara matriks dari 16 tombol tekan (*push button*) yang berfungsi sebagai input data. Untuk menerjemahkan penekanan pada salah satu tombol, *Keypad* akan disambung ke Port 1.0 s/d Port 1.7 dari mikrokontroller AT89S8253. Tombol-tombol tersebut disusun secara matriks dengan 4 baris dan 4 kolom. Rangkaian tombol *Keypad* dapat dilihat pada gambar berikut :



Gambar 3-6. Rangkaian *Keypad*

3.7 Perancangan Rangkaian *TOMBOL PILIH*

Tombol Pilih merupakan rangkaian yang disusun secara matriks dari 4 tombol tekan (*push button*) yang berfungsi sebagai input data. Untuk menerjemahkan penekanan pada salah satu tombol, *Tombol Pilih* akan disambung ke Port 0.0 s/d Port 0.7 dari mikrokontroller AT89S8253. Rangkaian *Tombol Pilih* dapat dilihat pada gambar berikut :



Gambar 3-7. Rangkaian *Tombol Pilih*

3.8 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.*

3.8.1 Program Aplikasi Mikrokontroller

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

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.

$$Baud\ rate = \frac{2^{SMOD}}{32} \text{ X } \frac{Frekuensi\_osilator}{12x[256-(TH1)]}$$

$$9600 = \frac{2^{0}}{32} \text{ X } \frac{11.059x10^{6}}{12x[256-(TH1)]}$$

$$[256\text{-}(TH1)] = \frac{1}{32} \text{ X } \frac{11.059x10^{6}}{12X9600}$$

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

$$TH1 = 256 - 3 = 253_{D} = 0xFD_{H}$$

3.8.2 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 mikrokontroller sebagai saluran masukan dan keluaran, sehingga perangkat lunak dalam program komputer perlu diketahui terlebih dahulu aturan-aturan atau

protokol komunikasi yang digunakan untuk mengatur jalanya komunikasi antara komputer dengan mikrokontroller. 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.

3.8.3 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 :

3.8.3.1 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 teeks 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.

3.8.3.2 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.

3.8.3.3 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.

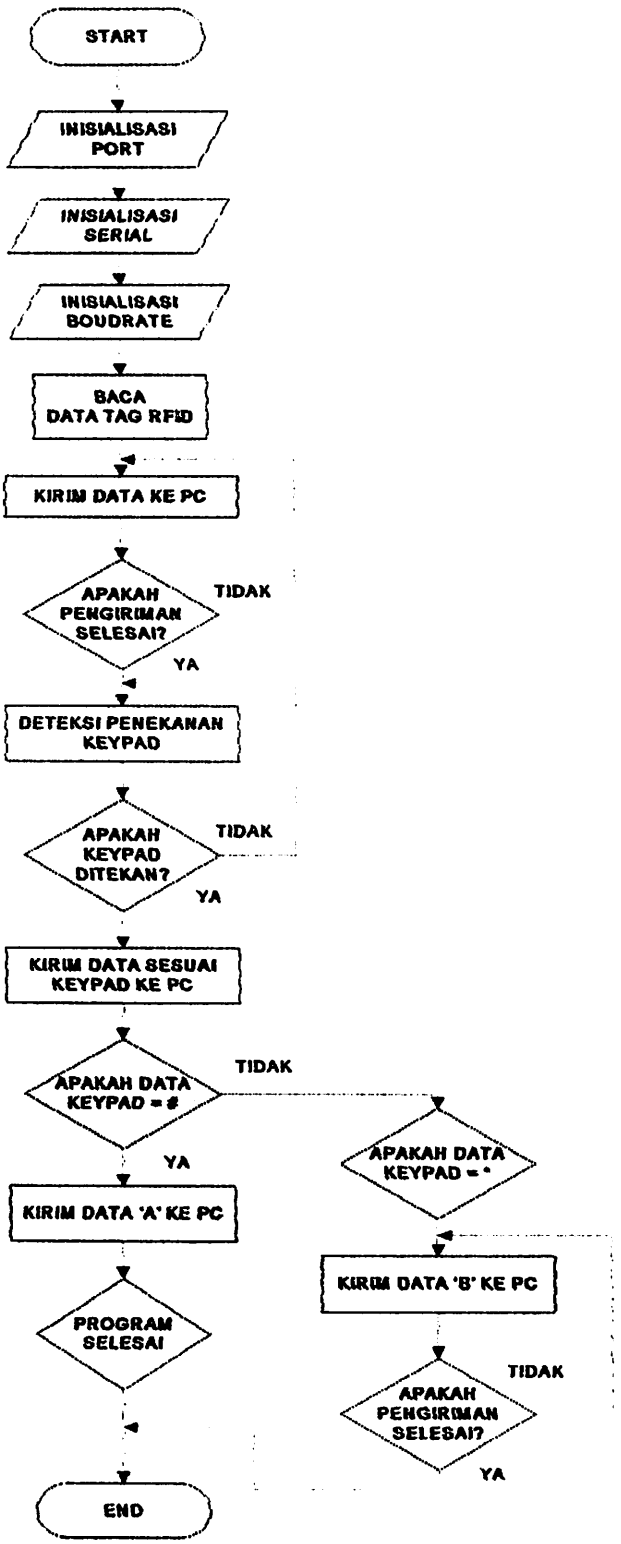
- ***DBNavigator***

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

- ***NOTEBOOK***

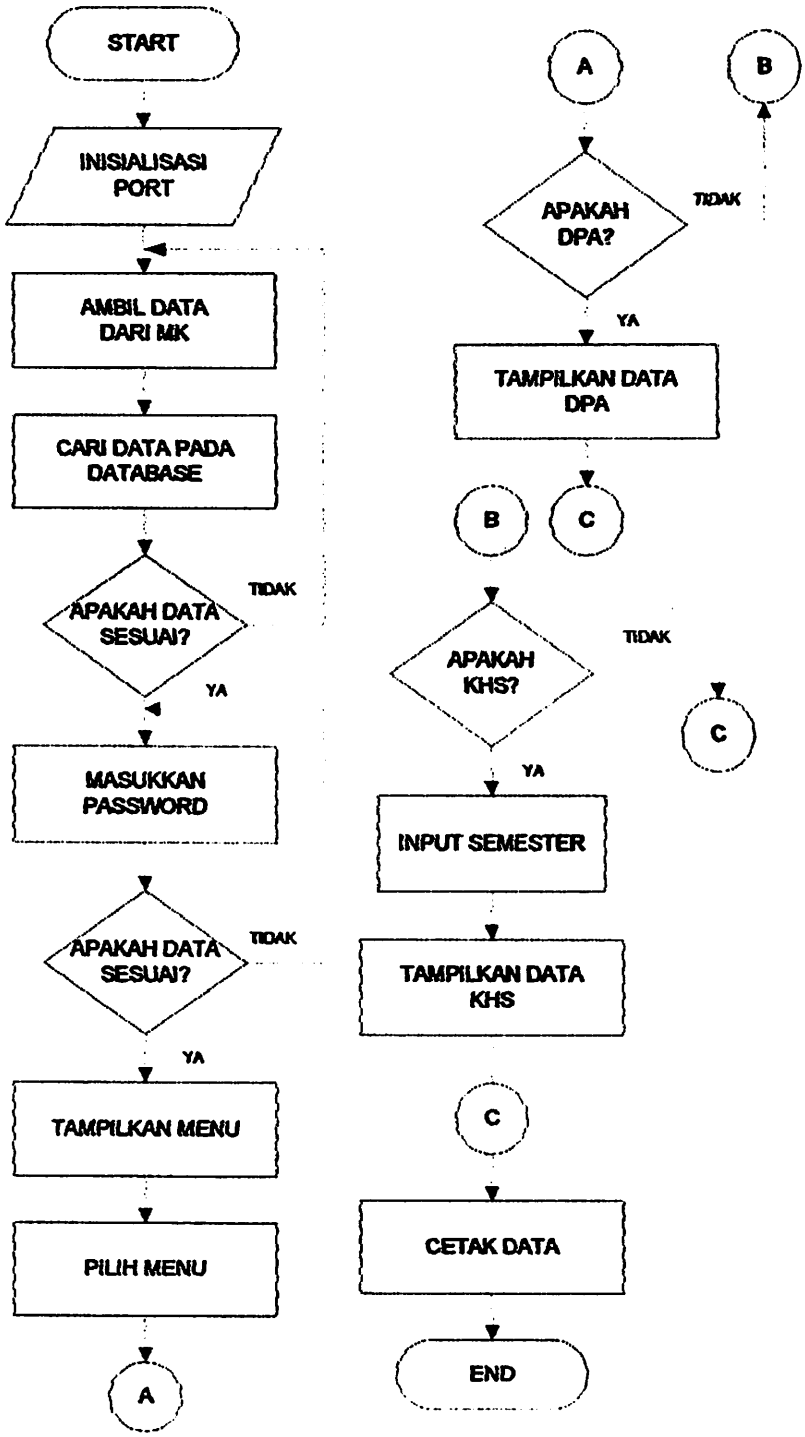
Komponen ini digunakan untuk membuat tumpukan lembar – lembar yang digunakan oleh komponen TabSet.

3.8.4 Flowchart Program Mikrokontroller



Gambar 3-8. Diagram Alir Mikrokontroller

3.8.5 Flowchart Program Komputer

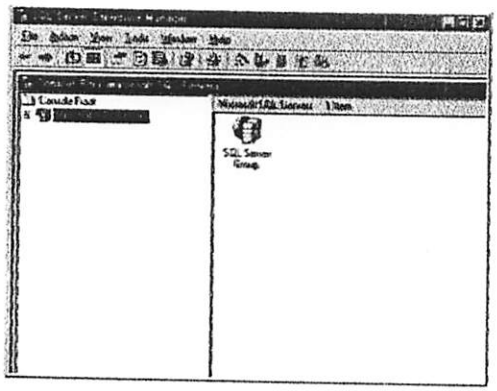


Gambar 3-9. Diagram Alir Komputer

3.8.6 Perencanaan Database

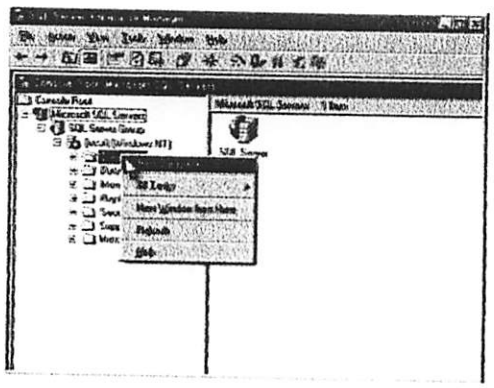
Program aplikasi yang digunakan dalam laporan skripsi ini menggunakan program delphi 7 dimana pada database menggunakan tabel dengan tipe tabel *SQL Server*. proses perencanaannya sebagai berikut:

1) Tampilkan lembar kerja SQL Server, seperti yang ditampilkan dibawah ini:



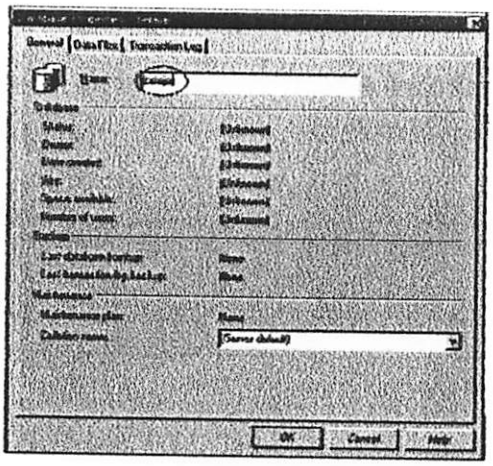
Gambar 3-10. Awal data base SQL Server

2) Click kanan pada data base dan pilih **New Database**, seperti gambar dibawah:



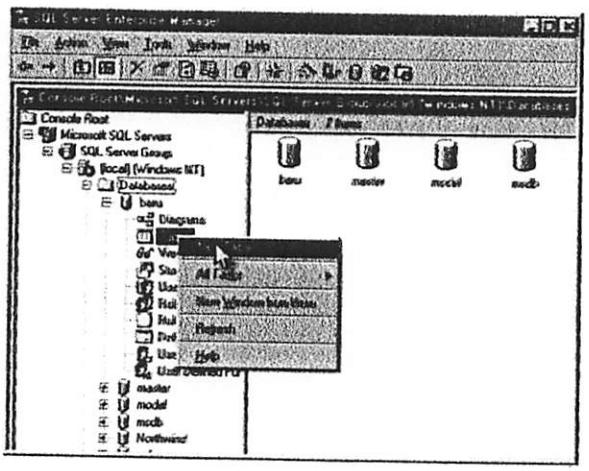
Gambar 3-11. Membuat Database baru pada SQL Server

3) Tuliskan nama dari Database yang baru misalnya **belajar**, misalnya seperti gambar dibawah:



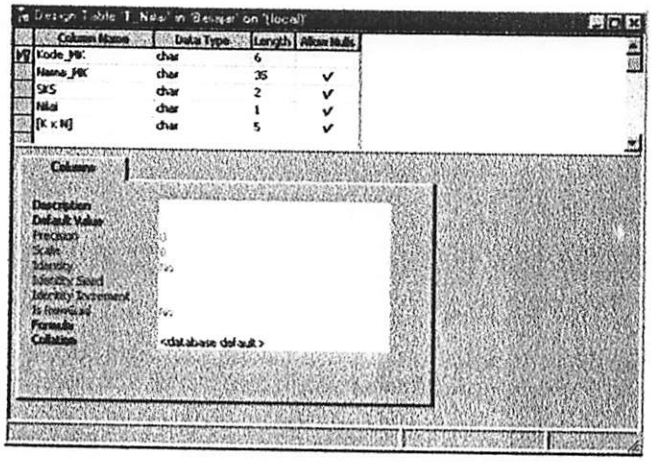
Gambar 3-12. Membuat nama dari Database yang baru

4) Pilih Table lalu click kanan kemudian pilih → **New Table,** misalnya seperti gambar dibawah:



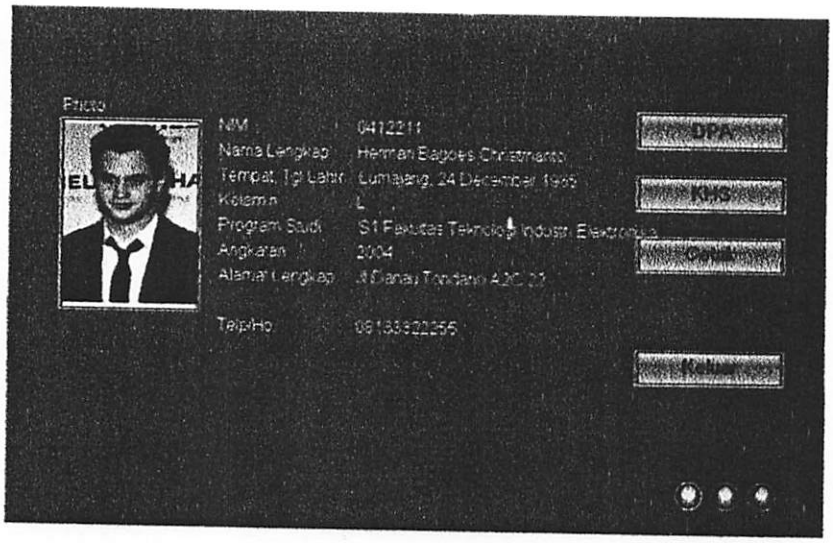
Gambar 3-13. Membuat table baru pada Database.

5) Berikut contoh tabel yang telah dibuat.



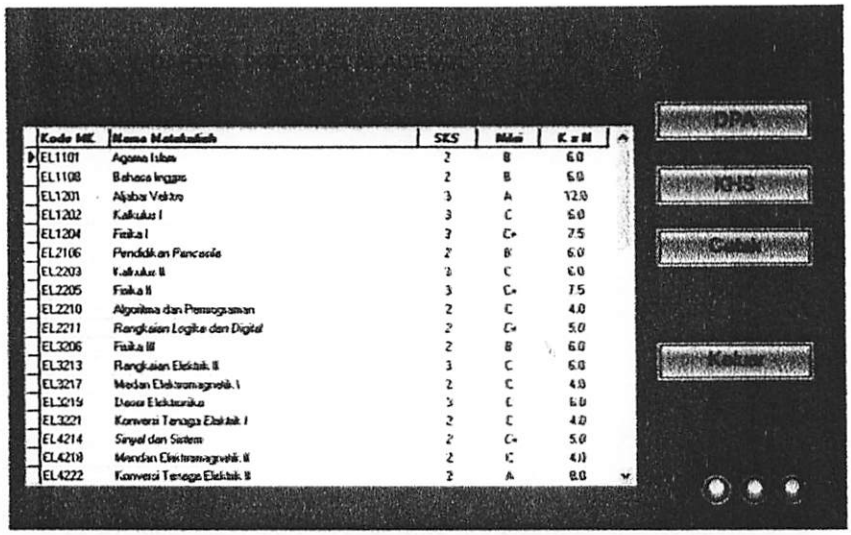
Gambar 3-14. Tampilan Database.

6) Tampilan Biodata Mahasiswa.



Gambar 3-15. Tampilan Biodata Mahasiswa.

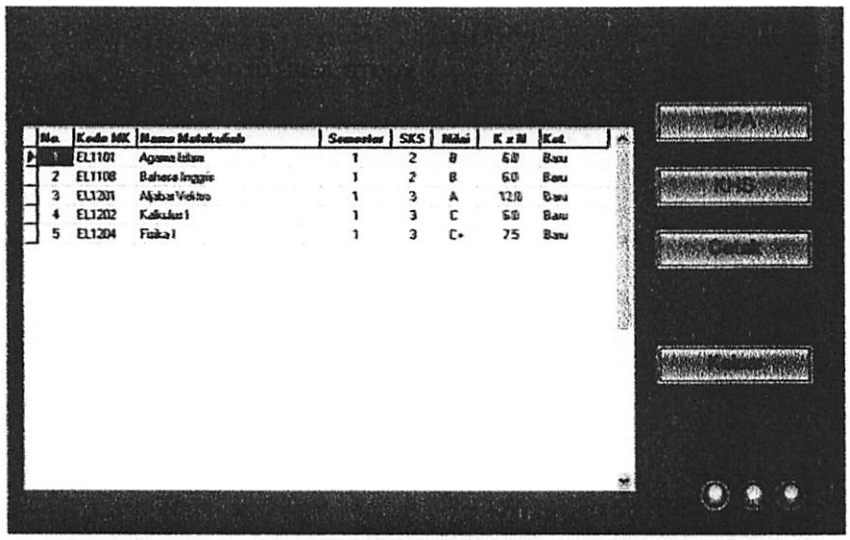
7) Tampilan Daftar Prestasi Akademik Mahasiswa



Kode MK	Nama Matakuliah	SKS	Nilai	K x N
EL1101	Agama Islam	2	B	6.0
EL1108	Bahasa Inggris	2	B	6.0
EL1201	Aljabar Vektor	3	A	12.0
EL1202	Kalkulus I	3	C	6.0
EL1204	Fisika I	3	C+	7.5
EL2106	Pendidikan Pancasila	2	B	6.0
EL2203	Kalkulus II	3	C	6.0
EL2205	Fisika II	3	C+	7.5
EL2210	Algoritma dan Pemrograman	2	C	4.0
EL2211	Rangkaian Logika dan Digital	2	C+	5.0
EL3206	Fisika III	2	B	6.0
EL3213	Rangkaian Elektrik II	3	C	6.0
EL3217	Medan Elektromagnetik I	2	C	4.0
EL3219	Dasar Elektronika	3	C	6.0
EL3221	Konversi Tenaga Elektrik I	2	C	4.0
EL4214	Sinyal dan Sistem	2	C+	5.0
EL4218	Medan Elektromagnetik II	2	C	4.0
EL4222	Konversi Tenaga Elektrik II	2	A	8.0

Gambar 3-16. Tampilan Delphi Mahasiswa.

8) Tampilan Kartu Hasil Study Mahasiswa



No.	Kode MK	Nama Matakuliah	Semester	SKS	Nilai	K x N	Ket.
1	EL1101	Agama Islam	1	2	B	6.0	Baru
2	EL1108	Bahasa Inggris	1	2	B	6.0	Baru
3	EL1201	Aljabar Vektor	1	3	A	12.0	Baru
4	EL1202	Kalkulus I	1	3	C	6.0	Baru
5	EL1204	Fisika I	1	3	C+	7.5	Baru

Gambar 3-17. Tampilan Delphi Mahasiswa.

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

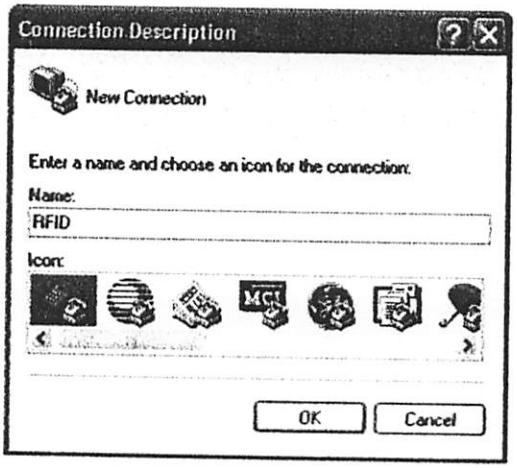
Tujuan dari pengujian ini adalah untuk mengetahui *tag* RFID bisa dibaca oleh *reader* RFID. Adapun cara pengujianya adalah dengan merangkai rangkaian RFID dan kemudian menghubungkan ke COM 1 PC. Untuk menguji *reader* bisa membaca kartu RFID dilakukan melalui *Hyper Terminal.*

4.2.2. Prosedur Pengujian

a. Menghubungkan rangkaian RFID ke COM 1 PC.

b. Membuka Hyper Terminal (Start → all program → accessories → Communication → hyperterminal).

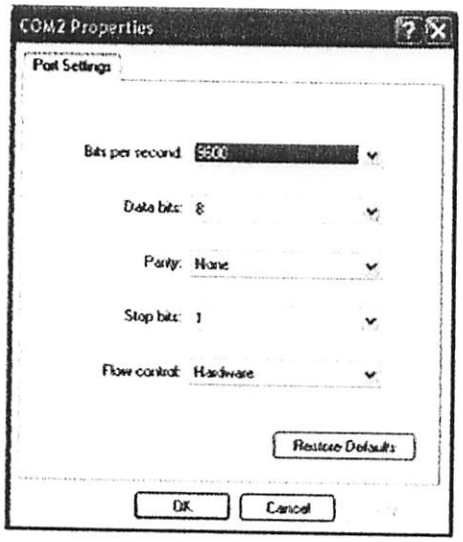
c. Memberi nama dan memilih *icon* pada *Connection Desert.*



Gambar 4-1. Kotak Dialog Connection Description

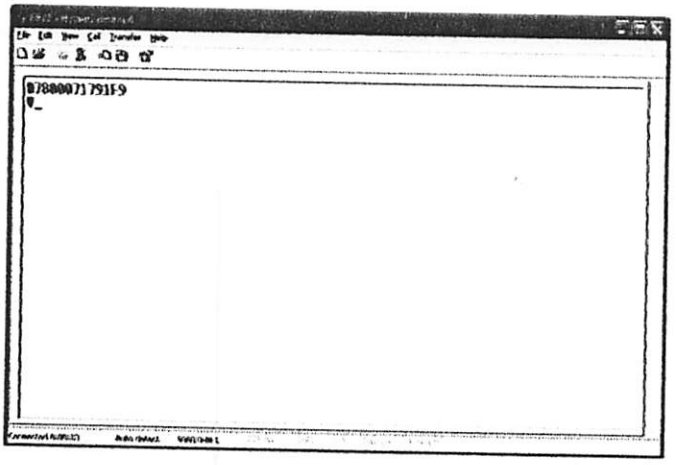
d. Memilih COM 1 pada kotak dialog *connect to*.

Gamabar 4-2. Kotak Connect To COM.



Gambar 4-3. Kotak 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										Error (%)
	1	2	3	4	5	6	7	8	9	10	
1 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
2 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
3 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
4 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
5 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
6 cm	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	0%
7 cm	✓	✓	✓	×	×	×	×	×	×	×	0%
8 cm	×	×	×	×	×	×	×	×	×	×	0%
9 cm	×	×	×	×	×	×	×	×	×	×	0%
10cm	×	×	×	×	×	×	×	×	×	×	0%

Menentukan nilai error adalah dengan menggunkan rumus di bawah ini:

$$Error = \frac{\text{Nilai Referensi} - \text{Nilai Percobaan}}{\text{Nilai Referensi}} \text{ x } 100\%$$

$$\text{Error rata-rata} = \frac{\text{Jumlah prosentase error}}{\text{Jumlah percobaan}}$$

$$= \frac{0+0+0+0-0+0+0+0-0+0}{10}$$

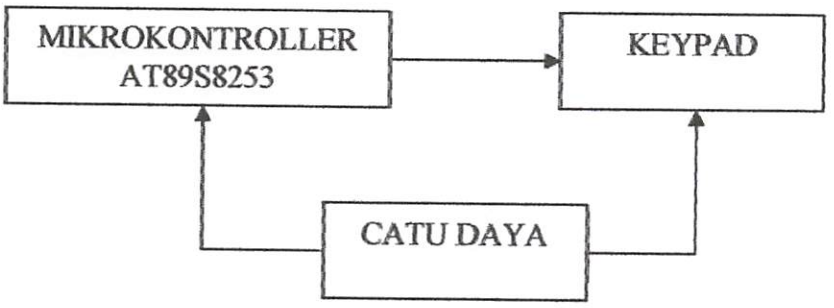
$$= \frac{0+0+0+0-0+0+0+0-0+0}{10}$$

$$= 0\,\%$$

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 KEYPAD

Tujuan dari pengujian ini adalah untuk mengetahui apakah rangkaian ini dapat mengkodekan tombol – tombol keypad yang ditekan menjadi data 4 bit yang bersesuaian , adapun procedure pengujianya yaitu:



Gambar 4-5. Blok Pengujian KEYPAD

1. Menghubungkan Mikrokontroller dengan Keypad dan diberi tegangan atau catu daya.
2. Mengisi mikrokontroller dengan program seperti di bawah ini.

❖ Program Pengujian KEYPAD

```
ORG     0000H
AJMP    MULAI
ORG     0013H
PIL0:
    CJNE  A,#00000001B,PILL1
    MOV   DATAKEY1,B
    JMP   TAMPIL

PIL1:
    CJNE  A,#00000010B,PIL2
    MOV   DATAKEY2,B
    JMP   TAMPIL

PIL2:
    CJNE  A,#00000011B,PIL3
    MOV   DATAKEY3,B
    JMP   TAMPIL
PIL3:
    CJNE  A,#00000100B,PIL4
    MOV   DATAKEY4,B
    JMP   TAMPIL
PIL4:
    CJNE  A,#00000101B,PIL5
    MOV   DATAKEY5,B
    JMP   TAMPIL
```

```
PIL2:
    CJNE A,#00000011B,PIL3
    MOV  DATAKEY3,B
    JMP  TAMPIL
PIL3:
    CJNE A,#00000100B,PIL4
    MOV  DATAKEY4,B
    JMP  TAMPIL
PIL4:
    CJNE A,#00000101B,PIL5
    MOV  DATAKEY5,B
    JMP  TAMPIL
PIL5:
   CJNE  A,#00000110B,PILL6
   MOV   DATAKEY6,B
   JMP   TAMPIL
PILL6:
   CJNE  A,#00000111B,PIL7
   MOV   DATAKEY7,B
   JMP   TAMPIL
PIL7:
   CJNE  A,#00001000B,PIL8
   MOV   DATAKEY8,B
   JMP   TAMPIL
PIL8:
   CJNE  A,#00001001B,PIL10
   MOV   DATAKEY9,B
   JMP   TAMPIL
```

```
PIL10:
    CJNE   A,#00001010B,PIL11
    MOV    DATAKEY10,B
    JMP    TAMPILL
PIL11:
    CJNE   A,#00001011B,PIL12
    MOV    DATAKEY11,B
    JMP    TAMPIL
PIL12:
    CJNE   A,#00001100B,PIL13
    MOV    DATAKEY12,B
    JMP    TAMPIL
PIL13:
    CJNE   A,#00001101B,PIL14
    MOV    DATAKEY13,B
    JMP    TAMPILL
PIL14:
    CJNE   A,#00001110B,PIL15
    MOV    DATAKEY14,B
    MOV    lokasi,#0
    JMP    TAMPIL1
PIL15:
    CJNE   A,#00001111B,PIL16
    MOV    DATAKEY15,B
    JMP    TAMPIL
PIL16:
    CJNE   A,#00010000B,PIL17
    MOV    DATAKEY16,B
    MOV    lokasi,#0
    JMP    TAMPIL1
```

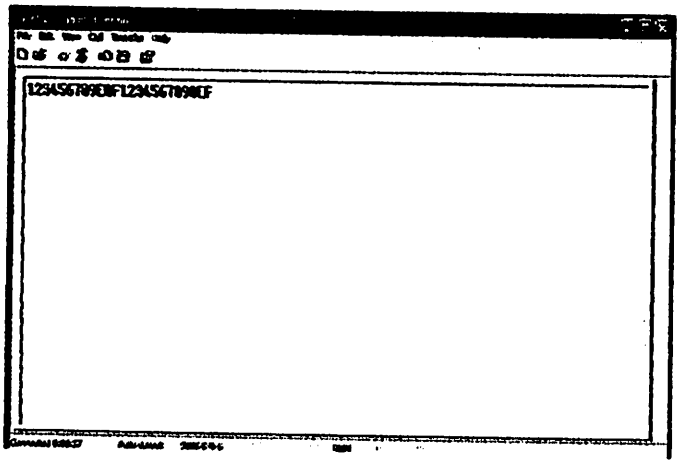
```
PIL17:
    MOV   lokasi,#0
    JMP   TAMPIL
TAMPIL1:
    JMP   TAMPILKAN_PC
TAMPIL:
    SETB  ES
RETI
```

4.3.1. Hasil Pengujian

Dari tabel dibawah dapat dilihat bahwa L3-L0 menunjukkan data keluaran yang dihasilkan setelah program dijalankan.

Tabel 4-2. Hasil Pengujian KEYPAD

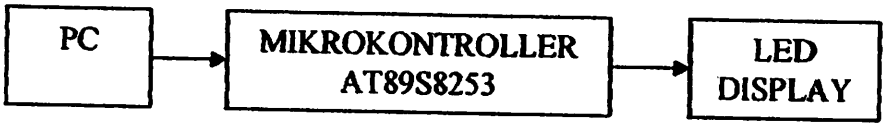
Lambang	L7	L6	L5	L4	L3	L2	L1	L0
0	*Off*	*Off*	*On*	*On*	*Off*	*Off*	*Off*	*Off*
1	*Off*	*Off*	*On*	*On*	*Off*	*Off*	*Off*	*On*
2	*Off*	*Off*	*On*	*On*	*Off*	*Off*	*On*	*Off*
3	*Off*	*Off*	*On*	*On*	*Off*	*Off*	*On*	*On*
4	*Off*	*Off*	*On*	*On*	*Off*	*On*	*Off*	*Off*
5	*Off*	*Off*	*On*	*On*	*Off*	*On*	*Off*	*On*
6	*Off*	*Off*	*On*	*On*	*Off*	*On*	*On*	*Off*
7	*Off*	*Off*	*On*	*On*	*Off*	*On*	*On*	*On*
8	*Off*	*Off*	*On*	*On*	*On*	*Off*	*Off*	*Off*
9	*Off*	*Off*	*On*	*On*	*On*	*Off*	*Off*	*On*
*	*Off*	*Off*	*On*	*On*	*On*	*Off*	*On*	*Off*
#	*Off*	*Off*	*On*	*On*	*On*	*Off*	*On*	*On*



Gambar 4-6. Tampilan Pengujian Keypad

4.4. Pengujian Sistem Mikrokontroler

Tujuan dari pengujian ini adalah Untuk mengetahui kondisi awal dari mikrokontroller apakah sudah sesuai dengan yang direncanakan.



Gambar 4-7. Diagram blok Pengujian Mikrokontroler

❖ Program Pengujian Mikrokontroller

```
        ORG     0000H
        JMP     START
START:  MOV     A,#0FH
        MOV     P1,A
        CALL    TUNDA
        MOV     A,#F0H
        MOV     P1,A
        JMP     START
TUNDA:  MOV     R3,#0FFH
TUNDA1: MOV     R2,#0FFH
```

```
DJNZ    R2,$
MOV     R1,#0FH
DJNZ    R1,$
DJNZ    R3,TUNDA1
RET
END
```

4.4.1. Hasil Pengujian

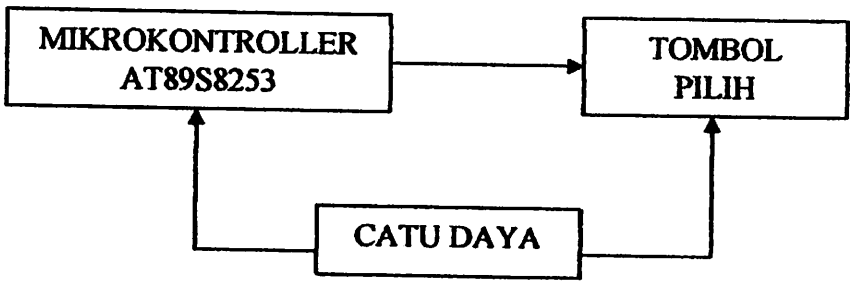
Hasil pengujian ditunjukkan dalam Tabel 4.3. terlihat bahwa *port* 1 memberikan logika 0FH dan F0H secara bergantian sesuai dengan isi program.

Tabel 4-3. Hasil Pengujian Sistem Mikrokontroler

Kondisi	Keluaran pada LED Display							
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Satu	1	1	1	1	0	0	0	0
Dua	0	0	0	0	1	1	1	1

4.5. Pengujian Sistem Tombol Pilih

Tujuan dari pengujian ini adalah untuk mengetahui apakah tombol pilih dapat berfungsi menampilkan data yang diinginkan dengan baik atau tidak, adapun procedure pengujianya yaitu:



Gambar 4-8. Diagram blok Pengujian Tombol Pilih

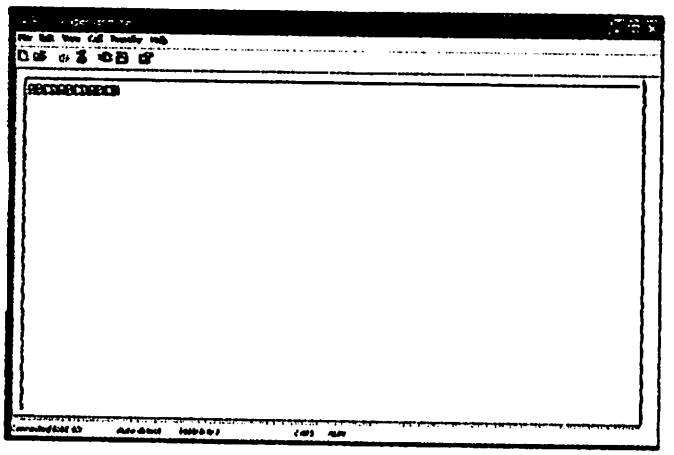
1. Menghubungkan Mikrokontroller dengan Keypad dan diberi tegangan atau catu daya.
2. Mengisi mikrokontroller dengan program seperti di bawah ini.

❖ Program Pengujian Tombol Pilih

```
ORG     0000H
AJMP    MULAI
 JNB    P0.0,kirim_data_A
 JNB    P0.1,kirim_data_B
 JNB    P0.2,kirim_data_C
 JNB    P0.3,kirim_data_D
 kirim_data_A:
        MOV A,#'A'
        CALL    DATA_M
        CALL    delay2
        JMP     data_mati
 kirim_data_B:
        MOV     A,#'B'
        CALL    DATA_M
        CALL    delay2
        JMP     data_mati
 kirim_data_C:
        MOV     A,#'C'
        CALL    DATA_M
        CALL    delay2
        JMP     data_mati
kirim_data_D:
        MOV     A,#'D'
        CALL    DATA_M
        CALL    delay2
        JMP     data_mati
```

4.5.1. Hasil Pengujian

Dari gambar dibawah dapat dilihat hasil pengujian tombol pilih yang di koneksikan ke Hyper Terminal yang dihasilkan setelah program dijalankan.

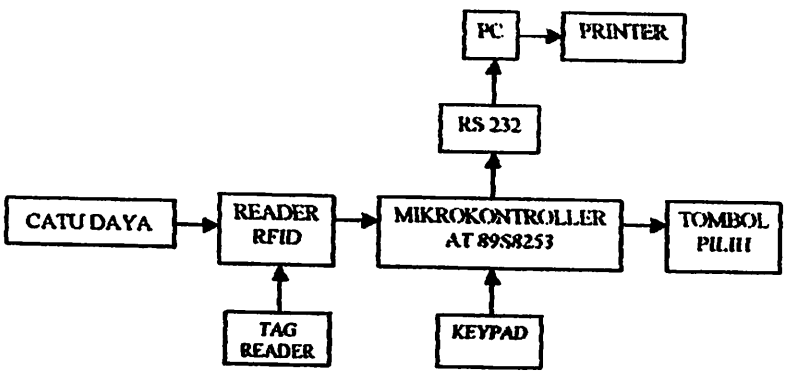


Gambar 4-9. Tampilan Pengujian Tombol Pilih

4.6. Pengujian Sistem Keseluruhan

4.6.1. Tujuan

Tujuan dari pengujian sistem ini adalah untuk mengetahui apakah alat yang dibuat dan dirancang dapat bekerja sesuai yang diiginkan dan juga untuk mengetahui *error* yang terjadi. Adapun prosedure pengujian sistem ini adalah sebagai berikut:



Gambar4-10. Blok pengujian Sistem Keseluruhan

4.6.2. Prosedur Pengujian

1 Prosedur pertama dalam melakukan pengujian alat dengan menghubungkan keseluruhan rangkaian sesuai dengan diagram blok

2 Proses kedua adalah dengan menjalankan program Delphi 7.

3 Setelah program Delphi 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.

4 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, Nim, Foto, Angkatan, Jenis Kelamin Mahasiswa, dll akan tampil di PC malalui kabel serial, tapi jika code yang dimasukan salah maka identitas Mahasiswa tidak tampil di PC malalui kabel serial.

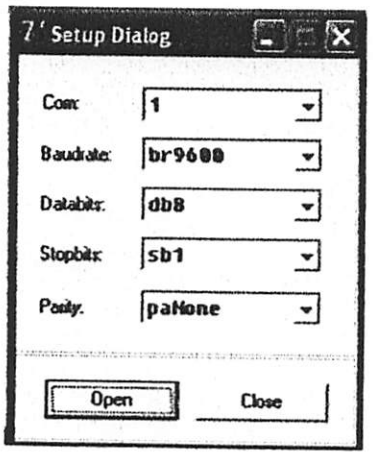
4.6.3. Hasil Pengujian

1. Tampilan Informasi Pada Program Delphi
   a. Menampilkan Form Entri Data Mahasiswa.
   b. Menampilkan Form Biodata Mahasiswa.
   c. Menampilkan Form DPA dan KHS Mahasiswa.
   d. Menampilkan Form Report.

e. Menampilkan Form Pengaturan Port dan Baud Rate

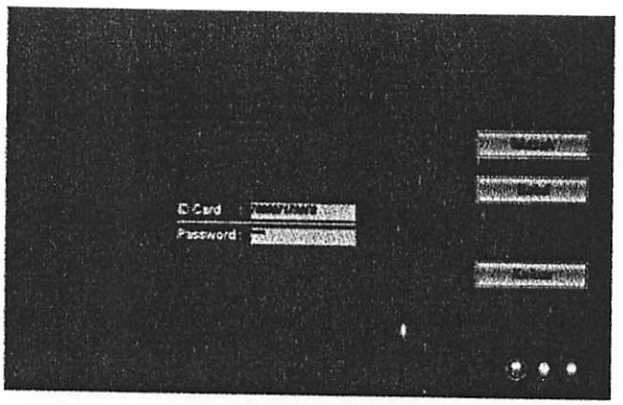
4.7. Proses Identifikasi dan Sistem Informasi Nilai Akademik Mahasiswa Berbasis RFID

a. Pada gambar 4.10. menunjukan settingan Port dan Baud Rate yang digunakan pada PC yang harus disesuaikan.



Gambar 4-11. Tampilan settingan Port dan Baud Rate

b. Setelah settingan Port dan Baud Rate selesai maka mahasiswa mescan *Tag* pada Reader dan memasukkan password. Maka di Monitor akan menampilkan seperti gambar 4.11.

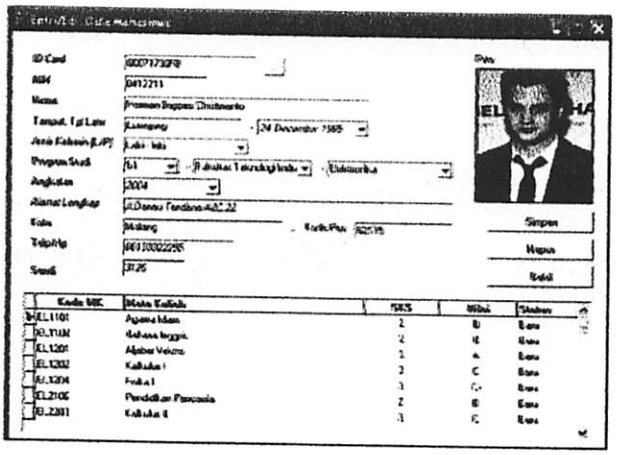


Gambar 4-12. Tampilan Monitor Setelah mahasiswa mescan *Tag*

c. Mikrokontroller akan menunggu ID dari kartu untuk dikirimkan ke PC melalui serial apakah ID tersebut terdaftar atau tidak, jika ID tersebut terdaftar maka PC mencocokan dengan database pada SQL Server. Pada layar monitor akan menampilkan ID, Nama, No telpon, Jenis Kelamin, dll seperti gambar 4.12.

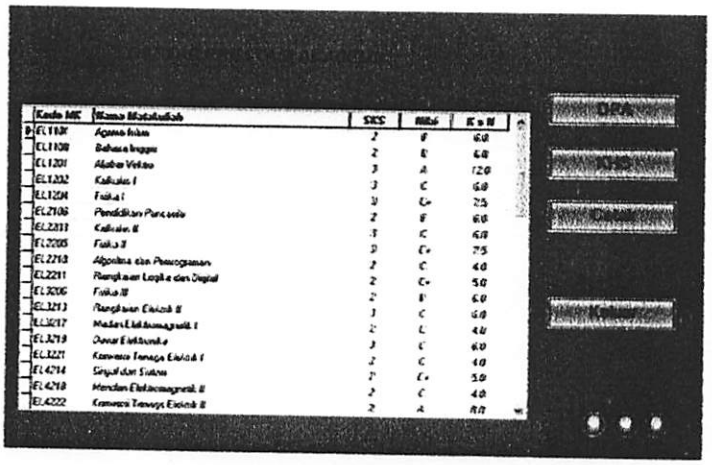
Gambar 4-13. Contoh Tampilan Identitas Mahasiswa

d. Kemudian untuk memasukkan data-data Mahasiswa, harus mengisi kode dari Tag, password, kode mata kuliah, dan nilai seperti yang ditunjukkan gambar 4.13.



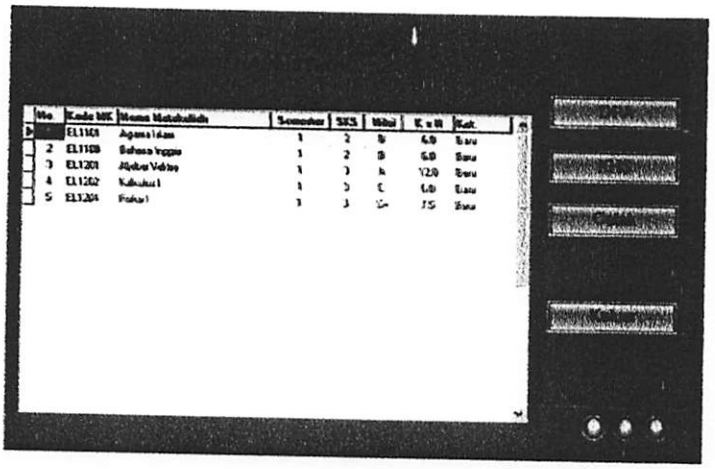
Gambar 4-14. Tampilan Membuat Data-Data Mahasiswa

e. Pada gambar 4.14. menunjukkan tampilan Daftar Prestasi Akademik setelah mahasiswa menekan tombol DPA



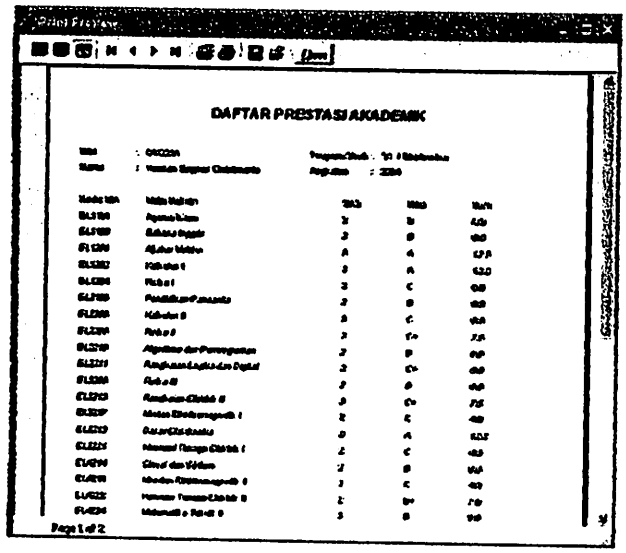
Gambar 4-15. Tampilan Daftar Prestasi Akademik Mahasiswa

f. Pada gambar 4.15. menunjukkan tampilan Kartu Hasil Study Semester setelah mahasiswa menekan tombol KHS.



Gambar 4-16. Tampilan Kartu Hasil Study Mahasiswa

g. Pada gambar 4.16. menunjukkan tampilan Print out DPA atau KHS setelah Mahasiswa menekan tombol cetak.



Gambar 4-17. Tampilan Print Out DPA/KHS Mahasiswa

4.8. Spesifikasi Alat

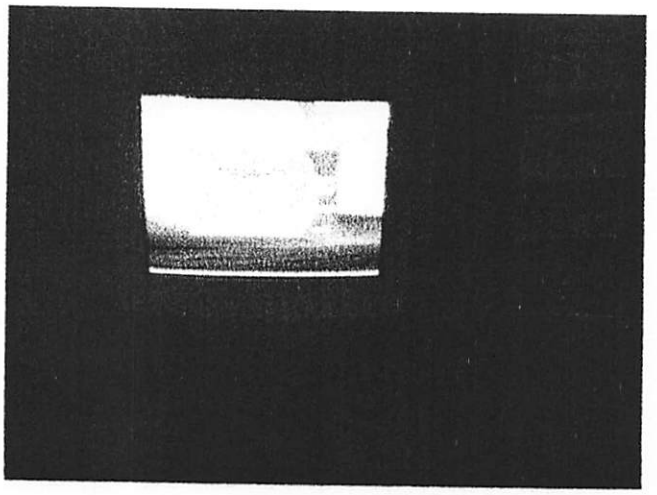
➢ **Alat Anjungan Mandiri Informasi Nilai Akademik**

- Dimensi : 18 x 13 x 5.5 cm
- Catu Daya : 5 volt
- Mikrokontroller AT89S8253
- Modul Reader RFID ID 12
- Tag RFID
- IC 74HC164
- BUZZER dan LED

Gambar 4-18. Alat Anjungan Mandiri Informasi Nilai Akademik

➢ **Alat Anjungan Mandiri Informasi Nilai Akademik keseluruhan**

- PC Celeron(R) prosesoor 2,666Hz.
- Delphi 7.
- Modul Reader RFID ID 12.
- Printer HP Deskjet 3900 Series.



Gambar 4-19. Alat Anjungan Mandiri Informasi Nilai Akademik keseluruhan

BAB V

PENUTUP

5.1. Kesimpulan

Dari pembuatan "Anjungan Mandiri Informasi Nilai Akademik Menggunkan RFID" ini maka diambil kesimpulan sebagai berikut:

1. Pada saat pengiriman software koneksi PC harus menyesuaikan kecepatan baudrate pada 9600Kbps pengaturan port yang digunakan pada PC agar dapat mensikronisasikan komunikasi data pada mikrokontroller dengan aplikasi pada PC.
2. Dari hasil pengujian, pembacaan RFID *tag* oleh Reader hanya dapat mencapai jarak 6 cm. Prensentase Error 0% dikarenakan pada waktu pengujian Tag tidak terdapat error.
3. Pada saat komunikasi dengan komputer, dibutuhkan kabel serial RS-232.

