PERENCANAAN DAN PEMBUATAN ALAT UNTUK MENDETEKSI VOLUME MINYAK PADA TANDON SPBU YANG DI TRANSMISIKAN VIA MODEM BERBASIS MIKROKONTROLLER AT89S52 DAN TERINTEGRASI DENGAN PC

RIF'AN RACHMANI
97.17.005

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK ELEKTONIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

LEMBAR PERSETUJUAN

PERENCANAAN DAN PEMBUATAN ALAT UNTUK MENDETEKSI VOLUME MINYAK PADA TANDON SPBU YANG DI TRANSMISIKAN VIA MODEM BERBASIS MIKROKONTROLLER AT89S52 DAN TERINTEGRASI DENGAN PC.

SKRIPSI

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

Disusun oleh :

RIF'AN RACHMANI

NIM : 97.17.005

Mengetahui	Diperiksa dan Disetujui
Ketua Jurusan T. Elektro S-1	**Dosen Pembimbing**
Ir. F. Yudi Limpraptono, MT	**Ir. Yusuf Ismail Nakhoda, MT**
NIP.Y. 103 9500 247	**NIP.Y. 101 8800 189**

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

2010

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

BERITA ACARA UJIAN SKRIPSI

FAKULTAS TEKNOLOGI INDUSTRI

Nama	: Rif'an Rachmani
NIM	: 97.17.005
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik Elektronika
Judul Skripsi	: **Perencanaan dan Pembuatan Alat Untuk Mendeteksi Volume Minyak Pada Tandon SPBU Yang Di Transmisikan Via Modem Berbasis Mikrokontroller AT89S52 Dan Terintegrasi Dengan PC**

Dipertahankan dihadapan Tim Penguji Skripsi Jenjang Program Strata Satu (S-1) pada :

Hari	: Selasa
Tanggal	: 09 Februari 2010
Dengan Nilai	: 86,25 (A)

Panitia Ujian Skripsi

Ketua Majelis Penguji

(Ir. H. Sidik Noertjahjono, MT.)
NIP.Y. 1028700163

Sekretaris Majelis Penguji

(Ir. F. Yudi Limpraptono, MT.)
NIP.Y. 1039500247

Anggota Penguji

Penguji I

(Ir. Eko Nurcahyo)
NIP.Y. 1028700172

Penguji II

(M. Ibrahim Ashari, ST. MT.)
NIP.P.1030100358

ABSTRAKSI

PERENCANAAN DAN PEMBUATAN ALAT UNTUK MENDETEKSI VOLUME MINYAK PADA TANDON SPBU YANG DI TRANSMISIKAN VIA MODEM BERBASIS MIKROKONTROLLER AT89S52 DAN TERINTEGRASI DENGAN PC.

(Rif'an Rachmani, 917005, Teknik Elektro/Elektronika S1, 97 Halaman)
(Dosen Pembimbing : Ir. Yusuf Ismail Nahkoda, MT)

Salah satu penunjang dalam kehidupan manusia adalah media transportasi, dimana media transportasi dapat mendistribusikan hampir seluruh kebutuhan dasar hidup manusia. Salah satu penunjang agar media transportasi dapat berfungsi adalah bahan bakar minyak.

Oleh sebab itu, dalam Tugas Akhir ini dibahas sebuah perencanaan dan pembuatan suatu sistem pemesanan otomatis jenis bahan bakar bensin (premium) dan solar oleh SPBU (Stasiun Pengisian Bahan Bakar Umum) kepada pihak distributor. Sistem otomatis yang digunakan adalah menggunakan mikrokontroller pada SPBU dan PC pada distributor, mikrokontroller yang digunakan berjenis AT89S52, sedangkan pada PC menggunakan program dengan bahasa pemograman Delphi, dimana data yang dikirimkan oleh mikrokontroller ditransmisikan kedalam jaringan telepon melalui modem analog eksternal.

Pada tahap pengujian, dilakukan dua (2) macam metode penerimaan data, yaitu dengan menggunakan hyperteminal serta program yang dibuat. Dimana pada hasil pengujian, terjadi kesalahan penerimaan data sebesar 11,67%.

Kata Kunci : Cara pemesanan otomatis jenis bahan bakar bensin (premium) dan solar pada SPBU via modem berbasis mikrokontroller dan terintegrasi dengan PC

One of proponent in the human life is transportation media, where it can distribute for almost all basic human life's need. Oil fuel is a kind of proponent for transportation media purpose.

Therefore, this thesis examine a planning and making an automatic ordering system with gasoline fuel (premium) and diesel fuel types by SPBU (Public Gas Station) to distributor party. This automatic system uses the microcontroller at SPBU and PC at distributor, which are used microcontroller AT89S52 type, whereas on the PC using a program with Delphi programming language, where the data sent by the microcontroller is transmitted into the telephone network via an external analog modem.

In the test phase, conducted two (2) different methods of data reception, using hyperteminal and programs created. In the test results, an error occurred receiving data for 11.67%.

Keyword: Way of automatic ordering with gasoline fuel (premium) and diesel fuel types at SPBU via modem base on the microcontroller and integrated by PC

KATA PENGANTAR

Puji syukur penulis panjatkan ke Hadirat Tuhan Yang Maha Esa, karena dengan rahmat dan hidayat-Nya penulis dapat menyelesaikan Laporan Skripsi dengan judul **"PERENCANAAN DAN PEMBUATAN ALAT UNTUK MENDETEKSI VOLUME MINYAK PADA TANDON SPBU YANG DI TRANSMISIKAN VIA MODEM BERBASIS MIKROKONTROLLER AT89S52 DAN TERINTEGRASI DENGAN PC".**

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

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. H. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1
4. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Dosen Pembimbing.
5. Kedua orang tua, yang telah memberikan do'a restu serta dukungan moril dan materi.
6. Semua pihak yang telah membantu dalam hal pengetikan dan pembuatan alat, baik hardware maupun software.

Penulis telah berusaha dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan skripsi ini. Untuk itu penulis mengharapkan saran-saran yang sekiranya dapat membantu kesempurnaan laporan skripsi ini.

Akhir kata, semoga laporan skripsi ini dapat bermanfaat bagi semua pihak.

Malang, Januari 2010

Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dengan semakin berkembangnya teknologi, mengakibatkan semakin meningkatnya mobilitas manusia. Maka disini dituntut untuk mampu menyediakan suatu fasilitas transportasi yang memadai dan mampu menjangkau seluruh lapisan masyarakat. Untuk itu diperlukan suatu jasa transportasi yang cukup besar.

Semakin bertumbuhnya jumlah pengguna jasa transportasi yang diakibatkan juga karena pertumbuhan penduduk dan berbagai aktifitas manusia. Oleh sebab itu diperlukan sumber daya pendukung yang mampu untuk mendukung masalah – masalah diatas.

Sumber daya pendukung itu adalah adanya minyak bumi yang diperlukan sebagai bahan bakar untuk semua jenis kendaraan yang umum dipakai. Dimana hasil olahan minyak bumi khususnya bensin atau solar dijadikan sebagai bahan bakar segala jenis alat transportasi. Dan kita juga mengetahui bahwa sumber minyak ini adalah sumber daya alam yang suatu saat dapat habis.

Menghadapi kenyataan – kenyataan diatas maka kita dituntut untuk mampu menggunakan sumber daya minyak secara effisien. Salah satu effisiensi ini dapat dilakukan pada system distribusi minyak tersebut, yaitu tepatnya pada Stasiun Pengisian Bahan Bakar Umum (SPBU).

Masalah yang sering dihadapi dalam pendistribusian minyak ke SPBU adalah adanya kasus – kasus keterlambatan dalam pengiriman bahan bakar dari distributor ke SPBU dikarenakan kurang cepatnya permintaan dari SPBU sehingga informasi menjadi terhalang.

Berdasarkan hal tersebut maka perlu dipasang suatu sistem otomatisasi untuk memberitahu pihak distributor bahwa volume pada tandon SPBU telah menipis yang kemudian mengirimkan informasi dari SPBU kepada pihak distributor melalui kabel telepon via modem sehingga dapat mengeffisiensikan waktu dan tenaga, serta pelanyanan dapat memuaskan.

1.2. Rumusan Masalah

Berdasarkan permasalah tersebut diatas, maka bagaimana cara membuat alat yang dapat memberikan fasilitas untuk memberitahukan kepada distributor bahan bakar bahwasanya terjadi kekosongan stok bahan bakar di tandon SPBU tertentu melalui modem dan sistem tersebut terintregrasi dengan PC *(Personal Computer).*

1.3. Batasan Masalah

Untuk memberikan pembahasan yang jelas maka diberikan ruang lingkup pembatasana masalah sebagai berikut :

- Perencanaan perangkat keras (*hardware*) meliputi Sensor Level Ketinggian Minyak, ADC0804, IC Mikrokontroler AT89S52, Modem Eksternal serta PC.

- Perencanaan dan pembuatan perangkat lunak (*software*) untuk IC *Microcontroller* AT89S52 dengan bahasa Assembler dan untuk PC menggunakan bahasa Delphi.
- Dalam rangkaian diangggap catu daya konstan dan tidak akan dibahas.
- Pengiriman informasi hanya dilakukan saat kondisi tandon dalam keadaan habis atau satu (1) kondisi keadaan.
- Jalur telekomunikasi yang digunakan dianggap normal (konstan) dan tidak akan dibahas.
- Jumlah dan kapasitas tandon pada tiap SPBU diasumsikan sama.

1.4. Tujuan

Tujuan dari perancangan dan pembuatan alat untuk mendeteksi volume minyak pada tendon SPBU yang ditransmisikan via modem berbasis Mikrokontroller AT89S52 dan terintegrasi dengan PC ini adalah untuk memberikan effisiansi kerja pihak pengelola SPBU dan distributor minyak khususnya untuk pihak Pertamina.

1.5. Metodologi

Untuk mencapai tujuan dari tugas akhir (Skripsi) ini, maka metodologi dalam tugas akhir ini adalah sebagai berikut :

a. Studi literatur
b. Studi lapangan
c. Perencanaan dan pembuatan perangkat keras (*hardware*) dan perangkat lunak (*software*).
d. Pengujian dan analisa alat yang dibuat

e. Penyusunan laporan tugas akhir.

1.6. Sistematika Pembahasan

Dalam penyusunan tugas akhir ini, sistematika pembahasan yang digunakan adalah sebagai beriut :

BAB I PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi dan sistematika pembahasan.

BAB II LANDASAN TEORI

Bab ini membahas mengenai dasar-dasar teori sebagai teori penunjang pada alat yang akan dirancang. Disamping itu juga sebagai pelengkap pemahaman mengenai bagian-bagian atau keseluruhan sistem.

BAB III PERENCANAAN DAN PEMBUATAN ALAT

Bab ini membahas menganai perencanaan dan pembuatan alat yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*)

BAB IV ANALISIS DAN PENGUJIAN ALAT

Bab ini mengulas tentang hasil pengujian dan analisa alat yang telah dibuat.

BAB V PENUTUP

Bab ini berisikan kesimpulan akhir dari keseluruhan alat yang telah dibuat dan saran dari penyusun demi perkembangan selanjutnya dari alat yang telah dibuat.

BAB II

LANDASAN TEORI

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. Transducer

Salah satu elemen penting pada sistem instrumentasi dan pengendali adalah transducer. Transducer adalah elemen yang akan mengubah energi nonlistrik menjadi energi listrik sehingga proses kontrol dapat dilakukan baik secara otomatik maupun manual.

2.2.1. Transducer sebagai Elemen Sistem Instrumentasi dan Pengendali Elektronik

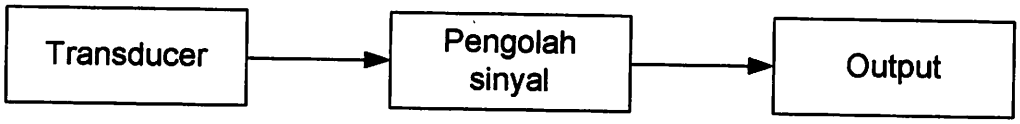
Sistem instrumentasi dan pengendali elektronik pada dasarnya terdiri dari 3 bagian pokok yaitu bagian masukan (*input*), pengkondisi sinyal (*signal conditioning*), dan keluaran (*output*). Ketiganya dapat bersama-sama melakukan proses pengendalian dan menampilkan hasilnya, baik digital maupun analog.

Bagian input mengambil besaran atau parameter nonlistrik yang terukur dan mengirimkan dalam bentuk besaran listrik yang sesuai ke bagian pengkondisi sinyal. Bagian pengkondisi sinyal melakukan proses mengkondisikan sinyal masukan ke dalam suatu format tertentu. Selanjutnya, sinyal yang sudah terkondisi dapat ditampilkan melalui bagian keluaran atau untuk melakukan proses gerak mekanik

sebuah mesin. Bagian output berfungsi untuk menampilkan sinyal keluaran baik berupa tampilan analog maupun digital.

Pada umumnya input pada sistem instrumentasi berupa besaran nonlistrik yang harus diubah ke dalam besaran listrik dengan menggunakan transducer. Oleh karena itu, sinyal ini dapat dikondisikan melalui proses elektronik. Dengan memahami fungsi kerja transducer tersebut maka transducer dapat didefinisikan sebagai piranti yang mengubah suatu bentuk energi ke bentuk energi yang lain.

Gambar 2.1 berikut menunjukkan blok diagram system pengendali atau instrumentasi elektronika.



Gambar 2.1 Blok Diagram Sistem Pengendali atau instrumentasi elektronika[2]

2.2.2. Klasifikasi Transducer

Pengklasifikasian transducer berikut ini berdasarkan pada prinsip kelistrikannya.

1. Transducer Pasif

Transducer ini tidak dapat menghasilkan tegangan sendiri tetapi dapat menghasilkan perubahan nilai resistansi, kapasitansi, atau induktansi apabila mengalami perubahan kondisi sekeliling.

Jika transducer ini mengalami perubahan kondisi pada lingkungan sekelilingnya maka nilai resistansinya akan berubah. Perubahan ini selanjutnya menyebabkan perubahan besar tegangan atau kuat arus yang dihasilkan transducer. Perubahan tegangan inilah yang dimanfaatkan untuk mengetahui keadaan yang ingin diukur.

a. Transducer Resistif

Pada tabel 2.1 menunjukkan jenis serta prinsip kerja pada transduser resistif.

Tabel 2.1 Jenis dan Prinsip Kerja Transducer Resistif[2]

Jenis Transducer	Prinsip Kerja	Penerapan
Potensiometer Resistif	Perubahan karena gerakan eksternal menjadi perubahan resistansi potensiometer atau rangkaian jembatan	Sensor tekanan, posisi
Strain Gage	Tekanan eksternal mengubah resistansi penghantar	Sensor berat, tekanan, posisi
RTD (*Resistance Temperatur Detector*)	Perubahan suhu mempengaruhi resistansi logam murni yang mempunyai koefisien suhu positif	Sensor suhu
Thermistor	Perubahan suhu mempengaruhi resistansi logam teroksidasi yang mempunyai koefisien suhu negative	Sensor suhu
Hygrometer Resistif	Resistasni elektrode turun bila kelembaban udara di sekelilingnya	Kelembaban

	naik atau bertambah	
Psychrometer	Perbedaan suhu pada elektrode kering dan elektrode basah menghasilkan perubahan tegangan	Kelembaban

b. Transducer Kapasitif dan Transducer Induktif

Prinsip kerja transducer ini mengubah perubahan besaran nonlistrik menjadi perubahan nilai kapasitansi atau nilai induktif. Pada tabel 2.2 berikut menunjukkan jenis dan prinsip kerja dari transduser kapasitif dan induktif.