5.2. Saran

Aplikasi Anjungan Mandiri Informasi Nilai Akademik menggunakan RFID berbasis AT89S8253 ini masih banyak keterbatasan. Sehingga nantinya diharapkan dapat dikembangkan untuk mengatasi keterbatasan tersebut. Pada saat pendeteksian *Tag* pada *Reader* dilakukan satu-persatu agar data yang diolah komputer tidak valid.

DAFTAR PUSTAKA

1. *Romy Budhi Widodo,Joseph Dedy Irawan, INTERFACING PARAREL DAN SERIAL MENGGUNAKAN DELPHI, Edisi Pertama,Graha*
2. *Panduan Dasar Mikrokontroler Keluarga MCS-51*, Edisi Pertama, Innovative Electronic, 2004
3. *Data Sheet RFID ID 12.* www.alldatasheet.com.
4. *AT89S8253 Data Sheet.* www.atmel.com
5. *Application Note (AN11).* www.digi-ware.com
6. *Application Note (AN13).* http://www.delta-electronics.com

LAMPIRAN

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo KM 2
MALANG

PERSETUJUAN PERBAIKAN SKRIPSI

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

Hari : Selasa
Tanggal : 17 Maret 2009

Telah dilaksanakan Perbaikan skripsi oleh saudara :

Nama : Herman Bagoes Christmanto
N I M : 04.12.211

Perbaikan tersebut meliputi :

No	Materi Perbaikan	Paraf Dosen
1	Ditambahkan Persen Error Jarak Optimal Data Tag Ke RFID	

Malang, Maret 2009

Disetujui Oleh

Penguji II

(M Ibrahim Ashari, ST, MT)

Mengetahui

Dosen Pembimbing I

(Joseph Dedy Irawan, ST, MT)

Dosen Pembimbing II

(I Komang Somawirata, ST, MT)

INSTITUT TEKNOLOGI NASIONAL MALANG
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 / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Hermah Bagus C
N I M :
Perbaikan meliputi :

ditambahkan Persen Error.
untuk jarak optimal data bisa ditransfer
dari tag ke RFID.

Malang, 17 Maret 2009.

(M. Ibrahim, ST, MT)

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

FORMULIR BIMBINGAN SKRIPSI

Nama : Herman Bagoes Christmanto
Nim : 04.12.211
Masa Bimbingan : 31-Oktober-2008 s/d 31-April -2009
Judul Skripsi : Perencanaan dan Pembuatan Anjungan Mandiri Informasi Akademik di ITN Malang menggunakan RFID Berbasis AT 89S8253 .

No	Tanggal	Uraian	Paraf Pembimbing
1	6/08/12	Konsultasi Bab I s/d Bab III	
2	02/09/02	konsultasi Bab III, IV & V	
3	5/09/02	Hilangkan "keluar" pada delphi (Software)	
4	4/09/02	tulis pada data Tag RFID yang dikirim (software)	
5	5/09/02	Batasi untuk ngeprint setiap mahasiswa	
6	6/09/02	Kesimpulan : · Bagaimana mendapatkan baudrate 9600 tsb?	
7	7/09/02	Flowchart Mikro diperbaiki !	
8	8/09/02	ACC ICOMPRE	
9			
10			

Malang,

Dosen pembimbing I

Joseph Dedy Irawan, ST,MT
NIP.132315178

Form S-4b

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

FORMULIR BIMBINGAN SKRIPSI

Nama	:	Herman Bagoes Christmanto
Nim	:	04.12.211
Masa Bimbingan	:	31-Oktober-2008 s/d 31-April-2009
Judul Skripsi	:	Perencanaan dan Pembuatan Anjungan Mandiri Informasi Akademik di ITN Malang menggunakan RFID Berbasis AT 89S8253 .

No	Tanggal	Uraian	Paraf Pembimbing
1	17/08/12	Konsultasi Bab I s/d Bab III - Gbr tanpa grid & file [illegible] - flowchart, [illegible]	[signature]
2	02/09/02	Konsultasi Bab III, IV & V	[signature]
3	3/09/02	nilai Resistor pada alat (hardware) disamakan pada laporan bab 3	[signature]
4	4/09/02	Analisa perhitungan Rb ?	[signature]
5	5/09/02	Cara mendapatkan 9600 kbps ?	[signature]
6	6/09/02	Laporan Skripsi ACC komplit	[signature]
7			
8			
9			
10			

Malang,

Dosen pembimbing II

[signature]

I Komang Somawirata, ST,MT
NIP.Y. 1030100361

Form S-4b

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Malang, 1 Desember 2008

Nomor : ITN- 489 /7/TA /2008
Lampiran :
Perihal : ***Bimbingan Skripsi***

Kepada : Yth. Sdr. **JOSEPH DEDY IRAWAN, ST, MT**
Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

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

Nama	: **HERMAN BAGOES CHRISTMANTO**
Nim	: **0412211**
Fakultas	: **Teknologi Industri**
Jurusan	: **Teknik Elektro S-1**
Konsentrasi	: **Teknik Elektronika**

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

31 OKTOBER 2008 S/D 31 APRIL 2009

Sebagai satu syarat untuk menempuh Ujian sarjana.
Demikian atas perhatian serta kerjasama yang baik kami ucapkan terima kasih

INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI JURUSAN TEKNIK ELEKTRO STRATA-1

Ketua Jurusan
Teknik Elektro S-1

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

Tindasan:
1. Mahasiswa yang Bersangkutan
2. Arsip

Form S-4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

BNI (PERSERO) MALANG
BANK NIAGA MALANG

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

Malang, 1 Desember 2008

Nomor : ITN- 490 /7/TA /2008
Lampiran :
Perihal : *Bimbingan Skripsi*

Kepada : Yth. Sdr. **I KOMANG SOMAWIRATA, ST, MT**
Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

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

Nama	: **HERMAN BAGOES CHRISTMANTO**
Nim	: **0412211**
Fakultas	: **Teknologi Industri**
Jurusan	: **Teknik Elektro S-1**
Konsentrasi	: **Teknik Elektronika**

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

31 OKTOBER 2008 S/D 31 APRIL 2009

Sebagai satu syarat untuk menempuh Ujian sarjana.
Demikian atas perhatian serta kerjasama yang baik kami ucapkan terima kasih

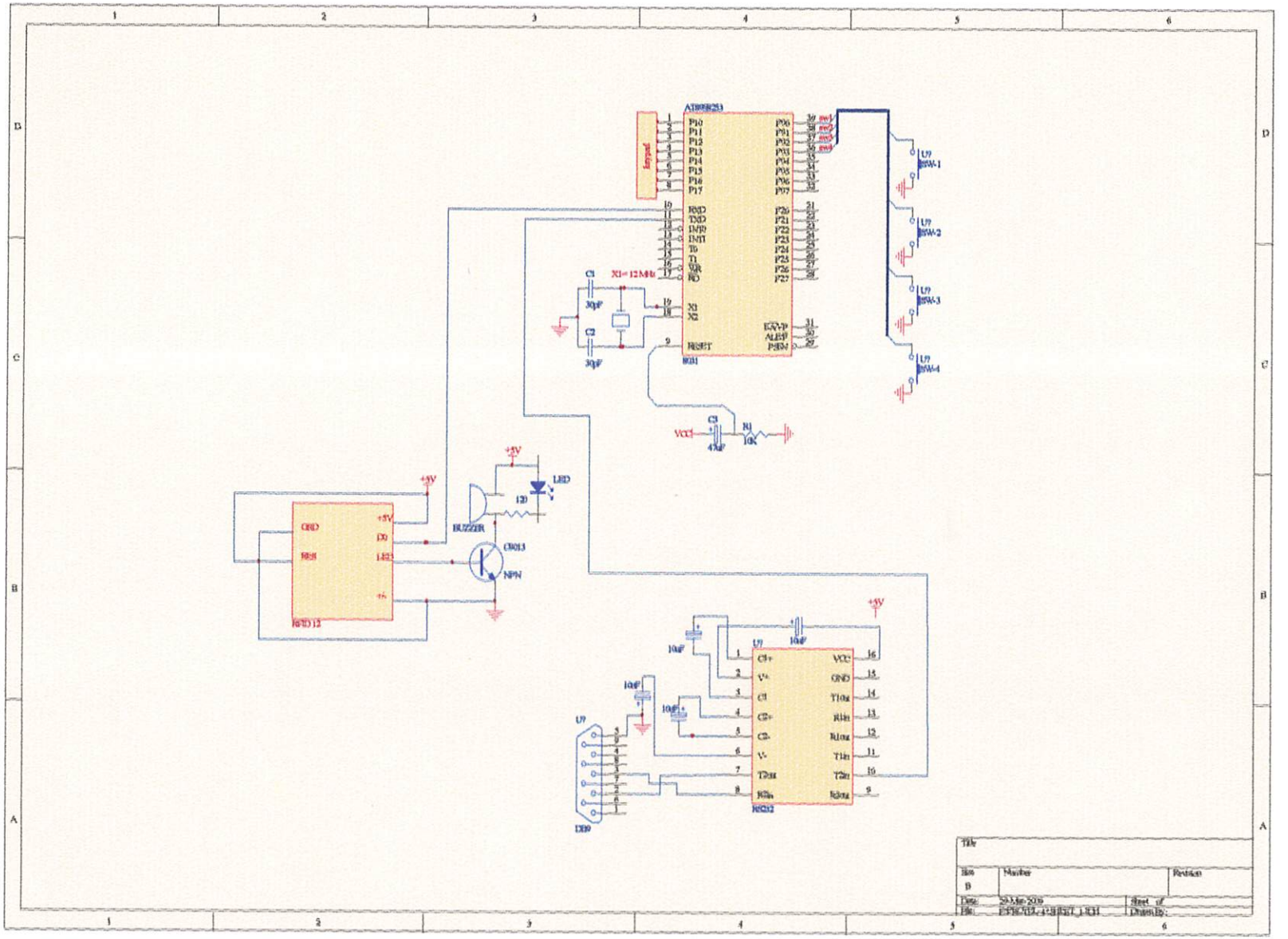
Ketua Jurusan
Teknik Elektro S-1

INSTITUT TEKNOLOGI NASIONAL MALANG FAKULTAS TEKNOLOGI INDUSTRI ITN JURUSAN TEKNIK ELEKTRO STRATA-1

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

Tindasan:
1. Mahasiswa yang Bersangkutan
2. Arsip

Form S-4a


AT89S52
Keypad
P10
P11
P12
P13
P14
P15
P16
P17
RESET
RST
X1
X2
C1
30pF
C2
30pF
XT=12MHz
C3
47uF
R1
10K
Vcc
SW-1
SW-2
SW-3
SW-4
+5V
LED
LED
BUZZER
C9013
NPN
16X2 LCD
DB9
MAX232
10uF
Title
Number
Revision
Size
B
Date:
Sheet of
File:
Drawn By:

```
unit uMaster;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ExtCtrls, jpeg, Buttons, ADODB, Grids, DBGrids,
  DBCtrls, VaClasses, VaComm, Mask, Math;

type
  TFormMaster = class(TForm)
    Image1: TImage;
    lblKeluar_Biodata: TLabel;
    lblDPA_Biodata: TLabel;
    lblKHS_Biodata: TLabel;
    lblCetak_Biodata: TLabel;
    ADOConnection1: TADOConnection;
    Query: TADOQuery;
    DSBiodata: TDataSource;
    PanelBiodata: TPanel;
    PanelLogin: TPanel;
    Image2: TImage;
    lblMasuk_Login: TLabel;
    lblBatal_Login: TLabel;
    lblKeluar_Login: TLabel;
    Label11: TLabel;
    Shape1: TShape;
    Label12: TLabel;
    Label13: TLabel;
```

Label14: TLabel;

Label7: TLabel;

Label6: TLabel;

Label5: TLabel;

PanelFoto: TPanel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

lblProdi: TLabel;

Label25: TLabel;

PanelUtama: TPanel;

Image4: TImage;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Edit2: TEdit;

DBText1: TDBText;

DBText2: TDBText;

DBText5: TDBText;

DBAngkatan: TDBText;

DBText9: TDBText;

DBText10: TDBText;

ADOQueryBiodata: TADOQuery;

lblTTL: TLabel;

PanelDPA: TPanel;

DBGridKHS: TDBGrid;

PanelKHS: TPanel;

ADOQueryKHS: TADOQuery;

ADOQueryDPA: TADOQuery;

DataSourceKHS: TDataSource;

DataSourceDPA: TDataSource;

DBGridDPA: TDBGrid;

ADOQueryKHSkode_mk: TStringField;

ADOQueryKHSsks: TBCDField;

ADOQueryKHSnilai: TStringField;

ADOQueryKHSstatus: TStringField;

ADOQueryKHSNo: TIntegerField;

ADOQueryKHSKxN: TFloatField;

ADOQueryDPAkode_mk: TStringField;

ADOQueryDPAnama_1: TStringField;

ADOQueryDPAsks: TBCDField;

ADOQueryDPAnilai: TStringField;

ADOQueryDPAKxN: TFloatField;

ADOQueryKHSsemester: TBCDField;

ADOQueryKHSnama: TStringField;

DBImage1: TDBImage;

VaComm1: TVaComm;

lbData: TLabel;

Edit1: TEdit;

Timer1: TTimer;

ADOQueryBiodataNIM: TStringField;

ADOQueryBiodataNama: TStringField;

```
ADOQueryBiodataTmp_Lahir: TStringField;
ADOQueryBiodataTgl_Lahir: TDateTimeField;
ADOQueryBiodataKelamin: TStringField;
ADOQueryBiodataAlamat: TStringField;
ADOQueryBiodataKota: TStringField;
ADOQueryBiodataKode_Pos: TBCDField;
ADOQueryBiodataTelpHP: TStringField;
ADOQueryBiodatafoto: TBlobField;
ADOQueryBiodataKode_Jen: TStringField;
ADOQueryBiodataAngkatan: TBCDField;
ADOQueryBiodataNama_Jur: TStringField;
ADOQueryBiodataNama_Fak: TStringField;
lblDPA: TLabel;
lblKHS: TLabel;
Timer2: TTimer;
lblEror: TLabel;
lblPrior: TLabel;
lblNext: TLabel;
procedure FormShow(Sender: TObject);
procedure lblDPA_BiodataClick(Sender: TObject);
procedure lblKHS_BiodataClick(Sender: TObject);
procedure lblCetak_BiodataClick(Sender: TObject);
procedure lblMasuk_LoginClick(Sender: TObject);
procedure lblKeluar_LoginClick(Sender: TObject);
procedure Label26Click(Sender: TObject);
procedure lblBatal_LoginClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Label28Click(Sender: TObject);
procedure Label27Click(Sender: TObject);
```

```
    procedure ADOQueryKHSCalcFields(DataSet: TDataSet);
    procedure ADOQueryDPACalcFields(DataSet: TDataSet);
    procedure ComboBox1Change(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure lblKeluar_BiodataClick(Sender: TObject);
    procedure VaComm1RxChar(Sender: TObject; Count: Integer);
    procedure Label29Click(Sender: TObject);
    procedure lbDataClick(Sender: TObject);
    procedure lblPriorClick(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure lblNextClick(Sender: TObject);
  private
    { Private declarations }
    i: integer;
    //pesan: string;
  public
    { Public declarations }
    pesan: string;
    Str, nim, sks : string;
    pilih    : integer; { 1: ID,
                          2: Sandi,
                          3: SKS}
  end;

const
  labelA = 'A';
  labelB = 'B';
```

```
  labelC = 'C';
  labelD = 'D';
  labelNext = 'E';
  labelPrior = 'F';

var
  FormMaster: TFormMaster;

implementation

uses uVaSetup, uData, uMK, uReport, uReportKHS, uBiodata, uSKS;

{$R *.dfm}

procedure TFormMaster.FormShow(Sender: TObject);
begin
  VaComm1.Open;
  FrmMain.Hide;
  with ADOConnection1 do
  begin
    ConnectionString:= 'Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist '+
                'Security Info=False;Initial Catalog=RFID;Data Source=(local)';
    Connected:= True;
  end;

  Pilih:= 1;
  i:= 0;
  PanelBiodata.Visible := False;
  PanelBiodata.Align   := alClient;
```

```
  PanelUtama.Visible  := False;
  PanelUtama.Align    := alClient;
  PanelLogin.Visible  := True;
  PanelLogin.Align    := alClient;
  PanelDPA.Top        := 112;
  PanelDPA.Left       := 16;
  PanelDPA.Width      := 577;
  PanelDPA.Height     := 345;
  PanelKHS.Top        := 112;
  PanelKHS.Left       := 16;
  PanelKHS.Width      := 577;
  PanelKHS.Height     := 345;
  PanelDPA.Visible    := False;
  PanelKHS.Visible    := False;
  Edit1.SetFocus;
end;

procedure TFormMaster.lblDPA_BiodataClick(Sender: TObject);
begin
  PanelBiodata.Visible := False;
  PanelLogin.Visible  := False;
  PanelUtama.Visible  := True;
  PanelDPA.Visible    := True;
  Label28Click(Sender);
end;

procedure TFormMaster.lblKHS_BiodataClick(Sender: TObject);
begin
  PanelBiodata.Visible := False;
```

```
  PanelLogin.Visible  := False;
  PanelUtama.Visible  := True;
  PanelKHS.Visible    := True;
  Label27Click(Sender);
end;

procedure TFormMaster.lblCetak_BiodataClick(Sender: TObject);
begin
  FormBiodata.QuickRep1.Print;
end;

procedure TFormMaster.lblMasuk_LoginClick(Sender: TObject);
var
  pass, kode : string;
begin
  Query.Close;
  Query.SQL.Text := 'SELECT Kode, Pass, nim FROM Mahasiswa WHERE Kode =
'+QuotedStr(Edit1.Text);
  Query.Open;

  if Query.RecordCount > 0 then
  begin
    nim:= Query.FieldByName('NIM').AsString;
    kode := Query.FieldByName('Kode').AsString;
    pass:= Query.fieldByName('Pass').AsString;

    if pass = trim(Edit2.Text) then
    begin
      with ADOQueryBiodata do
```

```
begin
 Close;
 SQL.Text := 'SELECT M.NIM, M.Nama, M.Tmp_Lahir, M.Tgl_Lahir, '+
             'M.Kelamin, M.Alamat, M.Kota, M.Kode_Pos, M.[Telp/HP], M.foto, '+
             'MM.* FROM Mahasiswa M, (Select M.Kode_Jen, M.Angkatan, '+
             'J.Nama_Jur, F.Nama_Fak FROM MHS_Jurusan M, Jurusan J, Fakultas F '+
             'WHERE M.Kode_Jur = J.Kode_jur AND M.Kode_Fak = '+
             'F.Kode_Fak) AS MM WHERE M.NIM = '+QuotedStr(Trim(nim));

 Open;

 if RecordCount > 0 then
 begin
  PanelBiodata.Visible  := True;
  PanelLogin.Visible    := False;
  PanelUtama.Visible    := False;
  lblProdi.Caption      := Trim(ADOQueryBiodataKode_Jen.Value)+' '+
                           Trim(ADOQueryBiodataNama_Fak.Value)+' '+
                           Trim(ADOQueryBiodataNama_Jur.Value);
  lblTTL.Caption        := Trim(ADOQueryBiodataTmp_Lahir.Value)+', '+
                           FormatDateTime('dd mmmm yyyy', ADOQueryBiodataTgl_Lahir.Value);
  lblEror.Visible:= False;
  Timer2.Enabled:= False;
 end else
 begin
  lblEror.Visible:= True;
  Timer2.Enabled:= True;
 end;
end;
```

```
    end else
    begin
      lblEror.Visible:= True;
      Timer2.Enabled:= True;
    end;
  end else
  begin
    lblEror.Visible:= True;
    Timer2.Enabled:= True;
  end;
end;

procedure TFormMaster.lblKeluar_LoginClick(Sender: TObject);
begin
  Close;
end;

procedure TFormMaster.Label26Click(Sender: TObject);
begin
  PanelUtama.Visible:= False;
  PanelBiodata.Visible := True;
end;

procedure TFormMaster.lblBatal_LoginClick(Sender: TObject);
begin
  pesan := '';
  Edit2.Text:= '';
end;
```

```
procedure TFormMaster.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Query.Active           := False;
  ADOConnection1.Connected := False;
  frmMain.Close;
end;

procedure TFormMaster.Label28Click(Sender: TObject);
begin
  PanelDPA.Visible := True;
  PanelKHS.Visible := False;
  lblDPA.Visible   := True;
  lblKHS.Visible   := False;

  with ADOQueryDPA do
  begin
    Close;
    SQL.Text := 'SELECT m.kode_mk, m.nama, m.sks, am.nilai '+
                'FROM matakuliah m, ambil_mk am '+
                'WHERE m.kode_mk = am.kode_mk '+
          'AND am.NIM = '+QuotedStr(trim(NIM));
    Open;
  end;
end;

procedure TFormMaster.Label27Click(Sender: TObject);
begin
  FormSKS.ShowModal;
end;
```

```
procedure TFormMaster.ADOQueryKHSCalcFields(DataSet: TDataSet);
begin
 if trim(ADOQueryKHSNilai.AsString) = 'A' then
  ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsInteger * 4;

 if trim(ADOQueryKHSNilai.AsString) = 'B' then
  ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsInteger * 3;

 if trim(ADOQueryKHSNilai.AsString) = 'C' then
  ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 2;

 if ADOQueryKHSNilai.AsString = 'B+' then
  ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 3.5;

 if ADOQueryKHSNilai.AsString = 'C+' then
  ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 2.5;

 ADOQueryKHSNo.Value := ADOQueryKHS.RecNo;
 if ADOQueryKHSNo.Value = -1 then
  ADOQueryKHSNo.Value:= 1;
end;

procedure TFormMaster.ADOQueryDPACalcFields(DataSet: TDataSet);
begin
 if trim(ADOQueryDPANilai.AsString) = 'A' then
  ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsInteger * 4;

 if trim(ADOQueryDPANilai.AsString) = 'B' then
```

```
  ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsInteger * 3;

 if trim(ADOQueryDPANilai.AsString) = 'C' then
  ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 2;

 if ADOQueryDPANilai.AsString = 'B+' then
  ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 3.5;

 if ADOQueryDPANilai.AsString = 'C+' then
  ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 2.5;
end;

procedure TFormMaster.ComboBox1Change(Sender: TObject);
begin
 Label27Click(Sender);
end;

procedure TFormMaster.Button6Click(Sender: TObject);
begin
 if (lblMasuk_Login.Transparent = False) or (i = 0) then
  lblMasuk_Login.OnClick(Sender) else
 if  lblBatal_Login.Transparent = False then
  lblBatal_Login.OnClick(Sender) else
 if lblKeluar_Login.Transparent = False then
  lblKeluar_Login.OnClick(Sender);
end;

procedure TFormMaster.lblKeluar_BiodataClick(Sender: TObject);
begin
```

```
  PanelLogin.Visible:= True;
  PanelBiodata.Visible := False;
  pesan := '';
  Pilih := 1;
  Edit1.Text:= '';
  Edit2.Text:= '';
  edit1.SetFocus;
  Timer1.Enabled:= True;
end;

procedure TFormMaster.VaComm1RxChar(Sender: TObject; Count: Integer);
var
  i: integer;
  tmp : string;
begin
  tmp:= VaComm1.ReadText;
  sleep(500);

  if labelA = trim(tmp) then
  begin
    if PanelLogin.Visible = true  then
      lblMasuk_LoginClick(Sender) else
    if PanelBiodata.Visible = true then
      lblDPA_BiodataClick(Sender) else
    if PanelUtama.Visible = True then
      Label28Click(Sender);
  end
  else if labelB = trim(tmp) then
  begin
```

```
  if PanelLogin.Visible = true  then
    lblBatal_Loginclick(Sender) else
  if PanelBiodata.Visible = true then
    lblKHS_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label27Click(Sender);
 end
 else if labelC = trim(tmp) then
 begin
  if PanelBiodata.Visible = true then
    lblCetak_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label29Click(Sender);
 end
 else if labelD = trim(tmp) then
 begin
  {if PanelLogin.Visible = true then
    lblKeluar_LoginClick(Sender) else}
  if PanelBiodata.Visible = true then
    lblKeluar_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label26Click(Sender);
 end
 else if labelNext = trim(tmp) then
 begin
  if PanelUtama.Visible = True then
    lblPriorClick(Sender);
 end
 else if labelPrior = trim(tmp) then
```

```
  begin
    if PanelUtama.Visible = True then
      lblNextClick(Sender);
  end
  else
  begin
    //baca data yang diterima oleh comport
    for l := 1 to Length(Tmp) do
      case Tmp[l] of
        #10:;                    //lewatkan
        #13:
      else                       //bukan #10 atau #13
        pesan := pesan + Tmp[l];
        case pilih of  //if Pilih = 1 then
          1: Edit1.Text := Trim(Pesan);
        //else if pilih = 2 then
          2: Edit2.Text := Trim(Pesan);
        end;
    end;
  end;
end;

procedure TFormMaster.Label29Click(Sender: TObject);
begin
  if PanelDPA.Visible then
    FormDPA.QuickRep1.Print
  else
    FormKHS.QuickRep1.Print;
end;
```

```
procedure TFormMaster.lbDataClick(Sender: TObject);
begin
  FormData.ShowModal;
end;

procedure TFormMaster.lblPriorClick(Sender: TObject);
begin
  ADOQueryDPA.Prior;
end;

procedure TFormMaster.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    Edit2.SetFocus;
end;

procedure TFormMaster.Timer1Timer(Sender: TObject);
var
  str: string;
begin
  if length(edit1.Text) > 10 then
  begin
    str:= edit1.Text;
    Delete(str, 1, 1);
    edit1.Text:= str;
  end else
  if length(edit1.Text) = 10 then
  begin
```

```
    pesan := '';
    Pilih := 2;
    edit2.SetFocus;
    Timer1.Enabled:= False;
  end;
end;

procedure TFormMaster.Timer2Timer(Sender: TObject);
begin
  if lblEror.Font.Color = clRed then
    lblEror.Font.Color:= clBlue else lblEror.Font.Color:= clRed;
end;

procedure TFormMaster.lblNextClick(Sender: TObject);
begin
  ADOQueryDPA.Next;
end;

end.
```

```
; LCD CONSTANTA
DISPCLR  EQU 00000001B
BLINK    EQU 00001101B
ENTRMOD  EQU 00000110B
DISPON   EQU 00001100B
CURSOR   EQU 00011100B
FUNCSET  EQU 00111000B
;
;DEVICE ADDRESS [LCD]
RS       BIT P2.6  ;LCD
E        BIT P2.7  ;LCD
MY_DATA      BIT P3.4
MY_COUNTER EQU 31H
MY_TEMP      EQU 32H
KEYDATA      EQU 33H
COD1   EQU 27H
COD2   EQU 28H
COD3   EQU 29H
DATAKEY0 EQU 34H
DATAKEY1 EQU 35H
DATAKEY2 EQU 36H
DATAKEY3 EQU 37H
DATAKEY4 EQU 38H
DATAKEY5 EQU 39H
DATAKEY6 EQU 40H
DATAKEY7 EQU 41H
DATAKEY8 EQU 42H
DATAKEY9 EQU 43H
DATAKEY10 EQU 44H
```

```
DATAKEY11 EQU 45H
DATAKEY12 EQU 46H
LOKASI    EQU 47H
DATAKEY13 EQU 48H
DATAKEY14 EQU 49H
DATAKEY15 EQU 50H
DATAKEY16 EQU 51H
DATAKEY17 EQU 52H
DATAKEY18 EQU 53H
DATAKEY19 EQU 54H
DATAKEY20 EQU 55H
DATAKEY21 EQU 56H
DATAKEY22 EQU 57H
DATAKEY23 EQU 58H
DATAKEY24 EQU 59H
DATAKEY25 EQU 60H
LOKASI1 EQU 61H
keybounc  equ   62H
DATAKEY26 EQU 63H
kolom1    bit   P1.4   ; kiri (1,4,7,redial)
kolom2    bit   P1.5
kolom3    bit   P1.6
baris1    bit   P1.0   ; atas (1,2,3)
baris2    bit   P1.1
baris3    bit   P1.2
baris4    bit   P1.3
;
keyport   equ   P1
      ORG 0000H       ; Mulai dari alamat reset
```

```
        AJMP   MULAI            ; Loncat ke label mulai
        ORG    0013H            ; Alamat interupt
        ORG    23H              ; alamat untuk serial comm. interupt
        CLR    ES               ; matikan interupt serial
        JNB    RI,$             ; tunggu sampai selesai menerima
        CLR    RI               ; clear receive interupt flag
        MOV    R1,SBUF
        MOV    B,R1
        MOV    A,R1
        MOV    A,LOKASI
        INC    A
        MOV    LOKASI,A
PIL0:
        CJNE   A,#00000001B,PILL1
        MOV    DATAKEY1,B
        JMP TAMPILL
PILL1:
        CJNE   A,#00000010B,PIL2
        MOV    DATAKEY2,B
        JMP TAMPILL
PIL2:
        CJNE   A,#00000011B,PIL3
        MOV    DATAKEY3,B
        JMP TAMPILL
PIL3:
        CJNE   A,#00000100B,PIL4
        MOV    DATAKEY4,B
        JMP TAMPILL
PIL4:
```

```
        CJNE   A,#00000101B,PIL5
        MOV    DATAKEY5,B
        JMP TAMPILL
PIL5:
        CJNE   A,#00000110B,PILL6
        MOV    DATAKEY6,B
        JMP TAMPILL
PILL6:
        CJNE   A,#00000111B,PIL7
        MOV    DATAKEY7,B
        JMP TAMPILL
PIL7:
        CJNE   A,#00001000B,PIL8
        MOV    DATAKEY8,B
        JMP TAMPILL
PIL8:
        CJNE   A,#00001001B,PIL10
        MOV    DATAKEY9,B
        JMP TAMPILL
PIL10:
        CJNE   A,#00001010B,PIL11
        MOV    DATAKEY10,B
        JMP TAMPILL
PIL11:
        CJNE   A,#00001011B,PIL12
        MOV    DATAKEY11,B
        JMP TAMPILL
PIL12:
        CJNE   A,#00001100B,PIL13
```

```
        MOV   DATAKEY12,B
        JMP TAMPILL
PIL13:
        CJNE   A,#00001101B,PIL14
        MOV   DATAKEY13,B
        JMP TAMPILL
PIL14:
        CJNE   A,#00001110B,PIL15
        MOV   DATAKEY14,B
        MOV   lokasi,#0
        JMP TAMPILL
PIL15:
        CJNE   A,#00001111B,PIL16
        MOV   DATAKEY15,B
        JMP TAMPILL
PIL16:
        CJNE   A,#00010000B,PIL17
        MOV   DATAKEY16,B
        MOV   lokasi,#0
        JMP TAMPILL1
PIL17:
        MOV   lokasi,#0
        JMP TAMPILL
TAMPILL1:
        JMP    TAMPILKAN_PC
TAMPILL:
        SETB  ES
        RETI
MULAI:
```

```
        CALL  init_serial
        JMP   oke
TUNDA_T:
        MOV   R7,#5
LOOP10:     MOV   R6,#5
LOOP9:      MOV   R5,#25
        DJNZ  R5,$
        DJNZ  R6,LOOP9
        DJNZ  R7,LOOP10
        RET
TUNDA_TAM:
        MOV   R7,#100
LOP10: MOV   R6,#100
LOP9:  MOV   R5,#100
        DJNZ  R5,$
        DJNZ  R6,LOP9
        DJNZ  R7,LOP10
        RET
init_serial:
        MOV   SCON,#50H
        MOV   TMOD,#20H
        MOV   TL1,#0FDH
        MOV   TH1,#0FDH
        SETB  TR1
        SETB  EA            ; enable interupt
        SETB  ES            ; enable serial interupt
        RET
OKE:
        MOV   DATAKEY0,#'0'
```

```
        MOV   DATAKEY1,#'0'
        MOV   DATAKEY2,#'0'
        MOV   DATAKEY3,#'0'
        MOV   DATAKEY4,#'0'
        MOV   DATAKEY5,#'0'
        MOV   DATAKEY6,#'0'
        MOV   DATAKEY7,#'0'
        MOV   DATAKEY8,#'0'
        MOV   DATAKEY9,#'0'
        MOV   DATAKEY10,#'0'
        MOV   DATAKEY11,#'0'
        MOV   DATAKEY12,#'0'
        MOV   LOKASI,#0
        MOV   DATAKEY13,#'0'
        MOV   DATAKEY14,#'0'
        MOV   DATAKEY15,#'0'
        MOV   DATAKEY16,#'0'
        MOV   DATAKEY26,#0
data_mati:
KEYPAD:
        JNB   P0.0,kirim_data_A
        JNB   P0.1,kirim_data_B
        JNB   P0.2,kirim_data_C
        JNB   P0.3,kirim_data_D
;*************************************

KEYLOOP:
ulang:  call  Keypad3x4
        mov   A,keydata
```

```
        cjne    A,#0FFH,tampil
        jmp     KEYPAD
kirim_data_A:
        MOV A,#'A'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_B:
        MOV A,#'B'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_C:
        MOV A,#'C'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_D:
        MOV A,#'D'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
TAMPIL:
        MOV A,keydata
        CALL  DATA_M
        CALL  delay1
        JMP     KEYPAD
Keypad3x4:
        mov     keybounc,#50
```

```
        mov     keyport,#0FFh
        clr     kolom1
ul1:    jb      baris1,key1
        djnz    keybounc,ul1
        mov     keydata,#'1'
        ret
key1:   jb      baris2,key2
        djnz    keybounc,key1
        mov     keydata,#'4'
        ret
key2:   jb      baris3,key3
        djnz    keybounc,key2
        mov     keydata,#'7'
        ret
key3:   jb      baris4,key4
        djnz    keybounc,key3
        mov     keydata,#'E'
        ret
key4:   setb    kolom1
        clr     kolom2
        jb      baris1,key5
        djnz    keybounc,key4
        mov     keydata,#'2'
        ret
key5:   jb      baris2,key6
        djnz    keybounc,key5
        mov     keydata,#'5'
        ret
key6:   jb      baris3,key7
```

```
        djnz    keybounc,key6
        mov     keydata,#'8'
        ret
key7:   jb      baris4,key8
        djnz    keybounc,key7
        mov     keydata,#'0'
        ret
key8:   setb    kolom2
        clr     kolom3
        jb      baris1,key9
        djnz    keybounc,key8
        mov     keydata,#'3'
        ret
key9:   jb      baris2,key10
        djnz    keybounc,key9
        mov     keydata,#'6'
        ret
key10:  jb      baris3,key11
        djnz    keybounc,key10
        mov     keydata,#'9'
        ret

key11:  jb      baris4,key12
        djnz    keybounc,key11
        mov     keydata,#'F'
        ret
key12:  mov     keydata,#0FFh
        ret
;*************************************
```

```
TAMPILKAN_PC:
        ;MOV A,DATAKEY1
        ;CALL DATA_M
        ;MOV A,DATAKEY2
        ;CALL DATA_M
        ;MOV A,DATAKEY3
        ;CALL DATA_M
        MOV A,DATAKEY4
        CALL  DATA_M
        MOV A,DATAKEY5
        CALL  DATA_M
        MOV A,DATAKEY6
        CALL  DATA_M
        MOV A,DATAKEY7
        CALL  DATA_M
        MOV A,DATAKEY8
        CALL  DATA_M
        MOV A,DATAKEY9
        CALL  DATA_M
        MOV A,DATAKEY10
        CALL  DATA_M
        MOV A,DATAKEY11
        CALL  DATA_M
        MOV A,DATAKEY12
        CALL  DATA_M
        MOV A,DATAKEY13
        CALL  DATA_M
        MOV A,DATAKEY14
        CALL  DATA_M
```

```
        ;MOV A,DATAKEY15
        ;CALL DATA_M
        ;MOV A,DATAKEY16
        ;CALL DATA_M
        CALL  delay2
        JMP TAMPILL
DATA_M:
        CLR   ES      ; matikan serial interupt saat mengirim
        MOV   SBUF,A          ; isi serial buffer dengan data yg dikirim
        JNB   TI,$    ; tunggu pengiriman selesai
        CLR   TI      ; clear transmit interupt flag
        MOV   50H,#50
TUNDAL1:
        MOV   51H,#50
        DJNZ  51H,$
        DJNZ  50H,TUNDAL1
        SETB  ES      ; hidupkan kembali serial interupt
        SETB  EA
        RET
;
delay:
        MOV   R1,#50
del1:
        mov   R2,#100
        djnz  R2,$
        djnz  R1,del1
        RET
delay2:
        MOV   R3,#100
```

```
del111a:
        MOV   R1,#100
del11a: MOV   R2,#100
        DJNZ  R2,$
        DJNZ  R1,del11a
        DJNZ  R3,del111a
        RET
delay1:
        MOV   R3,#50
del111:
        MOV   R1,#50
del11:  MOV   R2,#50
        DJNZ  R2,$
        DJNZ  R1,del11
        DJNZ  R3,del111
        ret
        END

$regfile = "89s8253.dat"
$crystal = 11059200
$baud = 9600
Baris1 Alias P1.4
Baris2 Alias P1.5
Baris3 Alias P1.6



Dim Kode As String * 24
Riset:
```

```
  Kode = ""
  Goto Kirimdata
Kirimdata:
  Input Kode Noecho
  Waitms 500
  If Kode = "QQ" Then
    Goto Riset
  Else
    Print "ID " ; Kode                    'If Pilihan = "Z" Then
    Goto Main
  End If
Main:
  Kode = ""
  P1 = &HFF
  Goto Ambil_key
Ambil_key:
  P1.0 = 0
  Waitms 20
  If Baris1 = 0 Then
    Print "1"
  End If
  If Baris2 = 0 Then
    Print "2"
  End If
  If Baris3 = 0 Then
    Print "3"
  End If
  Waitms 20
  P1.0 = 1
```

```
P1.1 = 0
Waitms 20
If Baris1 = 0 Then
  Print "4"
End If
If Baris2 = 0 Then
  Print "5"
End If
If Baris3 = 0 Then
  Print "6"
End If
Waitms 20
P1.1 = 1
P1.2 = 0
Waitms 20
If Baris1 = 0 Then
  Print "7"
End If

If Baris2 = 0 Then
  Print "8"
End If
If Baris3 = 0 Then
  Print "9"
End If
Waitms 20
P1.2 = 1
P1.3 = 0
Waitms 20
```

```
    If Baris1 = 0 Then
      Print "*"
    End If
    If Baris2 = 0 Then
      Print "0"
    End If
    If Baris3 = 0 Then
      Print "#"
    End If
    Waitms 20
    If P0.0 = 0 Then
      Print "A"                    'Masuk/DPA
    End If
    If P0.1 = 0 Then
      Print "B"                    'Batal/KHS
    End If
    If P0.2 = 0 Then
      Print "C"                    'Cetak
    End If
    If P0.3 = 0 Then                'Keluar
      Print "D"
      Waitms 20
      Goto Riset
    End If
  Waitms 20
  Goto Main
```

```
unit uMaster;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ExtCtrls, jpeg, Buttons, ADODB, Grids, DBGrids,
  DBCtrls, VaClasses, VaComm, Mask, Math;

type
  TFormMaster = class(TForm)
    Image1: TImage;
    lblKeluar_Biodata: TLabel;
    lblDPA_Biodata: TLabel;
    lblKHS_Biodata: TLabel;
    lblCetak_Biodata: TLabel;
    ADOConnection1: TADOConnection;
    Query: TADOQuery;
    DSBiodata: TDataSource;
    PanelBiodata: TPanel;
    PanelLogin: TPanel;
    Image2: TImage;
    lblMasuk_Login: TLabel;
    lblBatal_Login: TLabel;
    lblKeluar_Login: TLabel;
    Label11: TLabel;
    Shape1: TShape;
    Label12: TLabel;
    Label13: TLabel;
```

Label14: TLabel;

Label7: TLabel;

Label6: TLabel;

Label5: TLabel;

PanelFoto: TPanel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

Label23: TLabel;

lblProdi: TLabel;

Label25: TLabel;

PanelUtama: TPanel;

Image4: TImage;

Label26: TLabel;

Label27: TLabel;

Label28: TLabel;

Label29: TLabel;

Edit2: TEdit;

DBText1: TDBText;

DBText2: TDBText;

DBText5: TDBText;

DBAngkatan: TDBText;

DBText9: TDBText;

DBText10: TDBText;

ADOQueryBiodata: TADOQuery;

lblTTL: TLabel;

PanelDPA: TPanel;

DBGridKHS: TDBGrid;

PanelKHS: TPanel;

ADOQueryKHS: TADOQuery;

ADOQueryDPA: TADOQuery;

DataSourceKHS: TDataSource;

DataSourceDPA: TDataSource;

DBGridDPA: TDBGrid;

ADOQueryKHSkode_mk: TStringField;

ADOQueryKHSsks: TBCDField;

ADOQueryKHSnilai: TStringField;

ADOQueryKHSstatus: TStringField;

ADOQueryKHSNo: TIntegerField;

ADOQueryKHSKxN: TFloatField;

ADOQueryDPAkode_mk: TStringField;

ADOQueryDPAnama_1: TStringField;

ADOQueryDPAsks: TBCDField;

ADOQueryDPAnilai: TStringField;

ADOQueryDPAKxN: TFloatField;

ADOQueryKHSsemester: TBCDField;

ADOQueryKHSnama: TStringField;

DBImage1: TDBImage;

VaComm1: TVaComm;

lbData: TLabel;

Edit1: TEdit;

Timer1: TTimer;

ADOQueryBiodataNIM: TStringField;

ADOQueryBiodataNama: TStringField;

ADOQueryBiodataTmp_Lahir: TStringField;

ADOQueryBiodataTgl_Lahir: TDateTimeField;

ADOQueryBiodataKelamin: TStringField;

ADOQueryBiodataAlamat: TStringField;

ADOQueryBiodataKota: TStringField;

ADOQueryBiodataKode_Pos: TBCDField;

ADOQueryBiodataTelpHP: TStringField;

ADOQueryBiodatafoto: TBlobField;

ADOQueryBiodataKode_Jen: TStringField;

ADOQueryBiodataAngkatan: TBCDField;

ADOQueryBiodataNama_Jur: TStringField;

ADOQueryBiodataNama_Fak: TStringField;

lblDPA: TLabel;

lblKHS: TLabel;

Timer2: TTimer;

lblEror: TLabel;

lblPrior: TLabel;

lblNext: TLabel;

procedure FormShow(Sender: TObject);

procedure lblDPA_BiodataClick(Sender: TObject);

procedure lblKHS_BiodataClick(Sender: TObject);

procedure lblCetak_BiodataClick(Sender: TObject);

procedure lblMasuk_LoginClick(Sender: TObject);

procedure lblKeluar_LoginClick(Sender: TObject);

procedure Label26Click(Sender: TObject);

procedure lblBatal_LoginClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure Label28Click(Sender: TObject);

procedure Label27Click(Sender: TObject);

```
    procedure ADOQueryKHSCalcFields(DataSet: TDataSet);
    procedure ADOQueryDPACalcFields(DataSet: TDataSet);
    procedure ComboBox1Change(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure lblKeluar_BiodataClick(Sender: TObject);
    procedure VaComm1RxChar(Sender: TObject; Count: Integer);
    procedure Label29Click(Sender: TObject);
    procedure lbDataClick(Sender: TObject);
    procedure lblPriorClick(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Timer1Timer(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure lblNextClick(Sender: TObject);
  private
    { Private declarations }
    i: integer;
    //pesan: string;
  public
    { Public declarations }
    pesan: string;
    Str, nim, sks : string;
    pilih    : integer; { 1: ID,
                          2: Sandi,
                          3: SKS}
  end;

const
  labelA = 'A';
  labelB = 'B';
```

```
  labelC = 'C';
  labelD = 'D';
  labelNext = 'E';
  labelPrior = 'F';

var
  FormMaster: TFormMaster;

implementation

uses uVaSetup, uData, uMK, uReport, uReportKHS, uBiodata, uSKS;

{$R *.dfm}

procedure TFormMaster.FormShow(Sender: TObject);
begin
  VaComm1.Open;
  FrmMain.Hide;
  with ADOConnection1 do
  begin
    ConnectionString:= 'Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist '+
                'Security Info=False;Initial Catalog=RFID;Data Source=(local)';
    Connected:= True;
  end;

  Pilih:= 1;
  i:= 0;
  PanelBiodata.Visible := False;
  PanelBiodata.Align   := alClient;
```

```
  PanelUtama.Visible  := False;
  PanelUtama.Align    := alClient;
  PanelLogin.Visible  := True;
  PanelLogin.Align    := alClient;
  PanelDPA.Top        := 112;
  PanelDPA.Left       := 16;
  PanelDPA.Width      := 577;
  PanelDPA.Height     := 345;
  PanelKHS.Top        := 112;
  PanelKHS.Left       := 16;
  PanelKHS.Width      := 577;
  PanelKHS.Height     := 345;
  PanelDPA.Visible    := False;
  PanelKHS.Visible    := False;
  Edit1.SetFocus;
end;

procedure TFormMaster.lblDPA_BiodataClick(Sender: TObject);
begin
  PanelBiodata.Visible := False;
  PanelLogin.Visible  := False;
  PanelUtama.Visible  := True;
  PanelDPA.Visible    := True;
  Label28Click(Sender);
end;

procedure TFormMaster.lblKHS_BiodataClick(Sender: TObject);
begin
  PanelBiodata.Visible := False;
```

```
  PanelLogin.Visible   := False;
  PanelUtama.Visible   := True;
  PanelKHS.Visible     := True;
  Label27Click(Sender);
end;

procedure TFormMaster.lblCetak_BiodataClick(Sender: TObject);
begin
  FormBiodata.QuickRep1.Print;
end;

procedure TFormMaster.lblMasuk_LoginClick(Sender: TObject);
var
  pass, kode : string;
begin
  Query.Close;
  Query.SQL.Text := 'SELECT Kode, Pass, nim FROM Mahasiswa WHERE Kode =
'+QuotedStr(Edit1.Text);
  Query.Open;

  if Query.RecordCount > 0 then
  begin
    nim:= Query.FieldByName('NIM').AsString;
    kode := Query.FieldByName('Kode').AsString;
    pass:= Query.fieldByName('Pass').AsString;

    if pass = trim(Edit2.Text) then
    begin
      with ADOQueryBiodata do
```

```
begin
  Close;
  SQL.Text := 'SELECT M.NIM, M.Nama, M.Tmp_Lahir, M.Tgl_Lahir, '+
              'M.Kelamin, M.Alamat, M.Kota, M.Kode_Pos, M.[Telp/HP], M.foto, '+
              'MM.* FROM Mahasiswa M, (Select M.Kode_Jen, M.Angkatan, '+
              'J.Nama_Jur, F.Nama_Fak FROM MHS_Jurusan M, Jurusan J, Fakultas F '+
              'WHERE M.Kode_Jur = J.Kode_jur AND M.Kode_Fak = '+
              'F.Kode_Fak) AS MM WHERE M.NIM = '+QuotedStr(Trim(nim));

  Open;

  if RecordCount > 0 then
  begin
    PanelBiodata.Visible   := True;
    PanelLogin.Visible     := False;
    PanelUtama.Visible     := False;
    lblProdi.Caption       := Trim(ADOQueryBiodataKode_Jen.Value)+' '+
                              Trim(ADOQueryBiodataNama_Fak.Value)+' '+
                              Trim(ADOQueryBiodataNama_Jur.Value);
    lblTTL.Caption         := Trim(ADOQueryBiodataTmp_Lahir.Value)+', '+
                              FormatDateTime('dd mmmm yyyy', ADOQueryBiodataTgl_Lahir.Value);
    lblEror.Visible:= False;
    Timer2.Enabled:= False;
  end else
  begin
    lblEror.Visible:= True;
    Timer2.Enabled:= True;
  end;
end;
```

```
    end else
    begin
      lblEror.Visible:= True;
      Timer2.Enabled:= True;
    end;
  end else
  begin
    lblEror.Visible:= True;
    Timer2.Enabled:= True;
  end;
end;

procedure TFormMaster.lblKeluar_LoginClick(Sender: TObject);
begin
  Close;
end;

procedure TFormMaster.Label26Click(Sender: TObject);
begin
  PanelUtama.Visible:= False;
  PanelBiodata.Visible := True;
end;

procedure TFormMaster.lblBatal_LoginClick(Sender: TObject);
begin
  pesan := '';
  Edit2.Text:= '';
end;
```

```
procedure TFormMaster.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Query.Active            := False;
  ADOConnection1.Connected  := False;
  frmMain.Close;
end;

procedure TFormMaster.Label28Click(Sender: TObject);
begin
  PanelDPA.Visible := True;
  PanelKHS.Visible := False;
  lblDPA.Visible   := True;
  lblKHS.Visible   := False;

  with ADOQueryDPA do
  begin
    Close;
    SQL.Text := 'SELECT m.kode_mk, m.nama, m.sks, am.nilai '+
                        'FROM matakuliah m, ambil_mk am '+
                        'WHERE m.kode_mk = am.kode_mk '+
            'AND am.NIM = '+QuotedStr(trim(NIM));
    Open;
  end;
end;

procedure TFormMaster.Label27Click(Sender: TObject);
begin
  FormSKS.ShowModal;
end;
```

```
procedure TFormMaster.ADOQueryKHSCalcFields(DataSet: TDataSet);
begin
 if trim(ADOQueryKHSNilai.AsString) = 'A' then
   ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsInteger * 4;

 if trim(ADOQueryKHSNilai.AsString) = 'B' then
   ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsInteger * 3;

 if trim(ADOQueryKHSNilai.AsString) = 'C' then
   ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 2;

 if ADOQueryKHSNilai.AsString = 'B+' then
   ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 3.5;

 if ADOQueryKHSNilai.AsString = 'C+' then
   ADOQueryKHSKxN.Value := ADOQueryKHSSKs.AsFloat * 2.5;

 ADOQueryKHSNo.Value := ADOQueryKHS.RecNo;
 if ADOQueryKHSNo.Value = -1 then
   ADOQueryKHSNo.Value:= 1;
end;

procedure TFormMaster.ADOQueryDPACalcFields(DataSet: TDataSet);
begin
 if trim(ADOQueryDPANilai.AsString) = 'A' then
   ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsInteger * 4;

 if trim(ADOQueryDPANilai.AsString) = 'B' then
```

```
   ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsInteger * 3;

  if trim(ADOQueryDPANilai.AsString) = 'C' then
   ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 2;

  if ADOQueryDPANilai.AsString = 'B+' then
   ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 3.5;

  if ADOQueryDPANilai.AsString = 'C+' then
   ADOQueryDPAKxN.Value := ADOQueryDPASKs.AsFloat * 2.5;
end;

procedure TFormMaster.ComboBox1Change(Sender: TObject);
begin
 Label27Click(Sender);
end;

procedure TFormMaster.Button6Click(Sender: TObject);
begin
 if (lblMasuk_Login.Transparent = False) or (i = 0) then
  lblMasuk_Login.OnClick(Sender) else
 if lblBatal_Login.Transparent = False then
  lblBatal_Login.OnClick(Sender) else
 if lblKeluar_Login.Transparent = False then
  lblKeluar_Login.OnClick(Sender);
end;

procedure TFormMaster.lblKeluar_BiodataClick(Sender: TObject);
begin
```

```
  PanelLogin.Visible:= True;
  PanelBiodata.Visible := False;
  pesan := '';
  Pilih := 1;
  Edit1.Text:= '';
  Edit2.Text:= '';
  edit1.SetFocus;
  Timer1.Enabled:= True;
end;

procedure TFormMaster.VaComm1RxChar(Sender: TObject; Count: Integer);
var
  i: integer;
  tmp : string;
begin
  tmp:= VaComm1.ReadText;
  sleep(500);

  if labelA = trim(tmp) then
  begin
    if PanelLogin.Visible = true  then
      lblMasuk_LoginClick(Sender) else
    if PanelBiodata.Visible = true then
      lblDPA_BiodataClick(Sender) else
    if PanelUtama.Visible = True then
      Label28Click(Sender);
  end
  else if labelB = trim(tmp) then
  begin
```

```
  if PanelLogin.Visible = true  then
    lblBatal_Loginclick(Sender) else
  if PanelBiodata.Visible = true then
    lblKHS_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label27Click(Sender);
end
else if labelC = trim(tmp) then
begin
  if PanelBiodata.Visible = true then
    lblCetak_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label29Click(Sender);
end
else if labelD = trim(tmp) then
begin
  {if PanelLogin.Visible = true then
    lblKeluar_LoginClick(Sender) else}
  if PanelBiodata.Visible = true then
    lblKeluar_BiodataClick(Sender) else
  if PanelUtama.Visible = True then
    Label26Click(Sender);
end
else if labelNext = trim(tmp) then
begin
  if PanelUtama.Visible = True then
    lblPriorClick(Sender);
end
else if labelPrior = trim(tmp) then
```

```
  begin
   if PanelUtama.Visible = True then
    lblNextClick(Sender);
  end
  else
  begin
   //baca data yang diterima oleh comport
   for I := 1 to Length(Tmp) do
    case Tmp[I] of
     #10:;                    //lewatkan
     #13:
    else                      //bukan #10 atau #13
     pesan := pesan + Tmp[I];
     case pilih of  //if Pilih = 1 then
      1: Edit1.Text := Trim(Pesan);
     //else if pilih = 2 then
      2: Edit2.Text := Trim(Pesan);
     end;
    end;
  end;
end;

procedure TFormMaster.Label29Click(Sender: TObject);
begin
  if PanelDPA.Visible then
   FormDPA.QuickRep1.Print
  else
   FormKHS.QuickRep1.Print;
end;
```

```
procedure TFormMaster.lbDataClick(Sender: TObject);
begin
  FormData.ShowModal;
end;

procedure TFormMaster.lblPriorClick(Sender: TObject);
begin
  ADOQueryDPA.Prior;
end;

procedure TFormMaster.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if key = #13 then
    Edit2.SetFocus;
end;

procedure TFormMaster.Timer1Timer(Sender: TObject);
var
  str: string;
begin
  if length(edit1.Text) > 10 then
  begin
    str:= edit1.Text;
    Delete(str, 1, 1);
    edit1.Text:= str;
  end else
  if length(edit1.Text) = 10 then
  begin
```

```
    pesan := '';
    Pilih := 2;
    edit2.SetFocus;
    Timer1.Enabled:= False;
  end;
end;

procedure TFormMaster.Timer2Timer(Sender: TObject);
begin
  if lblEror.Font.Color = clRed then
    lblEror.Font.Color:= clBlue else lblEror.Font.Color:= clRed;
end;

procedure TFormMaster.lblNextClick(Sender: TObject);
begin
  ADOQueryDPA.Next;
end;

end.
```

```
; LCD CONSTANTA
DISPCLR  EQU 00000001B
BLINK    EQU 00001101B
ENTRMOD  EQU 00000110B
DISPON   EQU 00001100B
CURSOR   EQU 00011100B
FUNCSET  EQU 00111000B
;
;DEVICE ADDRESS [LCD]
RS       BIT P2.6  ;LCD
E        BIT P2.7  ;LCD
MY_DATA      BIT P3.4
MY_COUNTER EQU 31H
MY_TEMP      EQU 32H
KEYDATA      EQU 33H
COD1     EQU 27H
COD2     EQU 28H
COD3     EQU 29H
DATAKEY0 EQU 34H
DATAKEY1 EQU 35H
DATAKEY2 EQU 36H
DATAKEY3 EQU 37H
DATAKEY4 EQU 38H
DATAKEY5 EQU 39H
DATAKEY6 EQU 40H
DATAKEY7 EQU 41H
DATAKEY8 EQU 42H
DATAKEY9 EQU 43H
DATAKEY10 EQU 44H
```

```
DATAKEY11 EQU 45H
DATAKEY12 EQU 46H
LOKASI      EQU 47H
DATAKEY13  EQU 48H
DATAKEY14  EQU 49H
DATAKEY15  EQU 50H
DATAKEY16  EQU 51H
DATAKEY17  EQU 52H
DATAKEY18  EQU 53H
DATAKEY19  EQU 54H
DATAKEY20  EQU 55H
DATAKEY21  EQU 56H
DATAKEY22  EQU 57H
DATAKEY23 EQU 58H
DATAKEY24 EQU 59H
DATAKEY25 EQU 60H
LOKASI1 EQU 61H
keybounc    equ    62H
DATAKEY26 EQU 63H
kolom1      bit    P1.4    ; kiri (1,4,7,redial)
kolom2      bit    P1.5
kolom3      bit    P1.6
baris1      bit    P1.0    ; atas (1,2,3)
baris2      bit    P1.1
baris3      bit    P1.2
baris4      bit    P1.3
;
keyport     equ    P1
        ORG    0000H       ; Mulai dari alamat reset
```

```
        AJMP   MULAI            ; Loncat ke label mulai
        ORG    0013H            ; Alamat interupt
        ORG    23H              ; alamat untuk serial comm. interupt
        CLR    ES               ; matikan interupt serial
        JNB    RI,$             ; tunggu sampai selesai menerima
        CLR    RI               ; clear receive interupt flag
        MOV    R1,SBUF
        MOV    B,R1
        MOV    A,R1
        MOV    A,LOKASI
        INC    A
        MOV    LOKASI,A
PIL0:
        CJNE   A,#00000001B,PILL1
        MOV    DATAKEY1,B
        JMP TAMPILL
PILL1:
        CJNE   A,#00000010B,PIL2
        MOV    DATAKEY2,B
        JMP TAMPILL
PIL2:
        CJNE   A,#00000011B,PIL3
        MOV    DATAKEY3,B
        JMP TAMPILL
PIL3:
        CJNE   A,#00000100B,PIL4
        MOV    DATAKEY4,B
        JMP TAMPILL
PIL4:
```

```
        CJNE   A,#00000101B,PIL5
        MOV    DATAKEY5,B
        JMP TAMPILL
PIL5:
        CJNE   A,#00000110B,PILL6
        MOV    DATAKEY6,B
        JMP TAMPILL
PILL6:
        CJNE   A,#00000111B,PIL7
        MOV    DATAKEY7,B
        JMP TAMPILL
PIL7:
        CJNE   A,#00001000B,PIL8
        MOV    DATAKEY8,B
        JMP TAMPILL
PIL8:
        CJNE   A,#00001001B,PIL10
        MOV    DATAKEY9,B
        JMP TAMPILL
PIL10:
        CJNE   A,#00001010B,PIL11
        MOV    DATAKEY10,B
        JMP TAMPILL
PIL11:
        CJNE   A,#00001011B,PIL12
        MOV    DATAKEY11,B
        JMP TAMPILL
PIL12:
        CJNE   A,#00001100B,PIL13
```

```
        MOV   DATAKEY12,B
        JMP TAMPILL
PIL13:
        CJNE   A,#00001101B,PIL14
        MOV   DATAKEY13,B
        JMP TAMPILL
PIL14:
        CJNE   A,#00001110B,PIL15
        MOV   DATAKEY14,B
        MOV   lokasi,#0
        JMP TAMPILL
PIL15:
        CJNE   A,#00001111B,PIL16
        MOV   DATAKEY15,B
        JMP TAMPILL
PIL16:
        CJNE   A,#00010000B,PIL17
        MOV   DATAKEY16,B
        MOV   lokasi,#0
        JMP TAMPILL1
PIL17:
        MOV   lokasi,#0
        JMP TAMPILL
TAMPILL1:
        JMP    TAMPILKAN_PC
TAMPILL:
        SETB  ES
        RETI
MULAI:
```

```
        CALL  init_serial
        JMP   oke
TUNDA_T:
        MOV   R7,#5
LOOP10:     MOV   R6,#5
LOOP9:      MOV   R5,#25
        DJNZ  R5,$
        DJNZ  R6,LOOP9
        DJNZ  R7,LOOP10
        RET
TUNDA_TAM:
        MOV   R7,#100
LOP10: MOV   R6,#100
LOP9:  MOV   R5,#100
        DJNZ  R5,$
        DJNZ  R6,LOP9
        DJNZ  R7,LOP10
        RET
init_serial:
        MOV   SCON,#50H
        MOV   TMOD,#20H
        MOV   TL1,#0FDH
        MOV   TH1,#0FDH
        SETB  TR1
        SETB  EA            ; enable interupt
        SETB  ES            ; enable serial interupt
        RET
OKE:
        MOV   DATAKEY0,#'0'
```

```
        MOV   DATAKEY1,#'0'
        MOV   DATAKEY2,#'0'
        MOV   DATAKEY3,#'0'
        MOV   DATAKEY4,#'0'
        MOV   DATAKEY5,#'0'
        MOV   DATAKEY6,#'0'
        MOV   DATAKEY7,#'0'
        MOV   DATAKEY8,#'0'
        MOV   DATAKEY9,#'0'
        MOV   DATAKEY10,#'0'
        MOV   DATAKEY11,#'0'
        MOV   DATAKEY12,#'0'
        MOV   LOKASI,#0
        MOV   DATAKEY13,#'0'
        MOV   DATAKEY14,#'0'
        MOV   DATAKEY15,#'0'
        MOV   DATAKEY16,#'0'
        MOV   DATAKEY26,#0
data_mati:
KEYPAD:
        JNB   P0.0,kirim_data_A
        JNB   P0.1,kirim_data_B
        JNB   P0.2,kirim_data_C
        JNB   P0.3,kirim_data_D
;**************************************

KEYLOOP:
ulang:  call  Keypad3x4
        mov   A,keydata
```

```
        cjne    A,#0FFH,tampil
        jmp     KEYPAD
kirim_data_A:
        MOV A,#'A'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_B:
        MOV A,#'B'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_C:
        MOV A,#'C'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
kirim_data_D:
        MOV A,#'D'
        CALL  DATA_M
        CALL  delay2
        JMP     data_mati
TAMPIL:
        MOV A,keydata
        CALL  DATA_M
        CALL  delay1
        JMP     KEYPAD
Keypad3x4:
        mov     keybounc,#50
```

```
        mov     keyport,#0FFh
        clr     kolom1
ul1:    jb      baris1,key1
        djnz    keybounc,ul1
        mov     keydata,#'1'
        ret
key1:   jb      baris2,key2
        djnz    keybounc,key1
        mov     keydata,#'4'
        ret
key2:   jb      baris3,key3
        djnz    keybounc,key2
        mov     keydata,#'7'
        ret
key3:   jb      baris4,key4
        djnz    keybounc,key3
        mov     keydata,#'E'
        ret
key4:   setb    kolom1
        clr     kolom2
        jb      baris1,key5
        djnz    keybounc,key4
        mov     keydata,#'2'
        ret
key5:   jb      baris2,key6
        djnz    keybounc,key5
        mov     keydata,#'5'
        ret
key6:   jb      baris3,key7
```

```
        djnz    keybounc,key6
        mov     keydata,#'8'
        ret
key7:   jb      baris4,key8
        djnz    keybounc,key7
        mov     keydata,#'0'
        ret
key8:   setb    kolom2
        clr     kolom3
        jb      baris1,key9
        djnz    keybounc,key8
        mov     keydata,#'3'
        ret
key9:   jb      baris2,key10
        djnz    keybounc,key9
        mov     keydata,#'6'
        ret
key10:  jb      baris3,key11
        djnz    keybounc,key10
        mov     keydata,#'9'
        ret

key11:  jb      baris4,key12
        djnz    keybounc,key11
        mov     keydata,#'F'
        ret
key12:  mov     keydata,#0FFh
        ret ,
;*************************************
```

```
TAMPILKAN_PC:
        ;MOV A,DATAKEY1
        ;CALL DATA_M
        ;MOV A,DATAKEY2
        ;CALL DATA_M
        ;MOV A,DATAKEY3
        ;CALL DATA_M
        MOV A,DATAKEY4
        CALL  DATA_M
        MOV A,DATAKEY5
        CALL  DATA_M
        MOV A,DATAKEY6
        CALL  DATA_M
        MOV A,DATAKEY7
        CALL  DATA_M
        MOV A,DATAKEY8
        CALL  DATA_M
        MOV A,DATAKEY9
        CALL  DATA_M
        MOV A,DATAKEY10
        CALL  DATA_M
        MOV A,DATAKEY11
        CALL  DATA_M
        MOV A,DATAKEY12
        CALL  DATA_M
        MOV A,DATAKEY13
        CALL  DATA_M
        MOV A,DATAKEY14
        CALL  DATA_M
```

```
        ;MOV A,DATAKEY15
        ;CALL DATA_M
        ;MOV A,DATAKEY16
        ;CALL DATA_M
        CALL  delay2
        JMP TAMPILL
DATA_M:
        CLR   ES      ; matikan serial interupt saat mengirim
        MOV   SBUF,A        ; isi serial buffer dengan data yg dikirim
        JNB   TI,$    ; tunggu pengiriman selesai
        CLR   TI      ; clear transmit interupt flag
        MOV   50H,#50
TUNDAL1:
        MOV   51H,#50
        DJNZ  51H,$
        DJNZ  50H,TUNDAL1
        SETB  ES      ; hidupkan kembali serial interupt
        SETB  EA
        RET
;
delay:
        MOV   R1,#50
del1:
        mov   R2,#100
        djnz  R2,$
        djnz  R1,del1
        RET
delay2:
        MOV   R3,#100
```

```
del111a:
        MOV   R1,#100
del11a: MOV   R2,#100
        DJNZ  R2,$
        DJNZ  R1,del11a
        DJNZ  R3,del111a
        RET
delay1:
        MOV   R3,#50
del111:
        MOV   R1,#50
del11:  MOV   R2,#50
        DJNZ  R2,$
        DJNZ  R1,del11
        DJNZ  R3,del111
        ret
        END

$regfile = "89s8253.dat"
$crystal = 11059200
$baud = 9600
Baris1 Alias P1.4
Baris2 Alias P1.5
Baris3 Alias P1.6



Dim Kode As String * 24
Riset:
```

```
  Kode = ""
  Goto Kirimdata
Kirimdata:
  Input Kode Noecho
  Waitms 500
  If Kode = "QQ" Then
    Goto Riset
  Else
    Print "ID " ; Kode                    'If Pilihan = "Z" Then
    Goto Main
  End If
Main:
  Kode = ""
  P1 = &HFF
  Goto Ambil_key
Ambil_key:
  P1.0 = 0
  Waitms 20
  If Baris1 = 0 Then
    Print "1"
  End If
  If Baris2 = 0 Then
    Print "2"
  End If
  If Baris3 = 0 Then
    Print "3"
  End If
  Waitms 20
  P1.0 = 1
```

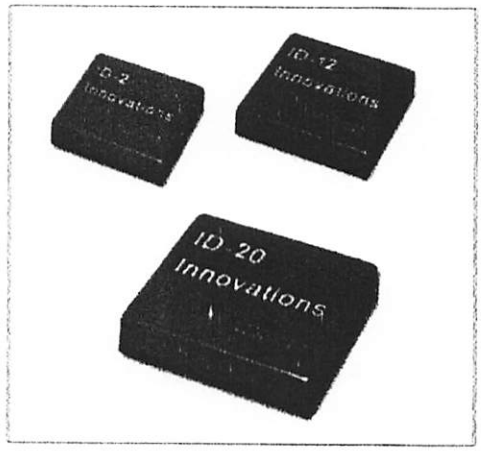
```
P1.1 = 0
Waitms 20
If Baris1 = 0 Then
  Print "4"
End If
If Baris2 = 0 Then
  Print "5"
End If
If Baris3 = 0 Then
  Print "6"
End If
Waitms 20
P1.1 = 1
P1.2 = 0
Waitms 20
If Baris1 = 0 Then
  Print "7"
End If

If Baris2 = 0 Then
  Print "8"
End If
If Baris3 = 0 Then
  Print "9"
End If
Waitms 20
P1.2 = 1
P1.3 = 0
Waitms 20
```

```
  If Baris1 = 0 Then
    Print "*"
  End If
  If Baris2 = 0 Then
    Print "0"
  End If
  If Baris3 = 0 Then
    Print "#"
  End If
  Waitms 20
  If P0.0 = 0 Then
    Print "A"                    'Masuk/DPA
  End If
  If P0.1 = 0 Then
    Print "B"                    'Batal/KHS
  End If
  If P0.2 = 0 Then
    Print "C"                    'Cetak
  End If
  If P0.3 = 0 Then                'Keluar
    Print "D"
    Waitms 20
    Goto Riset
  End If
Waitms 20
Goto Main
```

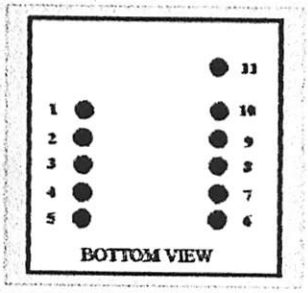
ID SERIES DATASHEET Mar 01, 2005

ID-2 / ID-12 / ID-20

The ID2. ID12 and ID20 are similar to the obsolete ID0, ID10 and ID15 MK(ii) series devices, but they have extra pins that 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

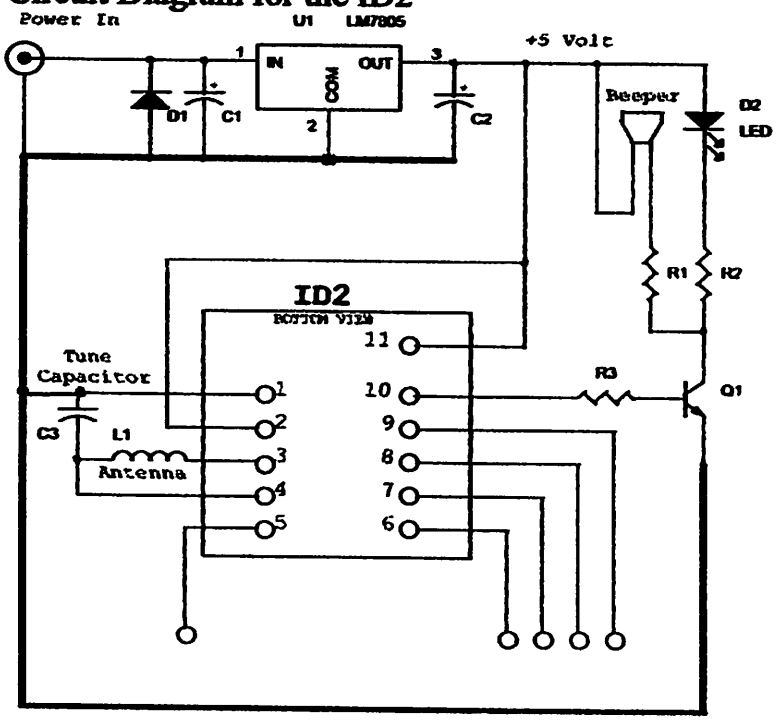
Parameters	ID-2	ID-12	ID-20
Read Range	N/A (no internal antenna)	12+ cm	16+ cm
Dimensions	21 mm x 19 mm x 6 mm	26 mm x 25 mm x 7 mm	40 mm x 40 mm x 9 mm
Frequency	125 kHz	125 kHz	125 kHz
Card Format	EM 4001 or compatible	EM 4001 or compatible	EM 4001 or compatible
Encoding	Manchester 64-bit, modulus 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

* Requires 4K7 Pull-up resistor to +5V

Circuit Diagram for the ID2



COMPONENT LIST

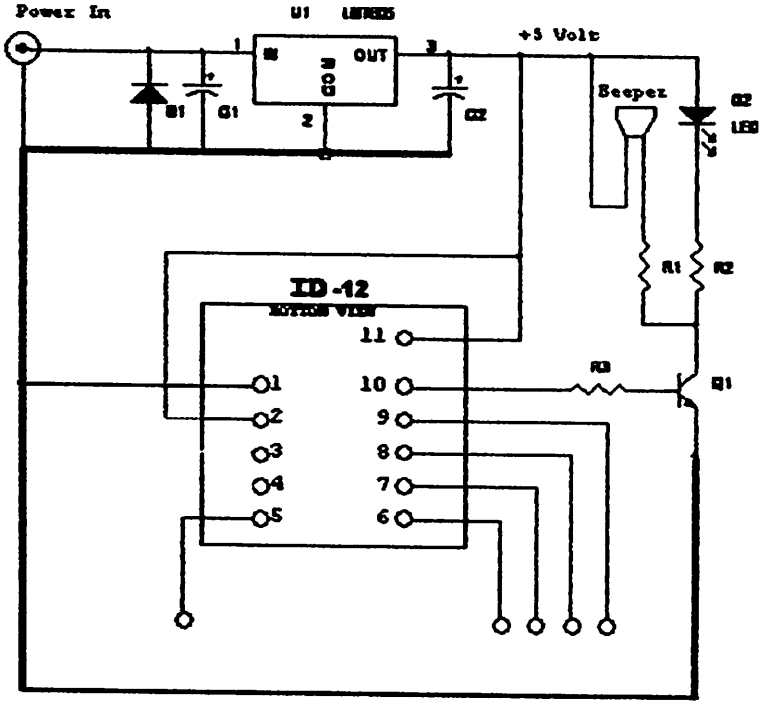
R1 = 100R
R2 = 1K
R3 = 1K
C1 = 100uF 16V
C2 = 100uF 10V
C3 = 1nF COG 100V *
Beeper = 2.7-3.5KHz 100R
D1 = 1N4001
D2 = GREEN LED
U1 = LM7805
Q1 = UTC8050 (NPN)
L1 = 640Uh

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

Circuit Diagram for the ID-12/ID20



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

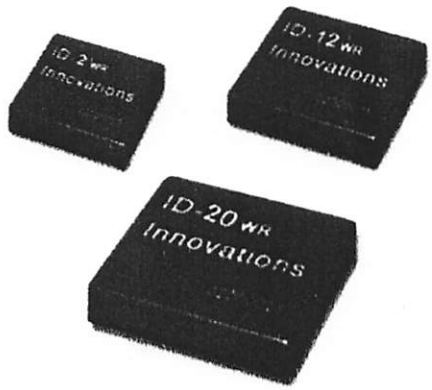
ID Innovations

Advanced Digital Reader Technology

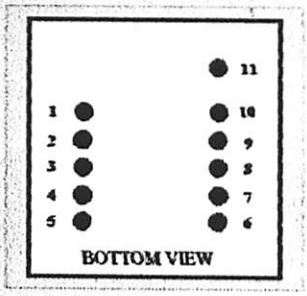
—Better by Design

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

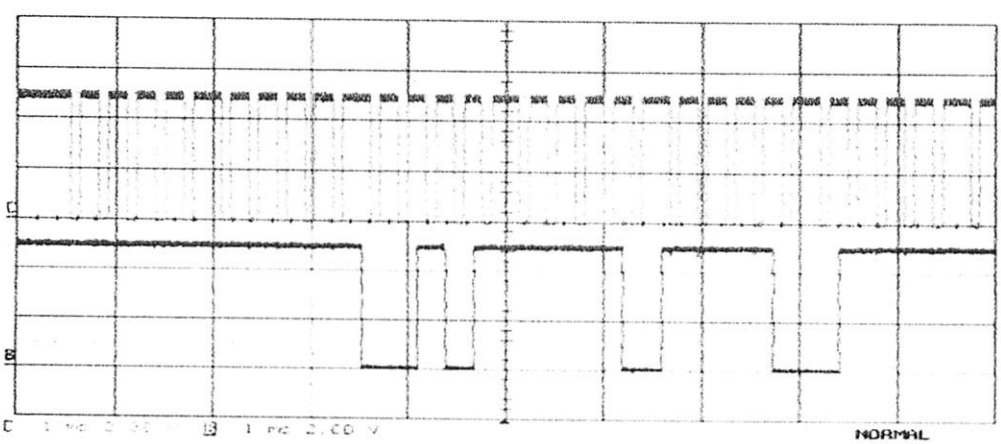
Operational and Physical Characteristics

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.

Magnetic Timing Diagram



DATA FORMATS

Output Data Structure – ASCII

STX (02h)	DATA (10 ASCII)	CHECK SUM (2 ASCII)	CR	LF	ETX (03h)

[The 1byte (2 ASCII characters) Check sum is the "Exclusive OR" of the 5 hex bytes (10 ASCII) Data characters.]

Output Data Structure – Wiegand26

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	E	E	E	E	E	E	E	E	E	E	E	E	O	O	O	O	O	O	O	O	O	O	O	O	P
Even parity (E)													Odd parity (O)												

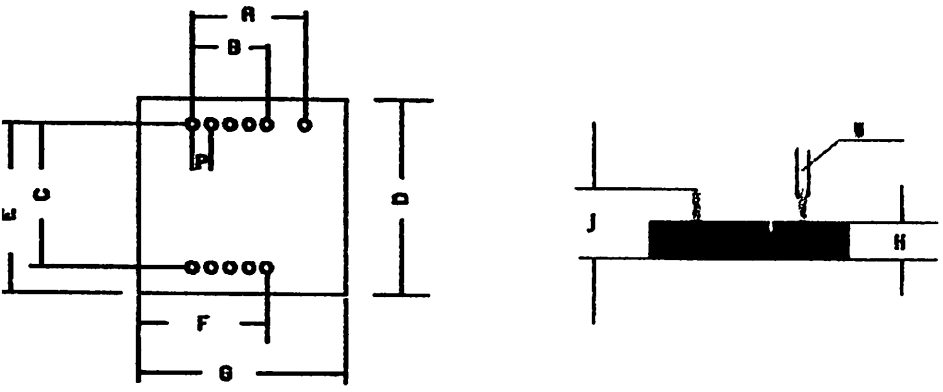
P = Parity start bit and stop bit

Output Data Magnetic ABA Track2

10 Leading Zeros	SS	Data	ES	LCR	10 Ending Zeros

[SS is the Start Character of 11010, ES is the end character of 11111, LRC is the Longitudinal Redundancy Check.]

Dimensions (Top View) (mm)



	ID-2			ID-12			ID-20		
	Nom.	Min.	Max.	Nom.	Min.	Max.	Nom.	Min.	Max
A	12.0	11.6	12.4	12.0	11.6	12.4	12.0	11.6	12.4
B	8.0	7.6	8.4	8.0	7.6	8.4	8.0	7.6	8.4
C	15.0	14.6	15.4	15.0	14.6	15.4	15.0	14.6	15.4
D	20.5	20.0	21.5	25.3	24.9	25.9	40.3	40.0	41.0
E	18.5	18.0	19.2	20.3	19.8	20.9	27.8	27.5	28.5
F	14.0	13.0	14.8	16.3	15.8	16.9	22.2	21.9	23.1
G	22.0	21.6	22.4	26.4	26.1	27.1	38.5	38.2	39.2
P	2.0	1.8	2.2	2.0	1.8	2.2	2.0	1.8	2.2
H	5.92	5.85	6.6	6.0	5.8	6.6	6.8	6.7	7.0
J	9.85	9.0	10.5	9.9	9.40	10.5	9.85	9.4	10.6
W	0.66	0.62	0.67	0.66	0.62	0.67	0.66	0.62	0.67

Note – measurements do not include any burring of edges.

Designing Coils for ID2

The recommended Inductance is 1.08mH to be used with an internal tuning capacitor of 1n5. In general the bigger the antenna the better, provided the reader is generating enough field strength to excite the tag. The ID-2 is relatively low power so a maximum coil size of 15x15cm is recommended if it is intended to read ISO cards. If the reader is intended to read glass tags the maximum coil size should be smaller, say 10x10cm.

There is a science to determine the exact size of an antenna but there are so many variables that in general it is best to get a general idea after which a degree of 'Try it and see' is unavoidable.

If the reader is located in a position where there is a lot of heavy interference then less range cannot be avoided. In this situation the coil should be made smaller to increase the field strength and coupling.

It is difficult to give actual examples of coils for hand wounding because the closeness and tightness of the winding will significantly change the inductance. A professionally wound coil will have much more inductance than a similar hand wound coil.

For those who want a starting point into practical antenna winding it was found that 63 turns on a 120mm diameter former gave an inductance of 1.08mH. For those contemplating adding an additional tuning capacitor it was found that 50 turns on a 120mm diameter former gave 700uH. The wire diameter is not important.
Anybody who wishes to be more theoretical we recommend a trip to the Microchip Website where we found an application sheet for Loop Antennas. http://ww1.microchip.com/downloads/cn/AppNotes/00831b.pdf

The Tuning Capacitor

It is recommended that the internal 1n% capacitor is used for tuning, however a capacitor may be also be added externally. The combined capacitance should not exceed 2n7. Do not forget that the choice of tuning capacitor can also substantially affect the quality of your system. The Id12 is basically an ID2 with an internal antenna. The loss in an ID12 series antenna is required to be fairly high to limit the series current. A low Q will hide a lot of the shortcomings of the capacitor, but for quality and reliability and repeatability the following capacitors are recommend.

Polypropylene	Good Readily available. Ensure AC voltage at 125kHz is sufficient.
COG/NPO	Excellent. Best Choice
Silver Mica	Excellent but expensive
Polycarbonate	Good Readily available. Ensure AC voltage at 125kHz is sufficient.

Voltage Working.

A capacitor capable of withstanding the RMS voltage at 125KHz MUST be chosen. The working voltage will depend on the coil design. I suggest the designer start with rugged 1n5 Polypropylene 630v capacitor to do his experiments and the come down to a suitable size/value. The capacitor manufacturer will supply information on their capacitors. Do not simply go by the DC voltage. This means little. A tolerance of 2% is preferable. A tolerance of 5% is acceptable.

Fine Tuning

We recommend using an oscilloscope for fine-tuning. Connect the oscilloscope to observe the 125KHz AC voltage across the coil. Get a sizeable piece of ferrite and bring it up to the antenna loop. If the voltage increases then you need more inductance (or more capacitance). If the voltage decreases as you bring the ferrite up to the antenna then the inductance is too great. If you have no ferrite then a piece of aluminum sheet may be used for testing in a slightly different way. Opposing currents will flow in the aluminum and it will act as a negative inductance. If the 125kH AC voltage increases as the aluminum sheet approaches the antenna then the inductance is too high. Note it may be possible that the voltage will first maximize then decrease. This simply means that you are near optimum tuning. If you are using ferrite then the coil is a little under value and if you are using an aluminum sheet then the coil is a over under value.

Pembacaan format data RFID

AN-05

Oleh: Tim Digiware

Saat ini 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. Di dalam artikel ini akan dibahas mengenai cara membaca format data yang dikeluarkan oleh RFID reader dengan format output ASCII.

Pendahuluan

RFID reader mempunyai banyak sekali tipe, antara lain: ID-10, ID-19, EM-13, dll. Biasanya RFID reader ini memiliki dua bentuk output serial yaitu: ASCII dan Wiegand 26-bit. Yang paling banyak digunakan adalah output dengan format ASCII, karena output ini sangat mudah untuk dihubungkan pada mikrokontroler atau PC menggunakan komunikasi serial UART.

Format Data ASCII

Output yang memiliki format ASCII memiliki struktur sebagai berikut:

02	10 data karakter ASCII	checksum	CR	LF	03

Gb1. Format data ASCII

Checksum merupakan hasil EXOR (Exclusive OR) dari 5 biner data byte. Untuk lebih jelasnya tentang cara pembacaan format ASCII, lihat contoh berikut.

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 ke dalam Format Data ASCII seperti gb 1. Kemudian ambil 10 data karakter ASCII. Dalam contoh ini berarti data tersebut adalah:

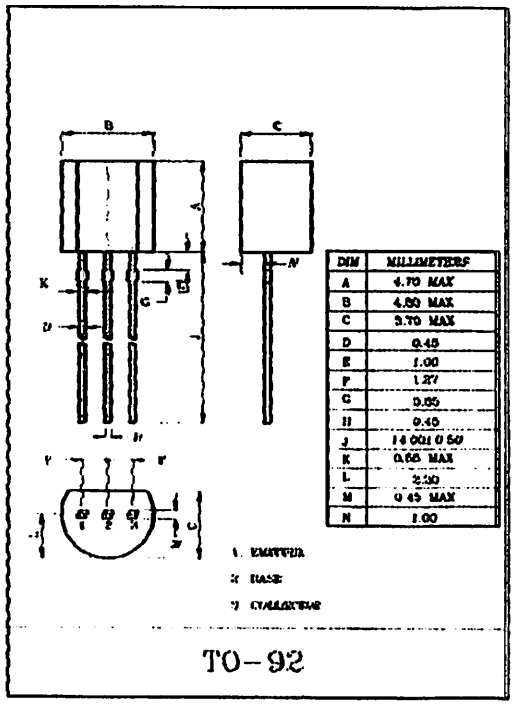
30	34	36	32	30	31	44	37	36	43	Data Heksa
		6	2	0	1	D	7	6	C	Data ASCII

Untuk data dengan warna biru merupakan data untuk jenis – jenis kartu dan tidak digunakan dalam proses konversi, yang akan dipakai disini adalah data yang ke 3 s/d 10. Hasil konversi dari data heksa ke dalam data ASCII adalah "6201D76C". Gabungkan karakter data ASCII menjadi bilangan Hexadesimal, kemudian konversikan bilangan hexadesimal tsb ke dalam desimal. Hasilnya sebagai berikut: 6201D76C H menjadi 1644287852 (ini merupakan nomor kartu sebenarnya yang tertera pada badan kartu tsb). Cara ini hanya berlaku pada kartu yang tidak dienkripsi.

Selamat mencoba.

NERAL PURPOSE APPLICATION.
ITCHING APPLICATION.

ATURES
xcellent h_{FE} Linearity.
omplementary to KTC9012



DIM	MILLIMETERS
A	4.70 MAX
B	4.80 MAX
C	3.70 MAX
D	0.45
E	1.00
F	1.27
G	0.85
H	0.45
J	1.00 ± 0.50
K	0.55 MAX
L	2.50
M	0.45 MAX
N	1.00

XIMUM RATINGS (Ta=25℃)

CHARACTERISTIC	SYMBOL	RATING	UNIT
llector-Base Voltage	V_{CBO}	40	V
llector-Emitter Voltage	V_{CEO}	30	V
nitter-Base Voltage	V_{EBO}	5	V
llector Current	I_C	500	mA
nitter Current	I_E	500	mA
llector Power Dissipation	P_C	625	mW
nction Temperature	T_j	150	℃
orage Temperature Range	T_{stg}	-55~150	℃

CTRICAL CHARACTERISTICS (Ta=25℃)

CHARACTERISTIC	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
llector Cut-off Current	I_{CBO}	V_{CB}=35V, I_E=0	-	-	0.1	μA
nitter Cut-off Current	I_{EBO}	V_{EB}=5V, I_C=0	-	-	0.1	μA
C Current Gain	h_{FE}(Note)	V_{CE}=1V, I_C=50mA	64	-	246	
llector-Emitter turation Voltage	$V_{CE(sat)}$	I_C=100mA, I_B=10mA	-	0.1	0.25	V
se-Emitter Voltage	V_{BE}	I_C=100mA, V_{CE}=1V	-	0.8	1.0	V
ansition Frequency	f_T	V_{CE}=6V, I_C=20mA, f=100MHz	140	-	-	MHz
llector Output Capacitance	C_{ob}	V_{CB}=6V, I_E=0, f=1MHz	-	7.0	-	pF

te : h_{FE} Classification D:64~91, E:78~112, F:96~135,
G:118~166, H:144~202, I:176~246

atures

- ompatible with MCS®-51 Products
- 2K Bytes of In-System Programmable (ISP) Flash Program Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 10,000 Write/Erase Cycles
- K Bytes EEPROM Data Memory
  - Endurance: 100,000 Write/Erase Cycles
- 4-byte User Signature Array
- .7V to 5.5V Operating Range
- ully Static Operation: 0 Hz to 24 MHz (In x1 and x2 Modes)
- hree-level Program Memory Lock
- 56 x 8-bit Internal RAM
- 2 Programmable I/O Lines
- hree 16-bit Timer/Counters
- ine Interrupt Sources
- nhanced UART Serial Port with Framing Error Detection and Automatic ddress Recognition
- nhanced SPI (Double Write/Read Buffered) Serial Interface
- ow-power Idle and Power-down Modes
- nterrupt Recovery from Power-down Mode
- rogrammable Watchdog Timer
- ual Data Pointer
- ower-off Flag
- lexible ISP Programming (Byte and Page Modes)
  - Page Mode: 64 Bytes/Page for Code Memory, 32 Bytes/Page for Data Memory
- our-level Enhanced Interrupt Controller
- rogrammable and Fuseable x2 Clock Option
- nternal Power-on Reset
- 2-pin PDIP Package Option for Reduced EMC Emission
- reen (Pb/Halide-free) Packaging Option

Description

AT89S8253 is a low-power, high-performance CMOS 8-bit microcontroller with
K bytes of In-System Programmable (ISP) Flash program memory and 2K bytes of
PROM data memory. The device is manufactured using Atmel's high-density non-
atile memory technology and is compatible with the industry-standard MCS-51
ruction set and pinout. The on-chip downloadable Flash allows the program mem-
to be reprogrammed in-system through an SPI serial interface or by a
ventional nonvolatile memory programmer. By combining a versatile 8-bit CPU
downloadable Flash on a monolithic chip, the Atmel AT89S8253 is a powerful
rocontroller which provides a highly-flexible and cost-effective solution to many
bedded control applications.

ATMEL®

8-bit Microcontroller with 12K Bytes Flash and 2K Bytes EEPROM

AT89S8253

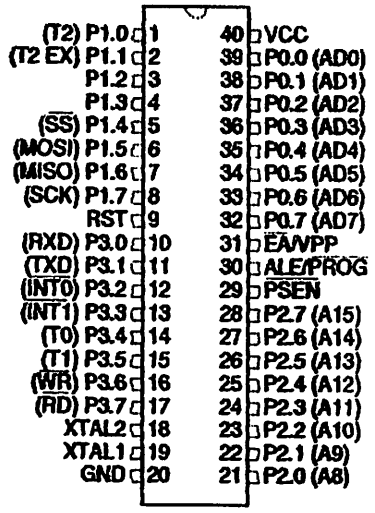
3286L-MICRO-6/08

The AT89S8253 provides the following standard features: 12K bytes of In-System Programmable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector, four-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8253 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next external interrupt or hardware reset.

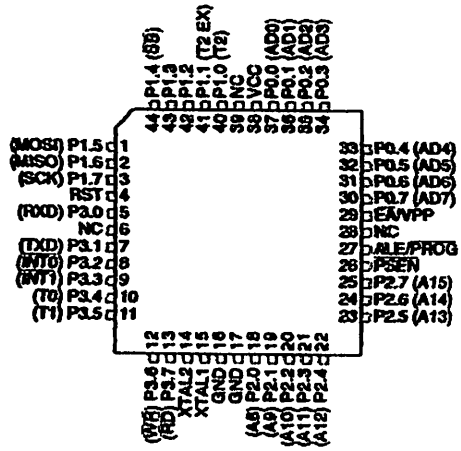
The on-board Flash/EEPROM is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from, unless one or more lock bits have been activated.

Pin Configurations

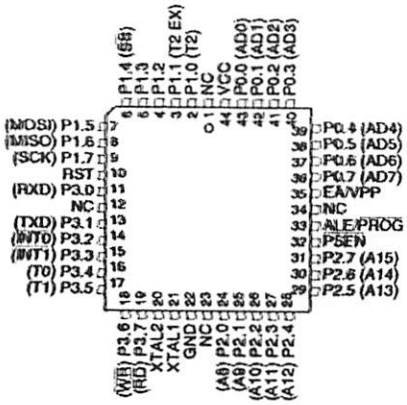
40P6 – 40-lead PDIP



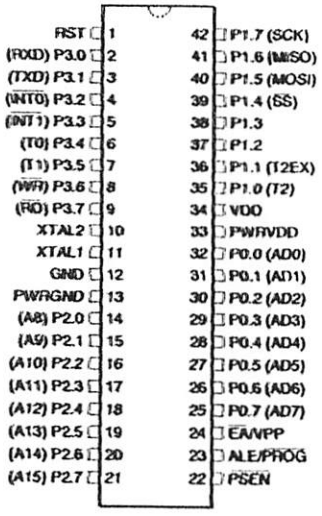
44A – 44-lead TQFP



44J – 44-lead PLCC



42PS6 – PDIP



Pin Description

VCC

Supply voltage (all packages except 42-PDIP).

GND

Ground (all packages except 42-PDIP; for 42-PDIP GND connects only the logic core and the embedded program/data memories).

VDD

Supply voltage for the 42-PDIP which connects only the logic core and the embedded program/data memories.

PWRVDD

Supply voltage for the 42-PDIP which connects only the I/O Pad Drivers.

The application board **must** connect both VDD and PWRVDD to the board supply voltage.

PWRGND

Ground for the 42-PDIP which connects only the I/O Pad Drivers. PWRGND and GND are weakly connected through the common silicon substrate, but not through any metal links. The application board must connect both GND and PWRGND to the board ground.

Port 0

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

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

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

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 1 output buffers can sink/source six TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the weak 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}, 150 µA typical) because of the weak internal pull-ups.

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

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

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

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

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source six TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the weak 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}, 150 µA typical) because of the weak internal pull-ups.

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

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

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source six TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the weak 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}, 150 µA typical) because of the weak internal 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 AT89S8253, as shown in the following table.

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

Note: 1. All pins in ports 1 and 2 and almost all pins in port 3 (the exceptions are P3.2 $\overline{INT0}$ and P3.3 $\overline{INT1}$) have their inputs disabled in the Power-down mode. Port pins P3.2 ($\overline{INT0}$) and P3.3 ($\overline{INT1}$) are active even in Power-down mode (to be able to sense an interrupt request to exit the Power-down mode) and as such still have their weak internal pull-ups turned on.

0 RST

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

1 ALE/$\overline{PROG}$

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

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

If desired, ALE operation can be disabled by setting bit 0 of the AUXR SFR at 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.

2 $\overline{PSEN}$

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

When the AT89S8253 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

3 $\overline{EA}$/VPP

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

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

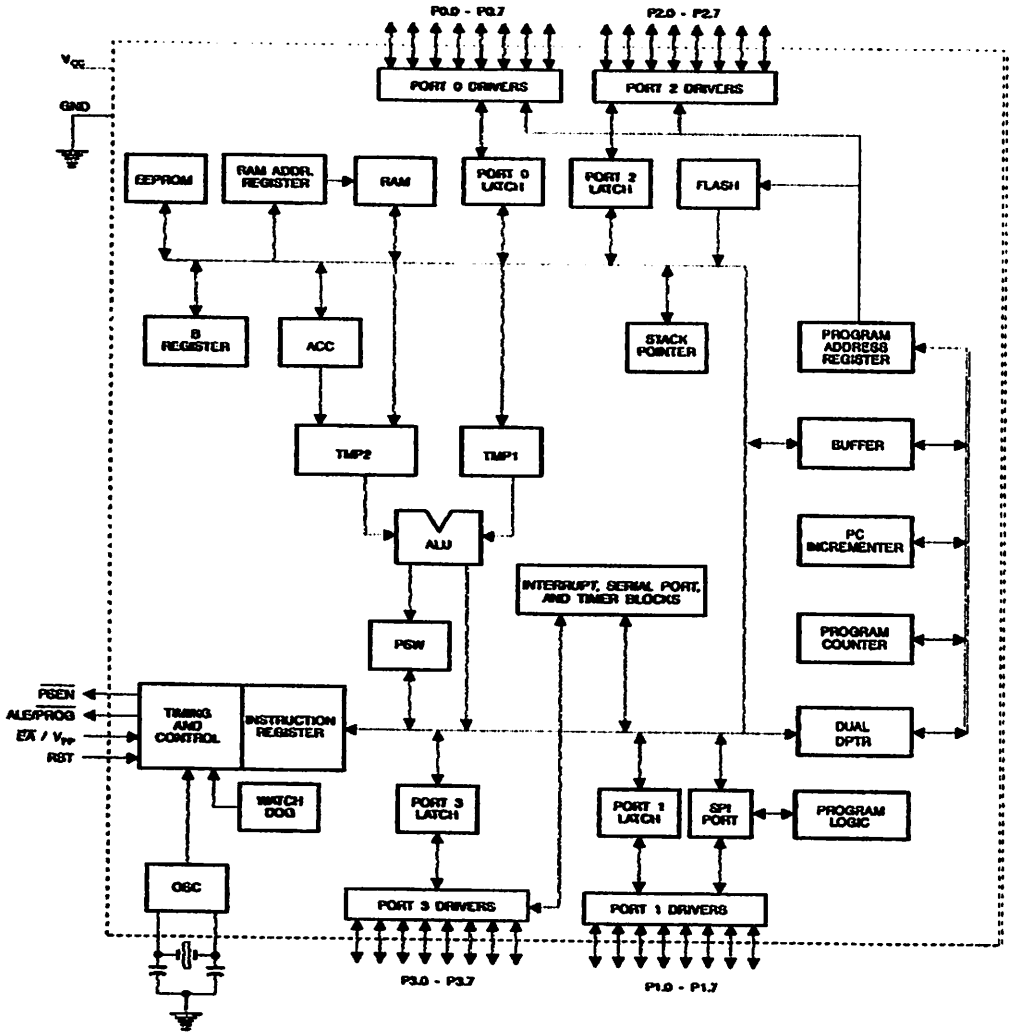
4 XTAL1

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

5 XTAL2

Output from the inverting oscillator amplifier.