Tabel 2.2 Jenis dan Prinsip Kerja Transducer Kapasitif dan Transducer Induktif[2]

Jenis Transducer	**Prinsip Kerja**	**Penerapan**
Transducer Kapasitif	Kapasitas antara dua dielektrik berubah karena kondisi fisik	Sensor tinggi cairan, sensor tekanan
Transducer Induktif LVDT (*Linear Variable Differential Transfomer*)	Perubahan posisi inti menyebabkan timbulnya tegangan pada kumparan sekunder	Sensor tekanan, posisi
Transducer tekanan	Perubahan tekanan fisis	Sensor tekanan

c. Transducer Photo

Transducer photo dapat mengubah besar arus listrik jika dikenai cahaya/sinar. Arus listrik inilah yang dimanfaatkan untuk mengetahui keadaan

yang ingin diukur. Pada tabel 2.3 berikut menunjukkan jenis serta prinsip kerja dari transduser photo.

Tabel 2.3 Jenis dan Prinsip Kerja Transducer Photo[2]

Jenis Transducer	Prinsip Kerja	Penerapan
Photoconductive	Kondiktivitas pada suatu bahan berubah bila terkena cahaya	
Photodiode	Arus reverse berubah sesuai intensitas cahaya pada diode tersebut	Saklar cahaya, sensor cahaya
Phototransistor	Intensitas cahaya yang jatuh pada transistor photo menyebabkan transistor dalam kondisi cut off atau saturasi	Saklar cahaya
Optocoupler	Mengubah pulsa menjadi sinar infra merah yang kemudian mentriger detector photo	Relay, saklar cahaya

2. Transducer Aktif

Transducer ini dapat menghasilkan energi listrik. Pada tabel 2.4 berikut berisi tentang jenis dan prinsip kerja transduser aktif

Tabel 2.4 Jenis dan Prinsip Kerja Transducer Aktif[2]

Jenis Transducer	Prinsip Kerja	Penerapan
Thermocouple dan Thermophile	Energi listrik muncul bila sambungan dua jenis semikonduktor logam yang berbeda dikenai panas	Sensor suhu, pancaran panas
Cell Photovoltaic	Energi listrik atau tegangan muncul bila sebuah hubungan semikonduktor mendapat pancaran sinar	Sensor cahaya, pembangkit tegangan energi sinar (*solar cell*)

Sensor yang digunakan untuk mengukur ketinggian minyak adalah Potensiometer. Untuk mengkonversi gerakan naik turun ketinggian minyak menjadi gerakan naik turun pada potensiometer, digunakan sebuah pelampung yang dirangkai dalam sebuah sistem lengan ayun (*Swing Arm*).

2.3. ADC 0809 (Analog to Digital Converter)

ADC adalah suatu alat yang digunakan untuk mengubah sinyal listrik analog menjadi diskrit yang diwakili oleh susunan bit-bit dengan kombinasi tertetu. Komponen mi bertugas unutk membantu komputer dalam penganbilan data analog, karena komputer bekerja dalam domain digital, yang hanya mampu membaca sinyal diskrit saja, sedangkan banyak sistem yang ada di luar komputer rnenggunakan sistem analog, sehingga sinyal analog harus diubah terlebih dahulu kedalam bentuk digital.

Salah satu contoh ADC adalah IC 0809, yaitu suatu rangkaian terintegrasi yang terdiri dan komponen penerirna data yang berupa CMOS terintegrasi,

multiplekser 8 ke 1 saluran dan kontrol logika. IC ini merupakan mikroprosesor yang mampu mengubah sinyal analog ke digital dengan metode berurutan (*Successive Aproximation*).

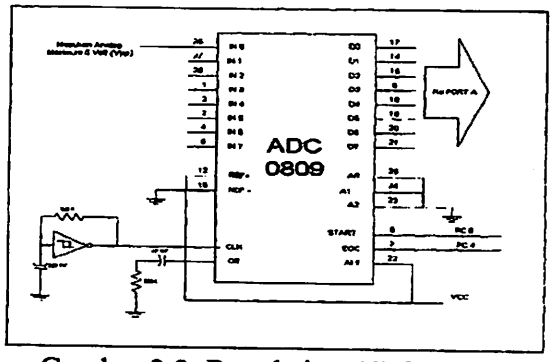
Dengan interface yang dilengkapi ADC mi dapat dibuat antara lain peralatan peralatan pengukuran besaran analog yaitu:

1. Nilai Resistor
2. RIFE Transistor
3. Besarnya Suhu
4. Kuat cahaya
5. Perekam. suara, dan lain-lain.

Spesifikasi ADC 0809 ini adalah:

1. Terdapat 8 masukan yang dapat diprogram
2. 8 Bit data keluaran
3. Tegangan yang dapat diukur maximal beramplitudo 5 V.

Pada gambar rangkaian 2.2 di bawah ini saat mulai konversi dikendalikan melalui port Co dan tanda akhir konver diterima oleh port C4. Sedangkan data konversi dihubungkan dengan Port A atau bisa juga Port A[13].



Gambar 2.2 Rangkaian ADC 0809

2.4. *Microcontoller* AT89S52

Microcontroller adalah pengembangan dari mikroprosesor yang sudah didesain dengan fungsi-fungsi khusus sehingga lebih mudah untuk diaplikasikan pada suatu sistem. *Microcontroller* bekerja sesuai dengan instruksi-instruksi yang diisikan ke dalamnya.

Sebuah *microcontroller* mempunyai komponen seperti halnya mikroprosesor, yaitu *Central Processing Unit (CPU), Arithmetic and Logic Unit* (ALU), *Program Counter (PC), Stack Pointer* (SP), dan register namun *microcontroller* juga memiliki komponen tambahan sejumlah *Random Access Memori* (RAM), *Read Only Memori* (ROM), dan *InputlOutput Port* (1/O port), dan *timer/counter* yang semuanya terdapat dalam satu keping *Integrated Circuit* (IC). Dengan kata lain, *microcontroller* memiliki prosesor, RAM, ROM, 1/O port, dan *timer/counter* yang terintegrasi menjadi satu IC.

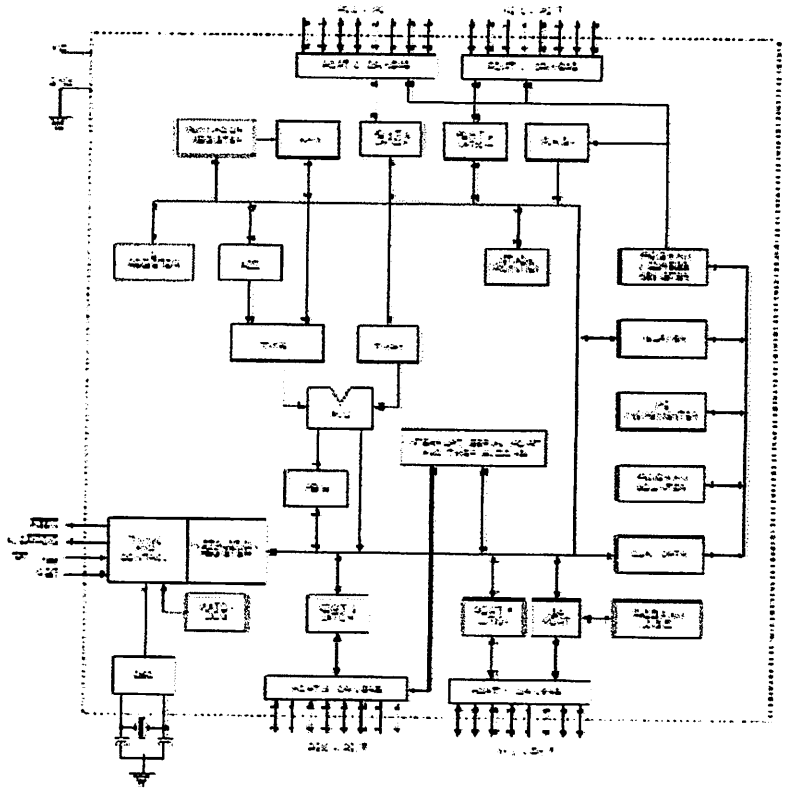
Fungsi masing-masing bagian pada *microcontroller* adalah:

- ALU : berfungsi sebagai pusat pengolahan proses perhitungan dan proses logika.
- *Accumulator* : berfungsi sebagai penampung nilai *variable* yang akan diproses oleh ALU sekaligus menyimpan hasil dari proses.
- *Working Register* : digunakaan untuk membantu *accumulator* menyimpan data selama proses berlangsung.
- PC : digunakan untuk menyimpan alamat program yang akan dieksekusi oleh CPU.
- SP : digunakan untuk menyimpan data dalam bentuk tumpukan *(stack)*. Register ini biasanya digunakan

untuk menyimpan. alamat program selama proses pemanggilan *subrutin* dan *interrupt.*

- RAM internal : digunakan untuk menyimpan data-data yang akan dimanipulasi oleh CPU.
- ROM internal : digunakan untuk menyimpan semua program yang akan dijalankan oleh *microcontroller.* Bagian ini berupa Flash ROM dimana data yang disimpan di dalamnya bersifat *non volatile* (isinya tidak hilang saat catu daya mati).
- *TimerlCounter* : digunakan untuk melakukan proses pewaktuan dan proses perhitungan denyut sinyal masukan.
- 1/O Port : digunakan untuk menyimpan dan menguatkan sinyal, baik yang akan masuk maupun yang akan keluar.
- *Interrupt Circuit* : digunakan untuk menangani semua proses *interrupt* yang masuk ke dalam CPU.
- *Clock Circuit* : *Clock Circuit* merupakan jantung dari *microcontroller,* Bagian ini akan memadukan semua proses yang berlangsung di dalam *microcontroller.*

Berikut ini adalah gambar blok diagram dari mikrokontroller AT89S52.



Gambar 2.3 Blok diagram AT89S52[5]

Microcontroller yang dipakai dalam proyek kali ini adalah AT89S52. AT89S52 termasuk salah satu jenis IC dari keluarga MCS-51 yang dikemas dalam standart *Dual IN Line Package* (DIL) 40 pin yang mempunyai susunan sendiri. *Microcontroller* ini diproduksi oleh ATMEL dengan karakteristik yang cocok dengan kumpulan intruksi dan pin standar keluarga MSC-52 buatan INTEL.

AT89S52 merupakan versi *electrically erasable and programable read only memory* (EEPROM) dari 8051AH. Memori program internalnya dapat diprogram dan dihapus, secara elektrik dengan teknologi *flash* EEPROM yang dapat menyimpan data meskipun catu daya pada IC dimatikan. Selain itu *microcontroller* jenis S atau ISP memiliki *built-in* rangkaian *downloader* yang memudahkan kita dalam memasukan kode-kode dalarn *microcontroller* ini. Data dikirim secara serial ke dalam *microcontroller* melalui pin-pin *Master In Slave Out* (MISO), *Master Out Slave In*

(MOSI) dengan sebuah sinyal kendali *Serial* Clock (SCK). AT89S52 memiliki ROM internal namun juga, dapat menggunakan EPROM ekstemal yang dihubungkan dengan *microcontroller* melalui port paralel (port 0 dan port 2).

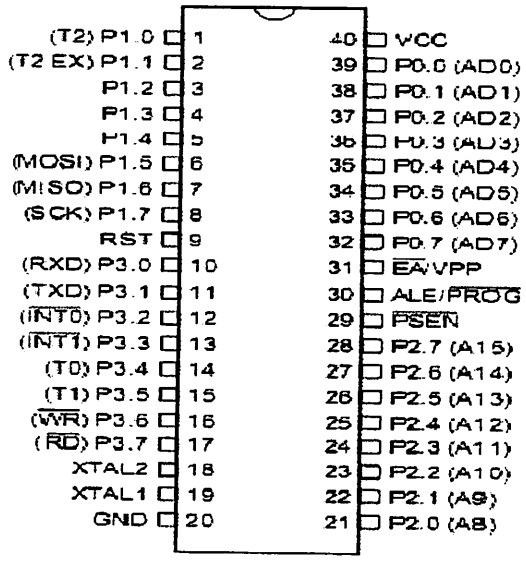
Dengan demikian, *microcontroller* memiliki kemampuan yang dapat lebih diandalkan sebagai sistem kontrol dibandingkan dengan mikroprosessor.

AT89S52 memiliki karakteristik sebagai berikut[5]:

- *In-System Programmable (ISP) Flash Memory* sebesar 4 kByte. Dengan menggunakan *flash chip* ini, mengijinkan program memori dapat diprogram ulang dalam sistem. Sehingga, *microcontroller* ini tidak membutuhkan *microcontroller* lain sebagai master untuk proses *download* program.
- Dapat diprogram dan dihapus berulang-ulang hingga +1000 kali.
- Memiliki 3 (tiga) level penguncian memori program
- Memiliki *Random Access Memory* (RAM) sebesar 128 Byte
- 32 jalur atau bit input dan output yang terbagi menjadi 4, yaitu Port 0, Port 1, Port 2, dan Port 3 yang dapat diprogram
- Memiliki *watchdog timer.*
- Memiliki 2 buah *data pointer.*
- Memiliki 2 buah *timer* dan *counter 16* bit
- Memiliki sebuah *full duplex* UART serial port.

Microcontroller tersebut mempunyai 40 kaki (Gambar 2.4), 32 kaki di antaranya adalah kaki untuk keperluan port paralel. Tiap port paralel terdiri dari 8 kaki, setiap kaki mewakili 1 jalur data paralel, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel, di mana setiap port paralel memiliki 8 jalur data, yang masing-masing dikenal sebagai Port 0, Port 1, Port 2 dan Port 3. Nomor

masing-masing jalur port paralel dimulai dari dari jalur 0 sampai 7, jalur pertama Port 0 disebut sebagai Po.0 dan jalur terakhir untuk Port 3 adalah P3.7[1].



Gambar 2.4 Konfigurasi pin-pin AT89S52[5]

2.4.1. Fungsi-fungsi Kaki (Pin)

- VCC : Sumber tegangan +5V DC. (pin 40)
- GND : Sumber tegangan *Ground* (pin 20)
- RST : Masukan reset. Jika diberikan kondisi '1' selama 2 siklus mesin ketika oscillator berjalan akan mengembalikan kondisi *microcontroller* yang bersangkutan ke kondisi reset. (pin 9)
- ALE/ PROG : Pin ALE atau. *Adreess Latch Enable* menghasilkan pulsa untuk mengunci alamat byte rendah *(low byte)* selama mengakses memori eksternal. Pin ini juga berfungsi sebagai masukan pulsa program atau PROG selama pemrograman *flash.* Pada operasi normal, ALE akan dipancarkan dengan laju 1/6 dari frekuensi kristal dan

dapat digunakan sebagai pewaktuan *(timing)* atau pendetakan *(clocking)* rangkaian eksternal. Catatan, ada satu pulsa yang dilompati selama pengaksesan memori data eksternal. Jika dikehendaki, operasi ALE bisa dimatikan dengan cara mengatur bit 0 dari SFR yang beralamatkan 8Eh. Jika diisi '1', ALE hanya akan aktif selama melakukan instruksi MOVX atau MOVC. Selain itu, kaki ini akan secara lemah *di-pulled high.* Mematikan bit ALE tidak berakibat apa-apa jika *microconfroller* sedang mengeksekusi program secara eksternal. (pin 30).

- PSEN : *Program Store Enable* merupakan sinyal baca pada memori program eksternal untuk mengizinkan program memori eksternal masuk ke dalam jalur bus untuk dieksekusi. Saat *microcontroller* menjalankan program dari memori eksternal, PSEN akan diaktifkan dua kali setiap siklus mesin, kecuali dua aktivasi PSEN dilompati (diabaikan) setiap kali mengakses memori data eksternal. (pin 29)

- *EA /VPP* : *External Access Enable. EA* harus diberi logika '0' jika *microcontroller* akan menjalankan instruksi dari memori program eksternal yang memiliki alamat mulai dari 0000h hingga FFFFh. Sebaliknya, EA bila diberi logika '1', mulai alamat 0000h sampai OFFFh yang dieksekusi adalah memori program internal, dan mulai alamat 1000h hingga FFFFh yang dieksekusi adalah memori program

ekstemal. Kaki ini juga berfungsi menerima tegangan 12 Volt (VPP) selama pemrograman flash. (pin 31)

- XTAL I : Masukan ke penguat osilator. Pin ini dihubungkan dengan kristal atau sumber osilator yang lain. (pin 19)
- XTAL2 : Keluaran dari penguat osilator. Pin ini dihubungkan dengan kristal atau *ground.* (pin 18)

2.4.2. Spesifikasi Masing-masing Port

Keempat port pada *microcontroller* dapat berkomunikasi secara dua arah dan masing-masing memiliki sebuah pengunci *(latch),* yang dalam program disebut dengan Register Fungsi Khusus (RFK atau SFR) sebagai PO, Pl, P2, dan P3. Selain itu masing-masing jalur port juga memiliki sebuah penggerak keluaran *(output driver)* dan sebuah penyangga masukan *(input buffer).*

Output driver dari Port 0 dan Port 2 serta *input buffer* dari Port 0 digunakan untuk mengakses memori eksternal. Pada aplikasi semacam ini, Port 0 mengeluarkan byte alamat memori eksternal rendah, *di-multipleks s*ecara waktu dengan byte yang akan dituliskan atau. dibaca (ke/dari memori eksternal). Port 2 mengeluarkan byte alamat memori eksternal tinggi, jika ukuran alamat yang digunakan adalah 16-bit[1].

A. Port 0

Port 0 terletak pada, pin 39 (P0.0) hingga pin 32 (P0.7).

Port 0 merupakan jalur keluaran maupun masukan (I/O) dua arah bertipe *open drain bidirectional.* Sebagai port keluaran, masing-masing jalur dapat menyerap arus *(sink)* delapan masukan TTL (sekitar 3,8 mA). Pada saat sinyal '1' diberikan ke jalur-jalur Port 0 ini, maka jalur-jalur Port 0 dapat digunakan sebagai masukan berimpedansi tinggi.

Port 0 dapat dikonfigurasikan sebagai jalur bus untuk alamat byte rendah *(low byte)* maupun data selama proses pengaksesan program eksternal dan memori data. Pada saat ini Port 0 memiliki pull-up internal.

Port 0 juga menerima kode-kode yang dikirim kepada *microcontroller* selama proses pemrograman dan mengirimkan kode-kode program yang telah tersimpan dalam flash selama proses verifikasi. Selama proses verifikasi program ini dibutuhkan pull-up eksternal.

B. Port 1

Port 1 terletak pada pin I (P1.0) hingga pin 8 (P1.7).

Port 1 merupakan jalur keluaran maupun masukan (1/0) dua arah yang dilengkapi dengan pull-up internal. Jika sinyal '1' diberikan ke kaki-kaki Port 1, maka masing-masing kaki akan *di-pulled high* dengan pullup internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, pin-pin Port I yang di *pulled low* secara eksternal akan memberikan arus (source) karena di-pulled *high* secara internal.

Port 1 juga menerima alamat byte rendah (*low byte*) selama pemrograman dan verifikasi flash karena beberapa pin Port I juga memiliki fungsi khusus yang terutama digunakan untuk keperluan pemrograman dan verifikasi flash (Tabel 2.5). P1.5 memiliki fungsi khusus sebagai pin Master Out Slave In (MOSI), P1.6 memiliki fungsi khusus sebagai pin Master In Slave Out (MISO), P1.7 memiliki fungsi khusus sebagai pin Serial Clock (SCK).

Tabel 2.5 Fungsi-fungsi khusus pin-pin Port I

Port Pin	Fungsi Khusus
P 1.5	MOSI (*Master Out Slave In*)
P 1.6	MISO (*Master In Slave Out*)
P1.7	SCK (*Serial Clock*)

C. Port 2

Port 2 terletak pada pin 20 (P2.0) hingga pin 28 (P2.7)

Port 2 merupakan jalur keluaran maupun masukan (1/0) dua arah yang dilengkapi dengan *pull-up* internal. Jika sinyal '1' diberikan ke kaki-kaki Port 2, maka masing-masing kaki akan di-pulled *high* dengan pullup internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, pin-pin Port 2 yang di *pulled low* secara eksternal akan memberikan arus (*source*) karena di-pulled *high* secara internal.

Port 2 akan memberikan alamat *byte* tinggi *(high byte*) selama pengambilan instruksi dari memori program eksternal dan selama pengaksesan memori data eksternal yang menggunakan pengalamatan 16-bit. Dalam aplikasi ini, Port 2 menggunakan pull-up internal yang kuat untuk menghasilkan sinyal '1'. Selama pengaksesan memori data eksternal yang menggunakan alamat 8-bit, Port 2 akan mengirimkan isi dari SFR P2.

Port 2 juga menerima alamat byte tinggi dan beberapa sinyal kontrol selama pemrograman dan verifikasi flash.

D. Port 3

Port 3 terletak pada pin 10 (P3. 0) hingga pin 17 (P3.7).

Port 3 merupakan jalur keluaran maupun masukan (I/O) dua arah yang dilengkapi dengan *pull-up* internal. Jika sinyal '1' diberikan ke kaki-kaki Port 3, maka masing-masing kaki akan *di-pulled high* dengan pullup internal sehingga dapat digunakan sebagai masukan. Sebagai masukan, pin-pin Port 3 yang di *pulled low* secara eksternal akan memberikan arus (*source*) karena di-*pulled high* secara internal.

Port 3, juga memiliki fungsi-fungsi khusus antara lain, sinyal kontrol (WR dan RD) dalam mengakses memory eksternal, kemudian beberapa fungsi khusus lainnya (Tabel 2.6). P3.0 memiliki fungsi khusus sebagai pin RXD (serial *input* port),

P3.1 memiliki fungsi khusus sebagai pin TXD (serial *output port),* P3.2 memiliki fungsi khusus sebagai pin INTI *(external interrupt 0),* P3.3 memiliki fungsi khusus sebagai pin INTI *(external interrupt 1),* P3.4 memiliki fungsi khusus sebagai pin TO *(timer 0 external input),* P3.5 memiliki fungsi khusus sebagai pin TI *(timer 1 external input),* P3.6 memiliki fungsi khusus sebagai pin WR *(external data memory* write *strobe),* P3.7 memiliki fungsi khusus sebagai pin RD *(external data memory read strobe.*

Tabel 2.6 Fungsi-fungsi Khusus pin-pin Port 3

Port Pin	Fungsi Khusus
P 3.0	RXD (*serial input port*)
P 3.1	TXD (*serial output port*)
P 3.2	INT0 (*external interrupt* 0)
P 3.3	INT1 (*external interrupt* 1)
P 3.4	T0 (*timer 0 external input*)
P 3.5	T1 (*timer 1 external input*)
P 3.6	WR (*external data memory write strobe*)
P 3.7	RD (*external data memory read strobe*)

2.4.3. Memori Program

Memori program internal AT89S52 berupa EEPROM dengan kapasitas sebesar 4 *kByte.* Padahal AT89S52 memiliki fasilitas pengalamatan hingga 16 bit atau sebesar 64 *kByte,* mulai alamat 0000H sampai dengan FFFFH. Maka dari itu, selain memori program internal, AT89S52 juga bisa dipasangi memori program eksternal.

Untuk mengakses memori program eksternal pin EA dihubungkan dengan *ground.* Enam belas jalur masukan/keluaran (pada Port 0 dan Port 2) difungsikan

sebagai jalur alamat, Port 0 mengeluarkan 8 bit alamat rendah (A7AO) dari PC. Pada saat Port 0 mengeluarkan alamat byte rendah maka sinyal ALE akan menahan alamat Port 2, yang merupakan alamat byte tinggi (A 15-A8). Port 0 dan Port 2 bersama-sama membentuk alamat sebesar 16 bit (A15-A8). Sinyal PSEN digunakan untuk membaca program eksternal.

2.4.4. Memori Data

AT89S52 memiliki 128 *Byte* RAM internal *(on chip* RAM). Pengalamatan RAM ini dapat dilakukan secara *direct* maupun *indirect.* Pada memori internal terdapat lokasi yang mempunyai fungsi khusus yang dapat digambarkan sebagai berikut:

- Alamat OOH - 1FH merupakan kumpulan dari 4 bank register umum yang dapat dipakai dalam program sebagai R0 - R7. Pemilihan bank register umum yang digunakan dapat diatur melalui register PSW pada register fungsi khusus *(Special Function Registers)*
- Alamat 20H - 2FH merupakan bagian internal memori yang dialamati perbit *(Bit Addressable).*
- Sedangkan sisanya mulai alamat 30H - 7FH merupakan memori bebas.

Selain mengakses memori data internal, *AT89S52* juga dapat mengakses memori data eksternal. Lebar jalur alamat yang dapat diakses adalah 16 bit yaitu mulai alamat 0000h sampai FFFFH. Susunan perangkat kerasnya sama dengan memori program eksternal, hanya saja sinyal PSEN tidak digunakan untuk membaca data tetapi yang digunakan adalah sinyal RD, sedangkan untuk menuliskan data ke memori ekstemal digunakan sinyal WR.

2.4.5. Special Function Registers (SFR)

Register fungsi khusus (SFR) merupakan bagian memori yang mempunyai fungsi khusus terhadap MCS-52, baik sebagai pengendali perangkat keras maupun perangkat lunak. Register ini menempati alamat interal memori mulai 80H - FFH. Dari 128 *Byte* lokasi memori tersebut tidak semuanya merupakan register fungsi khusus. Lokasi yang lain dapat kita gunakan sebagai tempat menyimpan data dengan kecepatan proses baca tulis yang lebih cepat dibanding dengan eksternal memori. Register fungsi khusus ini dibagi dalam dua bagian besar yaitu yang dapat diakses per-bit *(bit-addressable)* dan yang tidak dapat diakses per-bit *(not bit-addressable).* Dapat diakses per-bit maksudnya adalah pengaksesan register tersebut dimungkinkan untuk dimanipulasi per-bit. Sedangkan register fungsi khusus yang tidak dapat diakses per-bit harus dimanipulasi datanya secara keseluruhan (per-byte). Pemetaan alamat register fungsi khusus ini dapat dilihat pada Tabel 2.7.

Tabel 2.7 Pemetaan Register fungsi khusus AT89S52[13]

	8 Byte							
0F8H								FFH
0F0H	B							F7H
0E8H								EFH
0E0H	ACC							E7H
0D8H								DFH
0D0H	PSW							D7H
0C8H								CFH
0C0H								C7H
0B8H	IP							BFH
0B0H	P3							B7H

0A8H	IE							AFH
0A0H	P2							A7H
98H	SCON	SBUF						9FH
90H	P1							97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1		8FH
80H	P0	SP	DPL	DPH			PCON	87H

Setiap bit register mernpunyai fungsi tersendiri yang dapat dijelaskan sebagai berikut:

A. Register Program Status Word (PSW)/0D0h

Susunan bit dalam register program status word dapat dilihat pada tabel 2.8 berikut ini.

Tabel 2.8 Susunan bit dalam register PSW[13]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CY	AC	F0	RS1	RS0	OV	-	P

- CY : Carry flag.
- AC : Carry flag untuk operasi BCD (*Auxiliary Carry Flag*).
- F0 : Flag untuk kegunaan umum.
- RS1 : Untuk memilih bank -register umum bit ke 1.
- RSO : Untuk memilih bank register umumn bit ke 0.
- OV : *Overflow flag.*
- - : Tidak digunakan.
- *P* : *Parity flag,* yang proses set-resetnya dilakukan oleh perangkat keras yang menunjukkan data pada akumulator bernilai ganjil atau genap.

Register ini dapat diakses per-bit *(bit-addressable)*. PSW.3 dan PSW.4 atau RS0 dan RS1 merupakan kombinasi bit yang menentukan *bank register* yang dipakai. Kombinasi tersebut dapat dilihat pada Tabel 2.9.

Tabel 2.9 Kombinasi RS0 dan RS1 Sebagai Pemilih *Bank Register*[13]

RS1	RS0	Bank Register	Alamat
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

B. Register Power Control (PCON)/87h

Susunan bit dalam register power control ditunjukkan oleh Tabel 2.10 berikut.

Tabel 2.10 Susunan bit dalam register PCON[13]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	-	-	GF1	GF0	PD	IDL

- SMOD : Pengali dua kecepatan transmisi data serial.
- - : Tidak digunakan
- *GFI* : *Flag* untuk kegunaan umum.
- *GFO* : *Flag* untuk kegunaan umum.
- PD : Untuk mengaktifkan *mode Power Down.*
- IDL : Untuk mengaktifkan *mode Idle.*

Register ini tidak dapat diakses per-bit *(not bit-adrressable)*

C. Register *Timer/Counter Control* (TCON)/88h

Adapun susunan bit dalam register timer/counter control, dijelaskan oleh Tabel 2.11 di bawah ini.

Tabel 2.11 Susunan bit dalam register TCON[13]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
← Timer 1 →		← Timer 0 →		← Timer 1 →		← Timer 0 →	

Keterangan:

- TFI : *Overflow* flag *TimerlCounter* 1 (1 =overflow).
- TR1 : Enable *TimerlCounter 1.*
- TF0 : *Overflow* flag *TimerlCounter* 0 (1 =overflow).
- TR0 : Enable *TimerlCounter 0.*
- IEI : Petunjuk bila tetadi *interrupt* eksternal 1.
- IT1 : *Interrupt I type control* bit. Set/clear oleh program untuk menspesifikasi sisi turun/level rendah trigger dari interupsi eksternal.
- IE0 : Petunjuk bila terjadi *interrupt* eksternal 0.
- IT0 : *Interrupt 0 type control* bit. Set/clear oleh program untuk menspesifikasi sisi turun/level rendah trigger dari interupsi ekstemal.

Register ini bisa diakses secara per-bit (bit-adressable)

D. Register *Timer/Counter Mode Control* (TMOD)/89h

Susunan bit dalam register timer/counter mode control ditunjukkan oleh Tabel 2.12 berikut ini.

Tabel 2.12 Susunan bit dalam register TMOD[13]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
← Timer 1 →				← Timer 0 →			

- GATE : Bit pengatur sinyal clock. Jika GATE = 0, *Timer/Counter* akan berjalan saat bit TRx pada register MON bemilai 1. Jika GATE = l, Timer/Counter akan berjalan saat TRx = l atau INTx = 1.
- C/T : Untuk mengatur sumber sinyal clock yang diberikan kepada Timer/Counter. Jika C/T = 0, maka Timer akan aktif dengan sinyal clock diperoleh dari oscillator kristal yang frekuensinya sudah dibagi 12. Jika C/T=l, maka Counter akan aktif dengan sinyal clock diperoleh dari kaki T0 (untuk Timer 0) dan kaki T1 (untuk Timer 1).
- MO dan MI : Untuk menentukan mode Timer/Counter (Tabel 2.9).

Register ini dapat diakses per-bit (bit-addressable)

Untuk *setting* mode operasi timer/counter, Tabel 2.13 berikut ini akan menjelaskannya.