Block Diagram



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 5-1.

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 generally return random data, and write accesses will have an indeterminate effect.

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

ble 5-1. AT89S8253 SFR Map and Reset Values

F8H								0FFH
F0H	B 00000000							0F7H
E8H								0EFH
E0H	ACC 00000000							0E7H
D8H								0DFH
D0H	PSW 00000000					SPCR 00000100		0D7H
C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
C0H								0C7H
B8H	IP XX000000	SADEN 00000000						0BFH
B0H	P3 11111111							IPH XX000000 0B7H
A8H	IE 0X000000	SADDR 00000000	SPSR 000XXX00					0AFH
A0H	P2 11111111						WDTRST (Write Only)	WDTCON 0000 0000 0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111						EECON XX000011	97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXXXXXX0	CLKREG XXXXXXX0 8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR ########	PCON 00XX0000 87H

e: # means: 0 after cold reset and unchanged after warm reset.

Auxiliary Register

e AUXR Register contains a single active bit called DISALE.

ble 5-2. AUXR – Auxiliary Register

UXR Address = 8EH — Reset Value = XXXX XXX0B

ot Bit Addressable

	–	–	–	–	–	–	Intel_Pwd_Exit	DISALE
t	7	6	5	4	3	2	1	0

ymbol	Function
tel_Pwd_Exit	When set, this bit configures the interrupt driven exit from power-down to resume execution on the rising edge of the interrupt signal. When this bit is cleared, the execution resumes after a self-timed interval (nominal 2 ms) referenced from the falling edge of the interrupt signal.
SALE	When DISALE = 0, ALE is emitted at a constant rate of 1/6 the oscillator frequency (except during MOVX when 1 ALE pulse is missing). When DISALE = 1, ALE is active only during a MOVX or MOVC instruction.

Clock Register

e CLKREG register contains a single active bit called X2.

ble 5-3. CLKREG – Clock Register

LKREG Address = 8FH — Reset Value = XXXX XXX0B

ot Bit Addressable

	–	–	–	–	–	–	–	X2
t	7	6	5	4	3	2	1	0

ymbol	Function
2	When X2 = 0, the oscillator frequency (at XTAL1 pin) is internally divided by 2 before it is used as the device system frequency. When X2 = 1, the divider by 2 is no longer used and the XTAL1 frequency becomes the device system frequency. This enables the user to choose a 6 MHz crystal instead of a 12 MHz crystal, for example, in order to reduce EMI.

SPI Registers

Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (see Table 14-1 on page 25) and SPSR (see Table 14-2 on page 26). The SPI data bits are contained in the SPDR register. In normal SPI mode, writing the SPI data register during serial data transfer sets the Write Collision bit (WCOL) in the SPSR register. In enhanced SPI mode, the SPDR is also write double-buffered because WCOL works as a Write Buffer Full Flag instead of being a collision flag. The values in SPDR are not changed by Reset.

Interrupt Registers

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

IPH bits have the same functions as IP bits, except IPH has higher priority than IP. By using IPH in conjunction with IP, a priority level of 0, 1, 2, or 3 may be set for each interrupt.

Dual Data Pointer Registers

To facilitate accessing both internal EEPROM and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H - 83H and DP1 at 84H - 85H. Bit DPS = 0 in SFR EECON 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), 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.

Data Memory – EEPROM and RAM

The AT89S8253 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

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

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

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

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

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

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

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

During program execution mode (using the MOVX instruction) there is an auto-erase capability at the byte level. This means that the user can update or modify a single EEPROM byte location in real-time without affecting any other bytes.

The EEMWE bit in the EECON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 4 ms. The progress of EEPROM write can be monitored by reading the RDY/$\overline{BSY}$ bit (read-only) in SFR EECON. RDY/$\overline{BSY}$ = 0 means programming is still in progress and RDY/$\overline{BSY}$ = 1 means an EEPROM write cycle is completed and another write cycle can be initiated. Bit EELD in EECON controls whether the next MOVX instruction will only load the write buffer of the EEPROM or will actually start the programming cycle. By setting EELD, only load will occur. Before the last MOVX in a given page of 32 bytes, EELD should be cleared so that after the last MOVX the entire page will be programmed at the same time. This way, 32 bytes will only require 4 ms of programming time instead of 128 ms required in single byte programming.

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

Memory Control Register

The EECON register contains control bits for the 2K bytes of on-chip data EEPROM. It also contains the control bit for the dual data pointer.

ble 6-1. EECON – Data EEPROM Control Register

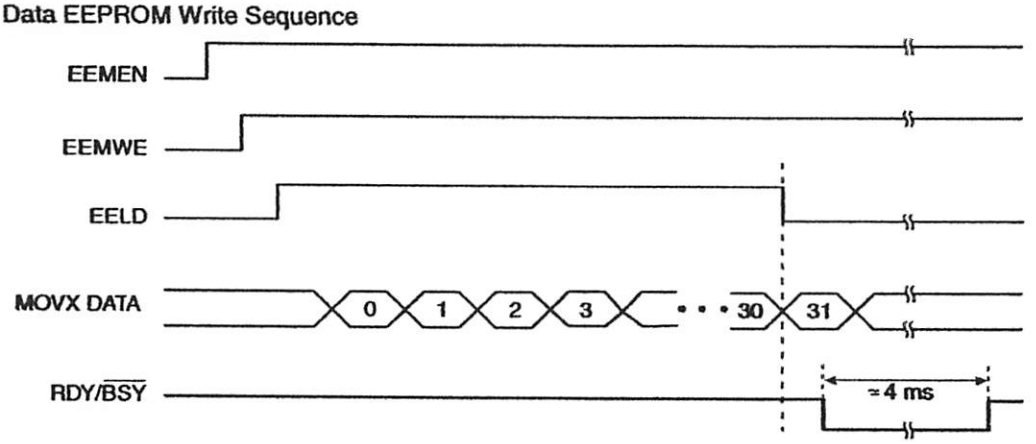
ECON Address = 96H — Reset Value = XX00 0011B

ot Bit Addressable

Bit	–	–	EELD	EEMWE	EEMEN	DPS	RDY/$\overline{BSY}$	$\overline{WRTINH}$
	7	6	5	4	3	2	1	0

ymbol	Function
ELD	EEPROM data memory load enable bit. Used to implement Page Mode Write. A MOVX instruction writing into the data EEPROM will not initiate the programming cycle if this bit is set, rather it will just load data into the volatile data buffer of the data EEPROM memory. Before the last MOVX, reset this bit and the data EEPROM will program all the bytes previously loaded on the same page of the address given by the last MOVX instruction.
EMWE	EEPROM data memory write enable bit. Set this bit to 1 before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to 0 after EEPROM write is completed.
EMEN	Internal EEPROM access enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory if the address used is less than 2K. When EEMEN = 0 or the address used is □ 2K, MOVX with DPTR accesses external data memory.
PS	Data pointer register select. DPS = 0 selects the first bank of data pointer register, DP0, and DPS = 1 selects the second bank, DP1.
DY/$\overline{BSY}$	RDY/$\overline{BSY}$ (Ready/$\overline{Busy}$) flag for the data EEPROM memory. This is a read-only bit which is cleared by hardware during the programming cycle of the on-chip EEPROM. It is also set by hardware when the programming is completed. Note that RDY/$\overline{BSY}$ will be cleared long after the completion of the MOVX instruction which has initiated the programming cycle.
$\overline{RTINH}$	$\overline{WRTINH}$ (Write Inhibit) is a READ-ONLY bit which is cleared by hardware when V_{cc} is too low for the programming cycle of the on-chip EEPROM to be executed. When this bit is cleared, an ongoing programming cycle will be aborted or a new programming cycle will not start.

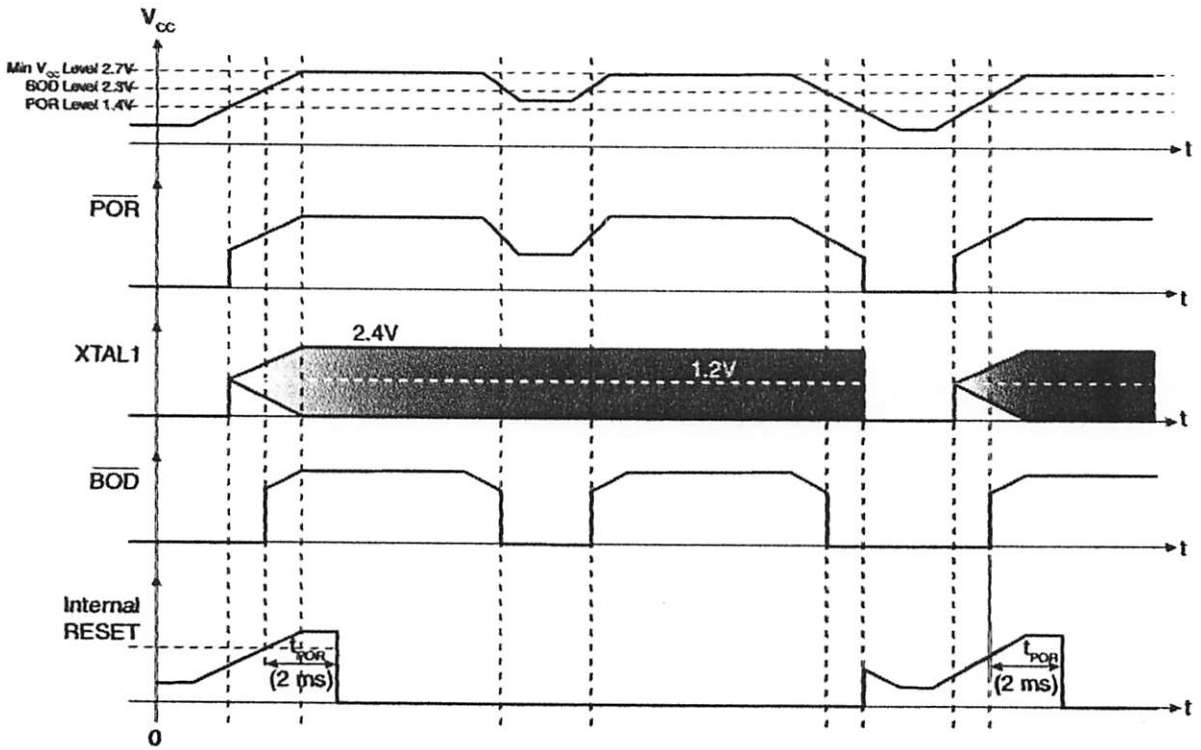
ure 6-1. Data EEPROM Write Sequence



Power-On Reset

A Power-On Reset (POR) is generated by an on-chip detection circuit. The detection level is nominally 1.4V. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the start-up reset or to detect a supply voltage failure in devices without a brown-out detector. The POR circuit ensures that the device is reset from power-on. When V_{CC} reaches the Power-on Reset threshold voltage, the POR delay counter determines how long the device is kept in POR after V_{CC} rise, nominally 2 ms. The POR signal is activated again, without any delay, when V_{CC} falls below the POR threshold level. A Power-On Reset (i.e. a cold reset) will set the POF flag in PCON.

ure 7-1. Power-up and Brown-out Detection Sequence



Brown-out Reset

The AT89S8253 has an on-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level of 2.4V (max). The trigger level for the BOD is nominally 2.2V. The purpose of the BOD is to ensure that if V_{CC} fails or dips while executing at speed, the system will gracefully enter reset without the possibility of errors induced by incorrect execution. When V_{CC} decreases to a value below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the BOD delay counter starts the MCU after the timeout period has expired in approximately 2 ms.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) counts instruction cycles. The prescaler bits, PS0, PS1 and PS2 in SFR WDTCON are used to set the period of the Watchdog Timer from 16K to 2048K instruction cycles. The available timer periods are shown in Table 8-1. The WDT time-out period is dependent upon the external clock frequency.

The WDT is disabled by Power-on Reset and during Power-down mode. When WDT times out without being serviced or disabled, an internal RST pulse is generated to reset the CPU. See Table 8-1 for the WDT period selections.

Table 8-1. Watchdog Timer Time-out Period Selection

WDT Prescaler Bits			Period (Nominal for F_{CLK} = 12 MHz)
PS2	PS1	PS0	
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Watchdog Control Register

WDTCON register contains control bits for the Watchdog Timer (shown in Table 8-2).

le 8-2. WDTCON – Watchdog Control Register

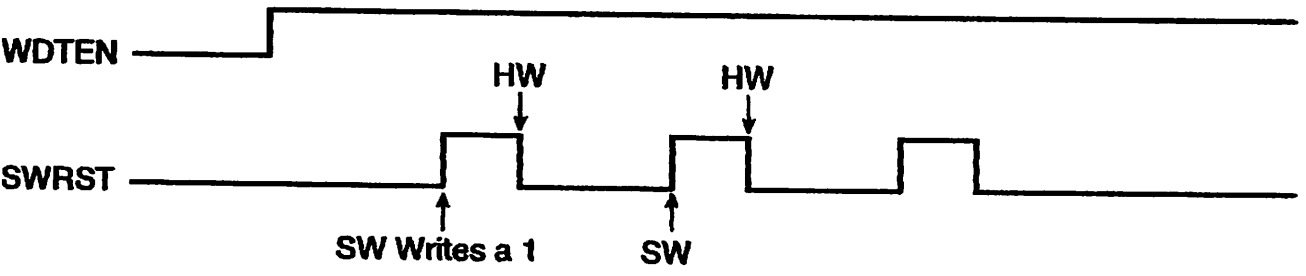
DTCON Address = A7H — Reset Value = 0000 0000B

t Bit Addressable

PS2	PS1	PS0	WDIDLE	DISRTO	HWDT	WSWRST	WDTEN
7	6	5	4	3	2	1	0

mbol	Function
:2 :1 :0	Prescaler bits for the watchdog timer (WDT). When all three bits are cleared to 0, the watchdog timer has a nominal period of 16K machine cycles, (i.e. 16 ms at a XTAL frequency of 12 MHz in normal mode or 6 MHz in x2 mode). When all three bits are set to 1, the nominal period is 2048K machine cycles, (i.e. 2048 ms at 12 MHz clock frequency in normal mode or 6 MHz in x2 mode).
DIDLE	Enable/disable the Watchdog Timer in IDLE mode. When WDIDLE = 0, WDT continues to count in IDLE mode. When WDIDLE = 1, WDT freezes while the device is in IDLE mode.
SRTO	Enable/disable the WDT-driven Reset Out (WDT drives the RST pin). When DISRTO = 0, the RST pin is driven high after WDT times out and the entire board is reset. When DISRTO = 1, the RST pin remains only as an input and the WDT resets only the microcontroller internally after WDT times out.
VDT	Hardware mode select for the WDT. When HWDT = 0, the WDT can be turned on/off by simply setting or clearing WDTEN in the same register (this is the software mode for WDT). When HWDT = 1, the WDT has to be set by writing the sequence 1EH/E1H to the WDTRST register (with address 0A6H) and after being set in this way, WDT cannot be turned off except by reset, warm or cold (this is the hardware mode for WDT). To prevent the hardware WDT from resetting the entire device, the same sequence 1EH/E1H must be written to the same WDTRST SFR before the timeout interval.
SWRST	Watchdog software reset bit. If HWDT = 0 (i.e. WDT is in software controlled mode), when set by software, this bit resets WDT. After being set by software, WSWRST is reset by hardware during the next machine cycle. If HWDT = 1, this bit has no effect, and if set by software, it will not be cleared by hardware.
DTEN	Watchdog software enable bit. When HWDT = 0 (i.e. WDT is in software-controlled mode), this bit enables WDT when set to 1 and disables WDT when cleared to 0 (it does not reset WDT in this case, but just freezes the existing counter state). If HWDT = 1, this bit is READ-ONLY and reflects the status of the WDT (whether it is running or not).

ire 8-1. Software Mode – Watchdog Timer Sequence



Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8253 operate the same way as Timer 0 and Timer 1 in the AT89S51 and AT89S52. For more detailed information on the Timer/Counter operation, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (see Table 10-2 on page 15). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 10-2.

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

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

Table 10-1. Timer 2 Operating Modes

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

le 10-2. T2CON – Timer/Counter 2 Control Register

CON Address = 0C8H — Reset Value = 0000 0000B

Addressable

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
7	6	5	4	3	2	1	0

mbol	Function
2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
F2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
LK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
LK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

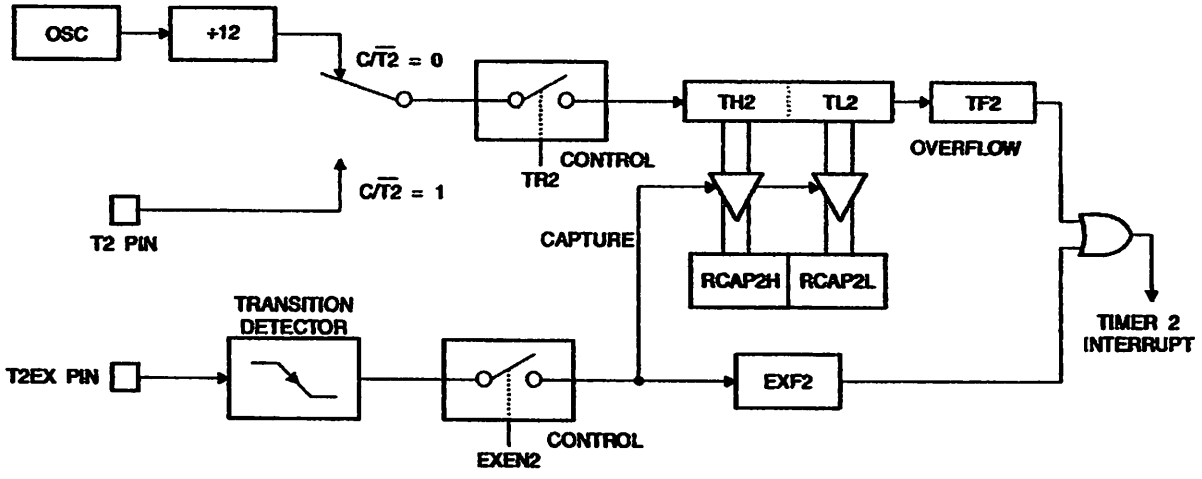
1 Timer 2 Registers

Control and status bits are contained in registers T2CON (see Table 10-2) and T2MOD (see Table 10-3) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

2 Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 10-1.

ure 10-1. Timer 2 in Capture Mode



.3 Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 10-3). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

ble 10-3. T2MOD – Timer 2 Mode Control Register

2MOD Address = 0C9H — Reset Value = XXXX XX00B

ot Bit Addressable

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

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

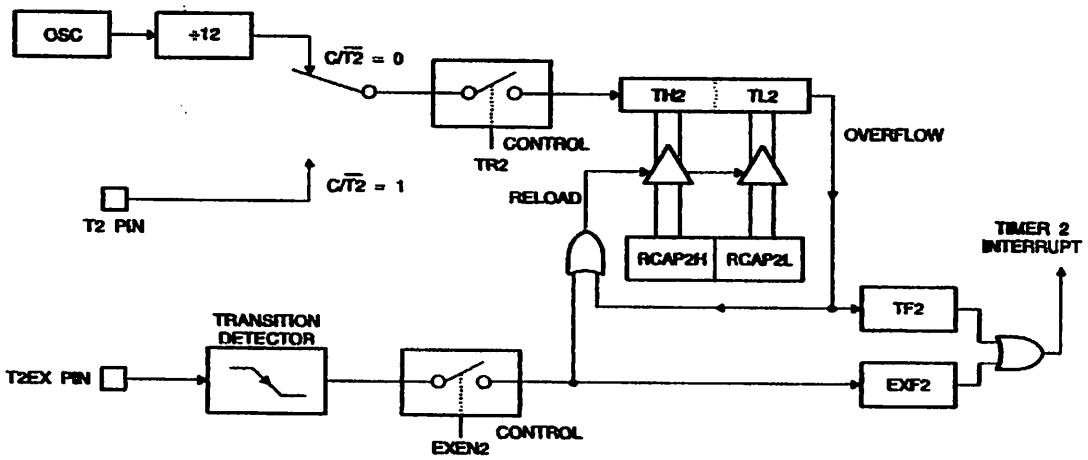
Figure 10-2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 10-3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

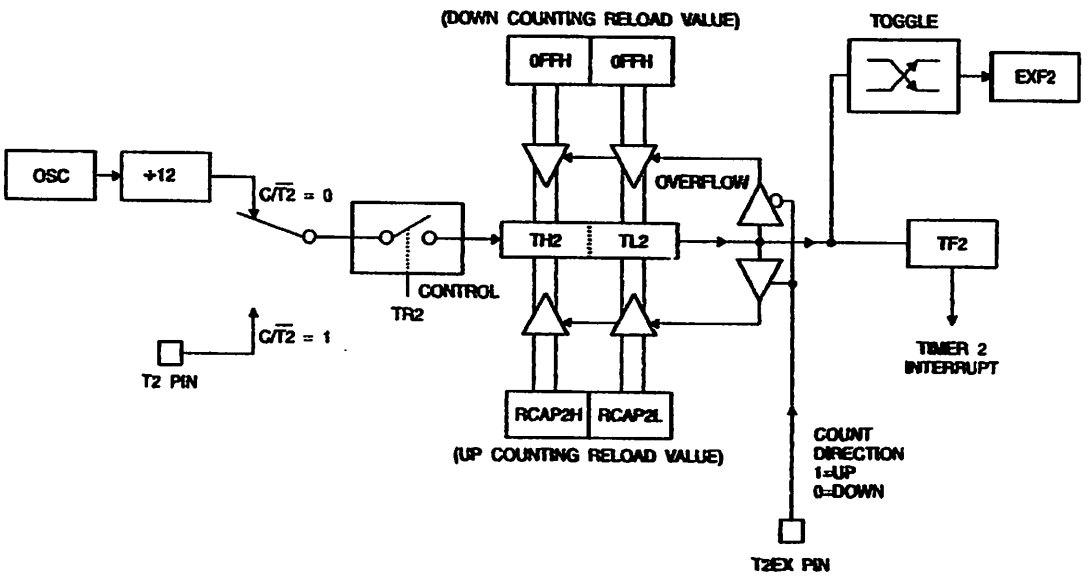
A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

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

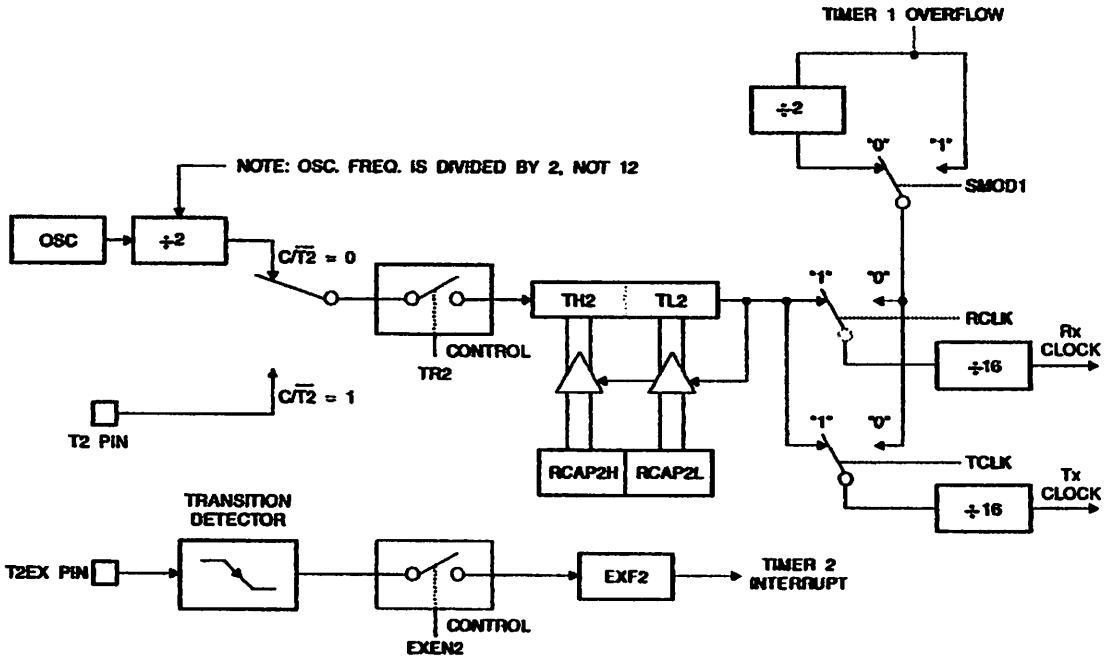
ure 10-2. Timer 2 in Auto Reload Mode (DCEN = 0)



ure 10-3. Timer 2 Auto Reload Mode (DCEN = 1 Timer 2 Auto Reload Mode (DCEN = 1)



ure 10-4. Timer 2 in Baud Rate Generator Mode



. Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 10-2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 10-4.

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

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

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

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

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

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

Timer 2 as a baud rate generator is shown in Figure 10-4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

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

. Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 12-1. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz (for a 16 MHz operating frequency).

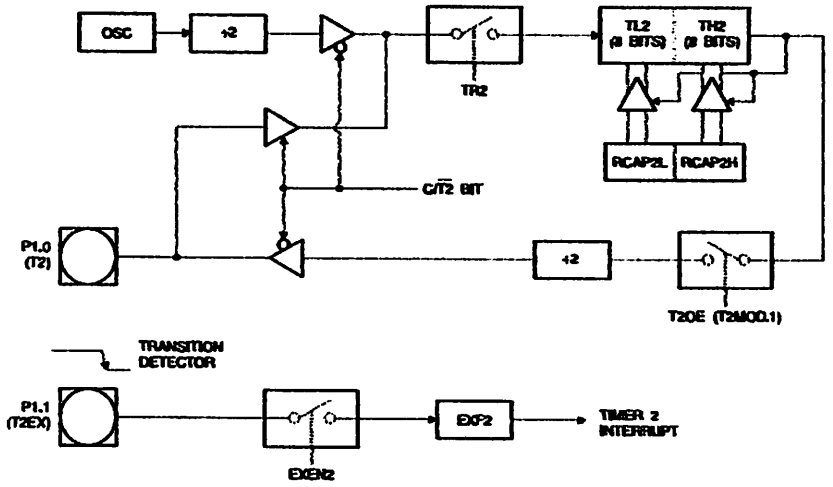
To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

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

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

ure 12-1. Timer 2 in Clock-out Mode



. UART

The UART in the AT89S8253 operates the same way as the UART in the AT89S51 and AT89S52. For more detailed information on the UART operation, please click on the document link below:

http://www.atmel.com/dyn/resources/prod_documents/DOC4316.PDF

1 Enhanced UART

In addition to all of its usual modes, the UART can perform framing error detection by looking for missing stop bits, and automatic address recognition. The UART also fully supports multiprocessor communication as does the standard 80C51 UART.

When used for framing error detect, the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE, SCON.7 can only be cleared by software.

.1 Automatic Address Recognition

Automatic Address Recognition is a feature which allows the UART to recognize certain addresses in the serial bit stream by using hardware to make the comparisons. This feature saves a great deal of software overhead by eliminating the need for the software to examine every serial address which passes by the serial port. This feature is enabled by setting the SM2 bit in SCON. In the 9-bit UART modes, mode 2 and mode 3, the Receive Interrupt flag (RI) will be automatically set when the received byte contains either the "Given" address or the "Broadcast" address. The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data.

The 8-bit mode is called mode 1. In this mode the RI flag will be set if SM2 is enabled and the information received has a valid stop bit following the 8 address bits and the information is either a Given or Broadcast address.

Mode 0 is the Shift Register mode and SM2 is ignored.

Using the Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the given slave address or addresses. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. SADEN is used to define which bits in the SADDR are to be used and which bits are "don't care". The SADEN mask can be logically ANDed with the SADDR to create the "Given" address which the master will use for addressing each of the slaves. Use of the Given address allows multiple slaves to be recognized while excluding others. The following examples will help to show the versatility of this scheme:

Slave 0

SADDR = 1100 0000

SADEN = 1111 1101

Given = 1100 00X0

Slave 1

SADDR = 1100 0000

SADEN = 1111 1110

Given = 1100 000X

In the previous example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires a 0 in bit 0 and it ignores bit 1. Slave 1 requires a 0 in bit 1 and bit 0 is ignored. A unique address for slave 0 would be 1100 0010 since slave 1 requires a 0 in bit 1. A unique address for slave 1 would be 1100 0001 since a 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 1100 0000.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Slave 0

SADDR = 1100 0000

SADEN = 1111 1001

Given = 1100 0XX0

Slave 1

SADDR = 1110 0000

SADEN = 1111 1010

Given = 1110 0X0X

Slave 2

SADDR = 1110 0000

SADEN = 1111 1100

Given = 1110 00XX

In the previous example the differentiation among the 3 slaves is in the lower 3 address bits. Slave 0 requires that bit 0 = 0 and it can be uniquely addressed by 1110 0110. Slave 1 requires that bit 1 = 0 and it can be uniquely addressed by 1110 and 0101. Slave 2 requires that bit 2 = 0 and its unique address is 1110 0011. To select Slaves 0 and 1 and exclude Slave 2, use address 1110 0100, since it is necessary to make bit 2 = 1 to exclude slave 2.

The Broadcast Address for each slave is created by taking the logical OR of SADDR and SADEN. Zeros in this result are trended as don't-cares. In most cases, interpreting the don't-cares as ones, the broadcast address will be FF hexadecimal.

Upon reset SADDR (SFR address 0A9H) and SADEN (SFR address 0B9H) are loaded with 0s. This produces a given address of all "don't cares" as well as a Broadcast address of all "don't cares". This effectively disables the Automatic Addressing mode and allows the microcontroller to use standard 80C51-type UART drivers which do not make use of this feature.

le 13-1. PCON – Power Control Register

ON Address = 87H — Reset Value = 00xx 0000B

Addressable

SMOD1	SMOD0	–	POF	GF1	GF0	PD	IDL
7	6	5	4	3	2	1	0

mbol	Function
MOD1	Double Baud Rate bit. Doubles the baud rate of the UART in Modes 1, 2, or 3.
MOD0	Frame Error Select. When SMOD0 = 1, SCON.7 is SM0. When SMOD0 = 1, SCON.7 is FE. Note that FE will be set after a frame error regardless of the state of SMOD0.
OF	Power Off Flag. POF is set to "1" during power up (i.e. cold reset). It can be set or reset under software control and is not affected by RST or BOD (i.e. warm resets).
F1, GF0	General-purpose Flags
)	Power-down bit. Setting this bit activates power-down operation.
L	Idle Mode bit. Setting this bit activates Idle mode operation

le 13-2. SCON – Serial Port Control Register

ON Address = 98H — Reset Value = 0000 0000B

Addressable

SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI
7	6	5	4	3	2	1	0

(SMOD0 = 0/1)[1]

mbol	Function
	Framing error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit. FE will be set regardless of the state of SMOD0.
M0	Serial Port Mode Bit 0, (SMOD0 must = 0 to access bit SM0)
M1	Serial Port Mode Bit 1 SM0 / SM1 / Mode / Description / Baud Rate[2] 0 / 0 / 0 / shift register / $f_{osc}/12$ 0 / 1 / 1 / 8-bit UART / variable 1 / 0 / 2 / 9-bit UART / $f_{osc}/64$ or $f_{osc}/32$ 1 / 1 / 3 / 9-bit UART / variable
M2	Enables the Automatic Address Recognition feature in modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
EN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
B8	The 9th data bit that will be transmitted in modes 2 and 3. Set or clear by software as desired.
B8	In modes 2 and 3, the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

es: 1. SMOD0 is located at PCON.6.

2. f_{osc} = oscillator frequency.

Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8253 and peripheral devices or between multiple AT89S8253 devices. The AT89S8253 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- Maximum Bit Frequency = f/4 (f/2 if in x2 Clock Mode)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates in Master Mode
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Double-Buffered Receive
- Double-Buffered Transmit (Enhanced Mode only)
- Wakeup from Idle Mode (Slave Mode only)

The interconnection between master and slave CPUs with SPI is shown in Figure 14-1. The four pins in the interface are Master-In/Slave-Out (MISO), Master-Out/Slave-In (MOSI), Shift Clock (SCK), and Slave Select ($\overline{SS}$). The SCK pin is the clock output in master mode, but is the clock input in slave mode. The MSTR bit in SPCR determines the directions of MISO and MOSI. Also notice that MOSI connects to MOSI and MISO to MISO. In master mode, $\overline{SS}$/P1.4 is ignored and may be used as a general-purpose input or output. In slave mode, $\overline{SS}$ must be driven low to select an individual device as a slave. When $\overline{SS}$ is driven high, the slave's SPI port is deactivated and the MOSI/P1.5 pin can be used as a general-purpose input.

Figure 14-1. SPI Master-Slave Interconnection

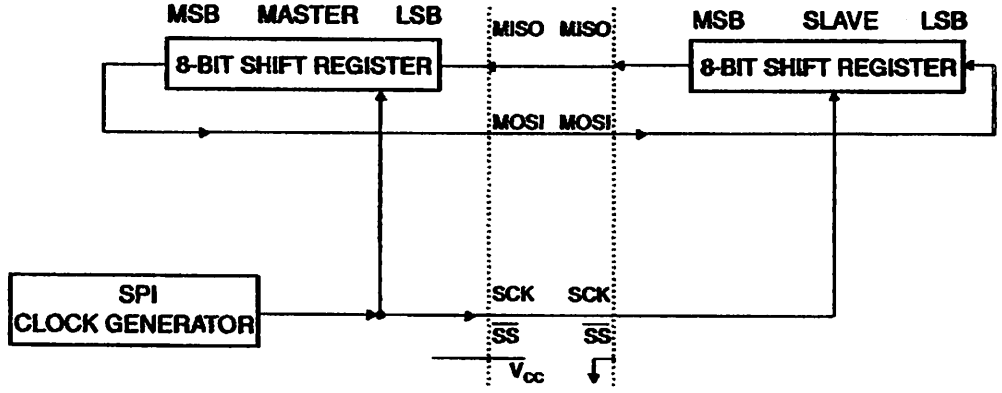
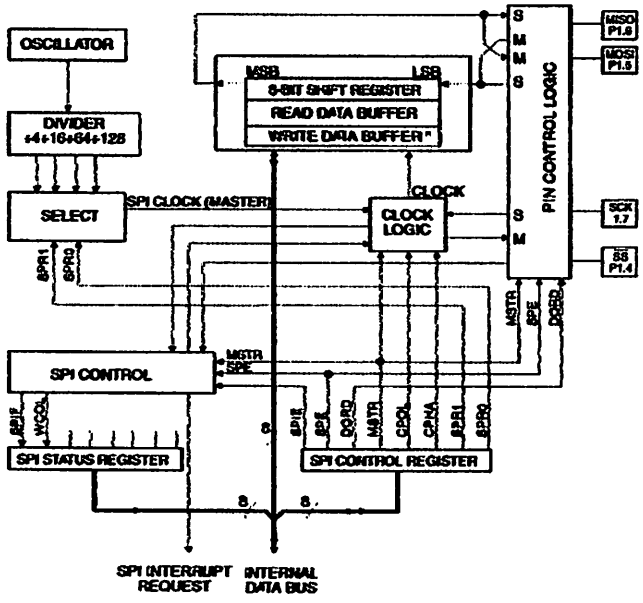


Figure 14-2. SPI Block Diagram



Note: 1. *The Write Data Buffer is only used in enhanced SPI mode.*

The SPI has two modes of operation: normal (non-buffered write) and enhanced (buffered write). In normal mode, writing to the SPI data register (SPDR) of the master CPU starts the SPI clock generator and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. Transmission may start after an initial delay while the clock generator waits for the next full bit slot of the specified baud rate. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF) and transferring the received byte to the read buffer (SPDR). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested. Note that SPDR refers to either the write data buffer or the read data buffer, depending on whether the access is a write or read. In normal mode, because the write buffer is transparent (and a write access to SPDR will be directed to the shift buffer), any attempt to write to SPDR while a transmission is in progress will result in a write collision with WCOL set. However, the transmission will still complete normally, but the new byte will be ignored and a new write access to SPDR will be necessary.

Enhanced mode is similar to normal mode except that the write buffer holds the next byte to be transmitted. Writing to SPDR loads the write buffer and sets WCOL to signify that the buffer is full and any further writes will overwrite the buffer. WCOL is cleared by hardware when the buffered byte is loaded into the shift register and transmission begins. If the master SPI is currently idle, i.e. if this is the first byte, then after loading SPDR, transmission of the byte starts and WCOL is cleared immediately. While this byte is transmitting, the next byte may be written to SPDR. The Load Enable flag (LDEN) in SPSR can be used to determine when transmission has started. LDEN is asserted during the first four bit slots of a SPI transfer. The master CPU should first check that LDEN is set and that WCOL is cleared before loading the next byte. In enhanced mode, if WCOL is set when a transfer completes, i.e. the next byte is available, then the SPI immediately loads the buffered byte into the shift register, resets WCOL, and continues transmission without stopping and restarting the clock generator. As long as the CPU can keep the write buffer full in this manner, multiple bytes may be transferred with minimal latency between bytes.

e 14-1. SPCR – SPI Control Register

CR Address = D5H — Reset Value = 0000 0100B

Bit Addressable

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

mbol	Function
IE	SPI interrupt enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts.
E	SPI enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel.
RD	Data order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission.
TR	Master/slave select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects slave SPI mode.
OL	Clock polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI clock phase and polarity control.
HA	Clock phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI clock phase and polarity control.
R0 R1	SPI clock rate select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, F_{osc}, is as follows: **SPR1SPR0SCK** 00f/4 (f/2 in x2 mode) 01f/16 (f/8 in x2 mode) 10f/64 (f/32 in x2 mode) 11f/128 (f/64 in x2 mode)

s: 1. Set up the clock mode before enabling the SPI: set all bits needed in SPCR except the SPE bit, then set SPE.
2. Enable the master SPI prior to the slave device.
3. Slave echoes master on next Tx if not loaded with new data.

ıle 14-2. SPSR – SPI Status Register

'SR Address = AAH

Reset Value = 000X XX00B

ıt Bit Addressable

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

'mbol	Function
'IF	SPI interrupt flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register followed by reading/writing the SPI data register.
COL	When ENH = 0: Write collision flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register followed by reading/writing the SPI data register. When ENH = 1: WCOL works in Enhanced mode as Tx Buffer Full. Writing during WCOL = 1 in enhanced mode will overwrite the waiting data already present in the Tx Buffer. In this mode, WCOL is no longer reset by the SPIF reset but is reset when the write buffer has been unloaded into the serial shift register.
)EN	Load enable for the Tx buffer in enhanced SPI mode. When ENH is set, it is safe to load the Tx Buffer while LDEN = 1 and WCOL = 0. LDEN is high during bits 0 - 3 and is low during bits 4 - 7 of the SPI serial byte transmission time frame.
SSO	Disable slave output bit. When set, this bit causes the MISO pin to be tri-stated so more than one slave device can share the same interface with a single master. Normally, the first byte in a transmission could be the slave address and only the selected slave should clear its DISSO bit.
NH	Enhanced SPI mode select bit. When ENH = 0, SPI is in normal mode, i.e. without write double buffering. When ENH = 1, SPI is in enhanced mode with write double buffering. The Tx buffer shares the same address with the SPDR register.

ıle 14-3. SPDR – SPI Data Register

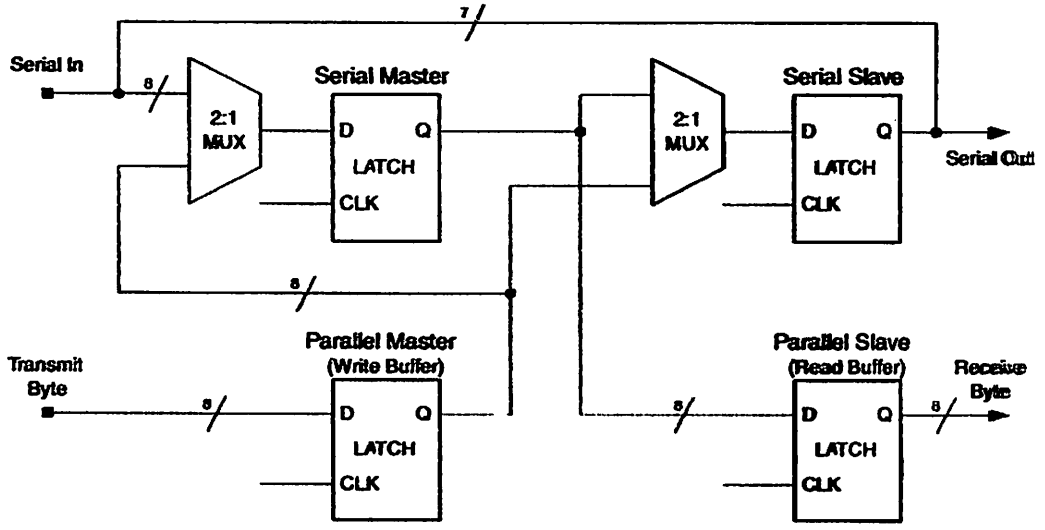
'DR Address = 86H

Reset Value = 00H (after cold reset)
unchanged (after warm reset)

ıt Bit Addressable

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

Figure 14-3. SPI Shift Register Diagram



The CPHA (Clock PHAse), CPOL (Clock POLarity), and SPR (Serial Peripheral clock Rate = baud rate) bits in SPCR control the shape and rate of SCK. The two SPR bits provide four possible clock rates when the SPI is in master mode. In slave mode, the SPI will operate at the rate of the incoming SCK as long as it does not exceed the maximum bit rate. There are also four possible combinations of SCK phase and polarity with respect to the serial data. CPHA and CPOL determine which format is used for transmission. The SPI data transfer formats are shown in Figure 14-4 and Figure 14-5. To prevent glitches on SCK from disrupting the interface, CPHA, CPOL, and SPR should be set up before the interface is enabled, and the master device should be enabled before the slave device(s).

Table 14-4. SPI Master Characteristics

Symbol	Parameter	Min	Max	Units
t_{CLCL}	Oscillator Period	41.6		ns
t_{SCK}	Serial Clock Cycle Time	$4t_{CLCL}$		ns
t_{SHSL}	Clock High Time	$t_{SCK}/2 - 25$		ns
t_{SLSH}	Clock Low Time	$t_{SCK}/2 - 25$		ns
t_{SR}	Rise Time		25	ns
t_{SF}	Fall Time		25	ns
t_{SIS}	Serial Input Setup Time	10		ns
t_{SIH}	Serial Input Hold Time	10		ns
t_{SOH}	Serial Output Hold Time		10	ns
t_{SOV}	Serial Output Valid Time		35	ns

Table 14-5. SPI Slave Characteristics

Symbol	Parameter	Min	Max	Units
t_{CLCL}	Oscillator Period	41.6		ns
t_{SCK}	Serial Clock Cycle Time	$4t_{CLCL}$		ns
t_{SHSL}	Clock High Time	$1.5\ t_{CLCL}$ - 25		ns
t_{SLSH}	Clock Low Time	$1.5\ t_{CLCL}$ - 25		ns
t_{SR}	Rise Time		25	ns
t_{SF}	Fall Time		25	ns
t_{SIS}	Serial Input Setup Time	10		ns
t_{SIH}	Serial Input Hold Time	10		ns
t_{SOH}	Serial Output Hold Time		10	ns
t_{SOV}	Serial Output Valid Time		35	ns
t_{SOE}	Output Enable Time		10	ns
t_{SOX}	Output Disable Time		25	ns
t_{SSE}	Slave Enable Lead Time	$4\ t_{CLCL}$ +50		ns
t_{SSD}	Slave Disable Lag Time	0		ns

Figure 14-4. SPI Master Timing (CPHA = 0)

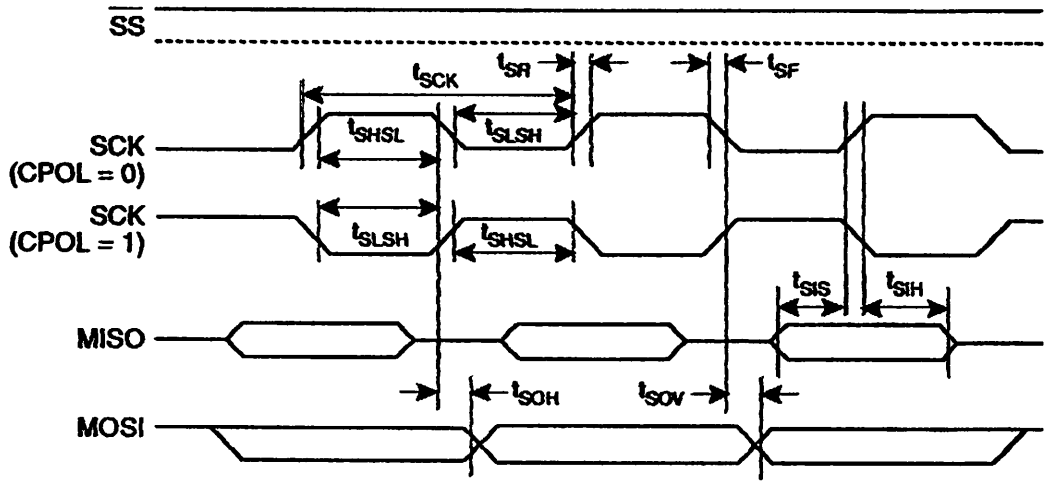


Figure 14-5. SPI Slave Timing (CPHA = 0)

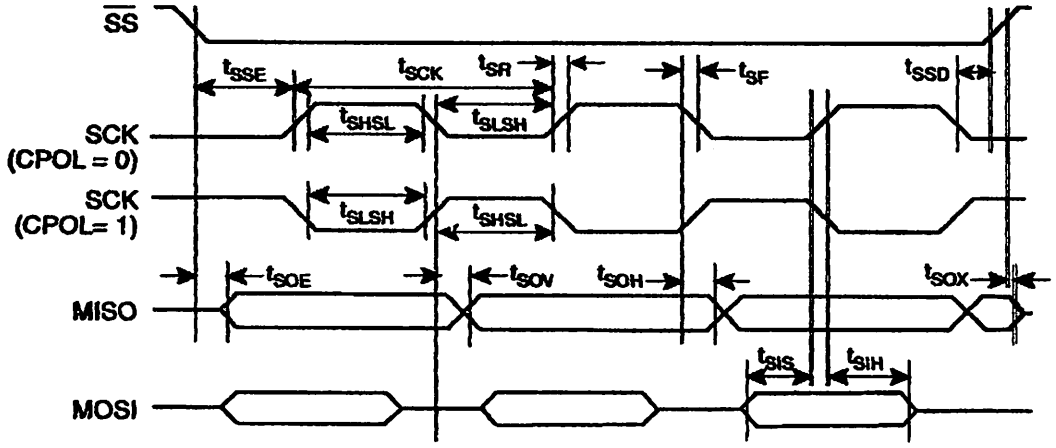


Figure 14-6. SPI Master Timing (CPHA = 1)

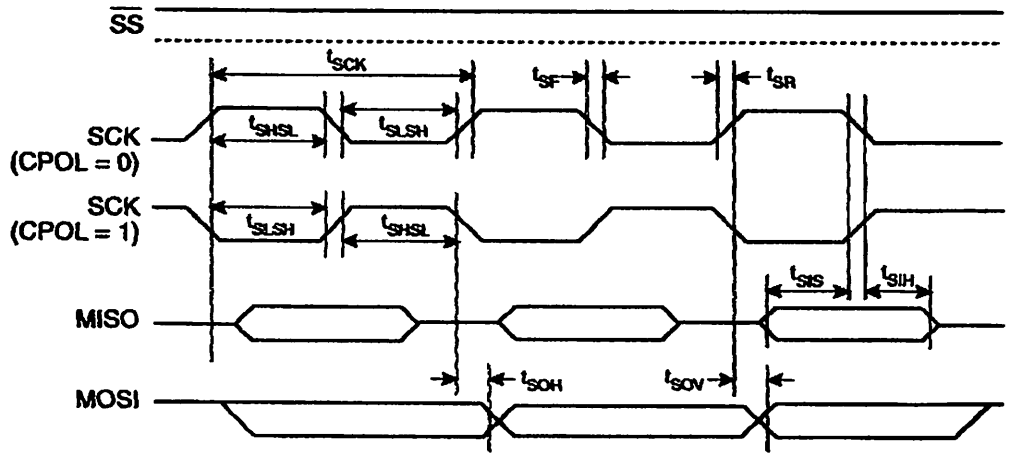
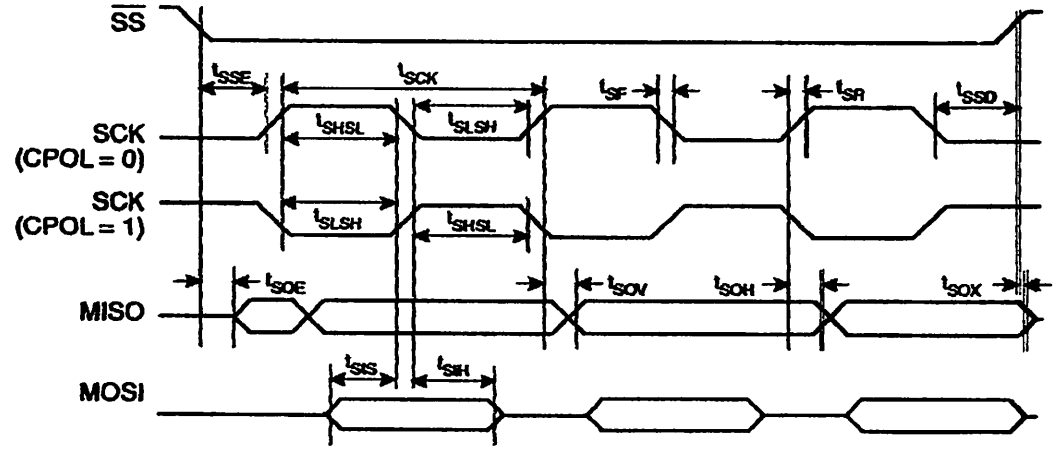
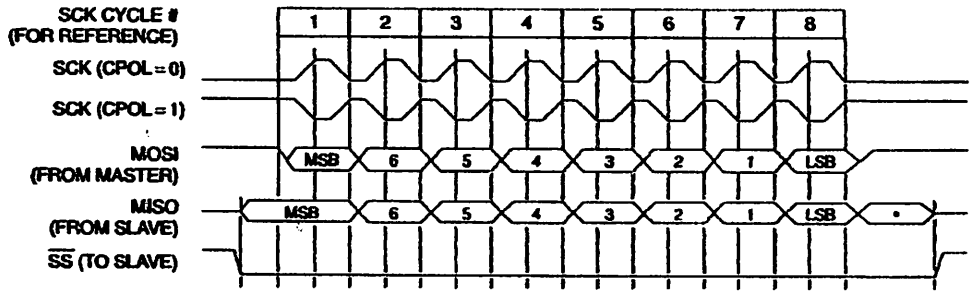


Figure 14-7. SPI Slave Timing (CPHA = 1)

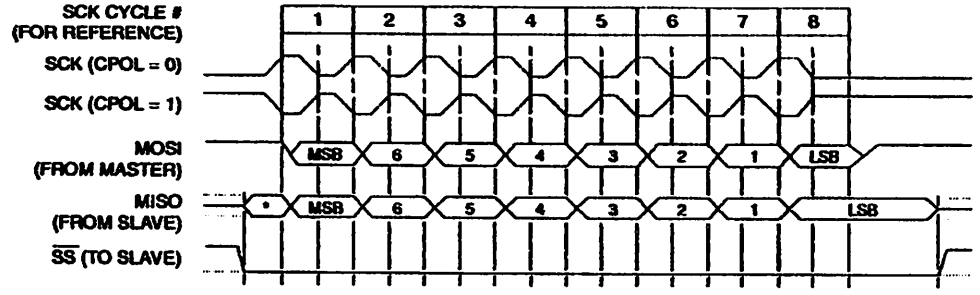


ure 14-8. SPI Transfer Format with CPHA = 0



: *Not defined but normally MSB of character just received

ure 14-9. SPI Transfer Format with CPHA = 1



: *Not defined but normally LSB of previously transmitted character

. Interrupts

The AT89S8253 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 15-1.