Tabel 2.13 Mode Operasi *Timer / Counter*

M1	M2	Mode	Operasi
0	0	0	*Timer / Counter* 13 bit
0	1	1	*Timer / Counter* 16 bit
1	0	2	*Timer* 8 bit *auto reload*
1	1	3	TL0 adalah *Timer / Counter* 8 bit yang dikontrol oleh kontrol bit *Timer* 0 (TF0)

			TH0 adalah *Timer / Counter* 8 bit yang dikontrol oleh kontrol bit *Timer* 1 (TF1)

E.Register Serial Port Control (SCON)

Tabel 2.14 berikut ini merupakan susunan bit dalam register serial port control.

Tabel 2.14 Susunan bit dalam register SCON[13]

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

- SMO dan SMI : Untuk menentukan mode kerja port serial (Tabel 2.11)
- SM2 : Untuk komunikasi multi prosesor.
- REN : Untuk mengaktifkan kemampuan port serial menerima data.
- TB8 : Bit ke-8 dari data yang akan dikirim (Mode 2 dan Mode 3).
- RB8 : Bit ke-8 dari data yang telah diterima. (Mode 2 dan Mode 3).
- TI : Bernilai 1 jika data telah dikirim.
- RI : Bernilai 1 jika data telah diterima.

Register ini dapat diakses per-bit *(bit-addressable)*.

Untuk *setting* mode port serial dapat dilihat pada Tabel 2.15 berikut ini.

Tabel 2.15 Mode Port Serial[13]

M1	M2	Mode	Keterangan	Baudrate
0	0	0	*Shift Register*	Tetap
0	1	1	UART 8 bit	Dapat diubah-ubah (dengan *Timer*)
1	0	2	UART 9 bit	*Timer* 8 bit *auto reload*
1	1	3	UART 9 bit	Dapat diubah-ubah (dengan *Timer*)

F. Register *Interrupt Enable* (IE)

Susunan bit dalam register *interrupt enable* ditunjukkan Tabel 2.16 berikut:

Tabel 2.16 Susunan bit dalam register IE[13]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EA	-	-	ES	ET1	EX1	ET0	EX0

Keterangan:

- EA : Enable semua interupsi (0 = non aktif, 1 = aktif).
- ES : Enable interupsi serial (0 = non aktif, 1 = aktif).
- ETI : Enable Timer I overflow (0 = non aktif, 1 = aktif).
- EXI : Enable eksternal interupsi 1 (0 = non aktif, 1 = aktif).
- ET0 : Enable Timer 0 overflow (0 =non aktif, 1 = aktif).
- EX0 : Enable ekstemal interupsi 0 (0 = non aktif, 1 =aktif).

Register ini dapat diakses per-bit *(bit-addressable)*.

G. Register *Interrupt Priority* (IP)

Susunan bit dalam register *interrupt priority* ditunjukkan oleh Tabel 2.17.

Tabel 2.17 Susunan bit dalam register IP[13]

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

Keterangan:

- PS : prioritas interupsi Serial (0 = prioritas rendah, 1 = prioritas tinggi).
- PT 1 : prioritas interupsi Timer 1 (0 = prioritas rendah, 1 = prioritas tinggi).
- PX1 : prioritas interupsi INT1 (0 = prioritas rendah, 1 = prioritas tinggi).
- PT0 : prioritas interupsi Timer 0 (0 = prioritas rendah, 1 = prioritas tinggi).
- PX0 : prioritas interupsi INT0 (0 = prioritas rendah, 1 = prioritas tinggi).

Register ini dapat diakses per-bit *(bit-addressable)*.

2.4.6. Reset

Reset dapat dilakukan secara manual maupun. otomatis saat catu daya dinyalakan. Saat terjadi reset isi register akan berubah seperti pada Tabel 2.18. Nilai PC menjadi 00h, nilai Port 0 sampai Port 3 menjadi 0FFh, nilai register Aux menjadi XXX00XX0b, nilai register SP menjadi 07h, nilai register PCON menjadi 0XXX0000b, dan nilai register-register lainnya menjadi 00h.

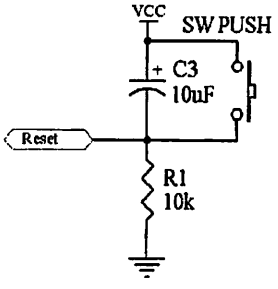
Reset terjadi dengan adanya logika '1' selama minimal 2 *cycle* pada kaki RST. Setelah kondisi pin RST kembali low, *microcontroller* akan mulai menjalankan program dari alamat 0000H.

Tabel 2.18 Nilai register *Microcontoller* saat reset[13]

Register	Nilai dalam kondisi reset
Program Counter (PC)	0000h
Aux	XXX00XX0b
Register B	00h
Accumulator (Acc)	00h
PSW	00h
Port 0 – Port 3	0FFh
IP	00h
IE	00h
SCON	00h
SBUF	00h
TCON	00h
TMOD	00h
THxTLx	00h
Stock Pointer (SP)	07h

DPTR	0000h
PCON	0XXX0000b

Gambar berikut ini menunjukkan skema rangkaian reset.



Gambar 2.5 Skema rangkaian reset

Gambar 2.5 merupakan gambar rangkaian reset yang bekerja secara otomatis saat catu daya diaktifkan. Saat catu daya diaktifkan, kapasitor CI, sesuai dengan sifat kapasitor, akan terhubung singkat. Arus mengalir dari VCC langsung ke pin RST sehingga pin tersebut berlogika '1'. Kemudian kapasitor terisi hingga tegangan pada kapasitor (VC), yaitu tegangan antara VCC dan kutub negatif kapasitor Cl dengan kutub positif kapasitor Cl dan resistor R2, mencapai VCC. Pada saat ini, otomatis tegangan yang masuk ke pin RST akan turun menjadi 0 sehingga pin RST akan berlogika '0' dan proses reset selesai.

Jika saklar S1 ditekan, reset kembali bekerja secara manual, aliran arus akan mengalir dari VCC melalui R1 menuju pin RST. Tegangan pada pin RST atau VR2 akan berubah menjadi:

$$VR2 = \frac{R2 \times VCC}{R1 + R2} = \frac{10k\Omega x 5V}{100\Omega + 10k\Omega} = 4{,}95V$$

Tegangan 4,95 volt pada pin RST menyebabkan pin ini berlogika '1' pada saat saklar tersebut ditekan. Saat saklar dilepas, aliran arus dari VCC melalui RI akan

terhenti, dan tegangan pada pin RST akan menurun menuju nol, sehingga logika pada pin ini menjadi '0'. Proses reset selesai.

2.4.7. Timer/Counter

AT89S52 dilengkapi dengan dua perangkat *timer/counter,* masing- masing dinamakan sebagai *timer/counter 0* dan *timer/counter* 1.

Untuk mengakses nilai *timer/counter* tersebut digunakan register khusus yang tersimpan dalam SFR. Nilai *timer/counter 0* diakses melalui register TL0 *(Timer 0 Low Byte,* memori internal alamat 8Ah) dan register TH0 *(Timer 0 High Byte,* memori internal alamat 8Ch). Nilai *timer/counter* I diakses melalui register TLI *(Timer I Low Byte,* memori internal alamat 8Bh) dan register THI *(Timer I High Byte,* memori internal alamat 8Dh).

Register THxTLx pada AT89S52 merupakan pencacah biner 16 bit naik *(count up binary counter)* yang mencacah dari 0000h sampai FFFFh, saat nilai register berubah dari FFFFh menjadi 0000h (overvlow), *microcontroller* akan menghasilkan sinyal *overflow.*

Untuk mengatur kerja *timer/counter* tersebut digunakan 2 register tambahan, yaitu register TCON (Tabel 2.11) dan register TMOD (Tabel 2.12)[1].

Rumus yang digunakan untuk menghitung periode osilasi atau yang biasa disebut siklus mesin *(machine cycle)* adalah:

$$T = \frac{1}{f}$$

Sehingga, dengan menggunakan Mode I dapat dihitung waktu tunda yang diperlukan dengan perhitungan berikut:

$$Delay = (65536 - n) \times T$$

Dengan *n* adalah nilai desimal dari nilai heksadesimal THxTLx.

2.4.8. Interupsi

Interupsi adalah suatu keadaan eksternal atau internal yang mengakibatkan *microcontroller* menunda proses yang sedang berjalan untuk menjalankan proses lain berdasarkan keadaan yang diberikan. *Microcontroller* AT89S52 menyediakan enam sumber interupsi, yaitu dua interupsi eksternal, dua interupsi timer, satu interupsi serial, dan satu lagi interupsi khusus, yaitu pada saat kondisi reset. Masing-masing sumber interupsi tersebut dapat diaktifkan dan dinonaktifkan sendiri-sendiri dengan mengatur bit-bit yang terkait dalam register IE *(Interrupt Enable)* di alamat memori internal A8h. Pengaturan bit pada register IE dapat dilihat di Tabel 2.16.

Program yang tergabung dalam sistem interupsi disebut *Interrupt Service Routine (ISR)* atau *interrupt handler.* Jika terjadi interupsi, *microcontroller* akan mengeksekusi ISR. Setiap program interupsi berada pada lokasi memori internal tertentu dan tidak berpindah-pindah. Tabel 2.19 menunjukkan alamat ISR pada memori internal *microcontroller.*

Tabel 2.19 Alamat ISR[13]

No	Nama	Alamat RAM	Media Interupsi
1.	Reset	0000h	*Power on Reset* (pin 9)
2.	INT0	0003h	Interupsi 0 *hardware* eksternal (pin 12)
3.	Timer 0	000Bh	*Overflow Timer* 0 (TF0)
4.	INT1	0013h	Interupsi 1 *hardware* eksternal (pin 13)
5.	Timer 1	001Bh	*Overflow Timer* 1 (TF1)
6.	Serial	0023h	Komunikasi Serial (RI/TI)

Secara normal, apabila dalam suatu proses yang sedang berjalan terdapat beberapa interupsi yang datang bersamaan, maka urutan prioritas interupsi yang

dikerjakan sesuai dengan urutan pada Tabel 2.15. *Interrupt* reset memiliki alamat 0000h yang terjadi setiap kondisi reset (pin 9), INTO memiliki alamat 0003h yang terjadi bila ada sinyal '0' di pin 12, timerO memiliki alamat 000Bh yang terjadi bila ada sinyal '1' di pin 14, INT1 memiliki alamat 0013h yang terjadi bila ada sinyal '0' di pin 13, timerl memiliki alamat 001Bh yang terjadi bila ada sinyal '1' di pin 15, dan serial memiliki alamat 0023h yang terjadi bila register bit RI atau TI bernilai '1'. Jika ingin mengatur prioritas dari suatu interupsi, dapat diatur dalam register *Interrupt Priority* (IP), dengan alamat memori internal B8h, yang konfigurasinya dapat dilihat pada Tabel 2.17.

2.4.9. Port Serial

AT89S52 mempunyai 1 pasang jalur komunikasi serial yang terdiri dari pin TXD, yang digunakan untuk mengirim data secara serial, dan pin RXD, yang digunakan untuk menerima data secara serial. Serial port pada AT89S52 mempunyai sistem transmisi data serial dengan tipe Full Duplex yaitu dapat mengirim dan menerima data pada waktu yang bersamaan.

Proses penerimaan dan pengiriman data secara serial melibatkan register SBUF (99h). Pada saat data akan dikirimkan, data diletakkan di SBUF. Begitu pula bila ada penerimaan data, data tersebut juga akan diletakkan di register SBUF.

Serial port pada AT89S52 mempunyai 4 mode operasi, yaitu mode 0, mode 1, mode 2, dan mode 3.

A. Mode 0

Mode ini bekerja secara sinkron. Pada mode 0 data yang dikirim mempunyai lebar 8 bit data. Data serial dikirim dan diterima melalui kaki P3.0 (Rx), sedangkan kaki P3.1 (Tx) digunakan untuk membangkitkan sinyal clock untuk sinkronisasi data

serial. I paket data berukuran 8 bit, dimulai dari bit ke-0 sampai dengan bit ke-7. Kecepatan pengiriman data *(baudrate)* adalah $\frac{1}{12}$ frekuensi kristal yang digunakan.

B. Mode I

Mode ini menggunakan komunikasi serial asinkron. Data dikirim melalui pin Tx dan diterima di pin Rx. 1 paket data memiliki ukuran 10 bit, diawali dengan 1 bit start, 8 bit data mulai dari bit ke-0 sampai dengan bit ke-7, dan 1 bit stop. Pada AT89S52, yang berfungsi sebagai penerima bit stop adalah bit RB8 pada register SCON. *Baudrate* dapat diatur sesuai keperluan. Mode 1, Mode 2, dan Mode 3 dikenal dengan *Universal Asynchronous Receiver/Transmitter* (UART).

C. Mode 2

Sama halnya seperti pada mode 1, mode ini juga menggunakan komunikasi serial asinkron. 1 paket data berukuran 11 bit, diawali dengan 1 bit start, 8 bit data mulai dari bit ke-0 sampai dengan bit ke-7, bit ke-8, dan 1 bit stop.

Pada AT89S52 yang berfungsi sebagai pengirim bit ke-8 adalah bit TB8 dalam register SCON sedangkan yang berfungsi sebagai penerima bit ke-8 adalah bit RB8 dalam register SCON, sedangkan bit stop tidak ditampung. Biasanya bit data ke-9 menunjukkan *parity check* dari 8 bit data. *Baudrate* bisa dipilih $\frac{1}{32}$ atau $\frac{1}{64}$ frekuensi kristal yang digunakan.

D. Mode 3:

Mode 3 ini mempunyai kesamaan dengan mode 2. 1 paket data berukuran 11 bit, diawali dengan 1 bit start, 8 bit data mulai dari bit ke-0 sampai dengan bit ke-7, bit ke-8, dan 1 bit stop. Namun *baudrate* pada mode 3 dapat diatur sesuai dengan keperluan[1].

Kecepatan dan keakuratan pengiriman data serial secara asinkron sangat dipengaruhi oleh *baudrate. Baudrate* adalah banyaknya bit yang dikirim setiap detik. Pada MCS-51, proses pengiriman data secara serial ditentukan oleh mode yang dipilih. *Baudrate* yang digunakan juga bersesuaian dengan mode yang digunakan. Pada mode 0 *baudrate* yang digunakan adalah

$$Baudrate = \frac{\text{frekuensi}}{12}$$

Baudrate yang digunakan pada mode 2 tergantung dari isi bit SMOD yang ada pada register PCON.

$$Baudrate = \frac{2^{SMOD}}{64} x\, frekuensi$$

Baudrate yang digunakan untuk mode 1 dan 3 dapat ditentukan melalui register *Timer* 1 dan juga bit SMOD pada register PCON.

$$Baudrate = \frac{2^{SMOD}}{32} \times \frac{frekuensi}{12 \times TH1}$$

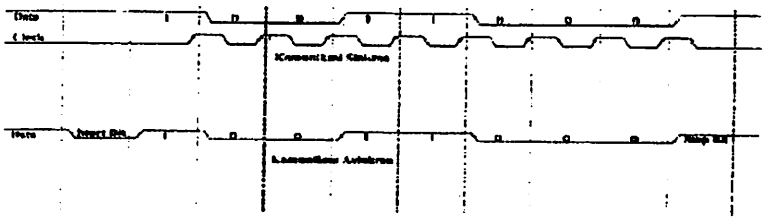
2.5. Komunikasi Serial

Komunikasi serial ada beberapa macam, dibagi berdasarkan cara transmisi, arah proses komunikasi, saluran komunikasi, dan sebagainya.