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 15-1 shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it may be used in future AT89 products.

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

The serial interrupt is the logical OR of bits RI and TI in register SCON and also bit SPIF in SPSR (if SPIE in SPCR is set). None of these flags is cleared by hardware when the service routine is vectored to. The service routine may have to determine whether the UART or SPI generated the interrupt.

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

errupt	Source	Vector Address
stem Reset	RST or POR or BOD	0000H
ternal Interrupt 0	IE0	0003H
ner 0 Overflow	TF0	000BH
ternal Interrupt 1	IE1	0013H
ner 1 Overflow	TF1	001BH
rial Port	RI or TI or SPIF	0023H

le 15-1. Interrupt Enable (IE) Register

Address = A8H — Reset Value = 0X00 0000B

Addressable

EA	–	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

mbol	Position	Function
	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.
2	IE.5	Timer 2 interrupt enable bit.
	IE.4	SPI and UART interrupt enable bit.
1	IE.3	Timer 1 interrupt enable bit.
1	IE.2	External interrupt 1 enable bit.
0	IE.1	Timer 0 interrupt enable bit.
0	IE.0	External interrupt 0 enable bit.
er software should never write 1s to reserved bits, because they may be used in future AT89 products.		

le 15-2. IP – Interrupt Priority Register

IP = B8H — Reset Value = XX00 0000B

Bit Addressable

	–	–	PT2	PS	PT1	PX1	PT0	PX0
Bit	7	6	5	4	3	2	1	0

mbol	Function
2	Timer 2 Interrupt Priority Low
	Serial Port Interrupt Priority Low
1	Timer 1 Interrupt Priority Low
1	External Interrupt 1 Priority Low
0	Timer 0 Interrupt Priority Low
0	External Interrupt 0 Priority Low

le 15-3. IPH – Interrupt Priority High Register

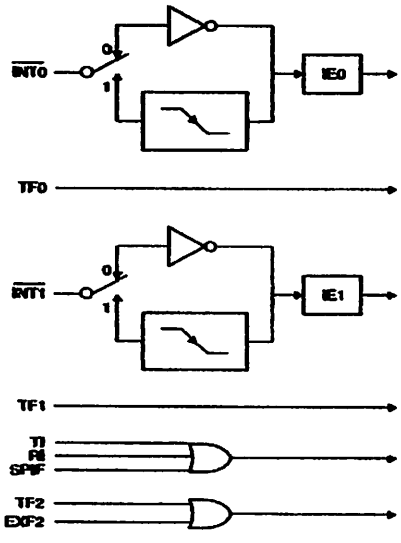
IPH = B7H — Reset Value = XX00 0000B

Not Bit Addressable

	–	–	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
Bit	7	6	5	4	3	2	1	0

mbol	Function
2H	Timer 2 Interrupt Priority High
H	Serial Port Interrupt Priority High
1H	Timer 1 Interrupt Priority High
1H	External Interrupt 1 Priority High
0H	Timer 0 Interrupt Priority High
0H	External Interrupt 0 Priority High

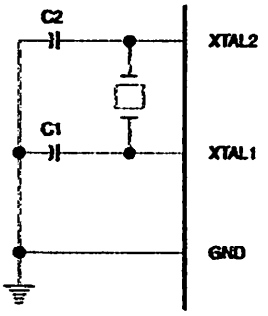
ure 15-1. Interrupt Sources



. 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 16-1. Either a quartz crystal or ceramic resonator may be used. Figures 16-2 and 16-3 illustrate the relationship between sample clock loading and the respective resulting clock amplitudes.

Figure 16-1. Oscillator Connections



Note: C1, C2 = 5 pF ± 5 pF for Crystals
= 5 pF ± 5 pF for Ceramic Resonators

Figure 16-2. Quartz Crystal Clock Source

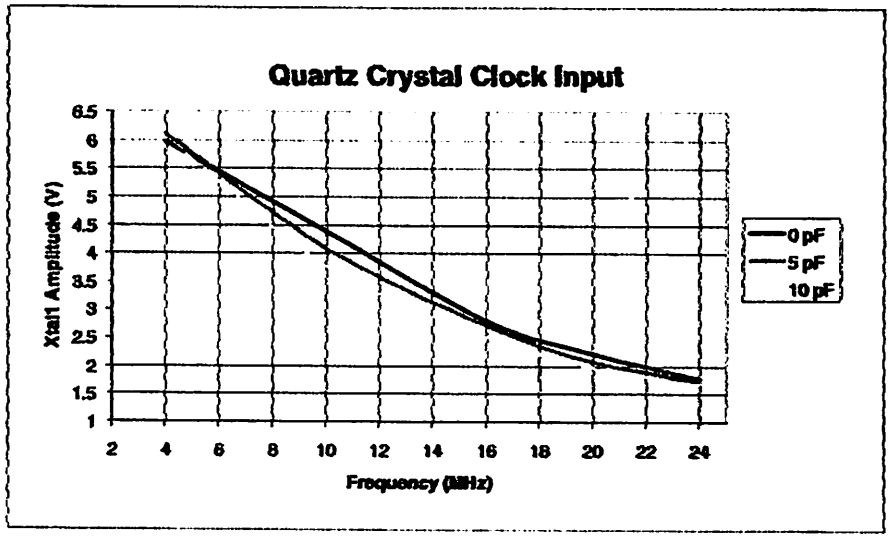
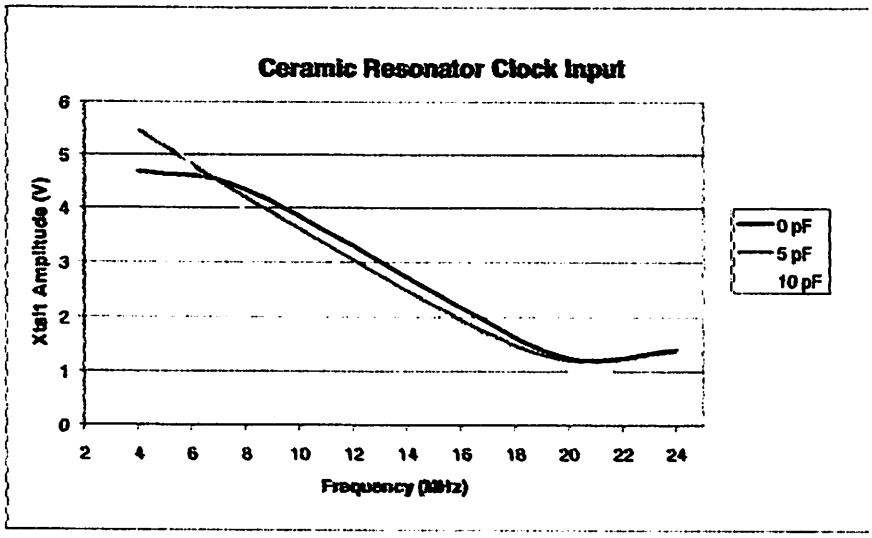
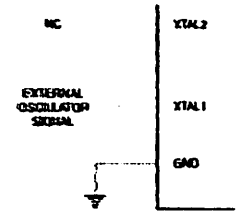


Figure 16-3. Ceramic Resonator Clock Source



To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 16-4.

Figure 16-4. External Clock Drive Configuration



. Idle Mode

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

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

Table 17-1. Status of External Pins During Idle and Power-down Modes

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

. Power-down Mode

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

To exit power-down via an interrupt, external interrupt pin P3.2 or P3.3 must be kept low for at least the specified required crystal oscillator start up time. Afterwards, the interrupt service routine starts at the rising edge of the external interrupt pin if the SFR bit AUXR.1 is set. If AUXR.1 is reset (cleared), execution starts after a self-timed interval of 2 ms (nominal) from the falling edge of the external interrupt pin.

The user should not attempt to enter (or re-enter) the power-down mode for a minimum of 4 µs until after one of the following conditions has occurred: Start of code execution (after any type of reset), or Exit from power-down mode.

. Program Memory Lock Bits

The AT89S8253 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 19-1.

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

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

Table 19-1. Lock Bit Protection Modes[1]

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

Note: 1. U = Unprogrammed
P = Programmed

Programming the Flash and EEPROM

Atmel's AT89S8253 Flash microcontroller offers 12K bytes of In-System reprogrammable Flash code memory and 2K bytes of EEPROM data memory.

The AT89S8253 is normally shipped with the on-chip Flash code and EEPROM data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a parallel programming mode and a serial programming mode. The serial programming mode provides a convenient way to reprogram the AT89S8253 inside the user's system. The parallel programming mode is compatible with conventional third-party Flash or EPROM programmers.

The code and data memory arrays are mapped via separate address spaces in the parallel and serial programming modes: 0000H to 2FFFH for code memory and 000H to 7FFH for data memory.

The code and data memory arrays in the AT89S8253 are programmed byte-by-byte or by page in either programming mode. To reprogram any non-blank byte in the parallel or serial mode, the user needs to invoke the Chip Erase operation first to erase both arrays since there is no built-in auto-erase capability.

Parallel Programming Algorithm: To program and verify the AT89S8253 in the parallel programming mode, the following sequence is recommended (see Figure 26-1):

1. Power-up sequence:
   a. Apply power between V_{CC} and GND pins.
   b. Set RST pin to "H".
   c. Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 ms.
2. Set $\overline{PSEN}$ pin to "L"
   a. ALE pin to "H"
   b. $\overline{EA}$ pin to "H" and all other pins to "H".
3. Raise $\overline{EA}$/VPP to 12V to enable Flash programming, erase or verification. Enable the P3.0 pull-up (10 KΩ typical) for RDY/$\overline{BSY}$ operation.
4. Apply the appropriate combination of "H" or "L" logic levels to pins P3.3, P3.4, P3.5, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
5. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
   a. Apply data to pins P0.0 to P0.7 for write code operation.
6. Pulse ALE/$\overline{PROG}$ once to load a byte in the code memory array, the data memory array, or the lock bits.
7. Repeat steps 5 and 6, changing the address and data for up to 64 bytes in the code memory page or 32 bytes in the data memory (EEPROM) page. When loading a page with individual bytes, the interval between consecutive byte loads should be no longer than 150 µs. Otherwise the device internally times out and assumes that the page load sequence is completed, rejecting any further loads before the page programming sequence has finished. This timing restriction also applies to Page Write of the 64-byte User Row.
8. After the last byte of the current page has been loaded, wait for 5 ms or monitor the RDY/$\overline{BUSY}$ pin until it transitions high. The page write cycle is self-timed and typically takes less than 5 ms.
9. To verify the last byte of the page just programmed, bring pin P3.4 to "L" and read the programmed data at pins P0.0 to P0.7.

10. Repeat steps 4 through 7 changing the address and data for the entire array or until the end of the object file is reached.
11. Power-off sequence:
    a. Tri-state the address and data inputs.
    b. Disable the P3.0 pullup used for RDY/$\overline{BUSY}$ operation.
    c. Set XTAL1 to "L".
    d. Set RST and $\overline{EA}$ pins to "L".
    e. Turn V_{CC} power off.

$\overline{Data}$ Polling: The AT89S8253 features $\overline{DATA}$ Polling to indicate the end of any programming cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last loaded byte will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{DATA}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{BSY}$ output signal. Pin P3.0 is pulled Low after ALE goes High during programming to indicate $\overline{BUSY}$. P3.0 is pulled High again when programming is done to indicate READY. P3.0 needs an external pullup (typical 10 KΩ) when functioning as RDY/$\overline{BSY}$.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel and serial programming modes.

Chip Erase: Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, Chip Erase is initiated by using the proper combination of control signals. The code and data arrays are written with all "1"s during the Chip Erase operation. The User Row will also be erased if the UsrRowProEn fuse (Fuse3) = 0 (enabled state).

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 also takes about 8 ms.

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

Serial Programming Fuse: A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can be disabled via both the Parallel/Serial Programming Modes, but can only be enabled via the Parallel mode.

The AT89S8253 is shipped with the Serial Programming Mode enabled.

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

(030H) = 1EH indicates manufactured by Atmel
(031H) = 73H indicates AT89S8253

. Programming Interface

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

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

. Serial Downloading

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

The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The code and data memory arrays have separate address spaces:

0000H to 2FFFH for code memory and 000H to 7FFH for data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 1.5 MHz.

. Serial Programming Algorithm

To program and verify the AT89S8253 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
   a. Apply power between VCC and GND pins.
   b. Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 12 MHz clock to XTAL1 pin and wait for at least 10 ms with RST pin high and P1.7 (SCK) low.

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 or data array is programmed one byte or one page at a time by supplying the address and data together with the appropriate Write instruction. The write cycle is self-timed and typically takes less than 4.0 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):

1. Set XTAL1 to "L" (if a crystal is not used).
2. Set RST to "L".
3. Turn V_{CC} power off.

. Serial Programming Instruction

) Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 24-1.

ble 24-1. Serial Programming Instruction Set

truction	Instruction Format					Operation
	Byte 1	Byte 2	Byte 3	Byte 4	Byte n	
gramming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx		Enable Serial Programming while RST is high
ip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx		Chip Erase both the 12K and 2K memory arrays
ite Program Memory rte Mode)	0100 0000	xx A13 A12 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 – Byte Mode
ad Program Memory rte Mode)	0010 0000	xx A13 A12 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 – Byte Mode
ite Program Memory ge Mode)	0101 0000	xx A13 A12 A11 A10 A9 A8	A7 A6 00 0000	Byte 0 ... Byte 63		Write data to Program Memory – Page Mode (64 bytes)
ad Program Memory ge Mode)	0011 0000	xx A13 A12 A11 A10 A9 A8	A7 A6 00 0000	Byte 0 ... Byte 63		Read data from Program Memory – Page Mode (64 bytes)
ite Data Memory rte Mode)	1100 0000	xxxx x A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0		Write data to Data Memory – Byte Mode
ad Data Memory rte Mode)	1010 0000	xxxx x A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0		Read data from Data Memory – Byte Mode
ite Data Memory ge Mode)	1101 0000	xxxx x A10 A9 A8	A7 A6 A5 0 0000	Byte 0 ... Byte 31		Write data to Data Memory – Page Mode (32 bytes)
ad Data Memory ge Mode)	1011 0000	xxxx x A10 A9 A8	A7 A6 A5 0 0000	Byte 0 ... Byte 31		Read data from Data Memory – Page Mode (32 bytes)
ite User Fuses	1010 1100	0001 FUSE4 FUSE3 FUSE2 FUSE1	xxxx xxxx	xxxx xxxx		Write user fuse bits (refer to next page for the fuse definitions)
ad User Fuses	0010 0001	xxxx xxxx	xxxx xxxx	xxxx FUSE4 FUSE3 FUSE2 FUSE1		Read back status of user fuse bits
ite Lock Bits	1010 1100	1110 0 LB3 LB2 LB1	xxxx xxxx	xxxx xxxx		Write the lock bits (write a "0" to lock)
ad Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxxx x LB3 LB2 LB1		Read back current status of the lock bits (a programmed lock bit reads back as a "0")
ite User Sgn. Byte	0100 0010	xxxx xxxx	xx A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0		
ad User Sgn. Byte	0010 0010	xxxx xxxx	xx A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0		
ite User Sgn. Page	0101 0010	xxxx xxxx	xxxx xxxx	Byte 0 ... Byte 63		
ad User Sgn. Page	0011 0010	xxxx xxxx	xxxx xxxx	Byte 0 ... Byte 63		
ad ATMEL Sgn. Byte	0010 1000	xxxx xxxx	xx A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0		Read Signature Byte

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

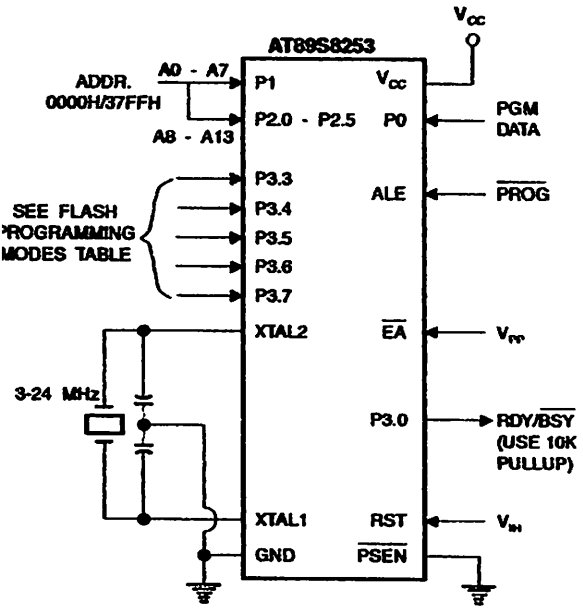
Page Read/Write, the data always starts from byte 0 to 31 or 63. After the command byte and upper address byte are ned, each byte thereafter is treated as data until all 32 or 64 bytes are shifted in/out. Then the next instruction will be fy to be decoded.

. Flash and EEPROM Parallel Programming Modes

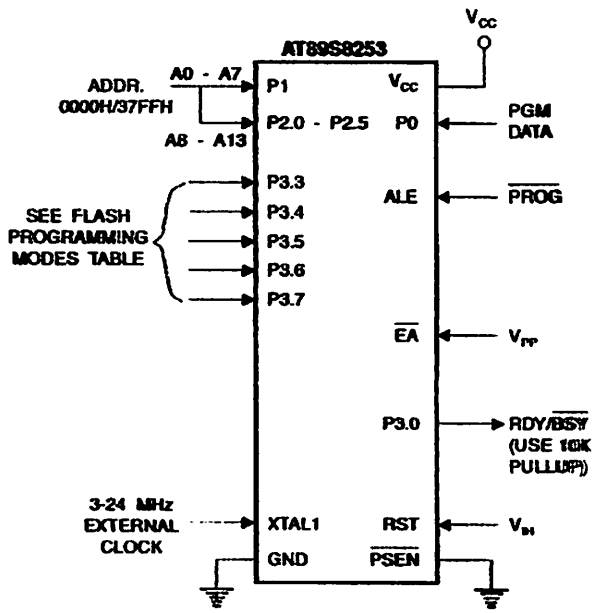
ıde		RST	PSEN	ALE	EA	P3.3	P3.4	P3.5	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0, P1.7:0
rial Prog. Modes[1]		H	h	h								
ip Erase[2]		H	L	1.0 µs	12V	H	L	H	L	L	X	X
ge Write[3][4][5]	12K Code	H	L	1.0 µs	12V	L	H	H	H	H	DI	ADDR
ad	12K Code	H	L	H	12V	L	L	H	H	H	DO	ADDR
ge Write[3][4][6]	2K Data	H	L	1.0 µs	12V	L	H	L	H	H	DI	ADDR
ad	2K Data	H	L	H	12V	L	L	L	H	H	DO	ADDR
ite Lock Bits[2][4]	Bit - 1	H	L	1.0 µs	12V	H	L	H	H	L	D0 = 0	X
	Bit - 2										D1 = 0	X
	Bit - 3										D2 = 0	X
ad Lock Bits	Bit - 1	H	L	H	12V	H	H	H	L	L	D0	X
	Bit - 2										D1	X
	Bit - 3										D2	X
ge Write[3][4][5]	User Row	H	L	1.0 µs	12V	H	L	H	H	H	DI	0 - 3FFH
ad	User Row	H	L	H	12V	L	L	H	L	H	DO	0 - 3FFH
ad	Sig. Row	H	L	H	12V	L	L	H	L	L	DO	0 - 3FFH
ite se[2][4][7] Fuse1	SerialPrgEn	H	L	1.0 µs	12V	L	H	H	L	H	D0 = 0	X
	SerialPrgDis										D0 = 1	X
Fuse2	x2 ClockEn										D1 = 0	X
	x2 ClockDis										D1 = 1	X
Fuse3	UsrRowPrgEn										D2 = 0	X
	UsrRowPrgDis										D2 = 1	X
Fuse4	External Clock En										D3 = 0	X
	Crystal Clock En										D3 = 1	X
ad Fuse	SerialPrg (Fuse1)	H	L	H	12V	H	H	H	L	H	D0	X
	x2 Clock (Fuse2)										D1	X
	UsrRow Prg (Fuse3)										D2	X
	Clock Select (Fuse4)										D2	X

es:
1. See detailed timing for Serial Programming Mode.
2. Internally timed for 8.0 ms.
3. Internally timed for 8.0 ms. Programming begins 150 µs (minimum) after the last write pulse.
4. P3.0 is pulled low during programming to indicate RDY/$\overline{BSY}$
5. 1 to 64 bytes can be programmed at a time per page.
6. 1 to 32 bytes can be programmed at a time per page.
7. Fuse Definitions:
   Fuse1 (Serial Programming Fuse): This fuse enables/disables the serial programming mode (ISP).
   Fuse2 (x2 Mode Selection Fuse): This fuse enables/disables the internal x2 clock mode.
   Fuse3 (User Row Access Fuse): This fuse enables/disables writing to the programmable user row.
   Fuse4 (Clock Selection Fuse): This fuse selects between an external clock source and a quartz crystal as the clock input.

ure 25-1. Programming the Flash/EEPROM Memory (Parallel Mode)

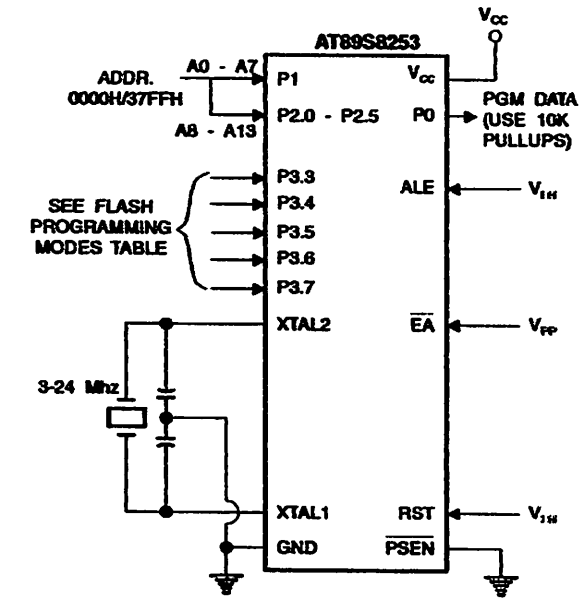


Oscillator Bypass Fuse (Fuse4) Off

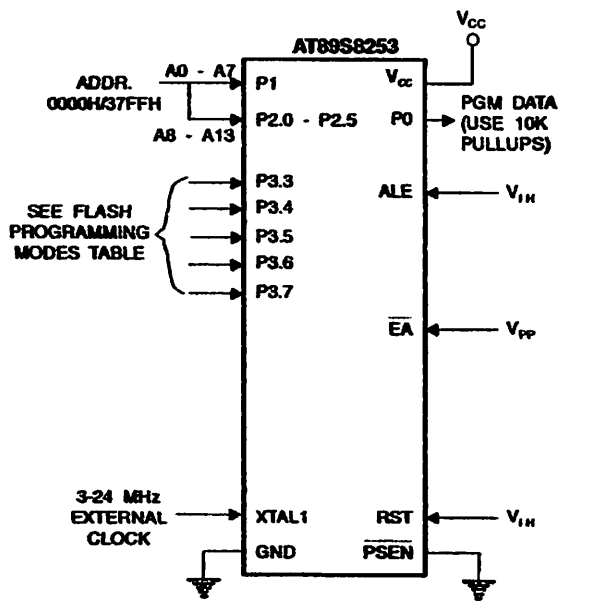


Oscillator Bypass Fuse (Fuse4) On

ure 25-2. Verifying the Flash/EEPROM Memory (Parallel Mode)

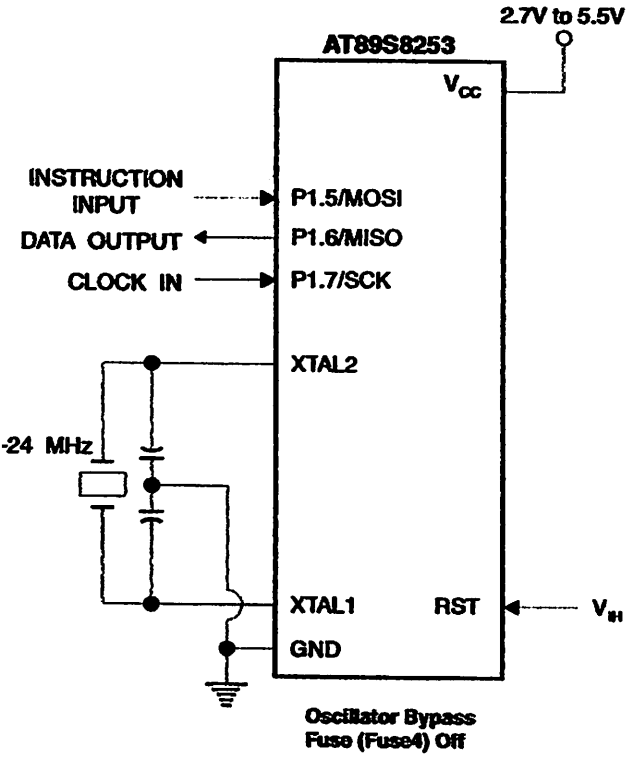


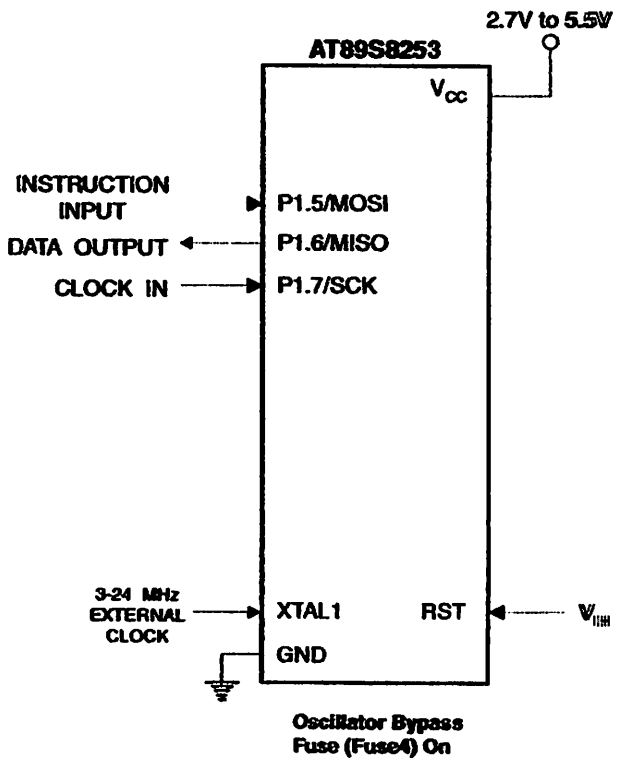
Oscillator Bypass Fuse (Fuse4) Off



Oscillator Bypass Fuse (Fuse4) On

ure 25-3. Flash/EEPROM Serial Downloading





. Flash Programming and Verification Characteristics – Parallel Mode

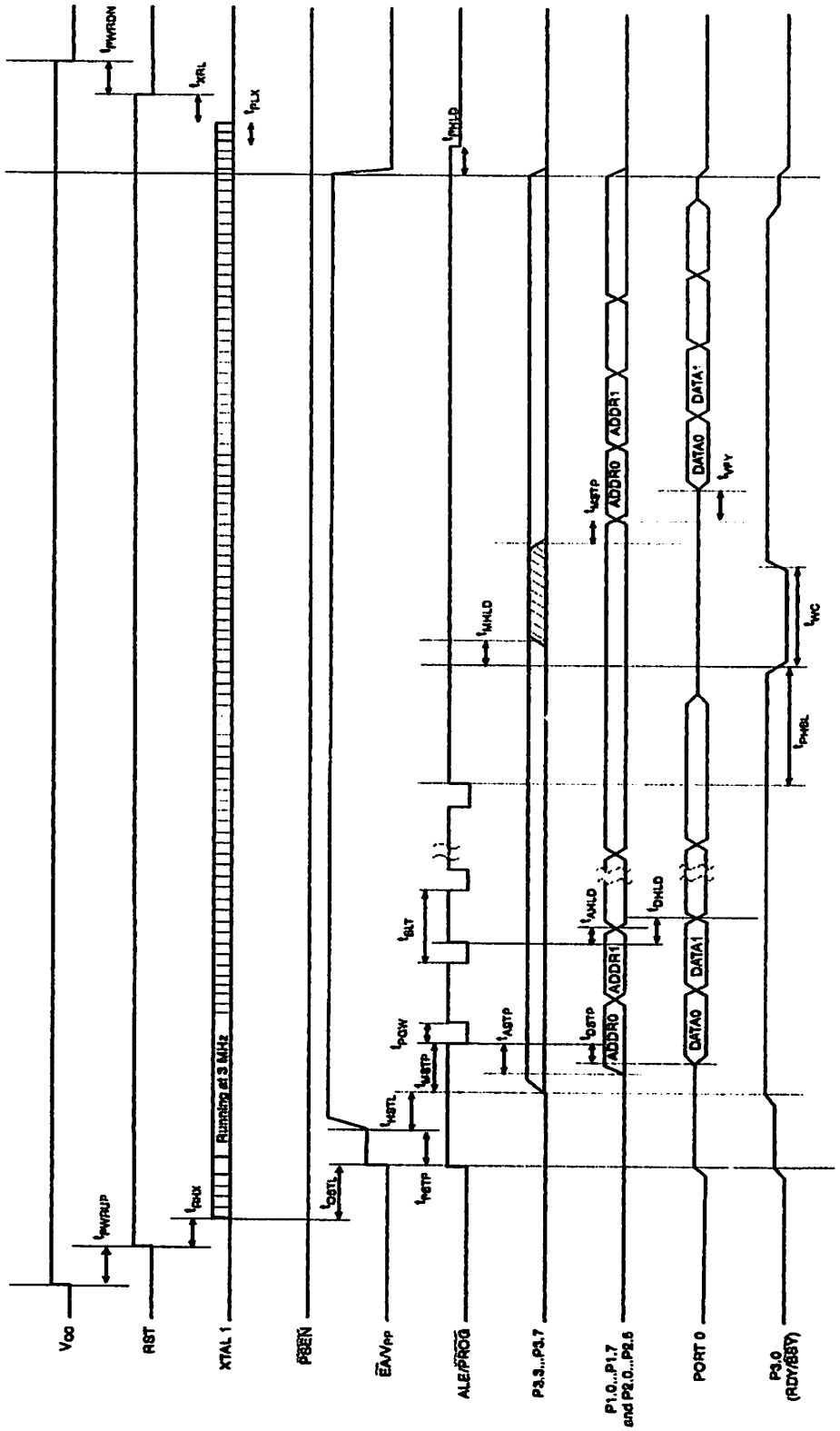
= 20°C to 30°C, V_{CC} = 4.0V to 5.5V

mbol	Parameter	Min	Max	Units
P	Programming Enable Voltage	11.5	12.5	V
	Programming Enable Current		1.0	mA
CLCL	Oscillator Frequency	3	24	MHz
VRUP	Power On to RST High (1)	10		µs
IX	RST High to XTAL Start	10		µs
STL	Oscillator Settling Time	10		ms
STL	High Voltage Settling Time	10		µs
STP	Mode Setup to $\overline{PROG}$ Low	1		µs
ITP	Address Setup to $\overline{PROG}$ Low	1		µs
DTP	Data Setup to $\overline{PROG}$ Low	1		µs
PW	$\overline{PROG}$ Width	1		µs
ALD	Address Hold after $\overline{PROG}$	1		µs
DLD	Data Hold after $\overline{PROG}$	1		µs
BT	Byte Load Period	1	150	µs
PBL	$\overline{PROG}$ High to $\overline{BUSY}$ Low		256	µs
WC	Write Cycle Time(2)		4.5	ms
BLD	Mode Hold After $\overline{BUSY}$ Low	10		µs
DVY	Address to Data Verify Valid		1	µs
PHP	$\overline{PROG}$ Setup to V_{PP} High	10		µs
PHLD	$\overline{PROG}$ Hold after V_{PP} Low	10		µs
XHX	$\overline{PROG}$ Low to XTAL Halt	1		µs
XHL	XTAL Halt to RST Low	1		µs
VRDN	RST Low to Power Off	1		µs

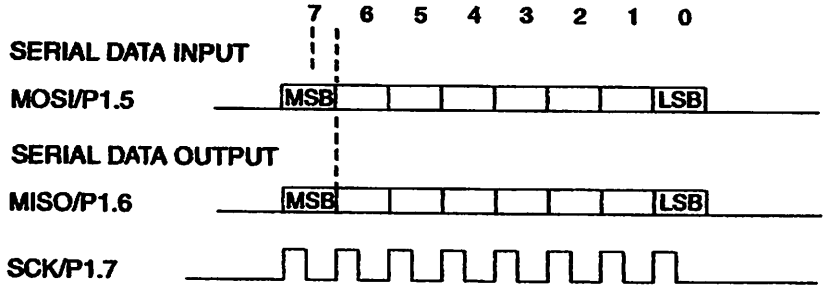
es: 1. Power On occurs once V_{CC} reaches 2.4V.

2. 9 ms if Chip Erase.

ure 26-1. Flash/EEPROM Programming and Verification Waveforms – Parallel Mode

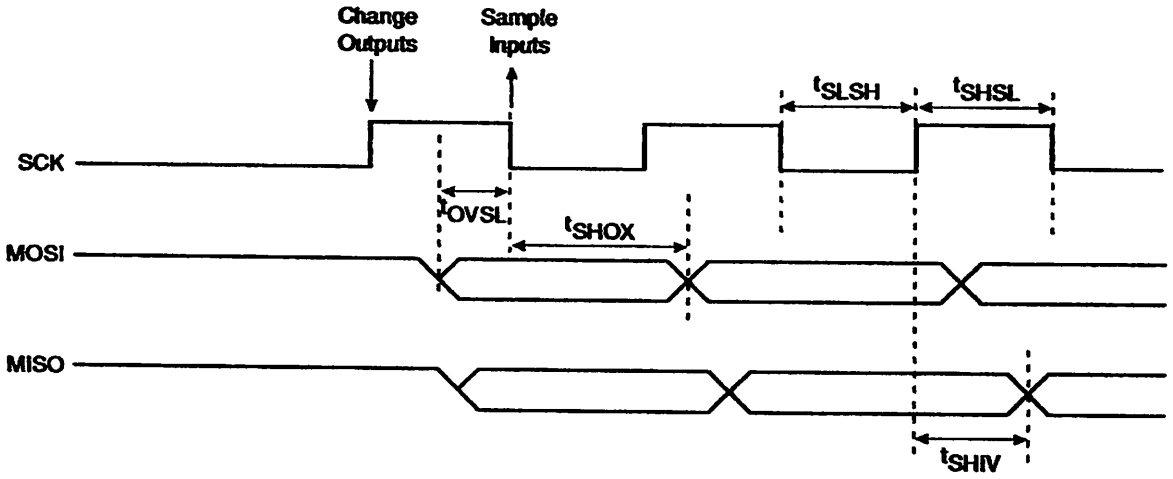


'. Serial Downloading Waveforms (SPI Mode 1 —> CPOL = 0, CPHA = 1)



'. Serial Programming Characteristics

ure 28-1. Serial Programming Timing



le 28-1. Serial Programming Characteristics, T_A = -40-C to 85-C, V_{CC} = 2.7V - 5.5V (Unless Otherwise Noted)

mbol	Parameter	Min	Typ	Max	Units
CLCL	Oscillator Frequency	3		24	MHz
CL	Oscillator Period	41.6		33.3	ns
ISL	SCK Pulse Width High	8 t_{CLCL}			ns
SH	SCK Pulse Width Low	8 t_{CLCL}			ns
SL	MOSI Setup to SCK Low	t_{CLCL}			ns
OX	MOSI Hold after SCK Low	2 t_{CLCL}			ns
IV	SCK High to MISO Valid	10	16	32	ns
ASE	Chip Erase Instruction Cycle Time			9	ms
C	Serial Page Write Cycle Time			4.5	ms

. Absolute Maximum Ratings*

›erating Temperature	-55°C to +125°C
orage Temperature	-65°C to +150°C
ltage on Any Pin th Respect to Ground	-1.0V to +7.0V
ıximum Operating Voltage	6.6V
: Output Current	15.0 mA

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

. DC Characteristics

› values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.7 to 5.5V, unless otherwise noted

›mbol	Parameter	Condition	Min	Max
	Input Low-voltage	(Except $\overline{EA}$)	-0.5V	0.2 V_{CC} - 0.1V
1	Input Low-voltage ($\overline{EA}$)		-0.5V	0.2 V_{CC} - 0.3V
ı	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9V	V_{CC} + 0.5V
ıı	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5V
L	Output Low-voltage[1]	I_{OL} = 10 mA, V_{CC} = 4.0V, T_A = 85°C		0.5V
H	Output High-voltage When Weak Pull Ups are Enabled (Ports 1, 2, 3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, T_A = 85°C	2.4V	
		I_{OH} = -25 µA, T_A = 85°C	0.75 V_{CC}	
		I_{OH} = -10 µA, T_A = 85°C	0.9 V_{CC}	
H1	Output High-voltage When Strong Pull Ups are Enabled (Port 0 in External Bus Mode, P1, 2, 3, ALE, $\overline{PSEN}$)	I_{OH} = -40 mA, T_A = 85°C	2.4V	
		I_{OH} = -25 mA, T_A = 85°C	0.75 V_{CC}	
		I_{OH} = -10 mA, T_A = 85°C	0.9 V_{CC}	
	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45V, V_{CC} = 5.5V, T_A = -40°C		-50 µA
	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	V_{IN} = 2V, V_{CC} = 5.5V, T_A = -40°C		-352 µA
	Input Leakage Current (Port 0, $\overline{EA}$)	0.45V< V_{IN} < V_{CC}		±10 µA
IST	Reset Pull-down Resistor		50 KΩ	150 KΩ
›	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10 pF
	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 5.5V, T_A = -40°C		10 mA
		Idle Mode, 12 MHz, V_{CC} = 5.5V, T_A = -40°C		3.5 mA
	Power-down Mode[2]	V_{CC} = 5.5V, T_A = -40°C		100 µA
		V_{CC} = 4.0V, T_A = -40°C		20 µA

›s: 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:15 mA,
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.

. AC Characteristics

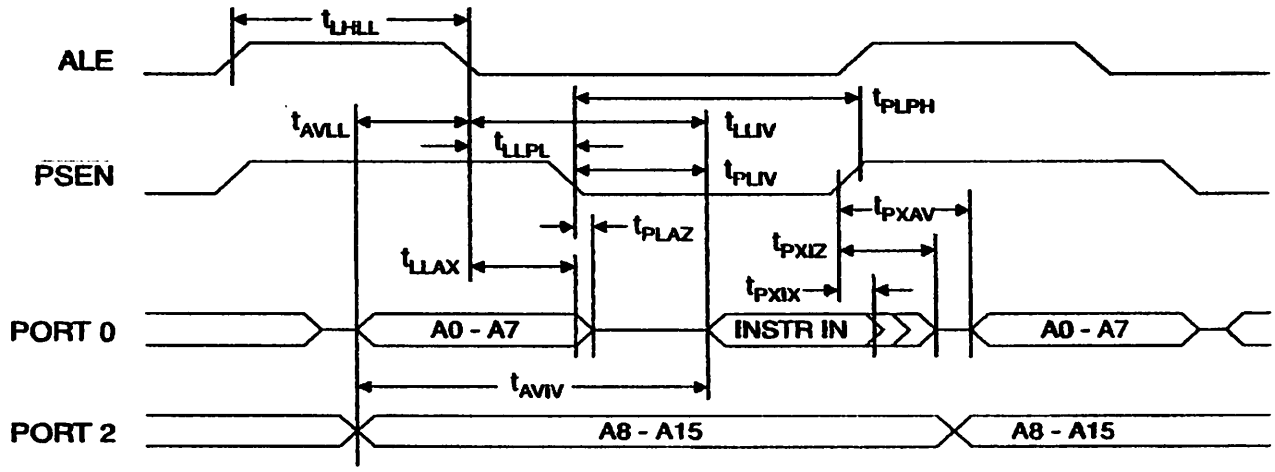
) values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.7 to 5.5V, unless otherwise noted.

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

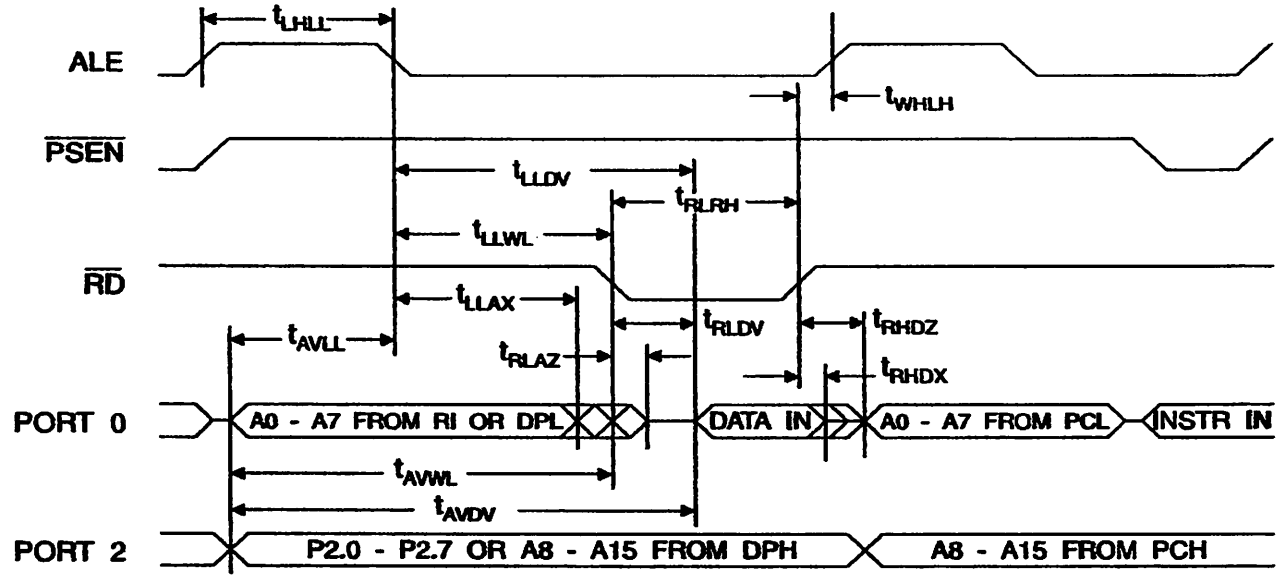
.1 External Program and Data Memory Characteristics

mbol	Parameter	Variable Oscillator		Units
		Min	Max	
CLCL	Oscillator Frequency	0	24	MHz
LL	ALE Pulse Width	$2t_{CLCL}$ - 12		ns
LL	Address Valid to ALE Low	t_{CLCL} - 12		ns
AX	Address Hold after ALE Low	t_{CLCL} - 16		ns
LV	ALE Low to Valid Instruction In		$4t_{CLCL}$ - 50	ns
PL	ALE Low to $\overline{PSEN}$ Low	t_{CLCL} - 12		ns
PH	$\overline{PSEN}$ Pulse Width	$3t_{CLCL}$ - 12		ns
PV	$\overline{PSEN}$ Low to Valid Instruction In		$3t_{CLCL}$ - 50	ns
PX	Input Instruction Hold after $\overline{PSEN}$	-10		ns
PZ	Input Instruction Float after $\overline{PSEN}$		t_{CLCL} - 20	ns
PAV	$\overline{PSEN}$ to Address Valid	t_{CLCL} - 4		ns
AIV	Address to Valid Instruction In		$5t_{CLCL}$ - 50	ns
PLAZ	$\overline{PSEN}$ Low to Address Float		20	ns
RLRH	$\overline{RD}$ Pulse Width	$6t_{CLCL}$		ns
WLWH	$\overline{WR}$ Pulse Width	$6t_{CLCL}$		ns
RLDV	$\overline{RD}$ Low to Valid Data In		$5t_{CLCL}$ - 50	ns
RHDX	Data Hold after $\overline{RD}$	0		ns
RHDZ	Data Float after $\overline{RD}$		$2t_{CLCL}$ - 20	ns
LLDV	ALE Low to Valid Data In		$8t_{CLCL}$ - 50	ns
AVDV	Address to Valid Data In		$9t_{CLCL}$ - 50	ns
LLWL	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	$3t_{CLCL}$ - 24	$3t_{CLCL}$	ns
AVWL	Address to $\overline{RD}$ or $\overline{WR}$ Low	$4t_{CLCL}$ - 12		ns
WLWX	Data Valid to $\overline{WR}$ Transition	$2t_{CLCL}$ - 24		ns
WLWH	Data Valid to $\overline{WR}$ High	$8t_{CLCL}$ - 24		ns
WHQX	Data Hold after $\overline{WR}$	$2t_{CLCL}$ - 24		ns
RLAZ	$\overline{RD}$ Low to Address Float		0	ns
WHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{CLCL} - 10	t_{CLCL} + 20	ns

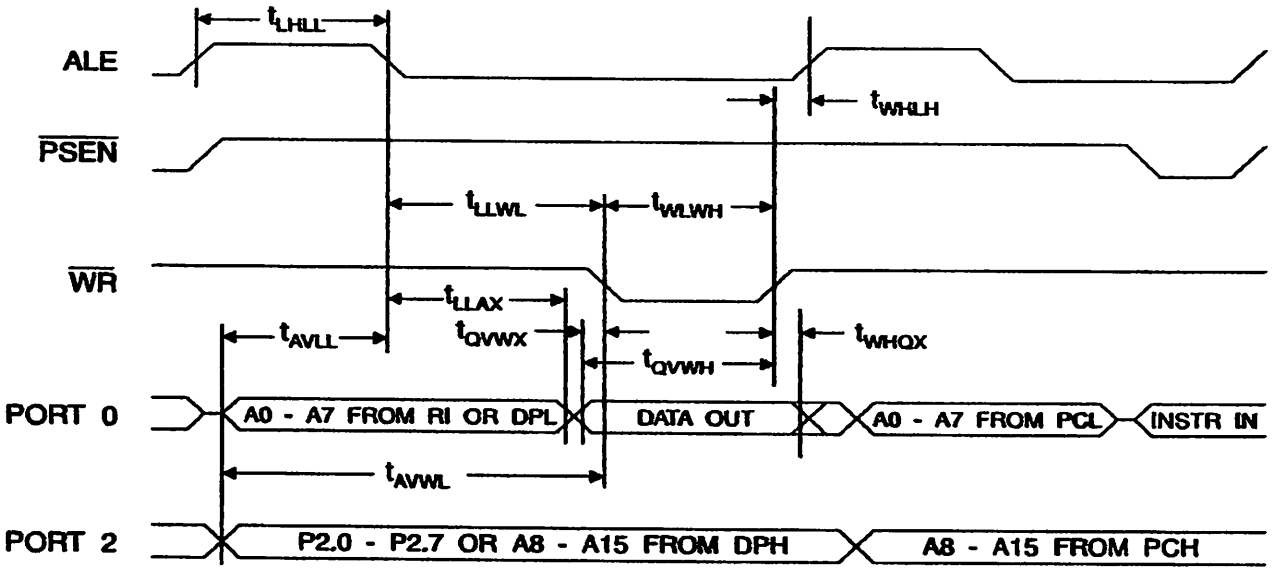
. External Program Memory Read Cycle



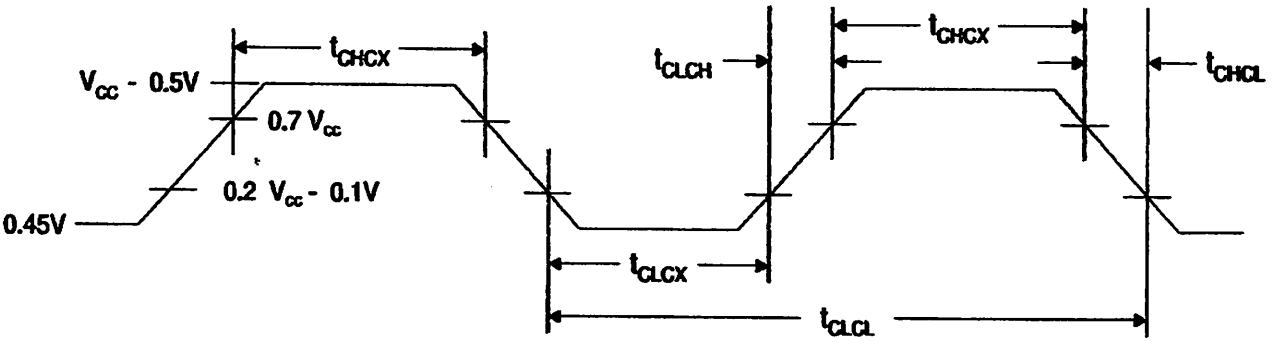
. External Data Memory Read Cycle



. External Data Memory Write Cycle



. External Clock Drive Waveforms



. External Clock Drive

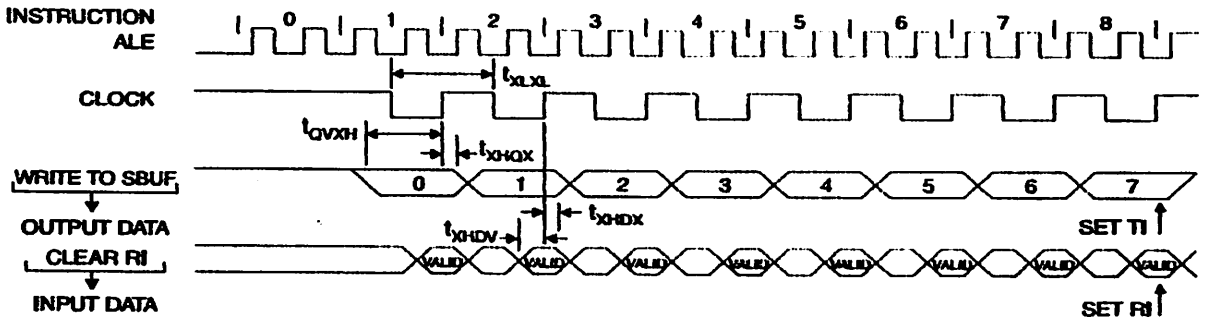
mbol	Parameter	V_{CC} = 2.7V to 5.5V		Units
		Min	Max	
CLCL	Oscillator Frequency	0	24	MHz
CL	Clock Period	41.6		ns
HCX	High Time	12		ns
CX	Low Time	12		ns
CH	Rise Time		5	ns
HCL	Fall Time		5	ns

'. Serial Port Timing: Shift Register Mode Test Conditions

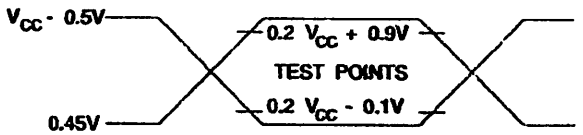
e values in this table are valid for V_{CC} = 2.7V to 5.5V and Load Capacitance = 80 pF.

ymbol	Parameter	Variable Oscillator		Units
		Min	Max	
LXL	Serial Port Clock Cycle Time	$12t_{CLCL}$-15		µs
VXH	Output Data Setup to Clock Rising Edge	$10t_{CLCL}$-15		ns
HQX	Output Data Hold after Clock Rising Edge	$2t_{CLCL}$-15		ns
HDX	Input Data Hold after Clock Rising Edge	t_{CLCL}		ns
HDV	Input Data Valid to Clock Rising Edge	0		ns

). Shift Register Mode Timing Waveforms



). AC Testing Input/Output Waveforms[1]

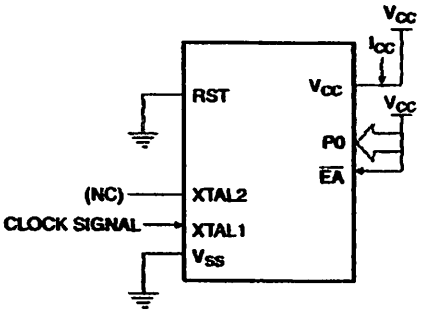


e: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

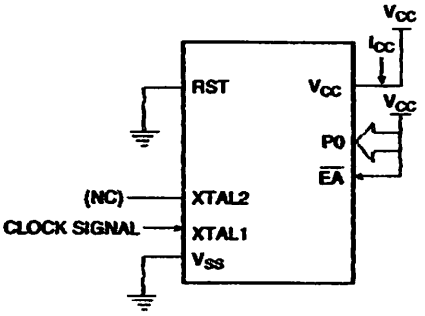
). Float Waveforms[1]

e: 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.

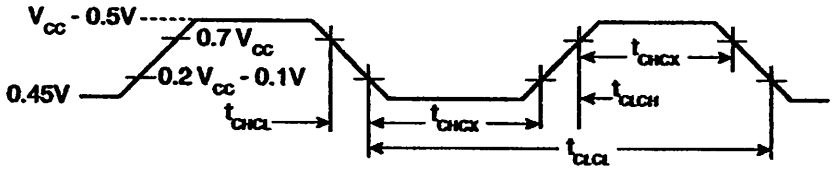
. I_{CC} Test Condition, Active Mode, All Other Pins are Disconnected



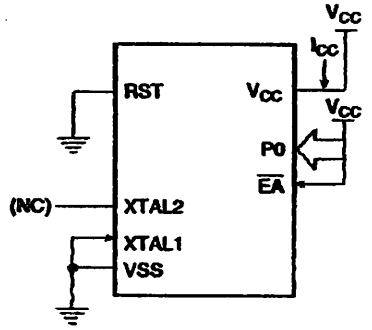
. I_{CC} Test Condition, Idle Mode, All Other Pins are Disconnected



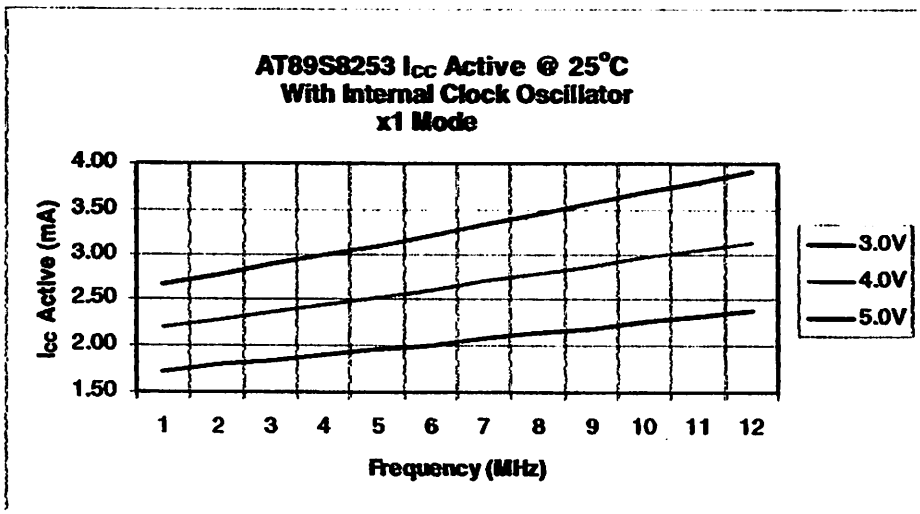
. Clock Signal Waveform for I_{CC} Tests in Active and Idle Modes, $t_{CLCH} = t_{CHCL} = 5$ ns

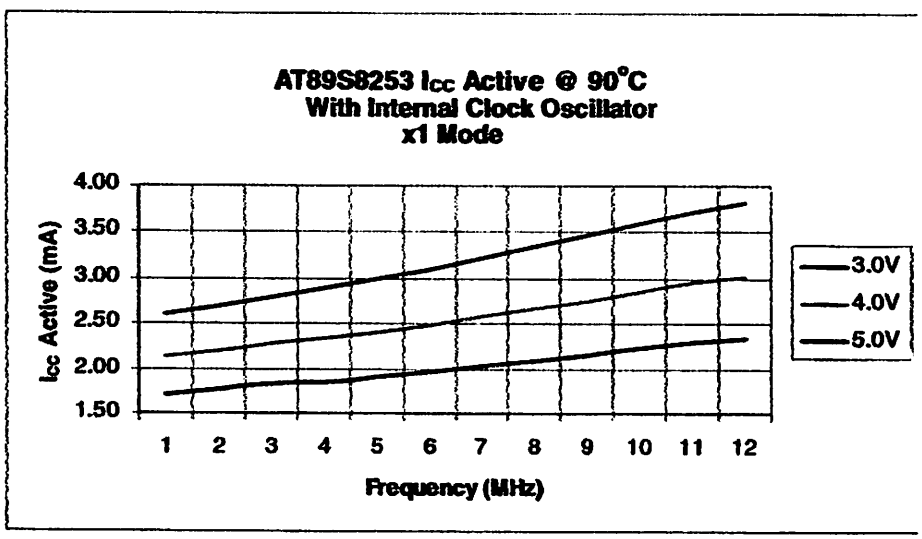


. I_{CC} Test Condition, Power-down Mode, All Other Pins are Disconnected, $V_{CC} = 2V$ to $5.5V$

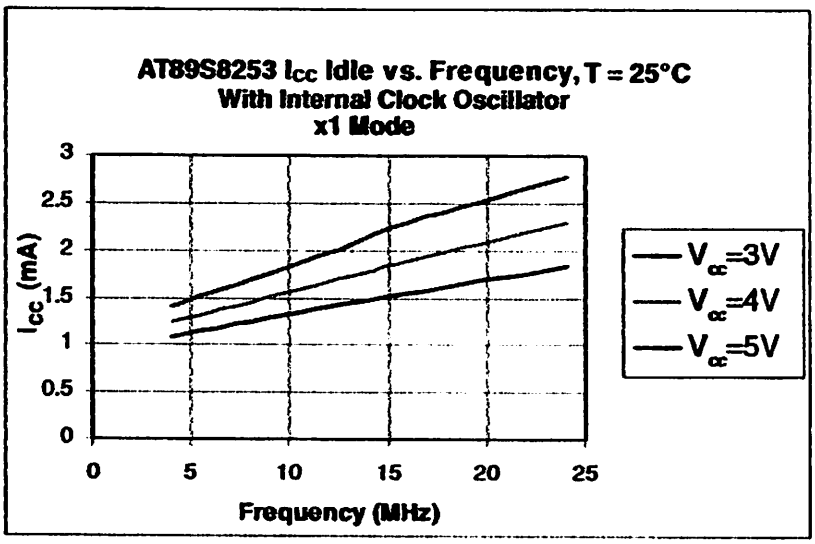


I_{CC} (Active Mode) Measurements

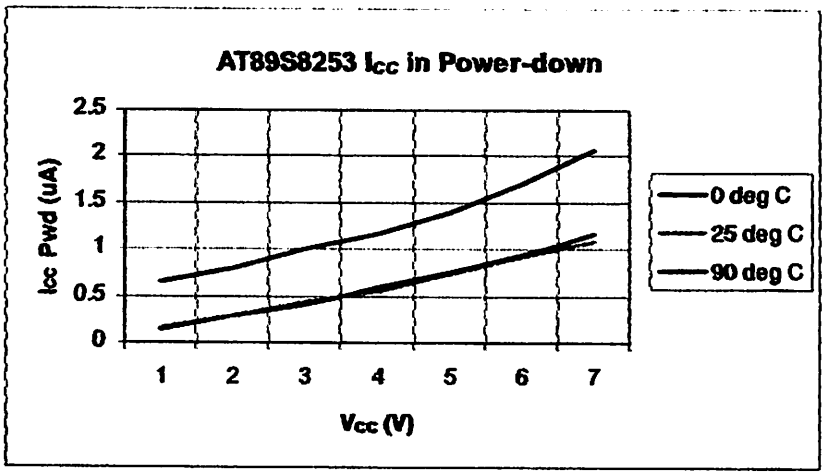




I_{CC} (Idle Mode) Measurements



I_{CC} (Power Down Mode) Measurements



. Ordering Information

1 Standard Package

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	2.7V to 5.5V	AT89S8253-24AC AT89S8253-24JC AT89S8253-24PC AT89S8253-24PSC	44A 44J 40P6 42PS6	Commercial (0°C to 70°C)
	2.7V to 5.5V	AT89S8253-24AI AT89S8253-24JI AT89S8253-24PI AT89S8253-24PSI	44A 44J 40P6 42PS6	Industrial (-40°C to 85°C)

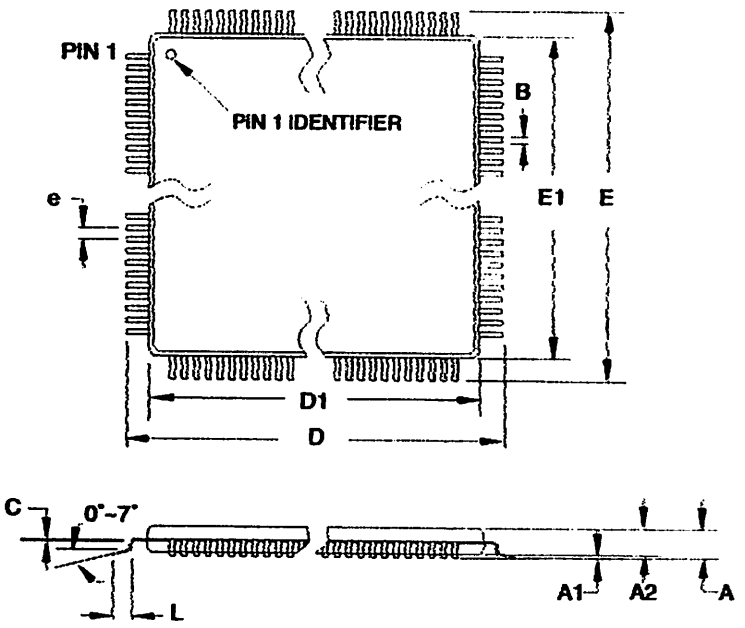
2 Green Package Option (Pb/Halide-free)

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	2.7V to 5.5V	AT89S8253-24AU AT89S8253-24JU AT89S8253-24PU AT89S8253-24PSU	44A 44J 40P6 42PS6	Industrial (-40°C to 85°C)

	Package Type
A	44-lead, Thin Plastic Gull Wing Quad Flat Package (TQFP)
J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
PS6	42-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)

. Package Information

.1 44A – TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

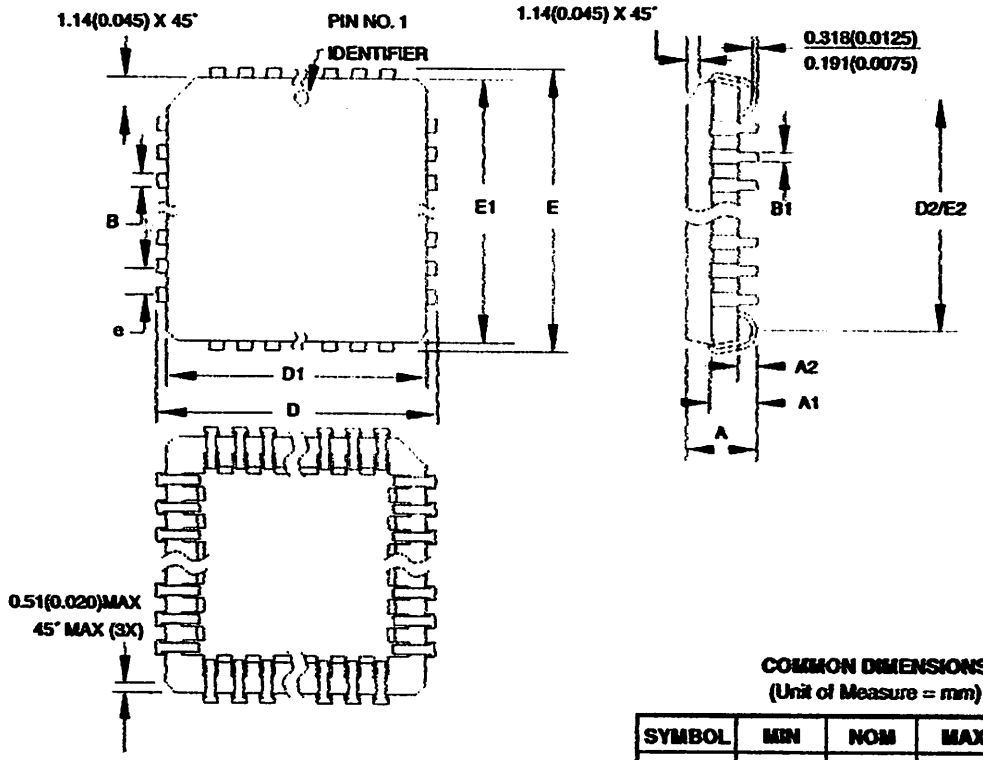
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

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

10/5/2001

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

.2 44J – PLCC



COMMON DIMENSIONS
(Unit of Measure = mm)

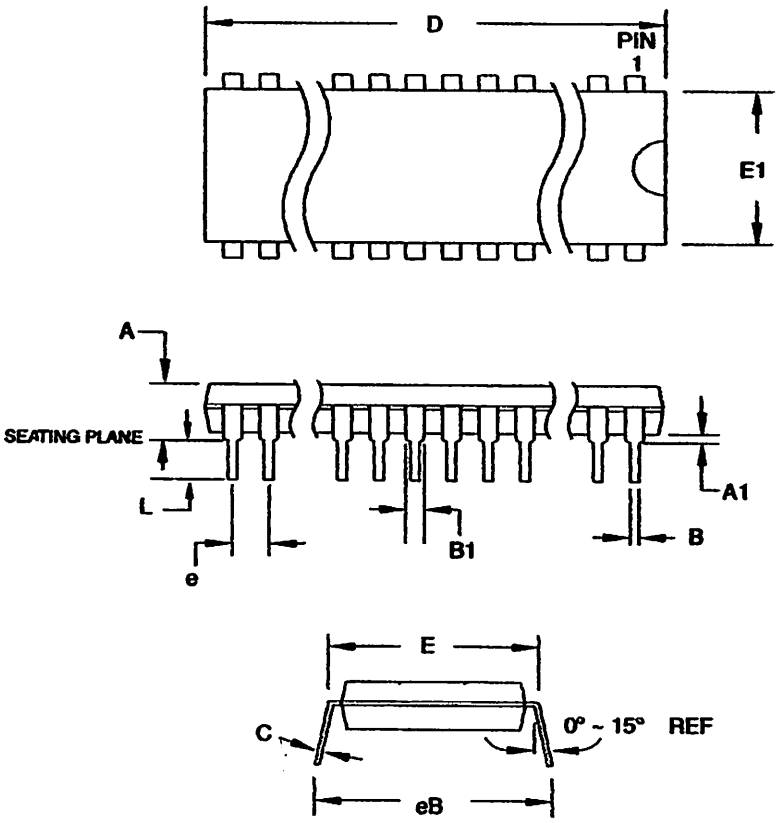
SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	–	4.572	
A1	2.286	–	3.048	
A2	0.508	–	–	
D	17.399	–	17.653	
D1	16.510	–	16.662	Note 2
E	17.399	–	17.653	
E1	16.510	–	16.662	Note 2
D2/E2	14.986	–	16.002	
B	0.660	–	0.813	
B1	0.330	–	0.533	
e	1.270 TYP			

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

10/04/01

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

.3 40P6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

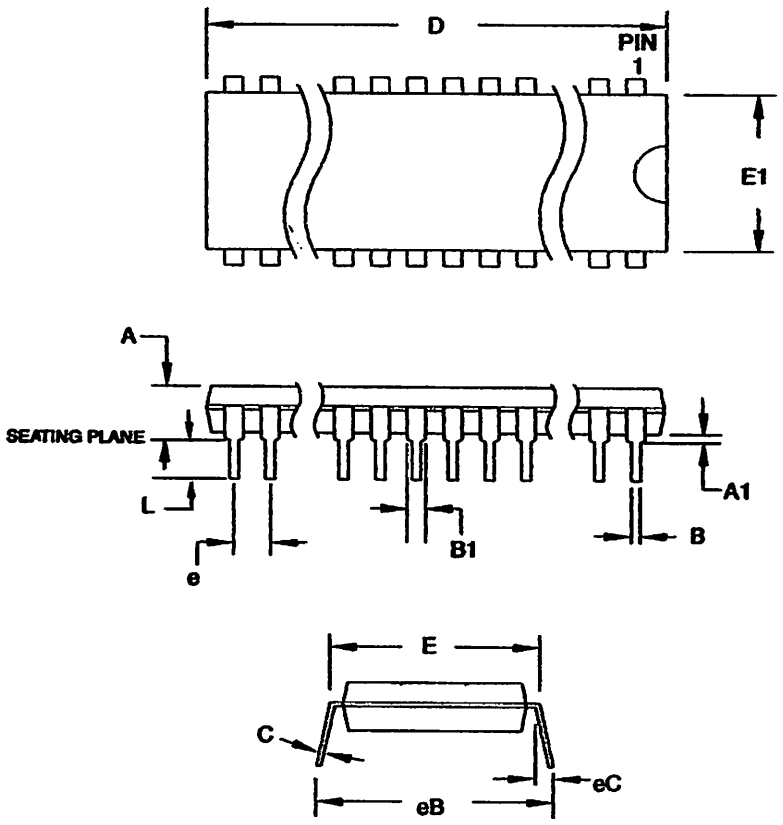
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

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

09/28/01

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

42PS6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = Inch)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	0.200	
A1	0.020	–	–	
D	1.440	1.450	1.460	Note 2
E	0.600	–	0.630	
E1	0.500	0.540	0.570	Note 2
B	0.015	0.018	0.022	
B1	0.035	0.040	0.045	
L	0.100	0.130	0.140	
C	0.009	0.010	0.015	
eB	–	–	0.730	
eC	0.000	–	0.060	
e	0.70 TYP			

Notes: 1. This package conforms to JEDEC reference MS-020, Variation AB.
2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

11/3/06

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	42PS6, 42-lead (Shrink 0.070"/0.600" Row Space) Plastic Dual Inline Package (PDIP)	42PS6	B

adquarters

nel Corporation
25 Orchard Parkway
n Jose, CA 95131
A
: 1(408) 441-0311
x: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

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

Product Contact

Web Site
www.atmel.com

Technical Support
mcu@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Requests
www.atmel.com/literature

19-4323; Rev 10; 8/01

MAXIM

+5V-Powered, Multichannel RS-232 Drivers/Receivers

General Description

The MAX220–MAX249 family of line drivers/receivers is intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where ±12V is not available.

These parts are especially useful in battery-powered systems, since their low-power shutdown mode reduces power dissipation to less than 5µW. The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use no external components and are recommended for applications where printed circuit board space is critical.

Applications

Portable Computers

Low-Power Modems

Interface Translation

Battery-Powered RS-232 Systems

Multidrop RS-232 Networks

Features

Superior to Bipolar

- **Operate from Single +5V Power Supply (+5V and +12V—MAX231/MAX239)**
- **Low-Power Receive Mode in Shutdown (MAX223/MAX242)**
- **Meet All EIA/TIA-232E and V.28 Specifications**
- **Multiple Drivers and Receivers**
- **3-State Driver and Receiver Outputs**
- **Open-Line Detection (MAX243)**

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX220CPE	0°C to +70°C	16 Plastic DIP
MAX220CSE	0°C to +70°C	16 Narrow SO
MAX220CWE	0°C to +70°C	16 Wide SO
MAX220C/D	0°C to +70°C	Dice*
MAX220EPE	-40°C to +85°C	16 Plastic DIP
MAX220ESE	-40°C to +85°C	16 Narrow SO
MAX220EWE	-40°C to +85°C	16 Wide SO
MAX220EJE	-40°C to +85°C	16 CERDIP
MAX220MJE	-55°C to +125°C	16 CERDIP

Ordering Information continued at end of data sheet.

**Contact factory for dice specifications.*

Selection Table

Part Number	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value (µF)	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
MAX220	+5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard pinout
MAX222	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
MAX223 (MAX213)	+5	4/5	4	1.0 (0.1)	Yes	✔	120	MAX241 and receivers active in shutdown
MAX225	+5	5/5	0	—	Yes	✔	120	Available in SO
MAX230 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
MAX231 (MAX201)	+5 and +7.5 to +13.2	2/2	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; same functions as MAX232
MAX232 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
MAX232A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
MAX233 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
MAX233A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
MAX234 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
MAX235 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
MAX236 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
MAX237 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
MAX238 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
MAX239 (MAX209)	+5 and +7.5 to +13.2	3/5	2	1.0 (0.1)	No	—	120	Standard +5/+12V or battery supplies; single-package solution for IBM PC serial port
MAX240	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
MAX241 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
MAX242	+5	2/2	4	0.1	Yes	✔	200	Separate shutdown and enable
MAX243	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
MAX244	+5	8/10	4	1.0	No	—	120	High slew rate
MAX245	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, two shutdown modes
MAX246	+5	8/10	0	—	Yes	✔	120	High slew rate, int. caps, three shutdown modes
MAX247	+5	8/9	0	—	Yes	✔	120	High slew rate, int. caps, nine operating modes
MAX248	+5	8/8	4	1.0	Yes	✔	120	High slew rate, selective half-chip enables
MAX249	+5	6/10	4	1.0	Yes	✔	120	Available in quad flatpack package

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC}) ..... -0.3V to +6V
Input Voltages
T_{IN} ..... -0.3V to (V_{CC} - 0.3V)
R_{IN} (Except MAX220) ..... ±30V
R_{IN} (MAX220) ..... ±25V
T_{OUT} (Except MAX220) (Note 1) ..... ±15V
T_{OUT} (MAX220) ..... ±13.2V
Output Voltages
T_{OUT} ..... ±15V
R_{OUT} ..... -0.3V to (V_{CC} + 0.3V)
Driver/Receiver Output Short Circuited to GND ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C) 889mW
20-Pin Plastic DIP (derate 8.00mW/°C above +70°C) 440mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
18-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
20-Pin Wide SO (derate 10.00mW/°C above +70°C) 800mW
20-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C) 800mW
18-Pin CERDIP (derate 10.53mW/°C above +70°C) 842mW
Operating Temperature Ranges
MAX2_ _AC_ _, MAX2_ _C_ _ ..... 0°C to +70°C
MAX2_ _AE_ _, MAX2_ _E_ _ ..... -40°C to +85°C
MAX2_ _AM_ _, MAX2_ _M_ _ ..... -55°C to +125°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10sec) ..... +300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $\overline{SHDN}$ or V_{CC} = 0V.
Note 2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V_{CC} = 5.0V		2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation			5	40	µA
	$\overline{SHDN}$ = 0V, MAX222/242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V_{CC} = 5.5V, $\overline{SHDN}$ = 0V, V_{OUT} = ±15V, MAX222/242			±0.01	±10	µA
	V_{CC} = $\overline{SHDN}$ = 0V, V_{OUT} = ±15V			±0.01	±10	
Data Rate				200	116	kb/s
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V			±7	±22	mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V_{CC} = 5V	All except MAX243 $R2_{IN}$	0.8	1.3		V
		MAX243 $R2_{IN}$ (Note 2)	-3			
RS-232 Input Threshold High	V_{CC} = 5V	All except MAX243 $R2_{IN}$		1.8	2.4	V
		MAX243 $R2_{IN}$ (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V_{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

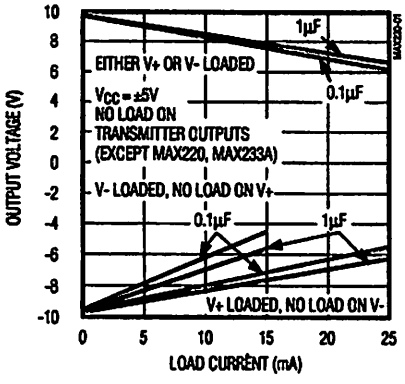
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
TTL/CMOS Output Leakage Current	$\overline{SHDN}$ = V_{CC} or $\overline{EN}$ = V_{CC} ($\overline{SHDN}$ = 0V for MAX222), 0V ≤ V_{OUT} ≤ V_{CC}			±0.05	±10	µA
$\overline{EN}$ Input Threshold Low	MAX242			1.4	0.8	V
$\overline{EN}$ Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
V_{CC} Supply Current ($\overline{SHDN}$ = V_{CC}), Figures 5, 6, 11, 19	No load	MAX220		0.5	2	mA
		MAX222/232A/233A/242/243		4	10	
	3kΩ load both inputs	MAX220		12		
		MAX222/232A/233A/242/243		15		
Shutdown Supply Current	MAX222/242	T_A = +25°C		0.1	10	µA
		T_A = 0°C to +70°C		2	50	
		T_A = -40°C to +85°C		2	50	
		T_A = -55°C to +125°C		35	100	
$\overline{SHDN}$ Input Leakage Current	MAX222/242				±1	µA
$\overline{SHDN}$ Threshold Low	MAX222/242			1.4	0.8	V
$\overline{SHDN}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	1.5	3	30	
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t_{PHLT}	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t_{PLHT}	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Receiver Propagation Delay RS-232 to TLL (normal operation), Figure 2	t_{PHLR}	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t_{PLHR}	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Receiver Propagation Delay RS-232 to TLL (shutdown), Figure 2	t_{PHLS}	MAX242		0.5	10	µs
	t_{PLHS}	MAX242		2.5	10	
Receiver-Output Enable Time, Figure 3	t_{ER}	MAX242		125	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}	MAX242		160	500	ns
Transmitter-Output Enable Time ($\overline{SHDN}$ goes high), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ goes low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + to - Propagation Delay Difference (normal operation)	t_{PHLT} - t_{PLHT}	MAX222/232A/233A/242/243		300		ns
		MAX220		2000		
Receiver + to - Propagation Delay Difference (normal operation)	t_{PHLR} - t_{PLHR}	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

Note 3: MAX243 $R2_{OUT}$ is guaranteed to be low when $R2_{IN}$ is ≥ 0V or is floating.

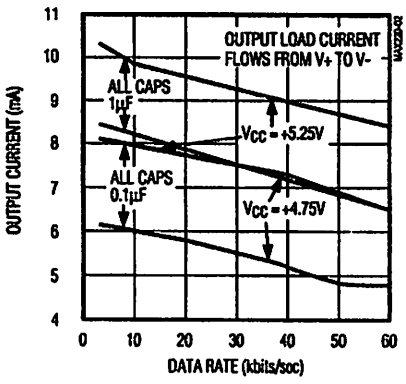
Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243

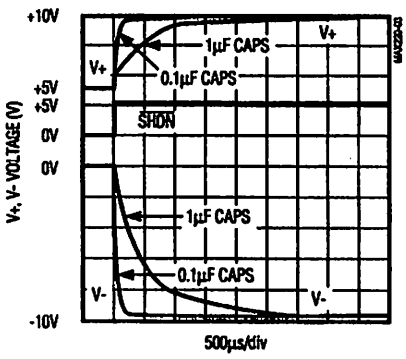
OUTPUT VOLTAGE vs. LOAD CURRENT



AVAILABLE OUTPUT CURRENT vs. DATA RATE



MAX222/MAX242 ON-TIME EXITING SHUTDOWN



ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230–MAX241

V_{CC} ..... -0.3V to +6V
V+ ..... (V_{CC} - 0.3V) to +14V
V- ..... +0.3V to -14V
Input Voltages
T_{IN} ..... -0.3V to (V_{CC} + 0.3V)
R_{IN} ..... ±30V
Output Voltages
T_{OUT} ..... (V+ + 0.3V) to (V- - 0.3V)
R_{OUT} ..... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_{OUT} ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C)....800mW
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)....842mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Plastic DIP (derate 9.09mW/°C above +70°C)..... 500mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C)..... 762mW
20-Pin Wide SO (derate 10 00mW/°C above +70°C)..... 800mW
24-Pin Wide SO (derate 11.76mW/°C above +70°C)..... 941mW
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
44-Pin Plastic FP (derate 11.11mW/°C above +70°C)..... 889mW
14-Pin CERDIP (derate 9.09mW/°C above +70°C)..... 727mW
16-Pin CERDIP (derate 10.00mW/°C above +70°C)..... 800mW
20-Pin CERDIP (derate 11.11mW/°C above +70°C)..... 889mW
24-Pin Narrow CERDIP
(derate 12.50mW/°C above +70°C) 1W
24-Pin Sidebraze (derate 20.0mW/°C above +70°C)..... 1.6W
28-Pin SSOP (derate 9.52mW/°C above +70°C)..... 762mW
Operating Temperature Ranges
MAX2 _ _ C _ _ ..... 0°C to +70°C
MAX2 _ _ E _ _ ..... -40°C to +85°C
MAX2 _ _ M _ _ ..... -55°C to +125°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10sec) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241

(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to ground		±5.0	±7.3		V
V_{CC} Power-Supply Current	No load, T_A = +25°C	MAX232/233		5	10	mA
		MAX223/230/234–238/240/241		7	15	
		MAX231/239		0.4	1	
V+ Power-Supply Current		MAX231		1.8	5	mA
		MAX239		5	15	
Shutdown Supply Current	T_A = +25°C	MAX223		15	50	µA
		MAX230/235/236/240/241		1	10	
Input Logic Threshold Low	T_{IN}; EN, $\overline{SHDN}$ (MAX233); $\overline{EN}$, SHDN (MAX230/235–241)				0.8	V
Input Logic Threshold High	T_{IN}		2.0			V
	EN, $\overline{SHDN}$ (MAX223); $\overline{EN}$, SHDN (MAX230/235/236/240/241)		2.4			
Logic Pull-Up Current	T_{IN} = 0V			1.5	200	µA
Receiver Input Voltage Operating Range			-30		30	V

ELECTRICAL CHARACTERISTICS—MAX223/MAX230–MAX241 (continued)

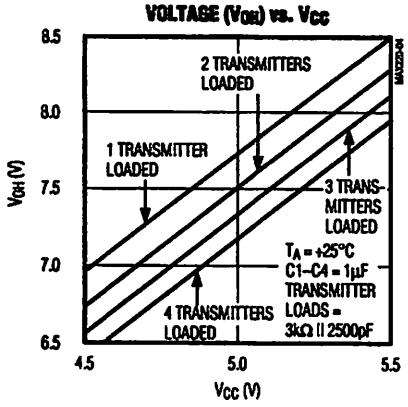
(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS			MIN	TYP	MAX	UNITS
RS-232 Input Threshold Low	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		0.8	1.2		V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)		0.6	1.5		
RS-232 Input Threshold High	T_A = +25°C, V_{CC} = 5V	Normal operation $\overline{SHDN}$ = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)			1.7	2.4	V
		Shutdown (MAX223) $\overline{SHDN}$ = 0V, EN = 5V ($R4_{IN}$, $R5_{IN}$)			1.5	2.4	
RS-232 Input Hysteresis	V_{CC} = 5V, no hysteresis in shutdown			0.2	0.5	1.0	V
RS-232 Input Resistance	T_A = +25°C, V_{CC} = 5V			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 1.6mA (MAX231/232/233, I_{OUT} = 3.2mA)					0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1mA			3.5	V_{CC} - 0.4		V
TTL/CMOS Output Leakage Current	0V ≤ R_{OUT} ≤ V_{CC}; EN = 0V (MAX223); $\overline{EN}$ = V_{CC} (MAX235–241)				0.05	±10	µA
Receiver Output Enable Time	Normal operation	MAX223			600		ns
		MAX235/236/239/240/241			400		
Receiver Output Disable Time	Normal operation	MAX223			900		ns
		MAX235/236/239/240/241			250		
Propagation Delay	RS-232 IN to TTL/CMOS OUT, C_L = 150pF	Normal operation			0.5	10	µs
		$\overline{SHDN}$ = 0V (MAX223)	t_{PHLS}		4	40	
			t_{PLHS}		6	40	
Transition Region Slew Rate	MAX223/MAX230/MAX234–241, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V			3	5.1	30	V/µs
	MAX231/MAX232/MAX233, T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 2500pF, measured from +3V to -3V or -3V to +3V				4	30	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V			300			Ω
Transmitter Output Short-Circuit Current					±10		mA

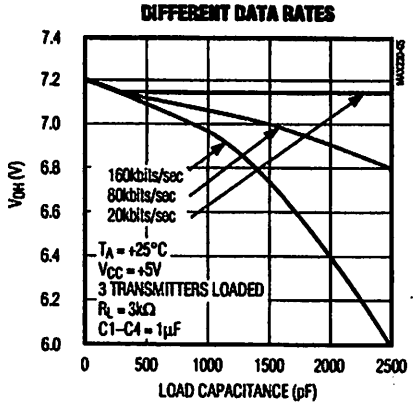
Typical Operating Characteristics

MAX223/MAX230–MAX241

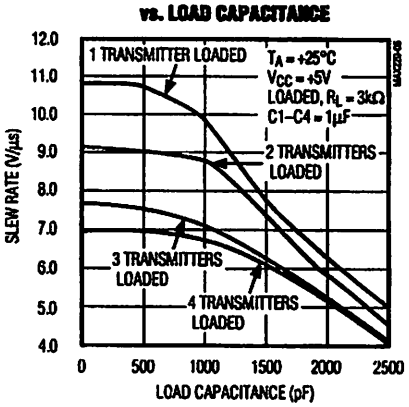
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. V_{CC}



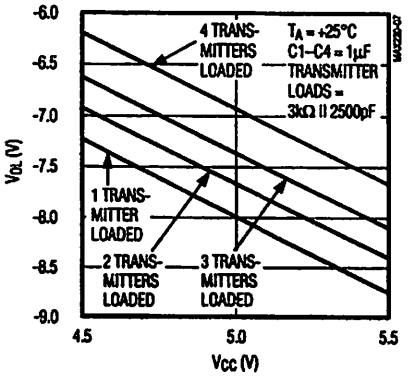
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



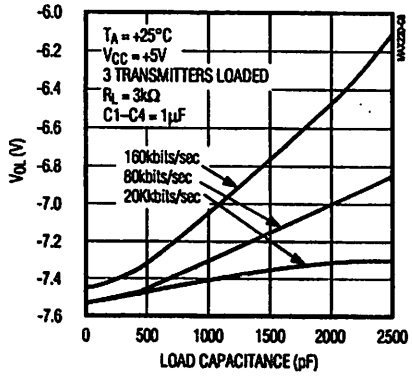
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



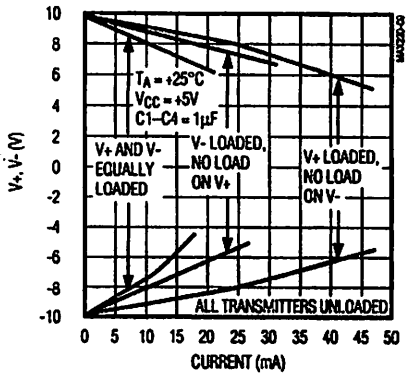
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. V_{CC}



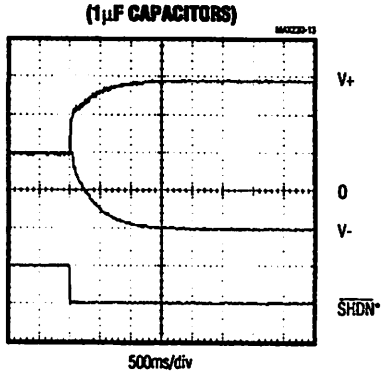
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CURRENT



V+, V- WHEN EXITING SHUTDOWN (1µF CAPACITORS)



*SHUTDOWN POLARITY IS REVERSED FOR NON MAX241 PARTS

ABSOLUTE MAXIMUM RATINGS—MAX225/MAX244–MAX249

Supply Voltage (V_{CC}) ..... -0.3V to +6V
nput Voltages
T_{IN}, $\overline{ENA}$, $\overline{ENB}$, $\overline{ENR}$, $\overline{ENT}$, $\overline{ENRA}$, $\overline{ENRB}$, $\overline{ENTA}$, $\overline{ENTB}$ ..... -0.3V to (V_{CC} + 0.3V)
R_{IN} ..... ±25V
T_{OUT} (Note 3) ..... ±15V
R_{OUT} ..... -0.3V to (V_{CC} + 0.3V)
Short Circuit (one output at a time)
T_{OUT} to GND ..... Continuous
R_{OUT} to GND ..... Continuous

Continuous Power Dissipation (T_A = +70°C)
28-Pin Wide SO (derate 12.50mW/°C above +70°C) 1W
40-Pin Plastic DIP (derate 11.11mW/°C above +70°C) ... 611mW
44-Pin PLCC (derate 13.33mW/°C above +70°C) 1.07W
Operating Temperature Ranges
MAX225C_ _, MAX24_C_ _ ..... 0°C to +70°C
MAX225E_ _, MAX24_E_ _ ..... -40°C to +85°C
Storage Temperature Range ..... -65°C to +160°C
Lead Temperature (soldering, 10sec) ..... +300°C

Note 4: Input voltage measured with transmitter output in a high-impedance state, shutdown, or V_{CC} = 0V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249

(MAX225, V_{CC} = 5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High			2	1.4		V
Logic Pull-Up/Input Current	Tables 1a–1d	Normal operation		10	50	µA
		Shutdown		±0.01	±1	
Data Rate	Tables 1a–1d, normal operation			120	64	kbits/sec
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±7.5		V
Output Leakage Current (shutdown)	Tables 1a–1d	$\overline{ENA}$, $\overline{ENB}$, $\overline{ENT}$, $\overline{ENTA}$, $\overline{ENTB}$ = V_{CC}, V_{OUT} = ±15V		±0.01	±25	µA
		V_{CC} = 0V, V_{OUT} = ±15V		±0.01	±25	
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V (Note 4)		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V		±7	±30		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±25	V
RS-232 Input Threshold Low	V_{CC} = 5V		0.8	1.3		V
RS-232 Input Threshold High	V_{CC} = 5V			1.8	2.4	V
RS-232 Input Hysteresis	V_{CC} = 5V		0.2	0.5	1.0	V
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I_{OUT} = -1.0mA		3.5	V_{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V_{OUT} = GND		-2	-10		mA
	Shrinking V_{OUT} = V_{CC}		10	30		
TTL/CMOS Output Leakage Current	Normal operation, outputs disabled, Tables 1a–1d, 0V ≤ V_{OUT} ≤ V_{CC}, $\overline{ENR}$_ = V_{CC}			±0.05	±0.10	µA

ELECTRICAL CHARACTERISTICS—MAX225/MAX244–MAX249 (continued)

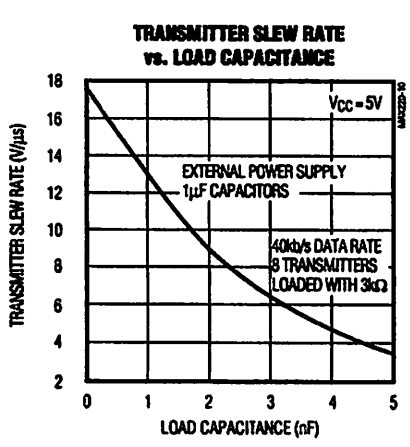
(MAX225, V_{CC} = 5.0V ±5%; MAX244–MAX249, V_{CC} = +5.0V ±10%, external capacitors C1–C4 = 1µF; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

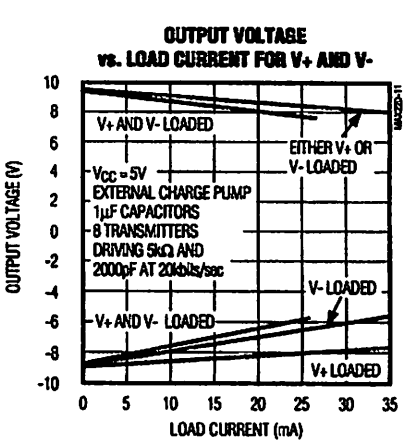
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
POWER SUPPLY AND CONTROL LOGIC						
Operating Supply Voltage		MAX225	4.75		5.25	V
		MAX244–MAX249	4.5		5.5	
V_{CC} Supply Current (normal operation)	No load	MAX225		10	20	mA
		MAX244–MAX249		11	30	
	3kΩ loads on all outputs	MAX225		40		
		MAX244–MAX249		57		
Shutdown Supply Current	T_A = +25°C			8	25	µA
	T_A = T_{MIN} to T_{MAX}				50	
Control Input	Leakage current				±1	µA
	Threshold low			1.4	0.8	V
	Threshold high		2.4	1.4		
AC CHARACTERISTICS						
Transition Slew Rate	C_L = 50pF to 2500pF, R_L = 3kΩ to 7kΩ, V_{CC} = 5V, T_A = +25°C, measured from +3V to -3V or -3V to +3V		5	10	30	V/µs
Transmitter Propagation Delay TLL to RS-232 (normal operation), Figure 1	t_{PHLT}			1.3	3.5	µs
	t_{PLHT}			1.5	3.5	
Receiver Propagation Delay TLL to RS-232 (normal operation), Figure 2	t_{PHLR}			0.6	1.5	µs
	t_{PLHR}			0.6	1.5	
Receiver Propagation Delay TLL to RS-232 (low-power mode), Figure 2	t_{PHLS}			0.6	10	µs
	t_{PLHS}			3.0	10	
Transmitter + to - Propagation Delay Difference (normal operation)	t_{PHLT} - t_{PLHT}			350		ns
Receiver + to - Propagation Delay Difference (normal operation)	t_{PHLR} - t_{PLHR}			350		ns
Receiver-Output Enable Time, Figure 3	t_{ER}			100	500	ns
Receiver-Output Disable Time, Figure 3	t_{DR}			100	500	ns
Transmitter Enable Time	t_{ET}	MAX246–MAX249 (excludes charge-pump start-up)		5		µs
		MAX225/MAX245–MAX249 (includes charge-pump start-up)		10		ms
Transmitter Disable Time, Figure 4	t_{DT}			100		ns

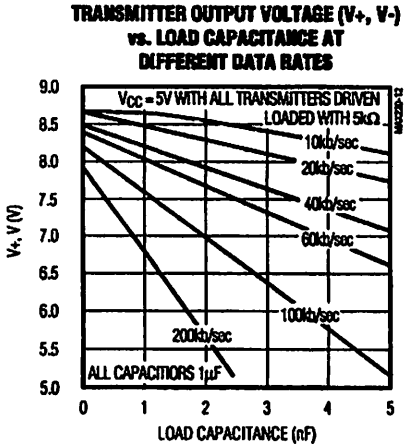
Note 5: The 300Ω minimum specification complies with EIA/TIA-232E, but the actual resistance when in shutdown mode or V_{CC} = 0V is 10MΩ as is implied by the leakage specification.