2.5.1. Metode Transmisi Data Serial

Transmisi data secara serial adalah transmisi data dimana data tersebut akan dikirimkan tiap bit dalam satu waktu. Terdapat *2* cara dalam mentransmisikan data secara serial, yaitu secara *Synchronous* dan *Asynchronous.*

Dari kedua cara tersebut yang membedakan adalah sinyal clock yang dipakai untuk sinkronisasi dalam mengirim data. Perbedaan dari kedua cara ini digambarkan jelas pada Gambar *2.6.*



Gambar 2.6 Komunikasi sinkron dan komunikasi asinkron

A. Komunikasi *Synchronous Serial*

Sinyal clock pada komunikasi *synchronous* (sinkron) diperlukan oleh peralatan penerima data untuk mengetahui adanya pengiriman setiap bit data, Tampak pada gambar 2.6 bahwa sinyal clock terpicu *(positive edge)* pada saat pengiriman bit yang pertama dan setiap perubahan bit data. Peralatan atau komponen penerima akan mengetahui adanya pengiriman bit yang pertama ataupun perubahan bit data dengan mendeteksi sinyal clock tersebut.

B. Komunikasi *Asynchronous Serial*

Komunikasi *asynchronous* (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 *baudrate* dari komunikasi data ini dibangkitkan oleh masing-masing baik penerima maupun pengirim data dengan frekuensi yang sama. Penerima hanya perlu mendeteksi adanya start bit sebagai awal pengiriman data, selanjutnya komunikasi data terjadi antar dua buah shift register yang ada pada pengirim maupun penerima. Setelah 8 bit data diterima, maka penerima akan menunggu adanya stop bit sebagai tanda bahwa 1 byte data telah terkirim dan penerima siap untuk menunggu pengiriman data berikutnya.

Pada aplikasinya proses komunikasi asinkron ini digunakan untuk mengakses komponen-komponen yang mempunyai fasilitas *Universal Asynchronous Receiver/Transmitter* (UART) seperti Port Serial pada Personal Computer (PC) atau Port Serial *microcontroller* yang lain.

2.5.2. Arah Proses Komunikasi Serial

Berdasarkan arahnya, proses komunikasi serial memiliki 3 metode, yaitu *Simplex, Half-Duplex* dan *Full-Duplex.*

A. Simplex

Metode komunikasi *simplex* adalah komunikasi searah, penerima hanya dapat bersifat pasif menerima data tanpa respon. Contoh dari komunikasi ini adalah siaran televisi atau radio.

B. Half-duplex

Metode komunikasi *half-duplex* sudah dapat berkomunikasi secara 2 arah, namun proses pengiriman dan penerimaannya, tidak dapat dilakukan secara bersamaan karena hanya memiliki 1 jalur yang dipakai bergantian. Contoh dari komunikasi ini adalah *handi-talkie (HT)* dan *walkie talkie.*

C. Full-duplex

Metode komunikasi full-duplex adalah metode komunikasi 2 arah yang proses pengiriman dan penerimaannya sudah dapat dilakukan secara bersamaan. Contohnya adalah telepon.

2.5.3. Saluran Komunikasi Serial

Sinyal digital dari IC TTL hanya bisa ditransmisikan paling jauh dua meter, untuk pengiriman yang lebih jauh sinyal harus diolah dan disalurkan dengan cara khusus, yaitu: output sinyal TTL harus di-buffer dengan rangkaian buffer. Tapi biaya pengolahan sinyal ini cukup mahal. Untuk mengirim. data secara paralel diperlukan satu saluran untuk setiap bit data, dengan demikian pengiriman data paralel 8 bit, membutuhkan biaya 8 kali lebih mahal dibandingkan dengan komunikasi data secara serial yang hanya perlu satu saluran saja.

Dengan demikian, meskipun kecepatan transmisi data dengan teknik komunikasi data secara paralel lebih cepat, teknik komunikasi data serial tetap dipilih untuk transmisi data jarak jauh.

Pada umumnya dikenal 3 macam saluran untuk transmisi data serial, yakni saluran yang mernenuhi ketentuan standard EIA RS232, saluran yang memenuhi ketentuan standard EIA RS422/485, dan saluran arus 20 mA (20 mA *current loop*[7].

A. Standard EU RS-232

Untuk memungkinkan kompabilitas di antara peralatan komunikasi data yang dibuat oleh berbagai pabrik dengan *standart interfacing* yang disebut RS232 telah ditetapkan oleh *Electronics Industries Association* (EIA) pada tahun 1960. Pada tahun 1963 standar itu dimodifikasi dan dikenal sebagai RS232A, RS232B, dan RS232C masing-masing diperkenalkan pada 1965 dan 1969. RS232 adalah standar *interfacing* I/O serial yang paling banyak digunakan. Tetapi, karena. standar itu ditetapkan lama sebelum kemunculan keluarga TTL *logic* sehingga level tegangan *input* dan output-nya tidak kompatibel dengan level tegangan TTL. Dalam RS232, bit '1' direpresentasikan dengan -3 sampai dengan -25 V, sedangkan bit 0 adalah +3 sampai dengan +25 V, sehingga -3 sampai +3 tidak didefinisikan. Karena itu, untuk menghubungkan suatu RS232 ke sistem berbasis mikroprosesor kita harus menggunakan konverter tegangan seperti MAX232 untuk mengkonversi level tegangan logika TTL menjadi level tegangan RS232 atau sebaliknya.

Komputer biasanya menggunakan *DB25-connector* untuk komunikasi serial, yang disebut sebagai DB-25SP untuk *connector (male)* dan DB-25S untuk *connector (female)*. Karena sernua pin DB-25 tidak digunakan, maka IBM PC mengenalkan versi DB-9 yang dapat dilihat di tabel 2.20.

Tabel 2.20 Fungsi pin-pin DB9 standar RS232

Pin	Description
1	*Data Carrier Detect* (DCD)
2	*Receive Data* (RxD)
3	*Transmitted Data* (TxD)
4	*Data Terminal Ready* (DTR)
5	*Signal Ground* (GND)
6	*Data Set Ready* (DSR)
7	*Request To Send* (RTS)
8	*Clear To Send* (CTS)
9	*Ring Indicator* (RI)

Dalam transmisi RS-232 urutan kontrol komunikasi dapat dideskripsikan sebagai berikut :

1. DTE mengirimkan sinyal *high* pada DTR untuk memberitahukan bahwa DTE aktif dan siap untuk proses transmisi.
2. DCE memberikan tanggapan dengan mengirirnkan sinyal pada DSR untuk memberitahukan bahwa DCE juga siap berkomunikasi
3. DTE mengirimkan sinyal pada RTS untuk memberitahukan bahwa DTE siap, melakukan proses transmisi
4. DCE menjawab dengan mengirimkan sinyal *high* pada CTS untuk memberitahukan bahwa DCE siap menerima data
5. Setelah ini data ditransmisikan melalui jalur TD dan RD
6. DTE mengirmkan sinyal. low pada DTR untuk memutuskan komunikasi

B. Standard EIA RS422/485

Saluran ini menggunakan cara tmnsmisi berimbang *(balance transmission),* sinyal TTL disalurkan lewat *IC Line Driver* yang mengeluarkan sinyal *differential* pada keluarannya, level logika tidak dinyatakan dengan perbedaan tegangan dengan *ground,* tapi dinyatakan dengan perbedaan tegangan antara kedua sinyal *output* itu, selanjutnya sinyal ini akan diterima oleh *IC Line Receiver* yang mempunyai *input differential* yang akan mengubah sinyal *differential* ke sinyal TTL.

Dengan cara semacam ini, jika dalam perjalanan sinyal menerima derau listrik, kedua jalur dalam saluran akan mendapat sinyal derau yang sama besarnya, sinyal derau yang sama besar ini bagi *input differential* di *IC Line Receiver* tidak akan dirasakan sebagai sinyal digital, sehingga saluran ini lebih kebal gangguan dibanding dengan saluran RS232.

Untuk meningkatkan kekebalannya terhadap ganguan, untuk saluran RS422 biasanya dipakai dua kabel yang dililit jadi satu, dengan maksud jika sampai mendapat gangguan maka gangguan yang diterima kedua kabel yang dililit itu benar-benar sama besarnya, sehingga selisih tegangan derau antara dua kabel benar-benar nol dan tidak dirasakan sebagai gangguan.

Dengan cara *balance transmission* ini, RS422 bisa dipakai untuk menyalurkan informasi dengan kecepatan yang lebih tinggi, dan jarak yang bisa dijangkau mencapai 1,2 km sedangkan RS232 hanya bisa dipakai sampai sekitar 15m[7].

C. Saluran arus 20 mA (20 mA *current loop*)

Saluran ini termasuk saluran yang sudah ketinggalan zaman dan sudah tidak banyak dipakai. Meskipun demikian, karena sifat saluran ini yang sangat kebal

terhadap gangguan saluran ini tetap.digunakan, terutarna pada lingkungan pabrik yang mengandung sumber gangguan derau listrik yang sangat besar.

20 mA Current Loop berbeda prinsip dengan. saluran RS232 dan RS422/485, karena. informasi disalurkan sebagai ada tidaknya arus. Sinyal TTL dipakai untuk mengatur transistor, jika transistor *on* maka arus akan mengalir dari V+ melalui resistor dan membuat LED di dalam Opto *Coupler* yang ada di bagian penerima menyala, yang pada gilirannya membuat transistor *on*. Bisa dibayangkan, sebesar-besamya tegangan derau yang diterima saluran, hampir tidak mungkin menimbulkan arus yang cukup besar sampai LED bisa menyala, dengan dernikian menjadikan saluran ini sangat kebal terhadap gangguan.

Tidak ada standar yang mengatur saluran arus ini, makin besar tegangan V+ makin besar pula arus yang mengalir (idealnya sampai 20 mA) maka saluran makin kebal terhadap gangguan. Kecepatan saluran sangat tergantung pada mutu *opto coupler* yang dipakai, meskipun demikian pada umumnya saluran. arus ini bisa dipakai dengan aman paling tidak sampai sejauh 300m[7].

2.6. *Light Emitting Diode* (LED)

Light Emitting Diode (LED) adalah sebuah komponen diode semikonduktor yang memancarkan spektrum cahaya inkoheren ketika mendapat bias maju dari kutub positif ke kutub negatif dari diode. Efek ini adalah bentuk dari *electroluminescence*[12].

LED terdiri dari sebuah keping bahan semikonduktor yang diisi dengan zat-zat lain untuk menghasilkan elektroda-elektroda berkutub positif dan negatif. Seperti diode pada umumnya, arus listrik bergerak dari elektroda berkutub positif, atau anoda, ke elektroda berkutub negatif, atau katoda. Baik elektron-elektron

maupun *hole-hole* bergerak dari masing-masing kutubnya menuju ke kutub lain yang memiliki perbedaan tegangan. Bila dalam proses pergerakan ini elektron dan hol bertemu, maka akan te~adi reaksi yang menimbulkan pelepasan energi dalam. bentuk *photon* (foton)[12].

Panjang gelombang yang dipancarkan oleh cahaya yang dilepaskan, dengan kata lain warna cahaya tersebut tergantung pada bahan yang dipakai sebagai elektroda. Bila elektroda yang dipakai adalah silikon atau germanium, bahan dari dioda biasa, elektron dan hole bersatu dalam. sebuah transisi yang non radiatif yang tidak menghasilkan emisi cahaya. Perkembangan LED dimulai dengan LED *infra-red* dan LED merah yang terbuat dari *gallium arsenide.* Perkembangan ilmu pengetahuan tentang unsur memungkinkan ditemukannya LED dengan warna-warna yang lain[12].

Untuk meningkatkan efisiensi LED, digunakan lapisan transparan yang ditunjang oleh lapisan yang memantulkan cahaya (reflektif), dan bahan yang dapat menyebarkan cahaya untuk memancarkan gelombang cahaya yang dihasilkan. Antara bahan-bahan yang digunakan tersebut harus benar-benar Cocok dengan unsur elektroda. yang dipakai, jika tidak, sebagian cahaya yang dihasilkan akan dipantulkan kembali, diserap, kemudian berubah bentuk menjadi panas, yang akan mengurangi efisiensi LED. Upaya-upaya lain untuk meningkatkan efisiensi LED adalah dengan kemasan yang berbentuk setengah bola dengan dioda yang diletakkan di tengah-tengahnya. Warna kernasan tidak terlalu berpengaruh terhadap warna cahaya yang dipanearkan. Desain LED dibuat sedemikian rupa agar dapat menyerap dan memancarkan lagi cahaya yang dipantulkan kembali *(photon recycling),* kemudian dengan memanipulasi struktur mikroskopik dari permukaan menjadi berbentuk seperti mata serangga *(moth eye)* untuk mengurangi efek pantulan balik. Akhir-akhir ini,

kristal fotonik *(photonic crystal)* juga digunakan untuk meningkatkan efisiensi LED[12].

Cahaya yang dikeluarkan oleh LED dapat berupa cahaya infra merah, mendekati ultra violet, maupun cahaya tarnpak. Jenis cahaya dan warna cahaya tampak yang dipancarkan oleh LED tergantung pada komposisi dan kondisi dari bahan semikonduktor yang dipakai. Kombinasi komposisi bahan dan warna LED yang dihasilkan dapat dilihat pada Tabel 2.21.

Tabel 2.21 Bahan semikonduktor dan warna LED yang dihasilkan[12]

Bahan	Warna
Alumunium gallium arsenide (AlGaAs)	Merah dan infra-merah
Alumunium gallium phosphide (AlGaP)	Hijau
Alumunium gallium indium phosphde (AlGaInP)	Jingga-merah, jingga, kuning, dan hijau terang
Gallium arsenide phosphde (GaAsP)	Merah, jingga-merah, dan kuning
Gallium phosphde (GaP)	Merah, kuning dan hijau
Gallium nitride (GaN)	Hijau, hijau emerald, dan biru serta putih (jika memilii AlGaN *quantum Barrier*)
Indium gallium nitride (InGaN)	Mendekati ultraviolet, hijau kebiru-biruan, dan biru (450-470nm)
Silicon carbide (SiC) (pelapis)	Biru
Silicon (Si) (pelapis)	Biru
Saphhire (Al_2O_3) (pelapis)	Biru
Zinc selenide	Biru

Diamond	Ultraviolet
Almunium nitride (AlN) *Almunium gallium nitride* (AlGaN) *Almunium gallium indium nitride* (AlGaInN)	Mendekati ultraviolet (dibawah 210nm)

Selain perbedaan dari jenis bahan, LED dengan cahaya yang berbeda juga memiliki tegangan kerja yang berbeda. Tegangan kerja untuk masing-masing jenis cahaya yang dikeluarkan LED dapat diilhat pada tabel 2.22.

Tabel 2.22 Tegangan kerja LED[12]

Cahaya	Tegangan (V)
Infra-merah	1.6
Merah	1.8 – 2.1
Orange	2.2
Kuning	2.4
Hijau	2.6
Biru	3.0 – 3.5
Putih	3.0 – 3.5
Ultraviolet	3.5

LED hanya akan menyala bila diberi polaritas yang tepat. Ketika tegangan yang berada di kutub positif dan negatif sudah benar, arus listrik pada LED akan terbias maju *(forward-biased)*. Sebaliknya, bila terbalik maka akan terjadi bisa mundur *(reverse-biased)*, dalam kondisi ini arus yang dapat lewat sangatlah kecil dan LED tidak menyala. LED dapat tetap beroperasi meskipun diberi arus AC, hanya saja LED tersebut baru menyala saat terjadi arus bias maju. Sehingga nantinya nyala LED

akan berkedip-kedip sesuai dengan frekuensi arus AC yang diberikan. LED memiliki ketentuan standar sehingga untuk menentukan polaritasnya bisa dilakukan dengan melihat ciri-ciri fisik LED seperti pada Tabel 2.23.

Tabel 2.23 Standar kutub positif dan negative pada LED[12]

Tanda	Positif (+)	Negatif (-)
Sambungan	Anoda	Katoda
Pin	Panjang	Pendek
Bentuk luar (tepi)	Melingkar	Rata
Bentuk dalam	Kecil	Besar
Warna kabel (standar)	Merah	Hitam

Meskipun LED dapat menerima arus bias mundur, namun LED memiliki tegangan balik *breakdown (reverse breakdown voltage)* rendah, sehingga bila diberi tegangan balik yang melewati batas tersebut LED akan rusak. LED juga memiliki batas, arus maksimum yang bisa diterima karena pada dasarnya, karakteristik LED seperti diode, tegangan bisa maju *(forward voltage)* LED selalu konstan, dan bahwa LED menyala karena ada arus yang mengalir. Pada umumnya, arus yang optimal untuk menyalakan LED adalah 20 mA. Untuk mengatur agar arus yang mengalir konstan, maka biasanya LED dirangkai seri dengan resistor.

Untuk mengetahui berapa. nilai R yang dipasang di RI, perlu mengetahui tegangan bias maju yang dibutuhkan oleh LED. Hal ini dapat diketahui lewat *datasheet* dari LED yang bersangkutan. Selain itu, tegangan bias maju LED juga dapat diukur dengan nilai R sembarang, misalnya 220 Ω, dan tegangan VCC sebesar 5 Volt. Setelah itu, ukur tegangan bias maju LED di titik B - C seperti pada voltmeter M2. Dari pengukuran ini akan diperoleh tegangan bias maju, misalnya 3 Volt. Di sini

berlaku hukum Kirchoff II, "Jumlah dari masing-masing tegangan yang jatuh pada sebuah rangkaian seri sarna besarnya dengan tegangan total yang melewati rangkaian seri tersebut." dengan kata lain, bila diterapkan pada rangkaian:

$$VCC = V_{AB} + V_{BC}$$

sehingga tegangan di titik A - B adalah:

$$V_{AB} = VCC - V_{BC}$$

$$V_{AB} = 5V - 3\text{V} = 2\text{ V}$$

Atau bila langsung dilakukan pengukuran tegangan di resistor R1 atau titik A - B seperti pada voltmeter MI pasti akan didapatkan hasil yang sama, yaitu 2 Volt. Dari sini diberlakukan hukum Kirchoff I, "Besarnya arus listrik yang masuk di titik percabangan sama dengan besarnya arus listrik yang keluar dari titik percabangan.", sehingga arus yang mengalir di resistor RI sama besamya dengan arus yang mengalir di LED DI. Jadi bila di resistor RI dilakukan perhitungan sesuai dengan rumus hukum Ohm (n):

$$R = V$$

$$\text{R}_1 = \frac{\text{V}_{\text{AB}}}{1} = \frac{2V}{20mA} = 100\Omega$$

Jadi supaya LED D1 mendapat arus sebesar 20 mA dengan tegangan sumber VCC 5 V, nilai resistor RI yang harus dipasang adalah sebesar 100 Ω.

Beberapa LED juga. dapat dihubungkan secara seri dengan satu buah resistor sebagai pembatas arus. Namun perlu diingat bahwa tegangan sumber yang diberikan harus lebih besar dari pada jumlah tegangan bias maju dari seluruh LED dalam satu rangkaian seri tersebut.

Mayoritas LED didesain dengan daya operasi antara, 30-60 mW. Bila masing-masing LED memiliki arus optimal sebesar 20 mA dan tegangan kerja LED

yang paling besar adalah 3,5 Volt untuk LED ultraviolet, dari persamaan 2.9 dapat diketahui bahwa daya yang dibutuhkan LED yang paling besar adalah:

20mA x 3.5V = 70m W

Dari perhitungan tersebut ternyata sebuah LED dengan tegangan kerja paling besar dan diberikan arus optimal hanya membutuhkan daya sebesar 70mW. Daya sebesar itu termasuk kecil. Meskipun untuk menambah intensitas cahaya diperlukan LED dalam jumlah yang cukup besar sekalipun, daya yang dibutuhkan masih tergolong kecil. Sehingga pantaslah jika nantinya LED digunakan sebagai sumber penerangan utama menggantikan lampu neon.

2.7. Osilator

Penggunaan utama dari umpan balik adalah dalam osilator, rangkaian yang membangkitkan sinyal output tanpat sinyal input. Dalam sebuah osilator, sebagian dari output diberikan kembali ke input; sinyal umpan balik ini adalah satu-satunya input ke penguat dalam.

2.7.1. Ide Dasar

Sebelum analisis matematis, marilah kita pelajari ide utama mengenai bagaimana sebuah osilator menghasilkan sinyalk output tanpa sinyal input luar, untuk memulainya. Sebuah sumber tegangan v yang menggerakkan terminal kesalahan (error terminals) dari penguat. Sinyal yang diperkuat Av menggerakkan rangkaian umpan balik untuk menimbulkan tegangan umpan balik ABv. Tegangan ini kembali ke titik x. Jika pergeseran fase melalui penguat dan rangkaian umpan balik benar, sinyal pada titik x akan tepat sefase dengan sinyal yang menggerakkan terminal

kesalahan dari penguat. Dinyatakan dengan ara lain, jika pergeseran fase keliling seluruh loop adalah 0°, kita dapatkan *umpan balik positif.*

2.7.2. Kristal Kuarts

Beberapa kristal yang ditemukan di alam menunjukkan efek piezoelektrik; jika anda memasang tegangan ac melalui kristal tersebut mereka akan bervibrasi pada frekuensi dari tegangan ac yang dipasang. Bahan utama yang menimbulkan efek piezoelektrik ini adalah kuarts, garam *Rochelle*, dan *tourmaline*.

Garam Rochelle mempunyai aktivitas piezoelektrik yang terbesar untuk suatu tegangan ac yang diberikan. Mereka bervibrasi lebih dari kuarts atau tourmaline. Secara mekanis, mereka adalah yang paling lemah; mereka mudah pecah. Garam Rochelle telah digunakan untuk membuat mikropon, pickup gramapon, headset dan pengeras suara.

Tourmaline menunjukkan aktivitas piezoelektrik yang terkecil, tetapi diantara ketiganya paling kuat. Kristal ini juga yang paling mahal. Kadang – kadang dia digunakan pada frekuensi yang sangat tinggi.

Kuarts adalah kompromi antara aktivitas piezoelektrik fari garam Rochelle dan kekuatan dari tourmaline. Karena tidak mahal dan dapat diperoleh di alam, kuarts digunakan secara luas untuk osilator RF dan filter.

Bentuk alami dari kuarts adalah prisma heksagonal dengan pyramid pada ujung – ujungnya. Untuk mendapatkan kristal yang berguna dari sini kita harus mengiris lempeng empat persegi panjang dari kristal alam. Lempeng tersebut tebalnya t. Jumlah lempeng yang kita peroleh dari kristal alam tergantung pada ukuran dari lempeng dan sudut pemotongan.

Ada sejumlah cara yang berbeda untuk memotng kristal alam; potongan tersebut mempunyai nama seperti potongan X, potongan Y, potongan XY dan potongan AT. Untuk tujuan kita, semua yang harus kita ketahui adalah potongan mempunuyai sifat *piezoelektrik* yang berbeda. (Katalog dari pabriknya biasanya merupakan sumber informasi yang paling baik mengenai potongan yang berbeda dan sifat-sifatnya).

Untuk pengggunaan dalam rangkaian elektronik, lempeng harus dipasang antara dua pelat. Dalam rangkaian ini jumlah dari vibrasi kristal tergantung pada frekuensi dari tegangan yang dipasang. Denagn mengubah frekuensi kita dapat menemukan frekuensi resonan di mana vibrasi kristal mencapai maksimum. Karena energi untuk vibrasi harus diberikan oleh sumber ac, arus ac menjadi maksimum pada tiap frekuensi resonan.

A. Frekuensi Dasar dan Nada Tambahan

Untuk waktu yang lama kristal dipotong dan dipasang untuk bervibrasi paling baik pada salah satu frekuensi resonan-nya, biasanya frekuensi dasar atau frekuensi yang terendah. Frekuensi resonan yang lebih tinggi disebut nada tambahan adalah hampir kelipatan eksak dari frekuensi dasar. Sebagai contoh sebuah kristal dengan frekuensi dasar 1 MHz mempunyai nada tambahan pertama mendekati 2 MHz, pada tambahan kedua mendekati 3 MHz dan seterusnya.

Rumus untuk frekuensi dasar dari Kristal adalah

$$f = \frac{K}{t}$$

Dimana K adalah sebuah konstanta yang tergantung pada potongan dan faktor-faktor lain, dan t adalah tebal kristal. Seperti kita lihat, frekuensi dasar berbanding terbalik terhadap tebal. Untuk alasan ini ada batas praktis mengenai berapa tingginya kita

dapat menaikkan frekuensi. Makin tipis kristal tersebut ; makin menjadi rapuh dan makin besar kemungkinannya untuk pecah karena vibrasi.

Kristal kuarts bekerja dengan baik sampai 10 MHz pada frekuensi dasar. Untuk mencapai frekuensi yang lebih tinggi kita dapat menggunakan kristal yang dipasang untuk bervibrasi pada nada tambahan dengan cara ini, kita dapat mencapai frekuensi sampai 100 MHz. Kadang-kadang tourmaline yang lebih mahal tetapi lebih kuat digunakan pada frekuensi yang lebih tinggi.

B. Rangkaian Ekivalen AC

Menyerupai apakah kristal tersebut sepanjang yang menyangkut sumber ac? Jika kristal yang dipasang tidak bervibrasi, hal ini ekivalen dengan kapasitansi Cm karena dia mempunyai dua pelat logam yang dipisahkan oleh dielektrik. Cm disebut kapasitansi pemasangan (*mounting capacitance*).

Tetapi, jika kristal bervibrasi, dia menyerupai rangkaian yang ditala. Rangkaian ekivalen ac dari kristal yang bervibrasi pada dekat frekuensi dasar. Harga tipikal dari L adalah dalam henry, Cs dalam pecahan dari pikofarad, harga-harga untuk satu kristal yang bisa diperoleh: L = 3 H, Cs = 0,05 pF, R = 2000 Ω, dan Cm = 10 pF. Yang mempengaruhi harga-harga tersebut antara lain adalah, potongan, tebal dan pemasangan lempeng (slab).

Ciri-ciri yang terkenal dari kristal dibandingkan dengan rangkaian tank LC yang diskrit adalah harga Q-nya yang sangat tinggi. Untuk harga-harga yang baru saja diberikan, kita dapat menghitung Q di atas 3000. Harga-harga Q dapat dengan mudah mencapai lebih dari 10.000. Di pihak lain, rangkaian tank LC jarang mempunyai Q di atas 100. Harga Q yang sangat tinggi dari kristal memungkinkan osilator dengan harga frekuensi yang sangat stabil.

C. Resonansi Seri dan Paralel

Di samping Q, L, Cs, R dan Cm dari kristal, ada dua karakteristik lain yang harus kita ketahui. Frekuensi resonan seri fs dari sebuah kristal adalah frekuensi resonan dari cabang LCR. Pada frekuensi ini arus cabang mencapai harga maksimum, karena L beresonansi dengan Cs. Rumus untuk frekuensi resonan seri adalah

$$f_s = \frac{1}{2\pi\sqrt{LC_s}}$$

Frekuensi resonan paralel fp dari kristal adalah frekuensi di mana arus sirkulasi atau arus loop dalam mencapai harga maksimum. Karena loop arus ini harus mengalir melalui kombinasi seri dari Cs dan Cm, maka C ekuivalen adalah

$$C_{loop} = \frac{C_m C_s}{C_m + C_s}$$

Dan frekuensi resonan paralel adalah

$$f_p = \frac{1}{2\pi\sqrt{LC_{loop}}}$$

Dua kapasitansi dalam hubungan seri selalu menghasilkan kapasitansi yang lebih kecil daripada salah satu dari keduanya; karena itu, C_{loop} lebih kecil daripada Cs dan fp lebih besar dari pada fs.

Dalam tiap kristal, Cs jauh lebih kecil daripada Cm. Misalnya, dengan harga-harga yang telah diberikan, Cs adalah 0,05 pF dan Cm sama dengan 10pF. Karena hal ini, Persamaan (19-10) memberikan harga dari Cloop hanya sedikit lebih kecil daripada Cs. Selanjutnya hal ini berarti fp hanya sedikit lebih besar daripada fs. Jika anda menggunakan kristal dalam sebuah rangkaian osilator, tambahan kapasitansi rangkaian muncul dalam hubungan cabang dengan Cm. Karena ini frekuensi osilasi akan terletak antara fs dan fp. Ini adalah keuntungan dari mengetahui harga dari fs dan fp ; mereka menset batas bawah dan batas ats frekuensi dari osilator kristal.

D. Stabilitas Kristal

Frekuensi dari sebuah osilator cenderung untuk berubah sedikit dengan waktu, drift ini ditimbulkan oleh temperatur, usia dan sebab-sebab lain. Dalam sebuah osilator kristal drift frekuensi dengan waktu kecil sekali, secara tipikal kurang dari 1 bagian dalam 10^6 (0,0001 persen) per hari. Stabilitas seperti ini penting dalam jam tangan elektronik mereka menggunakan osilator kristal kuarts sebagai alat pengatur waktu dasar.

Dengan menggunakan osilator kristal dalam tungku (oven) yang temperaturnya dikendalikan denan presisi, osilator kristal telah dibuat dengan drift frekuensi kurang dari 1 bagian dalam 10^{10} per hari. Stabilitas seperti ini diperlukan dalam standard waktu dna frekuensi. Untuk memberikan mengenai bagaimana keseksamaan 1 bagian dalam 10^{10} adalah, sebuah jam dengan drift ini akan memakan wkatu 300 tahun untuk lebih cepat atau terlambat 1 detik[8].

2.8. Modem

2.8.1. Mengenal Cara Kerja Modem[9]

Untuk dapat berselancar menggunakan Internet, salah satu komponen utama yang harus ada adalah modem.

Modem merupakan sebuah singkatan yang berarti modulator-demodulator. Namanya ini disesuaikan dengan fungsinya sebagai modulator dan demodulator. Dalam kerjanya, modem melakukan proses modulasi dan demodulasi terhadap data yang dipancarkan. Modem menerima rangkaian pulsa biner dari periferal komputer, kemudian memodulasi karakteristik sinyal analog (level tegangan, frekuensi atau fasa) agar dapat disalurkan melalui saluran telepon atau cable lines. Sedangkan pada si penerima, sinyal yang ditumpangi ini oleh rangkaian demodulator dipisahkan kembali

dari sinyal yang menumpanginya sehingga dapat dibaca oleh komputer, proses ini dinamakan demodulasi. Standarisasi dari modulasi dewasa ini berfungsi untuk mencapai kecepatan yang lebih baik lagi. Pada awalnya kecepatan dari modem ini adalah 300 bps dan dewasa ini telah mencapai 56 Kbps. Kecepatan modem itu sendiri sekarang ini sudah cukup cepat dibanding dahulu, tetapi untuk penggunaannya di Indonesia masih dibatasi dengan kurang bagusnya jaringan telepon yang tersedia. Saat ini kecepatan modem yang sering digunakan di Indonesia adalah sebesar 56Kbps, tetapi dengan kondisi jaringan telepon yang ada, kecepatan tersebut mungkin maksimal hanya sekitar 33.6 Kbps saja.