Typical Operating Characteristics

MAX225/MAX244–MAX249







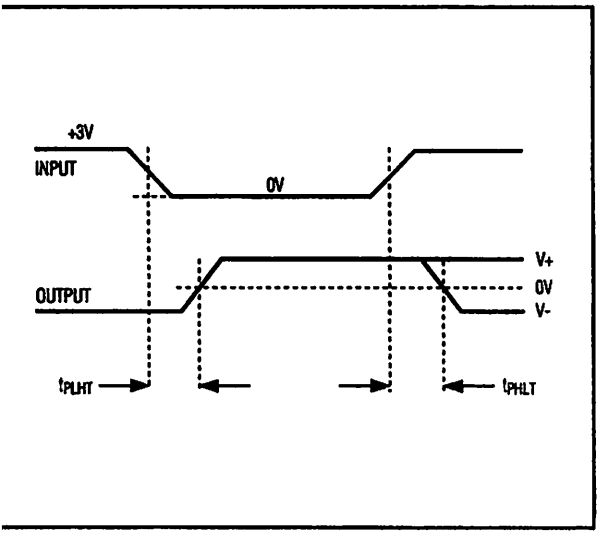


Figure 1. Transmitter Propagation-Delay Timing

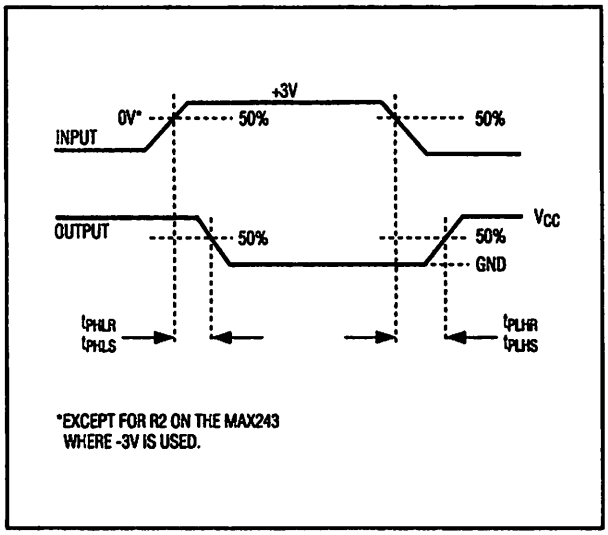


Figure 2. Receiver Propagation-Delay Timing

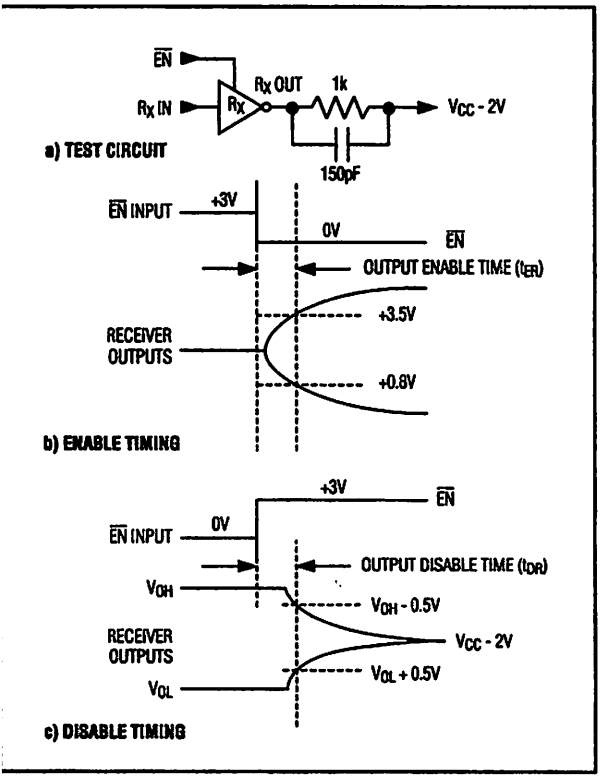


Figure 3. Receiver-Output Enable and Disable Timing

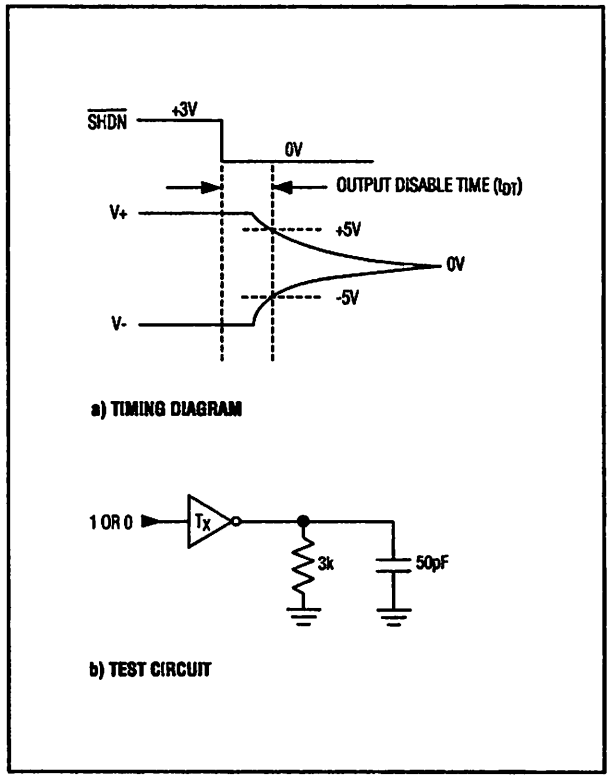


Figure 4. Transmitter-Output Disable Timing

Table 1a. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS	RECEIVERS
0	0	Normal Operation	All Active	All Active
0	1	Normal Operation	All Active	All 3-State
1	0	Shutdown	All 3-State	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State

Table 1b. MAX245 Control Pin Configurations

$\overline{ENT}$	$\overline{ENR}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All Active	RA1–RA4 3-State, RA5 Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All 3-State	All Low-Power Receive Mode	All Low-Power Receive Mode
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RB5 Low-Power Receive Mode

Table 1c. MAX246 Control Pin Configurations

$\overline{ENA}$	$\overline{ENB}$	OPERATION STATUS	TRANSMITTERS		RECEIVERS	
			TA1–TA4	TB1–TB4	RA1–RA5	RB1–RB5
0	0	Normal Operation	All Active	All Active	All Active	All Active
0	1	Normal Operation	All Active	All 3-State	All Active	RB1–RB4 3-State, RB5 Active
1	0	Shutdown	All 3-State	All Active	RA1–RA4 3-State, RA5 Active	All Active
1	1	Shutdown	All 3-State	All 3-State	RA1–RA4 3-State, RA5 Low-Power Receive Mode	RB1–RB4 3-State, RA5 Low-Power Receive Mode

Table 1d. MAX247/MAX248/MAX249 Control Pin Configurations

$\overline{ENTA}$	$\overline{ENTB}$	$\overline{ENRA}$	$\overline{ENRB}$	OPERATION STATUS		TRANSMITTERS		RECEIVERS	
					MAX247	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB5
					MAX248	TA1–TA4	TB1–TB4	RA1–RA4	RB1–RB4
					MAX249	TA1–TA3	TB1–TB3	RA1–RA5	RB1–RB5
0	0	0	0	Normal Operation		All Active	All Active	All Active	All Active
0	0	0	1	Normal Operation		All Active	All Active	All Active	All 3-State, except RB5 stays active on MAX247
0	0	1	0	Normal Operation		All Active	All Active	All 3-State	All Active
0	0	1	1	Normal Operation		All Active	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
0	1	0	0	Normal Operation		All Active	All 3-State	All Active	All Active
0	1	0	1	Normal Operation		All Active	All 3-State	All Active	All 3-State, except RB5 stays active on MAX247
0	1	1	0	Normal Operation		All Active	All 3-State	All 3-State	All Active
0	1	1	1	Normal Operation		All Active	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247
1	0	0	0	Normal Operation		All 3-State	All Active	All Active	All Active
1	0	0	1	Normal Operation		All 3-State	All Active	All Active	All 3-State, except RB5 stays active on MAX247
1	0	1	0	Normal Operation		All 3-State	All Active	All 3-State	All Active
1	0	1	1	Normal Operation		All 3-State	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247
1	1	0	0	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	Low-Power Receive Mode
1	1	0	1	Shutdown		All 3-State	All 3-State	Low-Power Receive Mode	All 3-State, except RB5 stays active on MAX247
1	1	1	0	Shutdown		All 3-State	All 3-State	All 3-State	Low-Power Receive Mode
1	1	1	1	Shutdown		All 3-State	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247

Detailed Description

The MAX220–MAX249 contain four sections: dual charge-pump DC-DC voltage converters, RS-232 drivers, RS-232 receivers, and receiver and transmitter enable control inputs.

Dual Charge-Pump Voltage Converter

The MAX220–MAX249 have two internal charge-pumps that convert +5V to ±10V (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double the +5V input to +10V on C3 at the V+ output. The second converter uses capacitor C2 to invert +10V to -10V on C4 at the V- output.

A small amount of power may be drawn from the +10V (V+) and -10V (V-) outputs to power external circuitry (see the *Typical Operating Characteristics* section), except on the MAX225 and MAX245–MAX247, where these pins are not available. V+ and V- are not regulated, so the output voltage drops with increasing load current. Do not load V+ and V- to a point that violates the minimum ±5V EIA/TIA-232E driver output voltage when sourcing current from V+ and V- to external circuitry.

When using the shutdown feature in the MAX222, MAX225, MAX230, MAX235, MAX236, MAX240, MAX241, and MAX245–MAX249, avoid using V+ and V- to power external circuitry. When these parts are shut down, V- falls to 0V, and V+ falls to +5V. For applications where a +10V external supply is applied to the V+ pin (instead of using the internal charge pump to generate +10V), the C1 capacitor must not be installed and the $\overline{SHDN}$ pin must be tied to V_{CC}. This is because V+ is internally connected to V_{CC} in shutdown mode.

RS-232 Drivers

The typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver and V_{CC} = +5V. Output swing is guaranteed to meet the EIA/TIA-232E and V.28 specification, which calls for ±5V minimum driver output levels under worst-case conditions. These include a minimum 3kΩ load, V_{CC} = +4.5V, and maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. The inputs of unused drivers can be left unconnected since 400kΩ input pull-up resistors to V_{CC} are built in (except for the MAX220). The pull-up resistors force the outputs of unused drivers low because all drivers invert. The internal input pull-up resistors typically source 12µA, except in shutdown mode where the pull-ups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically microamperes (maximum 25µA)—when in shutdown mode, in three-state mode, or when device power is removed. Outputs can be driven to ±15V. The power-supply current typically drops to 8µA in shutdown mode. The MAX220 does not have pull-up resistors to force the outputs of the unused drivers low. Connect unused inputs to GND or V_{CC}.

The MAX239 has a receiver three-state control line, and the MAX223, MAX225, MAX235, MAX236, MAX240, and MAX241 have both a receiver three-state control line and a low-power shutdown control. Table 2 shows the effects of the shutdown control and receiver three-state control on the receiver outputs.

The receiver TTL/CMOS outputs are in a high-impedance, three-state mode whenever the three-state enable line is high (for the MAX225/MAX235/MAX236/MAX239–MAX241), and are also high-impedance whenever the shutdown control line is high.

When in low-power shutdown mode, the driver outputs are turned off and their leakage current is less than 1µA with the driver output pulled to ground. The driver output leakage remains less than 1µA, even if the transmitter output is backdriven between 0V and (V_{CC} + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also zener clamped to approximately V_{CC} + 6V, with a series impedance of 1kΩ.

The driver output slew rate is limited to less than 30V/µs as required by the EIA/TIA-232E and V.28 specifications. Typical slew rates are 24V/µs unloaded and 10V/µs loaded with 3Ω and 2500pF.

RS-232 Receivers

EIA/TIA-232E and V.28 specifications define a voltage level greater than 3V as a logic 0, so all receivers invert. Input thresholds are set at 0.8V and 2.4V, so receivers respond to TTL level inputs as well as EIA/TIA-232E and V.28 levels.

The receiver inputs withstand an input overvoltage up to ±25V and provide input terminating resistors with

Table 2. Three-State Control of Receivers

PART	SHDN	$\overline{SHDN}$	EN	$\overline{EN(R)}$	RECEIVERS
MAX223	—	Low High High	X Low High	—	High Impedance Active High Impedance
MAX225	—	—	—	Low High	High Impedance Active
MAX235 MAX236 MAX240	Low Low High	—	—	Low High X	High Impedance Active High Impedance

nominal 5kΩ values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and EIA/TIA-232E.

The receiver input hysteresis is typically 0.5V with a guaranteed minimum of 0.2V. This produces clear output transitions with slow-moving input signals, even with moderate amounts of noise and ringing. The receiver propagation delay is typically 600ns and is independent of input swing direction.

Low-Power Receive Mode

The low-power receive-mode feature of the MAX223, MAX242, and MAX245–MAX249 puts the IC into shutdown mode but still allows it to receive information. This is important for applications where systems are periodically awakened to look for activity. Using low-power receive mode, the system can still receive a signal that will activate it on command and prepare it for communication at faster data rates. This operation conserves system power.

Negative Threshold—MAX243

The MAX243 is pin compatible with the MAX232A, differing only in that RS-232 cable fault protection is removed on one of the two receiver inputs. This means that control lines such as CTS and RTS can either be driven or left floating without interrupting communication. Different cables are not needed to interface with different pieces of equipment.

The input threshold of the receiver without cable fault protection is -0.8V rather than +1.4V. Its output goes positive only if the input is connected to a control line that is actively driven negative. If not driven, it defaults to the 0 or "OK to send" state. Normally, the MAX243's other receiver (+1.4V threshold) is used for the data line (TD or RD), while the negative threshold receiver is connected to the control line (DTR, DTS, CTS, RTS, etc.).

Other members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA-232E specifications. This means a receiver output goes high whenever its input is driven negative, left floating, or shorted to ground. The high output tells the serial communications IC to stop sending data. To avoid this, the control lines must either be driven or connected with jumpers to an appropriate positive voltage level.

Shutdown—MAX222–MAX242

On the MAX222, MAX235, MAX236, MAX240, and MAX241, all receivers are disabled during shutdown. On the MAX223 and MAX242, two receivers continue to operate in a reduced power mode when the chip is in shutdown. Under these conditions, the propagation delay increases to about 2.5µs for a high-to-low input transition. When in shutdown, the receiver acts as a CMOS inverter with no hysteresis. The MAX223 and MAX242 also have a receiver output enable input ($\overline{EN}$ for the MAX242 and EN for the MAX223) that allows receiver output control independent of $\overline{SHDN}$ (SHDN for MAX241). With all other devices, $\overline{SHDN}$ (SHDN for MAX241) also disables the receiver outputs.

The MAX225 provides five transmitters and five receivers, while the MAX245 provides ten receivers and eight transmitters. Both devices have separate receiver and transmitter-enable controls. The charge pumps turn off and the devices shut down when a logic high is applied to the ENT input. In this state, the supply current drops to less than 25µA and the receivers continue to operate in a low-power receive mode. Driver outputs enter a high-impedance state (three-state mode). On the MAX225, all five receivers are controlled by the $\overline{ENR}$ input. On the MAX245, eight of the receiver outputs are controlled by the $\overline{ENR}$ input, while the remaining two receivers (RA5 and RB5) are always active. RA1–RA4 and RB1–RB4 are put in a three-state mode when $\overline{ENR}$ is a logic high.

Receiver and Transmitter Enable Control Inputs

The MAX225 and MAX245–MAX249 feature transmitter and receiver enable controls.

The receivers have three modes of operation: full-speed receive (normal active), three-state (disabled), and low-power receive (enabled receivers continue to function at lower data rates). The receiver enable inputs control the full-speed receive and three-state modes. The transmitters have two modes of operation: full-speed transmit (normal active) and three-state (disabled). The transmitter enable inputs also control the shutdown mode. The device enters shutdown mode when all transmitters are disabled. Enabled receivers function in the low-power receive mode when in shutdown.

Tables 1a–1d define the control states. The MAX244 has no control pins and is not included in these tables.

The MAX246 has ten receivers and eight drivers with two control pins, each controlling one side of the device. A logic high at the A-side control input ($\overline{ENA}$) causes the four A-side receivers and drivers to go into a three-state mode. Similarly, the B-side control input ($\overline{ENB}$) causes the four B-side drivers and receivers to go into a three-state mode. As in the MAX245, one A-side and one B-side receiver (RA5 and RB5) remain active at all times. The entire device is put into shutdown mode when both the A and B sides are disabled ($\overline{ENA}$ = $\overline{ENB}$ = +5V).

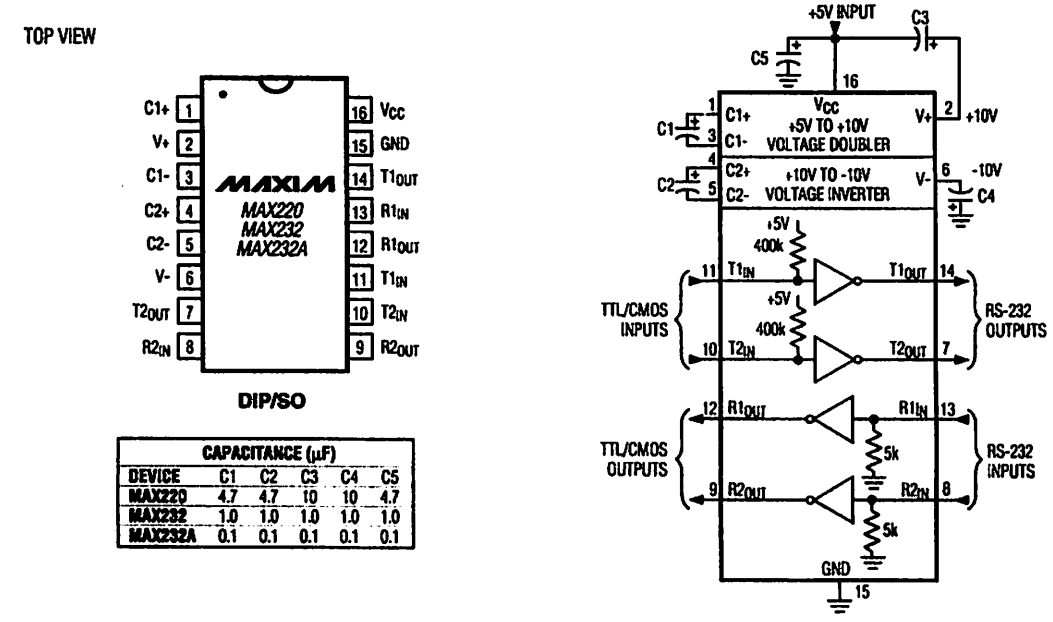
The MAX247 provides nine receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs each control four drivers. The ninth receiver (RB5) is always active. The device enters shutdown mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX248 provides eight receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control four drivers each. This part does not have an always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$.

The MAX249 provides ten receivers and six drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control five receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs control three drivers each. There is no always-active receiver. The device enters shutdown mode and transmitters go into a three-state mode with a logic high on both $\overline{ENTA}$ and $\overline{ENTB}$. In shutdown mode, active receivers operate in a low-power receive mode at data rates up to 20kbits/sec.

Applications Information

Figures 5 through 25 show pin configurations and typical operating circuits. In applications that are sensitive to power-supply noise, V_{CC} should be decoupled to ground with a capacitor of the same value as C1 and C2 connected as close as possible to the device.



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

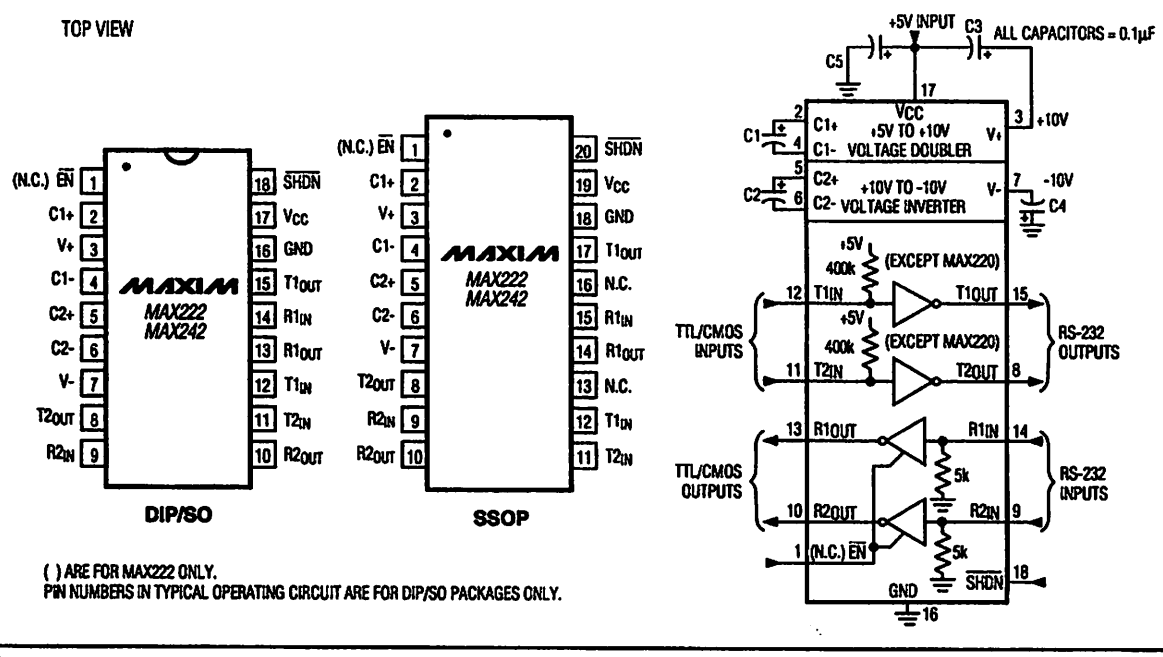
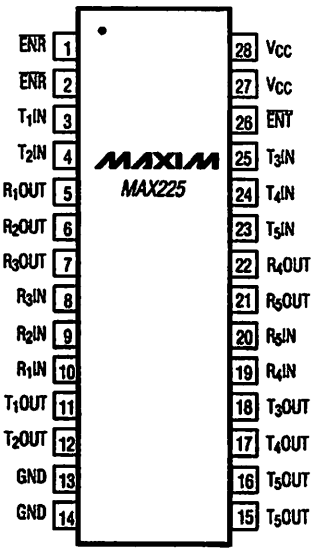


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

TOP VIEW



SO

MAX225 FUNCTIONAL DESCRIPTION

5 RECEIVERS
5 TRANSMITTERS
2 CONTROL PINS
1 RECEIVER ENABLE ($\overline{ENR}$)
1 TRANSMITTER ENABLE ($\overline{ENT}$)

PINS ($\overline{ENR}$, GND, V_{CC}, T_5OUT) ARE INTERNALLY CONNECTED.
CONNECT EITHER OR BOTH EXTERNALLY. T_5OUT IS A SINGLE DRIVER.

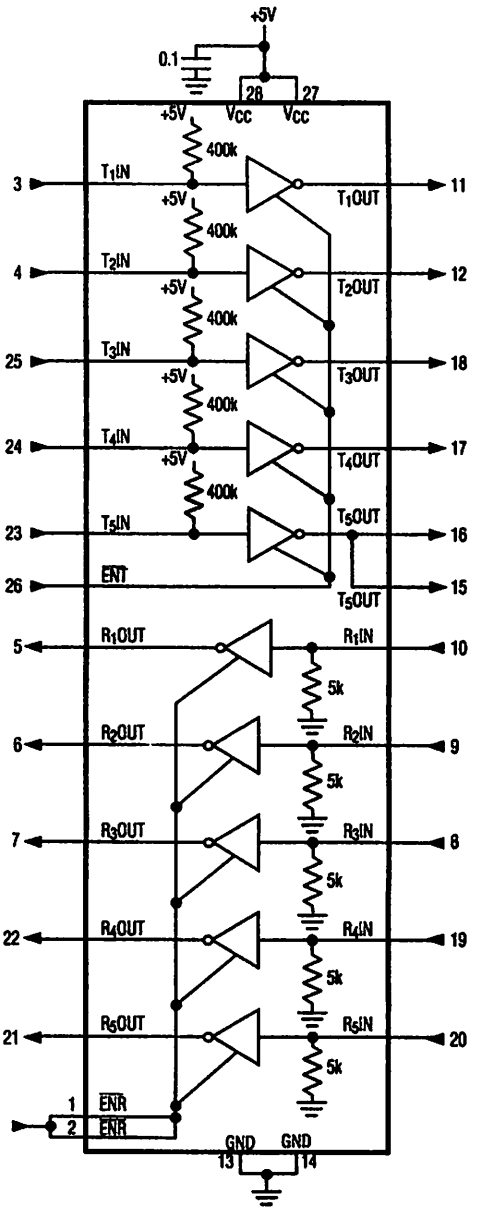


Figure 7. MAX225 Pin Configuration and Typical Operating Circuit

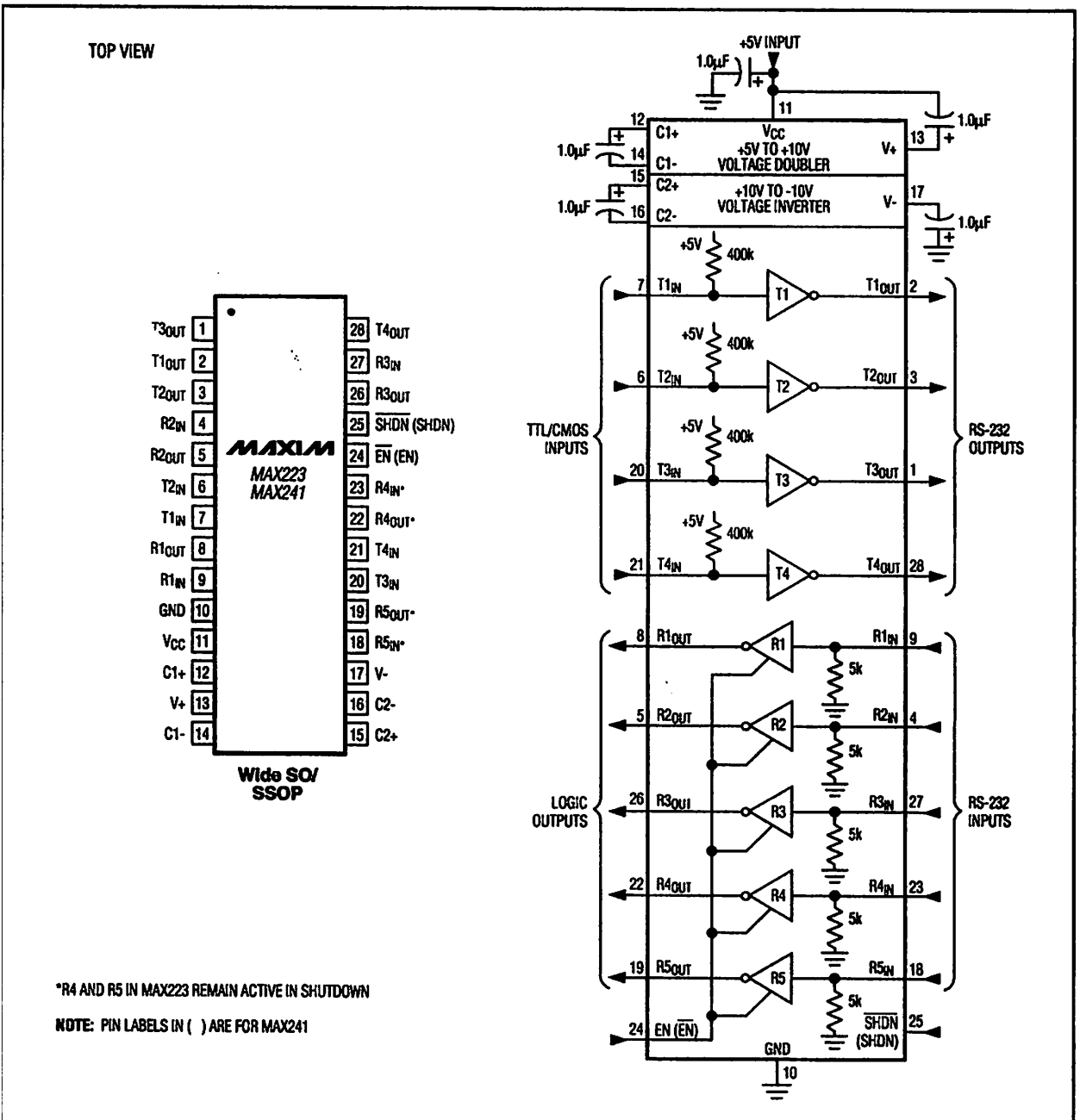


Figure 8. MAX223/MAX241 Pin Configuration and Typical Operating Circuit

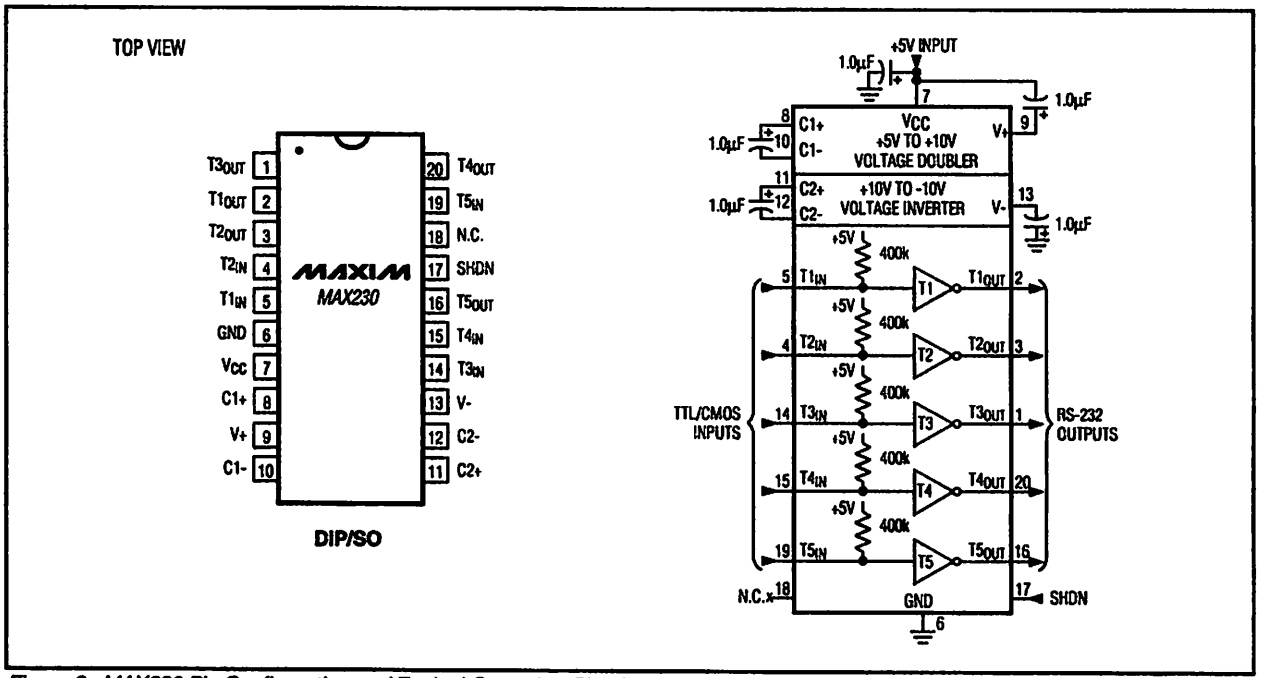


Figure 9. MAX230 Pin Configuration and Typical Operating Circuit

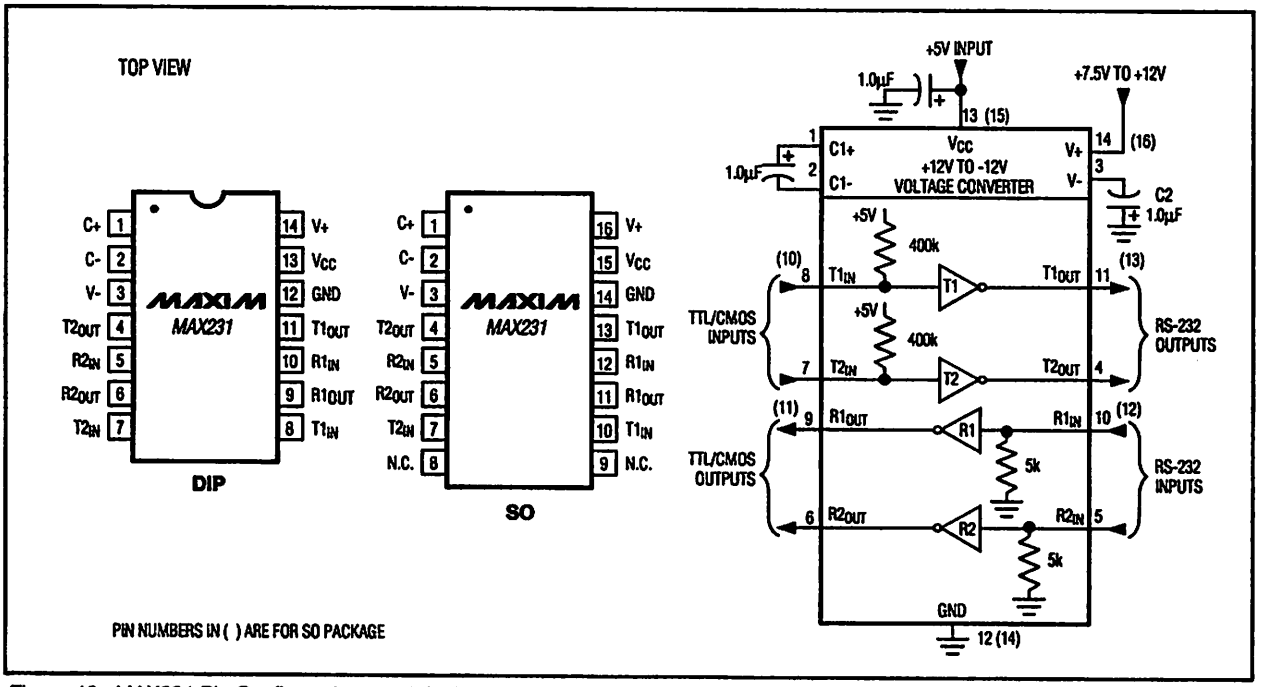


Figure 10. MAX231 Pin Configurations and Typical Operating Circuit

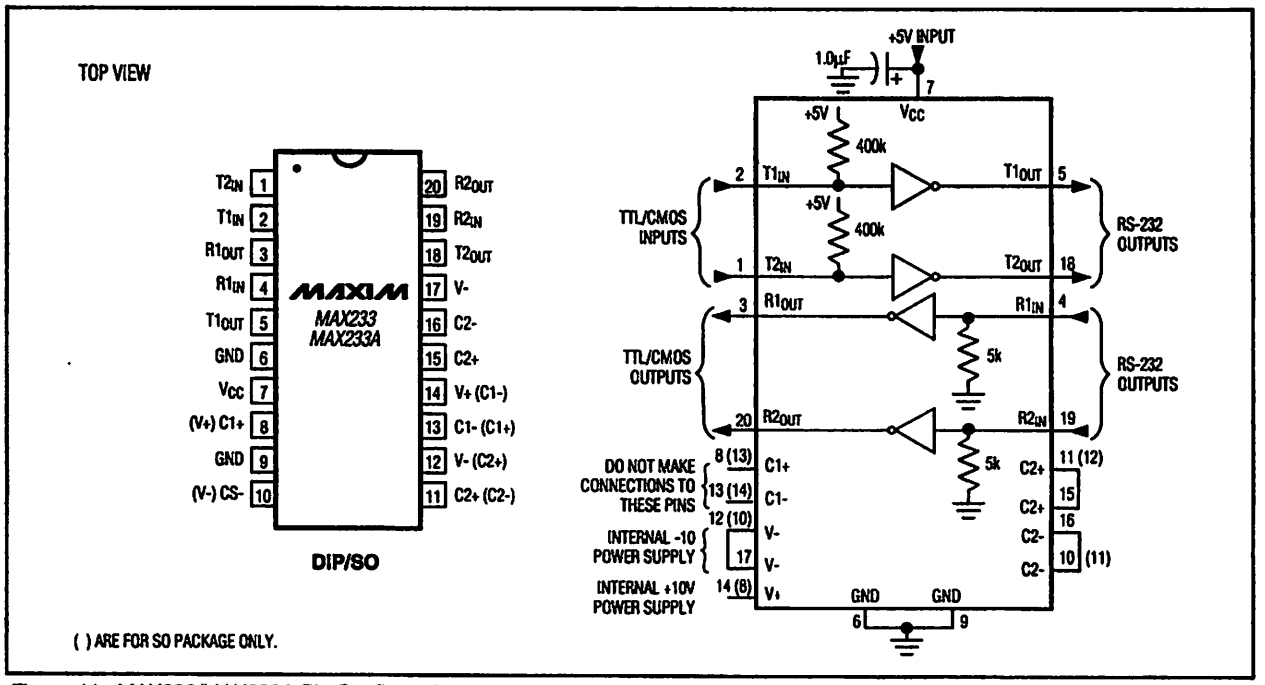


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

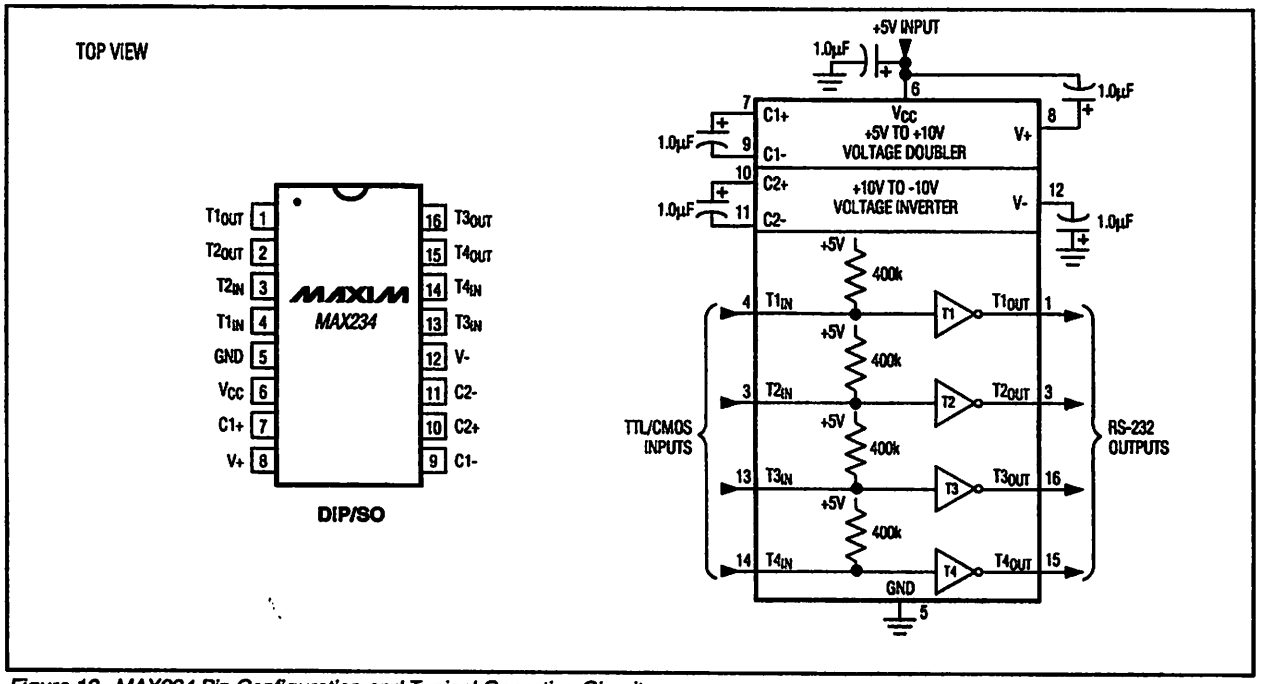


Figure 12. MAX234 Pin Configuration and Typical Operating Circuit

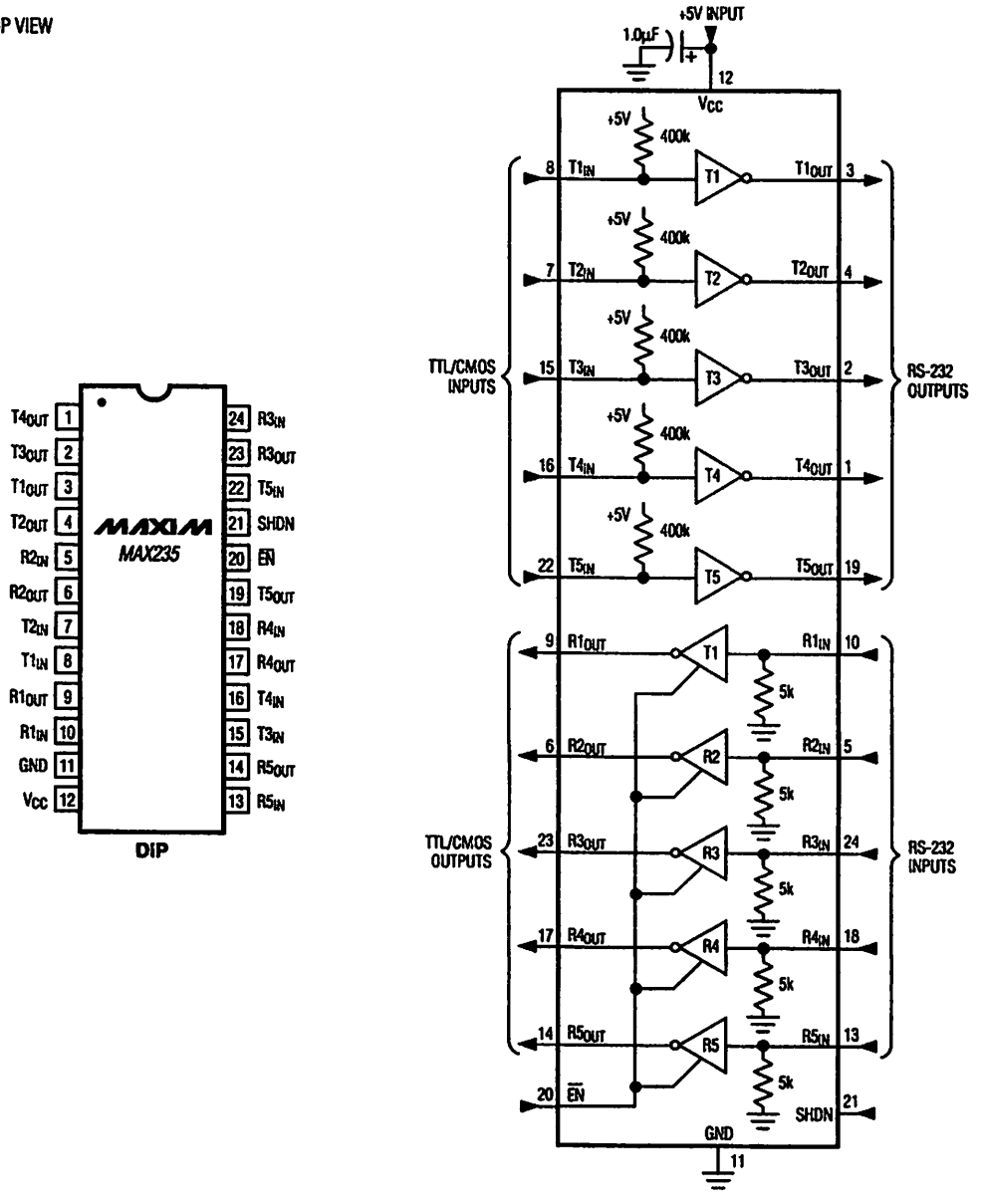


Figure 13. MAX235 Pin Configuration and Typical Operating Circuit

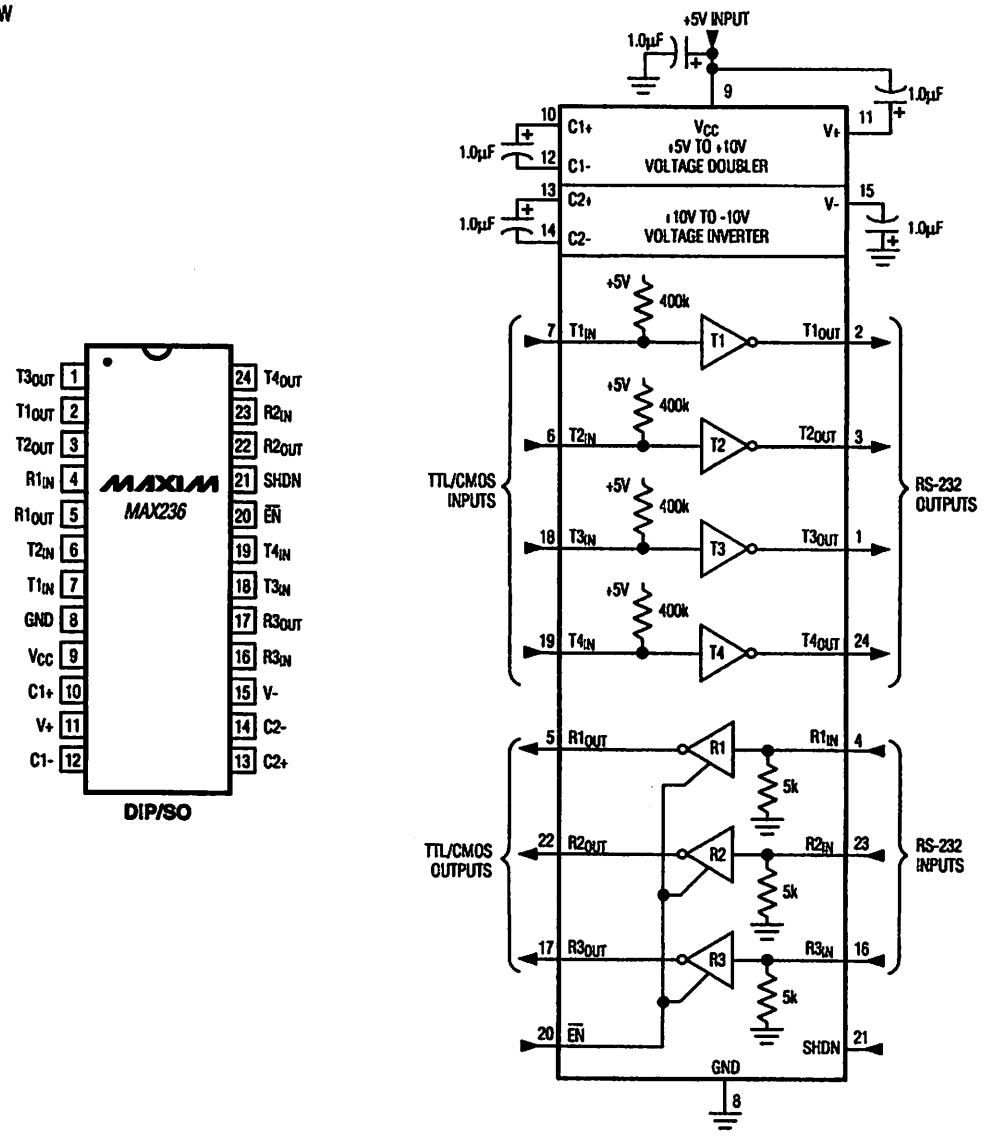


Figure 14. MAX236 Pin Configuration and Typical Operating Circuit

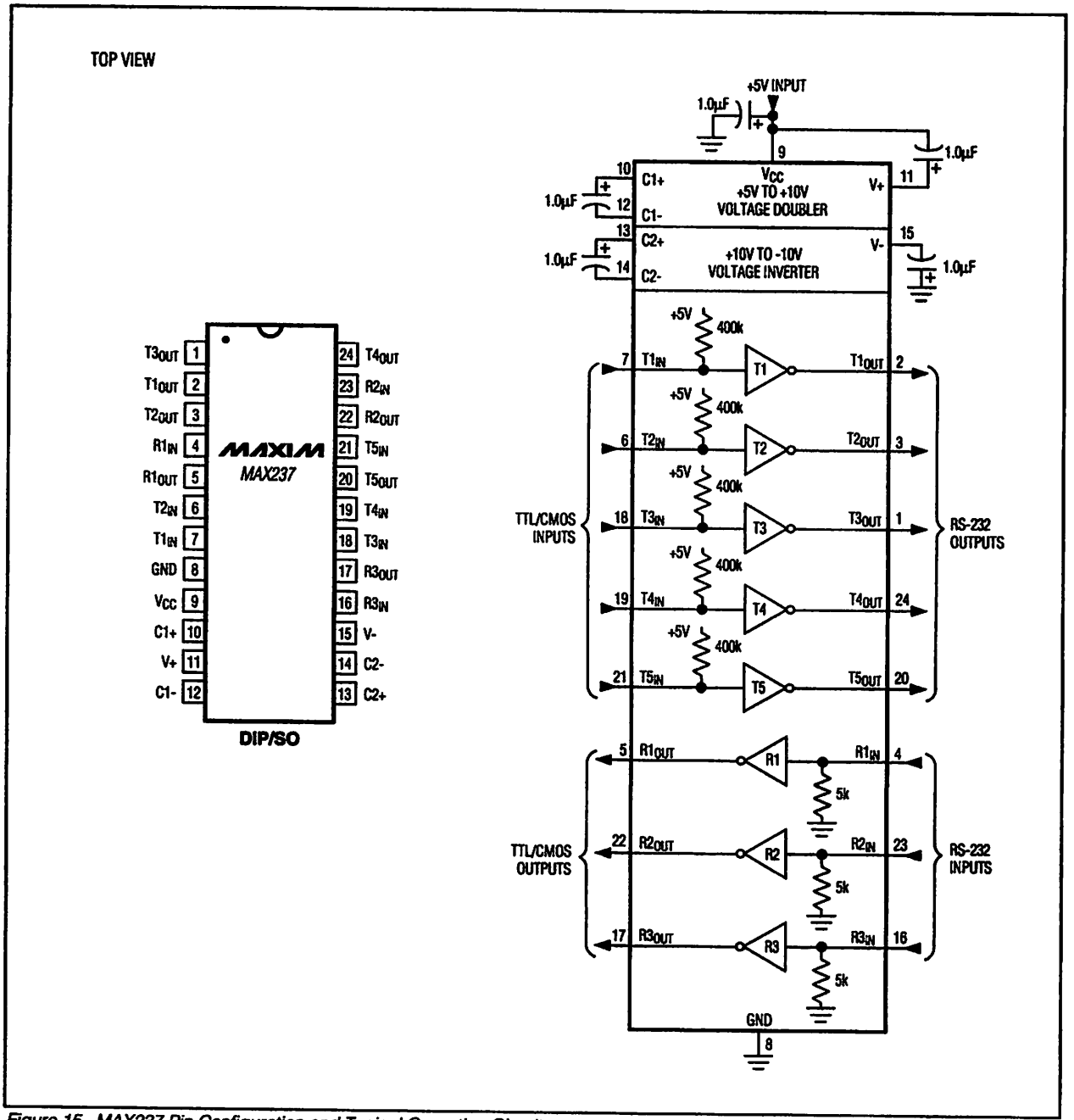
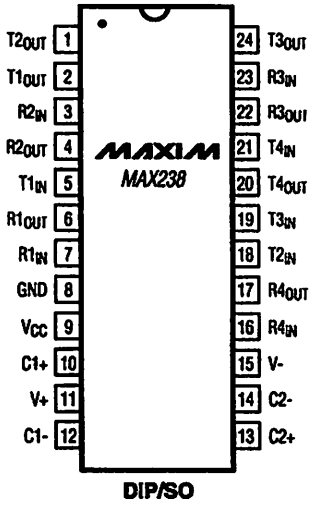