2.8.2. Jenis-Jenis Fisik Dan Fasilitas Dari Modem

Secara umum jenis fisik dari modem terdiri dari dua yaitu eksternal dan internal (onboard). Sesuai dengan namanya, keduanya dibedakan karena perletakannya yang berbeda. Modem eksternal merupakan modem yang terpisah dari PC atau dengan kata lain, modem ini tidak dipasangkan pada slot ekspansi yang ada pada mainboard. Modem eksternal lebih mahal dibandingkan modem internal, karena lebih aman dan lebih gampang diperbaiki jika rusak. Modem eksternal selain menggunakan casing dan power supply sendiri juga mempunyai alat pemonitoran dan tombol reset tersendiri sehingga lebih mudah dipantau keadaannya. Sedangkan dalam pemasangannya, modem eksternal ini tinggal menyambungkan kabel-kabel ke interface serial atau USB dari PC. Sedangkan modem internal (*onboard*), sesuai dengan namanya, diletakkan langsung di dalam PC atau laptop, modem langsung dipasangkan pada slot ekspansi yang ada pada mainboard. Kebanyakan modem internal sekarang ini menggunakan interface PCI. Dewasa ini modem telah berkembang dengan berbagai fasilitas yang cukup bermanfaat, misalnya voice

modem. Dengan adanya fasilitas voice modem ini, merubah fungsi modem bukan hanya sebagai penyambung ke internet tetapi lebih dari itu, modem dapat menjadi saluran radio, audio, percakapan telepon sampai streaming video. Malah saat ini sudah banyak modem yang mempunyai fasilitas yang dapat menjadi sebagai mesin penjawab telepon dan perekam suara. Selain itu modem mempunyai kemampuan menjadi fax modem. Fax modem ini bekerja seperti faksimil, dengan sebuah software yang dirancang khusus, dapat mengubah modem menjadi sebuah mesin faksimil, dimana modem merubah dokumen digital menjadi sinyal analog dan dibangkitkan kembali menjadi file gambar jika yang menerima kembali adalah fax modem atau menjadi dokumen tercetak jika yang menerima adalah mesin faksimil.

2.8.3. Cara Kerja Modem[3]

Kebanyakan modem yang digunakan di PC atau laptop dewasa ini adalah dengan menggunakan teknik asynchronous. *Asynchronous* ini maksudnya bahwa ketika modem ini mengirimkan data tanpa menggunakan clock untuk menyinkronisasikan kegiatan dari kedua sistem yang terhubung. Data dikirim dalam 1 byte yang berada dalam sebuah frame pada satu waktu. Frame tersebut berisikan sebuah start bit, data, dan biasanya satu atau lebih stop bit. Start dan stop bit inilah yang memberitahukan kapan dan dimana data tersebut. Karena fungsi inilah, si penerima akan tahu mana yang data dan mana yang noise, sehingga dapat diketahui mana yang dapat diterima atau tidak. Modem ini juga bisa menggunakan parity sebagai error detection. Ada dua parity yang digunakan, *odd* dan *even*. Jenis modem yang menggunakan parity ini sudah jarang digunakan pada masa sekarang ini. Standarisasi Sistem Transmisi Untuk standarisasi sistem transmisi dari modem, maka dua badan dunia yaitu CCITT (*Committee Consultative International Telegraphique*

et Telephonique) dan ITU (*International Telecommunication Union*), mengeluarkan sebuah standar yang dinamakan V-dot. Standar ini berhubungan dengan kecepatan kerja modem, tipe kompresi data dan penanganan kesalahan data. Misalnya, V22bis mengacu pada kecepatan modem 2,4 Kbps, V.32 yang diperkenalkan tahun 1984 mengacu pada kecepatan 9.6 Kbps, V.32bis pada tahun 1991 dengan kecepatan 14.4 Kbps. Standar V.34 pada tahun 1994 memperkenalkan kecepatan 28.8 Kbps, yang pada tahun 1996 diperbaharui dengan V.34+ dengan kecepatan 33.6 Kbps. Kemudian pada tahun 1998 dikembangkan V.90 dari ITU yang mempunyai kecepatan 42 kbps dan disempurnakan kecepatannya menjadi 55.6 Kbps. Penting dicacat bahwa batasan tersebut tidak berhubungan dengan kabel tembaga pada jaringan telepon umum, tetapi berhubungan dengan converter analog ke digital yang dipasang pada jaringan tersebut. Setiap kanal line telepon memiliki 4KHz bandwidth analog, yang setara dengan 64K bit/sec bandwidth digital. Dalam kondisi baik, sebuah line telepon, secara teori, mendukung 64K bit/sec. Pemrosesan sinyal yang kuat pada modem V.90 yang memungkinkan 56K bit/sec pada kanal saat ini. Kecepatan operasi sebuah modem tergantung pada rintangan yang harus diatasinya. Ada tiga rintangan utama untuk mendapatkan kecepatan yang maksimal yaitu port serial, kondisi saluran telepon dan jenis modem yang terpasang. Rintangan pertama berhubungan dengan port serial yang diakibatkan karena keterbatasan chip UART (*Universal Asynchronous Receiver/Transmitter*), yaitu chip yang menghubungkan antara port serial dan sistem bus PC/laptop. Sistem bus PCI bekerja dalam blok 32 bit, sedangkan kabel serial hanya menyalurkan bit-bit dalam satu rangkaian. UART harus menangani semua lalu-lintas data yang terjadi dan menyesuaikannya dengan prot serial tanpa boleh terhadi kemacetan. Rintangan kedua berhubungan dengan saluran telepon. Walaupun saluran telepon menggunakan sistem digital, tetapi masih ada elemen analog yang tersisa.

yang terdiri dari *full-rate* ADSL dan ADSL-lite atau G.lite. ADSL menawarkan kecepatan *downstream* hingga 8 MBps dan *upstream* 2 Mbsp, rata rata kecepatan yang digunakan secara efektif adalah 256 kbps hingga 1.5 mbps. Pada dasarnya teknologi DSL lebih menguntungkan dibandingkan koneksi Internet dial-up seperti koneksi yang jauh lebih cepat, stabil, lebih aman dan dapat melewatkan data serta suara pada sebuah jalur kabel. Jika koneksi diberlakukan selama 24 jam terus menerus maka biayanya juga jauh lebih murah. Misalnya, bila modem biasa dengan kecepatan 56kbps hanya dapat mendownload file 7 KB perdetik maka modem DSL mampu sekitar 1MB per detik. Adapun kelemahan DSL yaitu masalah jarak antara pengguna dengan sentral telepon. Semakin jauh jaraknya akan semakin lambat pula koneksinya, karena akan terjadi distorsi saat data lewat pada kabel telepon dan sinyal data akan banyak yang rusak. Karena itu, untuk memasang sebuah koneksi, ada beberapa hal yang perlu diperhatikan dalam penggunaan DSL ini yaitu pertama kita harus mengetahui apakah layanan tersebut ada di wilayah layanan DSL. Kedua, tentukan jenis modem DSL yang digunakan, hal ini berhubungan dengan jenis layanan DSL yang ditawarkan oleh ISP pilihan anda, beberapa perusahaan menyewakan modem tersebut dan termasuk dengan paket yang anda pilih, beberapa yang lain mungkin menjualkannya dengan harga promosi. Hal perlu diperhatikan, penggunaan modem eksternal memerlukan komputer yang dilengkapi dengan kartu Ethernet dan jika kita memilih paket full-rate, kita harus memasang splitter pada kabel telepon agar hubungan telepon tidak terganggu. Ketiga, biasanya perusahaan telepon bertindak sebagai penyedia jasa DSL dan ISP sekaligus, walaupun beberapa ISP yang menyediakan layanan DSL sendiri-sendiri, kita tetap membutuhkan jaringan telepon untuk mengaksesnya. Cable modem dewasa ini memiliki prospek yang cukup cerah, karena menawarkan kecepatan akses Internet yang lebih cepat dengan menggunakan

jaringan TV kabel. Perangkat kabel modem ini berupa sebuah kotak eksternal yang disambungkan langsung ke PC melalui *interface Ethernet*.Cable modem tipe lama menggunakan fasilitas jaringan dari tv cable. Sementara tipe yang lebih baru menggunakan *Hybrid Fibre-Coax* (HFC).

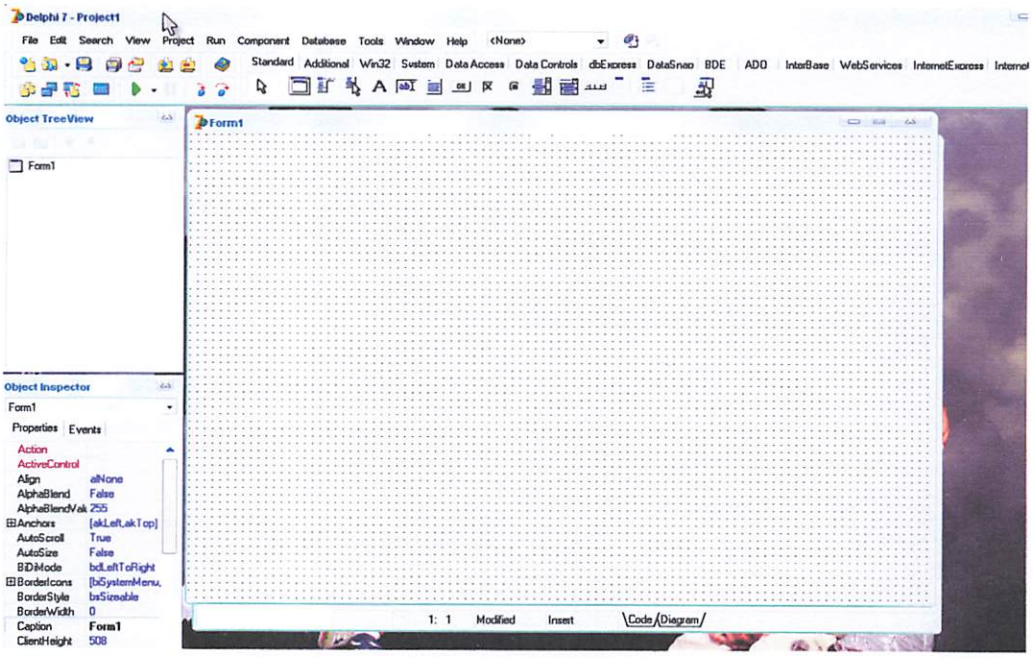
2.9. Borland Delphi

Secara umum, Borland Delphi adalah sebuah program untuk membuat aplikasi – aplikasi berbasis Windows. Bahasa pengembangan yang digunakan oleh Delphi adalah bahasa Pascal. Turbo Pascal dikenal dengan kelebihan dalam kecepatan eksekusi dan kompilasi, dibandingkan dengan bahasa pemrograman lain yang berkembang saat ini. *Integrated Development Enviroment* (IDE) yang diperkenalkan dan diterapkan oleh Turbo Pascal sangat memudahkan para 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.

2.9.1. IDE (*Integrated Development Enviroment*)

IDE adalah sebuah lingkungan yang berisi tool – tool yang diperlukan untuk desain, menjalankan dan mengetes sebuah aplikasi, disajikan dan terhubung dengan baik sehingga memudahkan pengembangan program. Di Delphi, tampilan *IDE* ditunjukkan oleh gambar 2.7.



Gambar 2.7 IDE (*Integrated Development Enviroment*)[10]

Dimana *IDE* terdiri dari:

a. *Main Window*

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

b. *Menu Utama*

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

c. *Toolbar*

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

d. *Form Designer*

Jendela kosong yang digunakan untuk merancang aplikasi Windows.

e. *Code Editor*

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

f. *Code Explorer*

Code explorer digunakan untuk memudahkan navigasi didalam file unit.

g. *Object Treeview*

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

2.9.2. Menu Borland Delphi

1. *Menu File*

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

2. *Menu Edit*

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

3. *Menu Search*

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

4. *Menu View*

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

5. *Menu Project*

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

6. *Menu Run*

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

7. ***Menu Component***

Berisi fasilitas untuk mengatur properti *Component 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***

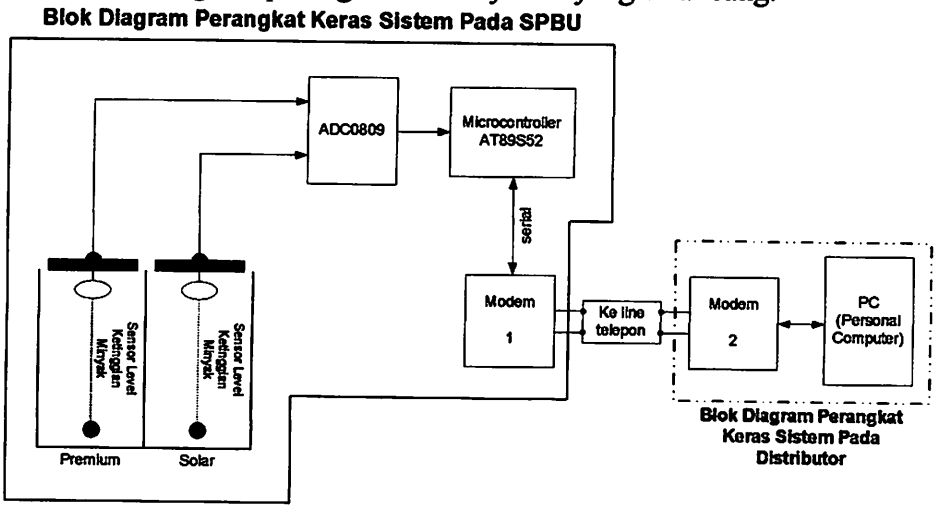
Berisi fasilitas menerima bantuan atau keterangan tentang Delphi.

BAB III

PERENCANAAN DAN PEMBUATAN ALAT

3.1. Blok Diagram Rangkaian

Berikut ini blok diagram perangkat keras system yang dirancang.



Gambar 3.1 Blok Diagram Perangkat Keras Sistem

- **Sensor Level Ketinggian Minyak** berupa potensiometer, digunakan untuk mendeteksi ketinggian minyak dalam tanki.
- **ADC0809** digunakan sebagai pengkonversi data analog menjadi data digital.
- **Mikrokontroller AT89S52**, merupakan rangkaian penterjemah sinyal dari ADC untuk PC.
- **Kabel Serial,** berfungsi sebagai penghubung dari mikrokontroller ke modem pertama.
- **Modem 1,** merupakan perangkat modulator demodulator yang menjadi media penghubung antara mikrokontroller dengan line telepon

- **Line telepon,** menjadi perangkat yang dipakai sebagai penghubung perangkat keras sistem di SPBU dengan sistem perangkat keras pada distributor.
- **Modem 2,** merupakan perangkat modulator demodulator yang menjadi media penghubung antara line telepon dengan PC (Personal Computer) user.
- **PC (Personal Computer),** adalah piranti yang digunakan distributor atau user di rumah atau tempat tertentu untuk memantau kondisi minyak dalam tanki di SPBU.

Untuk memperjelas prinsip kerja dan fungsi dari masing – masing blok diagram, maka pada perencanaan ini dibagi kedalam dua sub bab yaitu:

a. Perencanaan Hardware

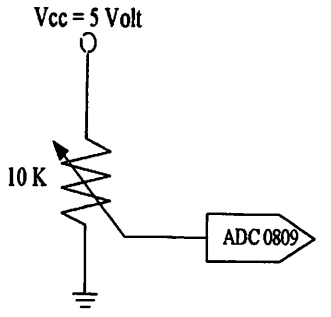
b. Perencanaan Software

3.2. Perencanaan Hardware

3.2.1. Sensor Level Ketinggian Minyak

Sensor level ketinggian minyak yang digunakan adalah potensiometer. Potensiometer dihubungkan dengan pelampung yang akan mengapung dipermukaan minyak dalam tangki. Dengan demikian, naik turunnya pelampung mempengaruhi besaran hambatan yang dihasilkan potensiometer. Perubahan tegangan yang dihasilkan inilah yang diinputkan ke ADC 0809.