Figure 15. MAX237 Pin Configuration and Typical Operating Circuit



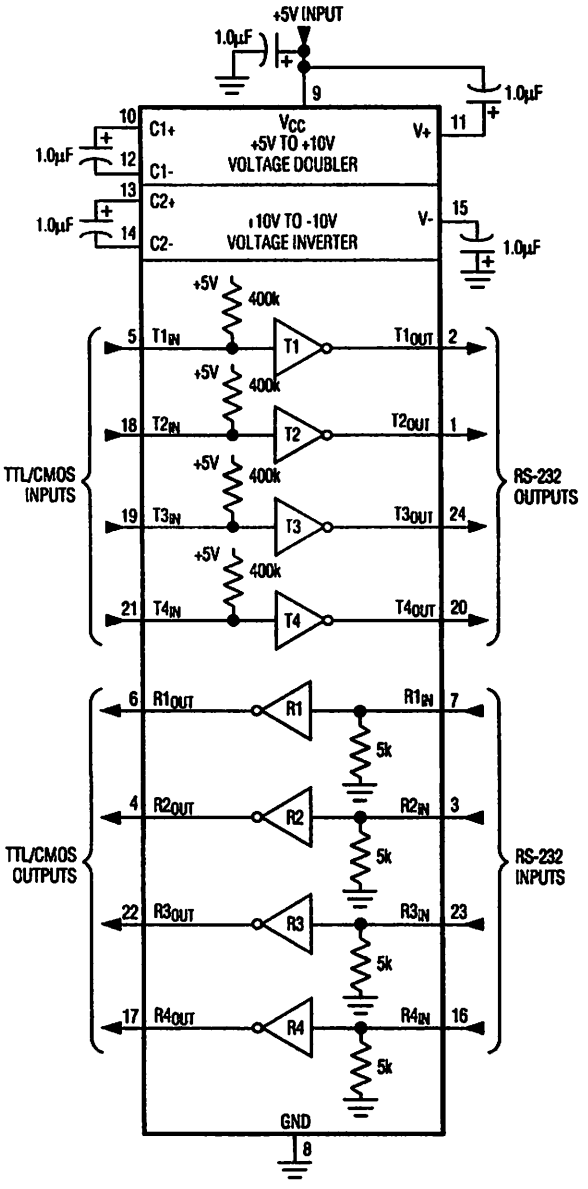
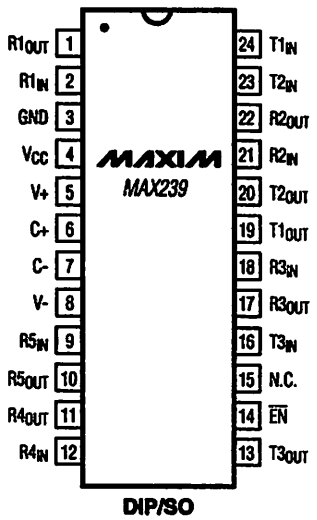


Figure 16. MAX238 Pin Configuration and Typical Operating Circuit



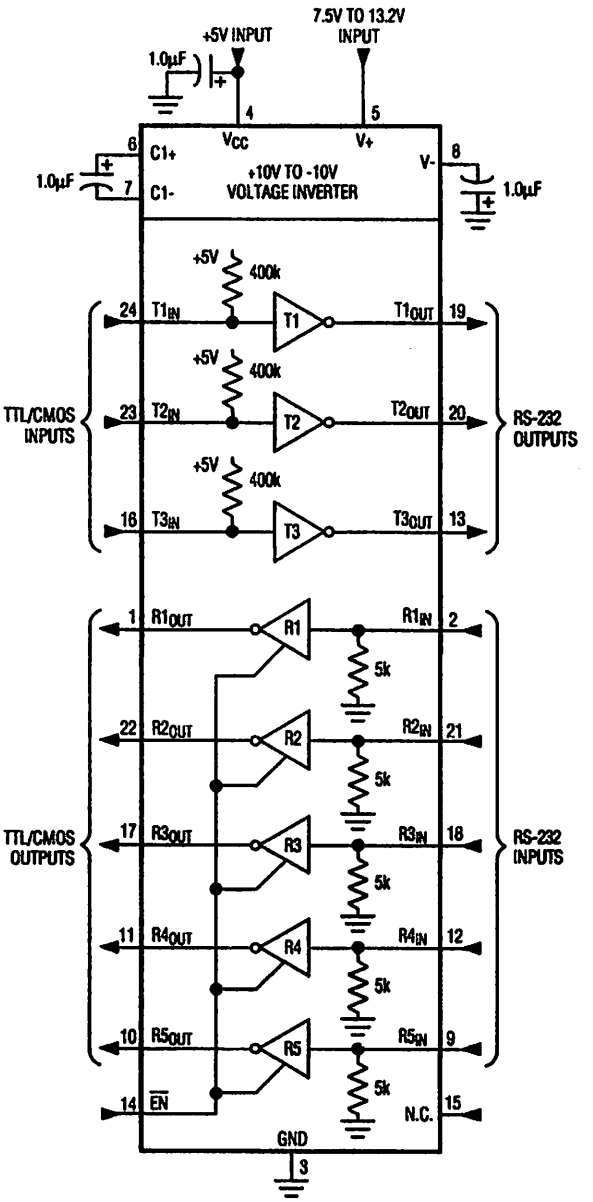


Figure 17. MAX239 Pin Configuration and Typical Operating Circuit

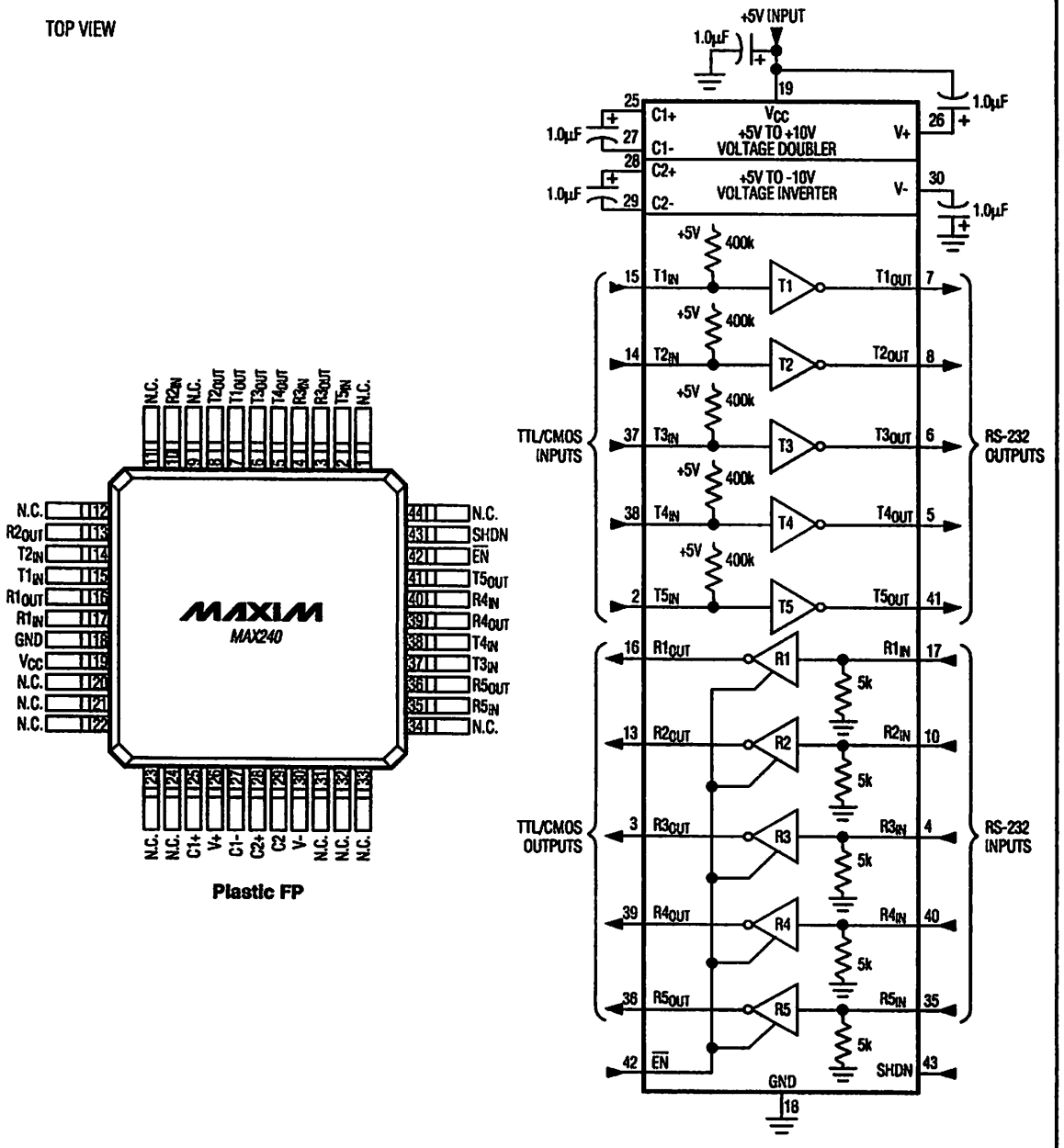
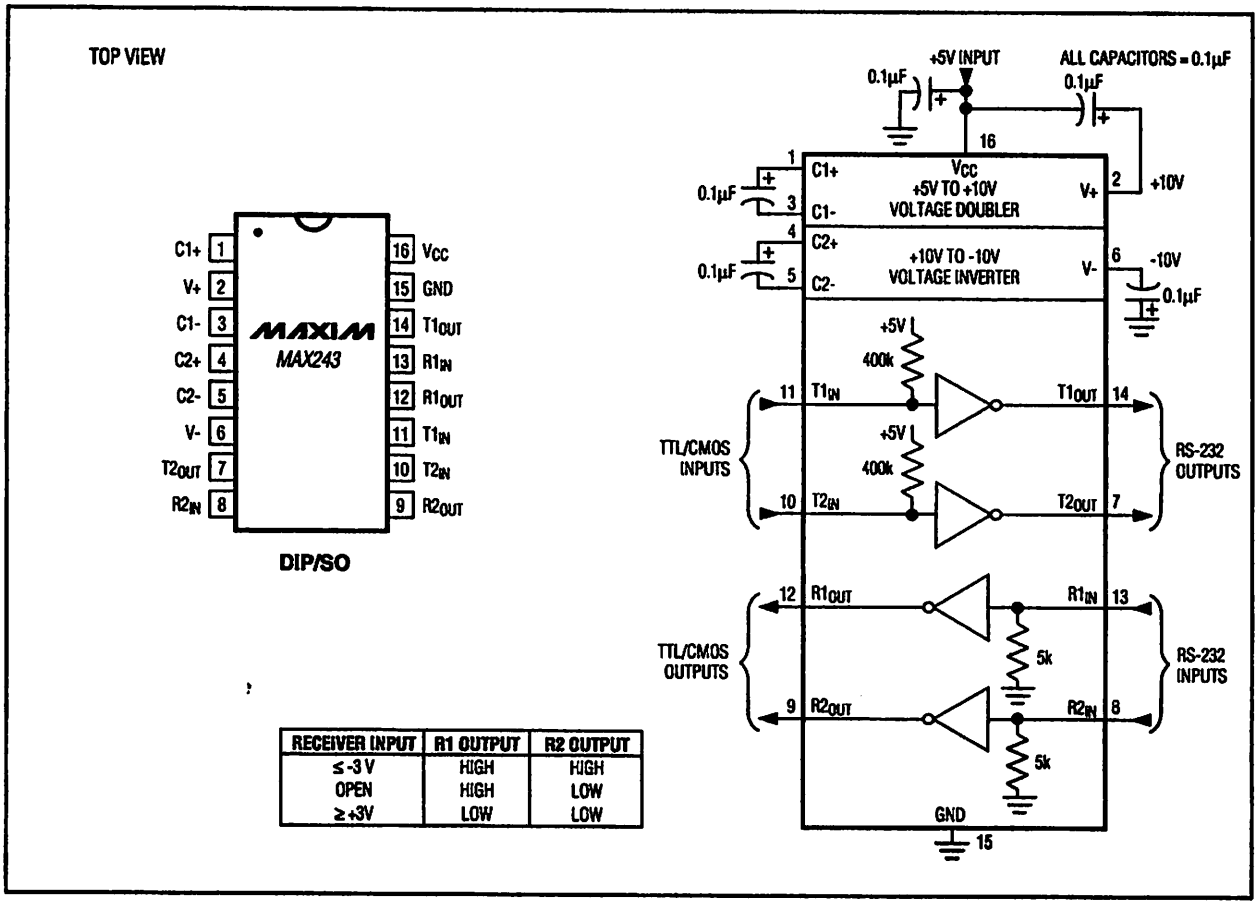
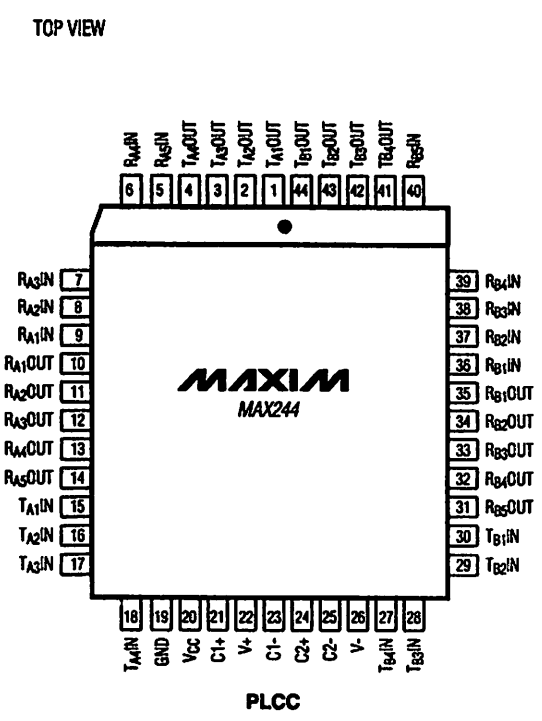


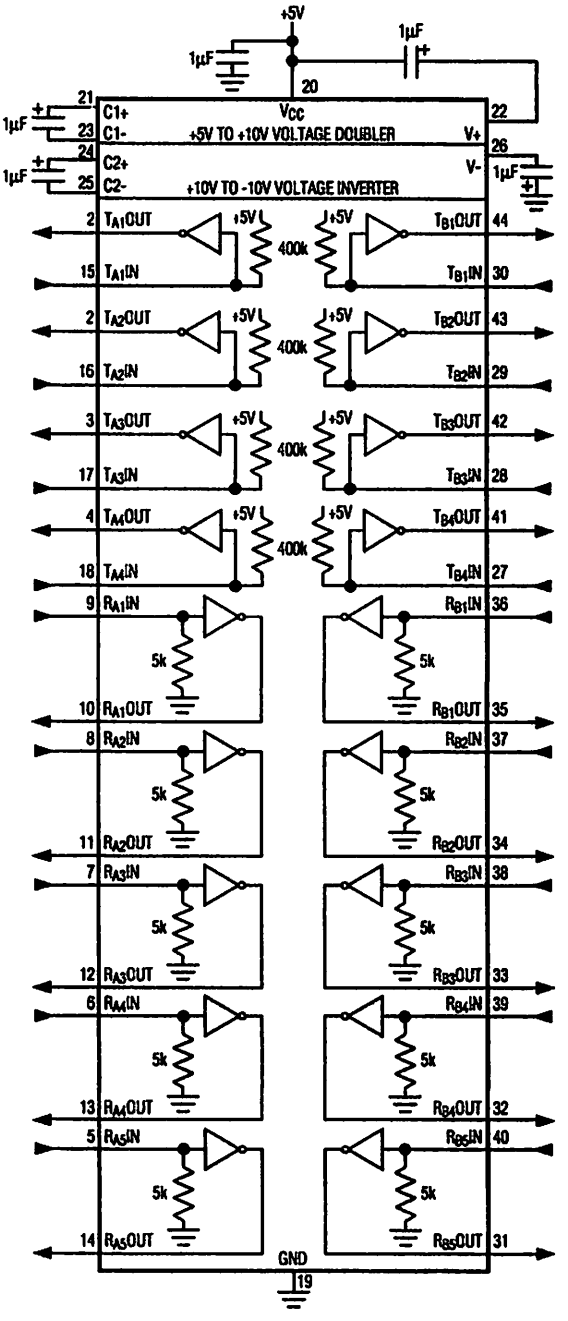
Figure 18. MAX240 Pin Configuration and Typical Operating Circuit



RECEIVER INPUT	R1 OUTPUT	R2 OUTPUT
≤ -3 V	HIGH	HIGH
OPEN	HIGH	LOW
≥ +3V	LOW	LOW

Figure 19. MAX243 Pin Configuration and Typical Operating Circuit





MAX249 FUNCTIONAL DESCRIPTION

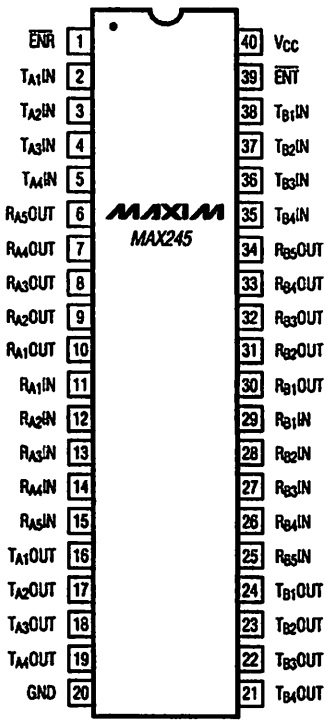
10 RECEIVERS
- 5 A-SIDE RECEIVER
- 5 B-SIDE RECEIVER

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

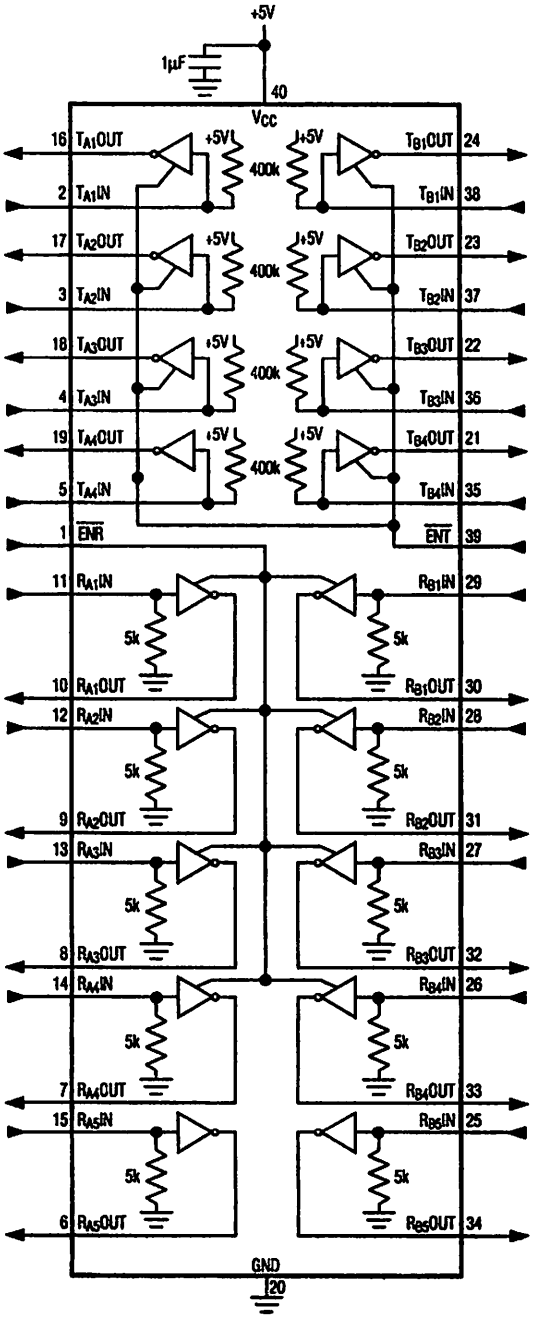
NO CONTROL PINS

Figure 20. MAX244 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP



MAX245 FUNCTIONAL DESCRIPTION

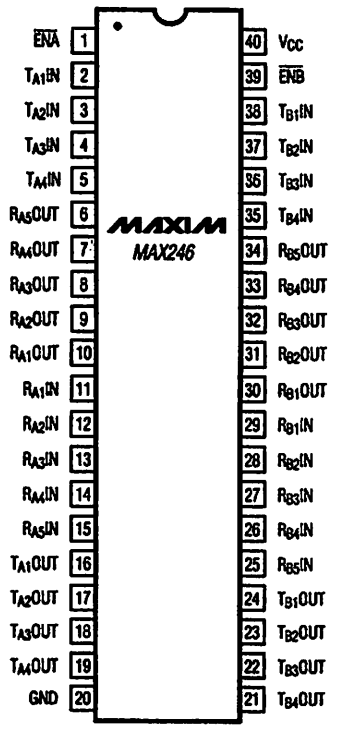
10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

8 TRANSMITTTERS
- 4 A-SIDE TRANSMITTERS

2 CONTROL PINS
- 1 RECEIVER ENABLE ($\overline{ENR}$)
- 1 TRANSMITTER ENABLE ($\overline{ENT}$)

Figure 21. MAX245 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP

MAX248 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

2 CONTROL PINS
- ENABLE A-SIDE ($\overline{ENA}$)
- ENABLE B-SIDE ($\overline{ENB}$)

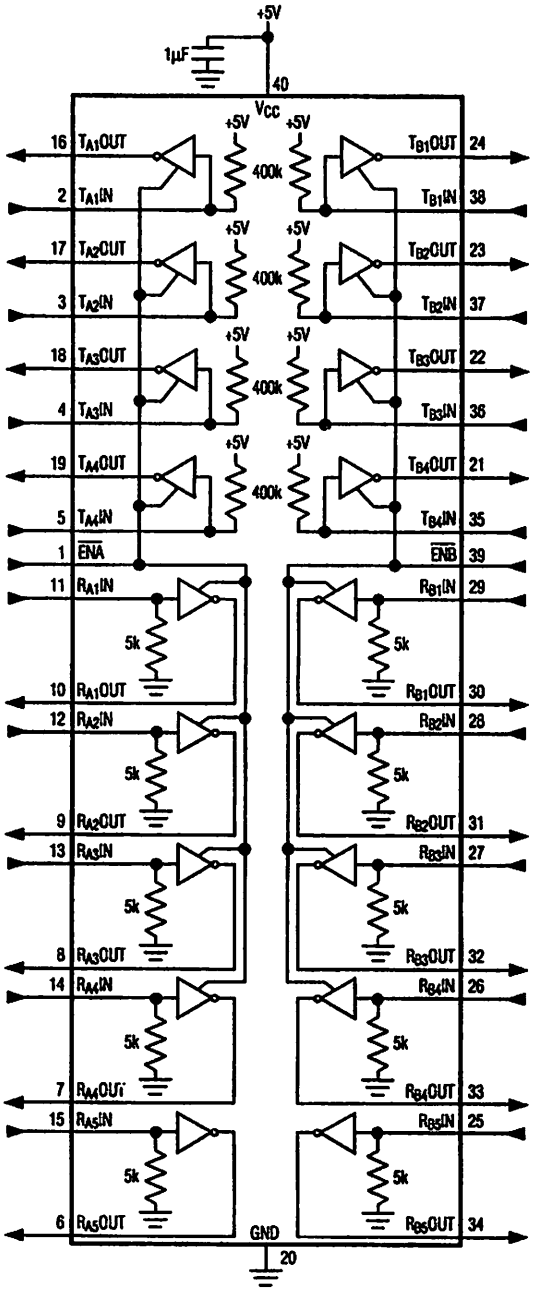
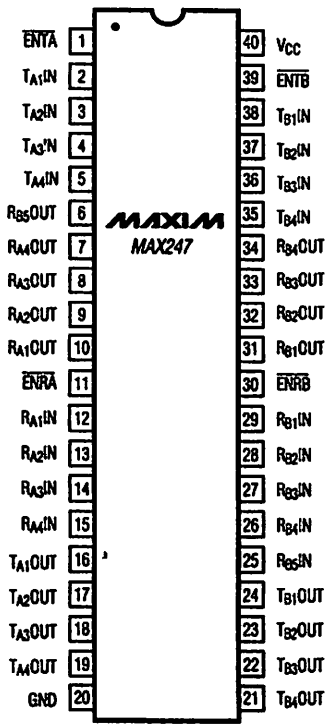


Figure 22. MAX246 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP

MAX247 FUNCTIONAL DESCRIPTION

9 RECEIVERS
- 4 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVERr B-SIDE ($\overline{ENTB}$)

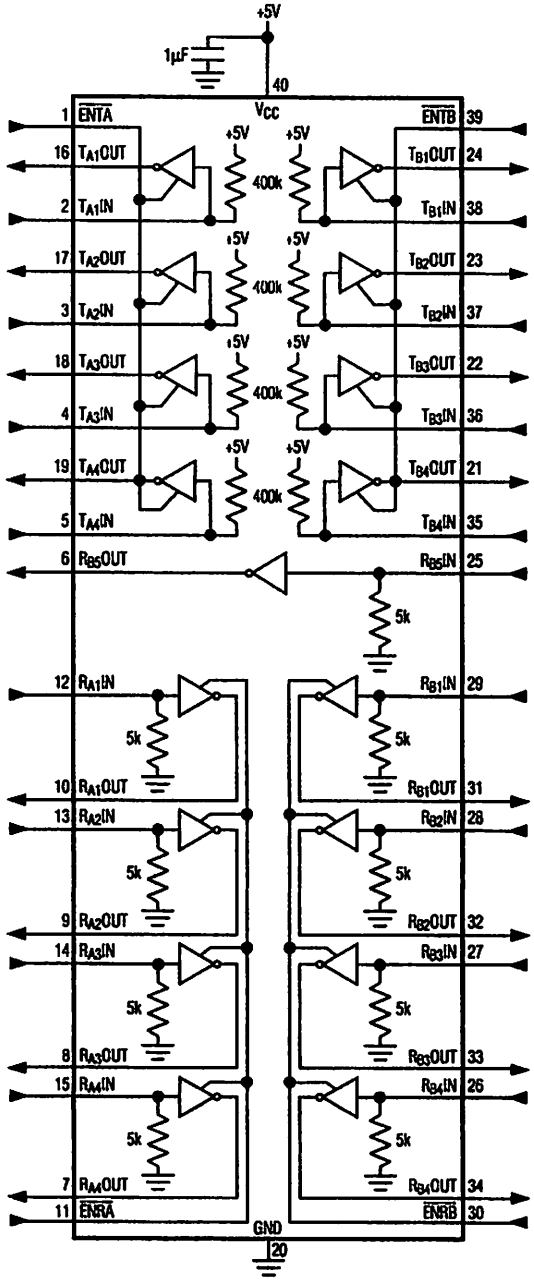
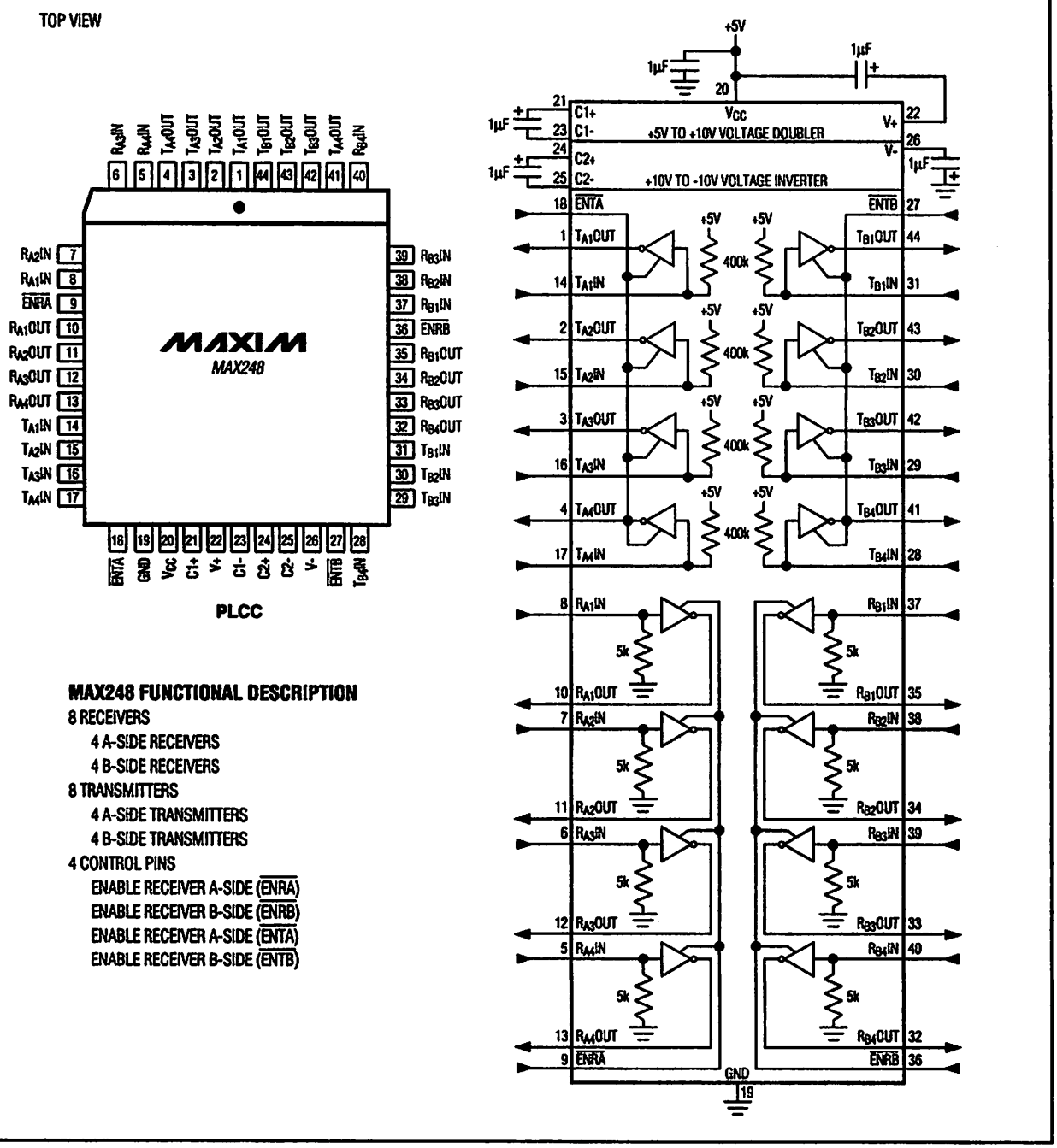


Figure 23. MAX247 Pin Configuration and Typical Operating Circuit



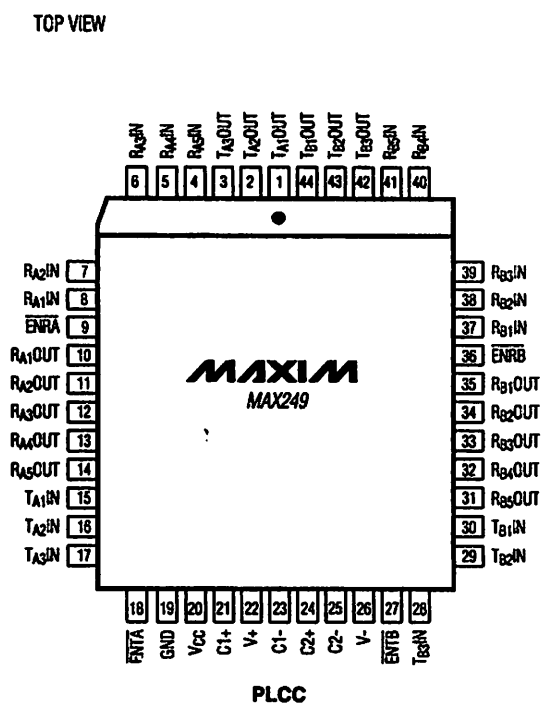
MAX248 FUNCTIONAL DESCRIPTION

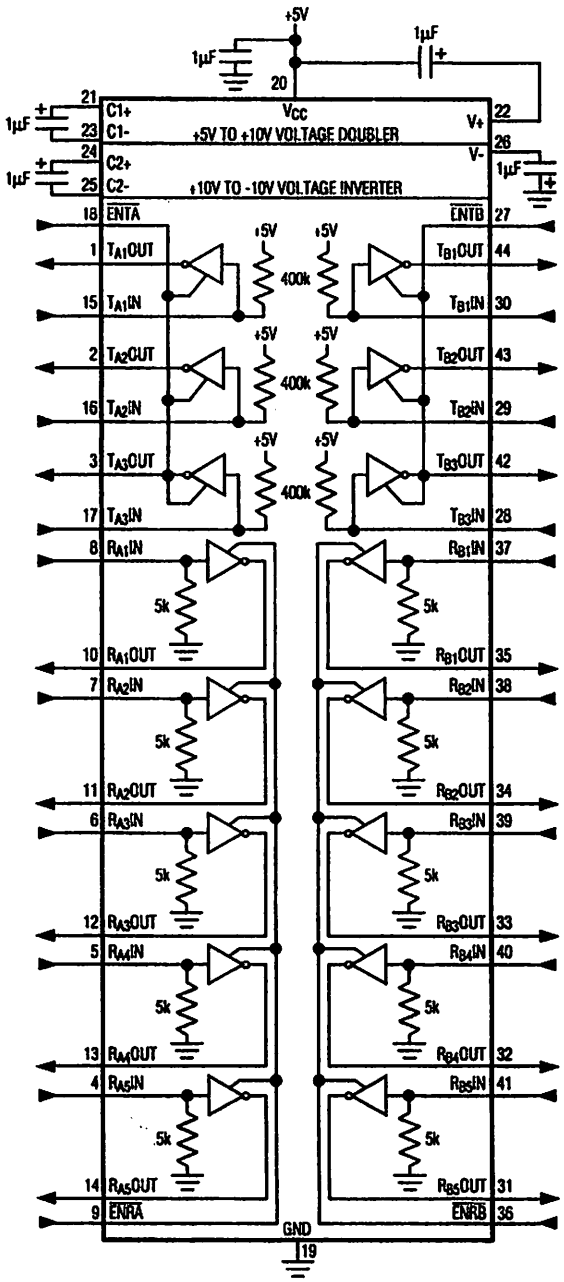
8 RECEIVERS
- 4 A-SIDE RECEIVERS
- 4 B-SIDE RECEIVERS

8 TRANSMITTERS
- 4 A-SIDE TRANSMITTERS
- 4 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENTB}$)

Figure 24. MAX248 Pin Configuration and Typical Operating Circuit





MAX249 FUNCTIONAL DESCRIPTION

10 RECEIVERS
- 5 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS

6 TRANSMITTERS
- 3 A-SIDE TRANSMITTERS
- 3 B-SIDE TRANSMITTERS

4 CONTROL PINS
- ENABLE RECEIVER A-SIDE ($\overline{ENRA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENRB}$)
- ENABLE RECEIVER A-SIDE ($\overline{ENTA}$)
- ENABLE RECEIVER B-SIDE ($\overline{ENTB}$)

Figure 25. MAX249 Pin Configuration and Typical Operating Circuit

+5V-Powered, Multichannel RS-232 Drivers/Receivers

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX222CPN	0°C to +70°C	18 Plastic DIP
MAX222CWN	0°C to +70°C	18 Wide SO
MAX222C/D	0°C to +70°C	Dice*
MAX222EPN	-40°C to +85°C	18 Plastic DIP
MAX222EWN	-40°C to +85°C	18 Wide SO
MAX222EJN	-40°C to +85°C	18 CERDIP
MAX222MJN	-55°C to +125°C	18 CERDIP
MAX223CAI	0°C to +70°C	28 SSOP
MAX223CWI	0°C to +70°C	28 Wide SO
MAX223C/D	0°C to +70°C	Dice*
MAX223EAI	-40°C to +85°C	28 SSOP
MAX223EWI	-40°C to +85°C	28 Wide SO
MAX225CWI	0°C to +70°C	28 Wide SO
MAX225EWI	-40°C to +85°C	28 Wide SO
MAX230CPP	0°C to +70°C	20 Plastic DIP
MAX230CWP	0°C to +70°C	20 Wide SO
MAX230C/D	0°C to +70°C	Dice*
MAX230EPP	-40°C to +85°C	20 Plastic DIP
MAX230EWP	-40°C to +85°C	20 Wide SO
MAX230EJP	-40°C to +85°C	20 CERDIP
MAX230MJP	-55°C to +125°C	20 CERDIP
MAX231CPD	0°C to +70°C	14 Plastic DIP
MAX231CWE	0°C to +70°C	16 Wide SO
MAX231CJD	0°C to +70°C	14 CERDIP
MAX231C/D	0°C to +70°C	Dice*
MAX231EPD	-40°C to +85°C	14 Plastic DIP
MAX231EWE	-40°C to +85°C	16 Wide SO
MAX231EJD	-40°C to +85°C	14 CERDIP
MAX231MJD	-55°C to +125°C	14 CERDIP
MAX232CPE	0°C to +70°C	16 Plastic DIP
MAX232CSE	0°C to +70°C	16 Narrow SO
MAX232CWE	0°C to +70°C	16 Wide SO
MAX232C/D	0°C to +70°C	Dice*
MAX232EPE	-40°C to +85°C	16 Plastic DIP
MAX232ESE	-40°C to +85°C	16 Narrow SO
MAX232EWE	-40°C to +85°C	16 Wide SO
MAX232EJE	-40°C to +85°C	16 CERDIP
MAX232MJE	-55°C to +125°C	16 CERDIP
MAX232MLP	-55°C to +125°C	20 LCC
MAX232ACPE	0°C to +70°C	16 Plastic DIP
MAX232ACSE	0°C to +70°C	16 Narrow SO
MAX232ACWE	0°C to +70°C	16 Wide SO
MAX232AC/D	0°C to +70°C	Dice*
MAX232AEPE	-40°C to +85°C	16 Plastic DIP
MAX232AESE	-40°C to +85°C	16 Narrow SO
MAX232AEWE	-40°C to +85°C	16 Wide SO
MAX232AEJE	-40°C to +85°C	16 CERDIP
MAX232AMJE	-55°C to +125°C	16 CERDIP
MAX232AMLP	-55°C to +125°C	20 LCC
MAX233CPP	0°C to +70°C	20 Plastic DIP
MAX233EPP	-40°C to +85°C	20 Plastic DIP
MAX233ACPP	0°C to +70°C	20 Plastic DIP
MAX233ACWP	0°C to +70°C	20 Wide SO
MAX233AEPP	-40°C to +85°C	20 Plastic DIP
MAX233AEWP	-40°C to +85°C	20 Wide SO
MAX234CPE	0°C to +70°C	16 Plastic DIP
MAX234CWE	0°C to +70°C	16 Wide SO
MAX234C/D	0°C to +70°C	Dice*
MAX234EPE	-40°C to +85°C	16 Plastic DIP
MAX234EWE	-40°C to +85°C	16 Wide SO
MAX234EJE	-40°C to +85°C	16 CERDIP
MAX234MJE	-55°C to +125°C	16 CERDIP
MAX235CPG	0°C to +70°C	24 Wide Plastic DIP
MAX235EPG	-40°C to +85°C	24 Wide Plastic DIP
MAX235EDG	-40°C to +85°C	24 Ceramic SB
MAX235MDG	-55°C to +125°C	24 Ceramic SB
MAX236CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX236CWG	0°C to +70°C	24 Wide SO
MAX236C/D	0°C to +70°C	Dice*
MAX236ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX236EWG	-40°C to +85°C	24 Wide SO
MAX236ERG	-40°C to +85°C	24 Narrow CERDIP
MAX236MRG	-55°C to +125°C	24 Narrow CERDIP
MAX237CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX237CWG	0°C to +70°C	24 Wide SO
MAX237C/D	0°C to +70°C	Dice*
MAX237ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX237EWG	-40°C to +85°C	24 Wide SO
MAX237ERG	-40°C to +85°C	24 Narrow CERDIP
MAX237MRG	-55°C to +125°C	24 Narrow CERDIP
MAX238CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX238CWG	0°C to +70°C	24 Wide SO
MAX238C/D	0°C to +70°C	Dice*
MAX238ENG	-40°C to +85°C	24 Narrow Plastic DIP

** Contact factory for dice specifications.*

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX238EWG	-40°C to +85°C	24 Wide SO
MAX238ERG	-40°C to +85°C	24 Narrow CERDIP
MAX238MRG	-55°C to +125°C	24 Narrow CERDIP
MAX239CNG	0°C to +70°C	24 Narrow Plastic DIP
MAX239CWG	0°C to +70°C	24 Wide SO
MAX239C/D	0°C to +70°C	Dice*
MAX239ENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX239EWG	-40°C to +85°C	24 Wide SO
MAX239ERG	-40°C to +85°C	24 Narrow CERDIP
MAX239MRG	-55°C to +125°C	24 Narrow CERDIP
MAX240CMH	0°C to +70°C	44 Plastic FP
MAX240C/D	0°C to +70°C	Dice*
MAX241CAI	0°C to +70°C	28 SSOP
MAX241CWI	0°C to +70°C	28 Wide SO
MAX241C/D	0°C to +70°C	Dice*
MAX241EAI	-40°C to +85°C	28 SSOP
MAX241EWI	-40°C to +85°C	28 Wide SO
MAX242CAP	0°C to +70°C	20 SSOP
MAX242CPN	0°C to +70°C	18 Plastic DIP
MAX242CWN	0°C to +70°C	18 Wide SO
MAX242C/D	0°C to +70°C	Dice*
MAX242EPN	-40°C to +85°C	18 Plastic DIP
MAX242EWN	-40°C to +85°C	18 Wide SO
MAX242EJN	-40°C to +85°C	18 CERDIP
MAX242MJN	-55°C to +125°C	18 CERDIP
MAX243CPE	0°C to +70°C	16 Plastic DIP
MAX243CSE	0°C to +70°C	16 Narrow SO
MAX243CWE	0°C to +70°C	16 Wide SO
MAX243C/D	0°C to +70°C	Dice*
MAX243EPE	-40°C to +85°C	16 Plastic DIP
MAX243ESE	-40°C to +85°C	16 Narrow SO
MAX243EWE	-40°C to +85°C	16 Wide SO
MAX243EJE	-40°C to +85°C	16 CERDIP
MAX243MJE	-55°C to +125°C	16 CERDIP
MAX244CQH	0°C to +70°C	44 PLCC
MAX244C/D	0°C to +70°C	Dice*
MAX244EQH	-40°C to +85°C	44 PLCC
MAX245CPL	0°C to +70°C	40 Plastic DIP
MAX245C/D	0°C to +70°C	Dice*
MAX245EPL	-40°C to +85°C	40 Plastic DIP
MAX246CPL	0°C to +70°C	40 Plastic DIP
MAX246C/D	0°C to +70°C	Dice*
MAX246EPL	-40°C to +85°C	40 Plastic DIP
MAX247CPL	0°C to +70°C	40 Plastic DIP
MAX247C/D	0°C to +70°C	Dice*
MAX247EPL	-40°C to +85°C	40 Plastic DIP
MAX248CQH	0°C to +70°C	44 PLCC
MAX248C/D	0°C to +70°C	Dice*
MAX248EQH	-40°C to +85°C	44 PLCC
MAX249CQH	0°C to +70°C	44 PLCC
MAX249EQH	-40°C to +85°C	44 PLCC

** Contact factory for dice specifications.*