Rangkaian sensor ketinggian minyak dapat digambarkan seperti Gambar 3.2 sebagai berikut:



Gambar 3.2 Rangkaian Sensor Ketinggian Minyak

Rangkaian sederhana ini dibuat 2 unit yang masing-masing diletakkan dalam tiap tangki. Dalam tugas akhir yang disusun penulis, ada 2 tangki minyak yang dipantau.

3.2.2. ADC 0809

Rangkaian ADC ini terdiri dari ADC 0809 yang merupakan ADC 8 bit dengan output paralel. Agar ADC ini dapat bekerja secara optimal maka diberi catu daya 5 volt. ADC 0809 telah dilengkapi clock internal yang dapat diaktifkan dengan menghubungkan tahanan dan kapasitor eksternal. Nilai tahanan R ditentukan sebesar 10 KΩ dan nilai kapasitor C sebesar 150 pF [4]. Sehingga memberikan frekuensi clock sebesar :

$$f = \frac{1}{1{,}1\,RC} = \frac{1}{1{,}1.10K\Omega\,.150pF}$$

$$= 606{,}06\ \text{KHz}$$

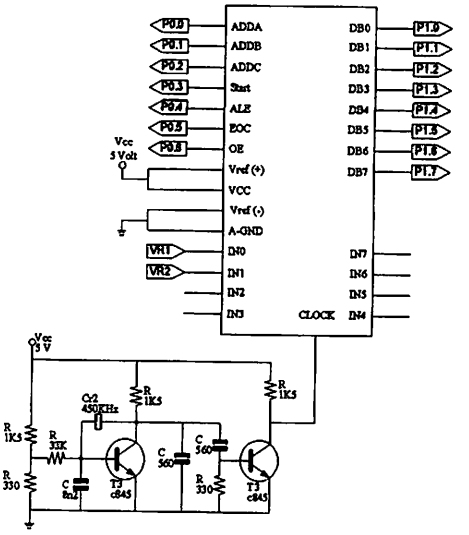
ADC ini dapat menerima tegangan input antara 0 – 5 volt. Pada perencanaan ini diberikan tegangan referensi (Vref) sebesar 2,5 volt. Maka resolusinya adalah :

Resolusi = Vref / 255

= 2,5 / 255

= 10 mV/step

Dari hasil perhitungan tersebut, disusunlah rangkaian *Analog to Digital* seperti pada Gambar 3.3 berikut:



Gambar 3.3 Rangkaian Analog To Digital

3.2.3. Kristal

Kristal yang akan digunakan di sini bergantung pada baudrate yang dibutuhkan oleh mikrokontroller. Dalam hal ini, baudrate yang dibutuhkan adalah sebesar 2400Kbps dan TH1=13 serta SMOD=0, maka diperlukan 2 persamaan berikut[1]:

$$\text{Baudrate} = \frac{2^{\text{SMOD}}}{32} \times (\text{laju limpahan Timer1})$$

$$\text{Baudrate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{frekuensi kristal}}{12 \times \text{TH1}}$$

Dari kedua persamaan di atas, didapat kesimpulan bahwa:

$$\text{laju limpahan Timer1} = \frac{\text{frekuensi kristal}}{12 \times \text{TH1}}$$

Berdasarkan persamaan-persamaan di atas, maka:

$$\text{Baudrate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{frekuensi kristal}}{12 \times \text{TH1}}$$

$$2400 = \frac{2^{0}}{32} \times (\text{laju limpahan Timer1})$$

$$\text{laju limpahan Timer1} = \frac{2400 \times 32}{2^{0}} = \frac{76800}{1} = 76800$$

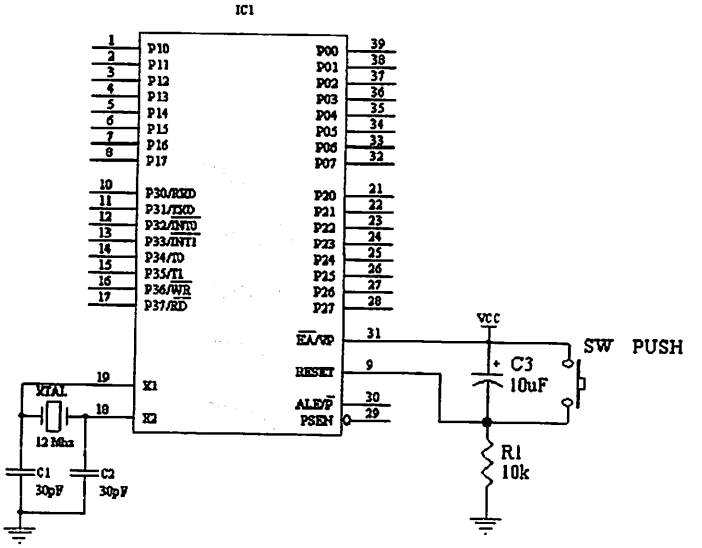
$$\text{laju limpahan Timer1} = \frac{\text{frekuensi kristal}}{12 \times \text{TH1}}$$

$$76800 = \frac{\text{frekuensi kristal}}{12 \times 13}$$

$$\text{frekuensi kristal} = 76800 \times 12 \times 13 = 11980800\text{Hz} \approx 12\text{MHz}$$

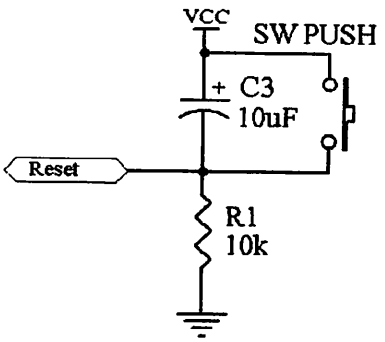
3.2.4. Mikrokontroller AT89S52

Mikrokontroller memerlukan beberapa komponen eksternal agar dapat berfungsi dengan baik, paling tidak harus dibuat rangkaian sistem minimumnya. Rangkaian sistem minimum ini terdiri atas sebuah mikrokontroller AT89S52, kristal 12MHz, tiga buah kapasitor yaitu dua buah kapasitor 30pF dan sebuah kapasitor 10μF, serta sebuah tahanan 10 kΩ. Rangkaian sistem minimum ditinjukkan dalam gambar 3.4 di bawah ini :



Gambar 3.4 Sistem minimum mikrokontroller AT89C52

Dari data *sheet*, untuk mereset mikrokontroler AT89S52, pin RST harus diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator). Untuk membangkitkan sinyal reset kapasitor dihubungkan dengan V_{CC} dan sebuah resistor dihubungkan dengan ground. Selain itu diantara kapasitor dipasang sebuah saklar (*push button*) untuk memberikan sinyal reset secara manual. Rangkaian reset ditunjukkan dalam Gambar 3.5.



Gambar 3.5 Rangkaian reset

Karena kristal yang dipergunakan sebesar 12 MHz, maka periode yang didapat sebesar:

$$T = \frac{1}{f}$$

$$= \frac{1}{12 \times 10^6} S$$

$$= 8{,}33 \times 10^{-8} S$$

Waktu logika tinggi yang dibutuhkan untuk mereset mikrokontroler adalah:

$$T_{reset} = T_{Osc} \times \textit{periode yang dibutuhkan}$$

$$= 8{,}33 \cdot 10^{-8} S \times 24$$

$$= 2uS$$

Jadi mikrokontroler membutuhkan waktu minimal 2 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 :

$$t = RC\, Ln \frac{5}{V_O(t)}$$

dengan nilai Vo adalah tegangan logika nominal yang diijinkan oleh pin RST (Atmel, 1997: 4-37).

Vo = 0,7 x V_{CC} = 0,7 x 5 volt = 3,5 volt.

$$t = RC\, Ln \frac{5}{3{,}5}$$

t = 0,357 $R.C$ = 0,357 x 8200Ω x 10.10^{-6} = 29,274 ms

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

3.2.5. Light Emitting Diode (LED)

16 buah LED dihubungkan dengan Port 2 mikrokontroller sebagai indikator level ketinggian minyak. 8 buah LED pertama menjadi indikator level ketinggian minyak pada tangki 1. Sedangkan 8 buah lainnya sebagai indikator level ketinggian minyak di tangki 2.

Pada umumnya, arus yang optimal untuk menyalakan LED adalah 20 mA. Untuk mengatur agar arus yang mengalir konstan, maka biasanya LED dirangkai seri dengan resistor.

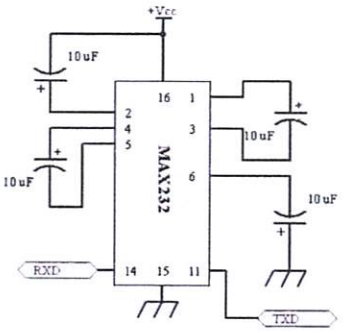
Untuk mengetahui berapa. nilai R maka:

$$V = I \times R$$
$$5 = 20.10^{-3} \times R$$
$$R = \frac{5}{20.10^{-3}}$$
$$R = 250\Omega$$

Namun pada kenyataannya, untuk R yang digunakan adalah 220Ω. Hal ini disebabkan karena nilai R yang umum dipasaran adalah 220Ω. Penggunaan nilai R yang lebih kecil, hanya berpengaruh pada nyala LED yang dalam hal ini, nyalanya menjadi lebih terang (maksimal).

3.2.6. Kabel Serial (Rangkaian MAXIM 232)

Data yang dikeluarkan dari pin TXD mikrokontroler AT89S52 masih berbentuk logika TTL, supaya spesifikasinya sesuai dengan protokol RS 232, maka harus dirubah terlebih dahulu ke logika RS 232 dengan menggunakan komponen MAXIM 232.

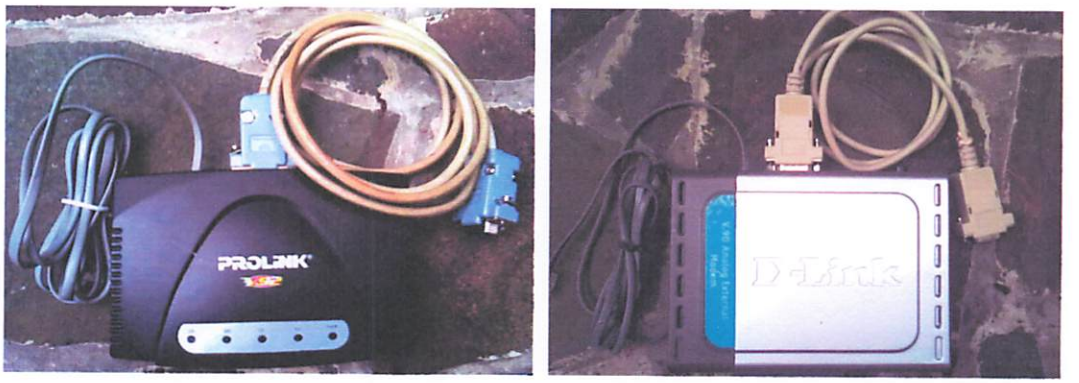


Gambar 3.6 Rangkaian MAXIM 232

Gambar 3.6 menunjukkan skema rangkaian MAXIM 232 yang akan digunakan untuk melakukan konversi dari tegangan TTL menjadi tegangan RS 232. Pin no 11/T1IN menerima tegangan TTL dari pin TXD mikrokontroler AT89S51 dan *pin* no 14/T1OUT akan dihubungkan ke pin RXD dari *port* RS 232.

3.2.7. Modem

Modem yang digunakan di sini ada 2 dengan model dan tipe yang berbeda serta berbeda fungsi. Modem 1 merupakan perangkat modulator demodulator yang menjadi media penghubung antara mikrokontroller dengan line telepon, sedangkan modem ke 2 merupakan perangkat modulator demodulator yang menjadi media penghubung antara line telepon dengan PC (Personal Computer) user.



Gambar 3.7 Foto Modem Yang Digunakan

3.2.8. Line Telepon

Line telepon merupakan sarana penghubung antar modem, dalam hal ini digunakanlah sebuah PABX sebagai pengganti media line telepon. Dimana Fungsi dari PABX tersebut menyerupai line telepon

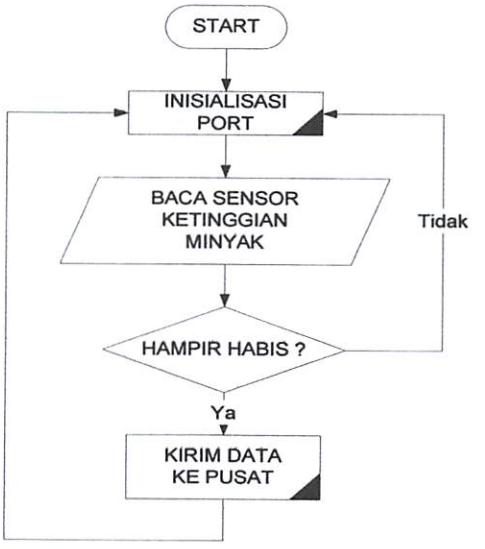
Gambar 3.8 PABX

3.3. Perencanaan Software

2 software yang digunakan dalam perencanaan software di sini adalah assembler untuk mikrokontroller dan Delphi untuk tampilan pada PC.

3.3.1. Perencanaan Software dalam Mikrokontroller AT89S52

Sebelum membuat program untuk mikrokontroller, terlebih dulu membuat diagram alir atau diagram blok seperti berikut.

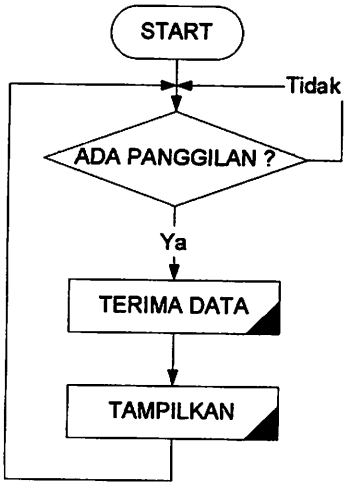


Gambar 3.9 Flowchart Software Pada Mikrokontroller AT89S52

Jadi fungsi mikrokontroller di sini adalah untuk mengambil data ketinggian level minyak dari ADC dan kemudian mengirimkan datanya ke PC melalui kabel serial, modem dan line telepon.
Listing program pada mikrokontroller dapat dilihat pada Lampiran 1.

3.3.2. Perencanaan Software Pada PC

Untuk merancang sebuah software, terlebih dulu dibuat Flowchart seperti pada gambar 3.10 berikut.



Gambar 3.10 Flowchart Software Pada PC (*Personal Computer*)

Software yang digunakan untuk membuat program dalam perancangan alat ini adalah Borland Delphi 7. Komponen penting yang digunakan adalah *Comp Port*. Komponen ini berfungsi untuk menghubungkan atau menampilkan data yang diterima modem, data yang diterima dapat ditampilkan dengan menggunakan *hyperterminal*. Sedangkan komponen-komponen lain yang digunakan adalah komponen-komponen standard seperti *Label, Edit, Panel, Label, GroupBox*, dan sebagainya, seperti yang tercantum dalam *listing program*. Tampilan program di sini memiliki beberapa menu. Dan listing-listing program serta databasenya dapat dilihat pada Lampiran 2.

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 Sensor Ketinggian Level Minyak

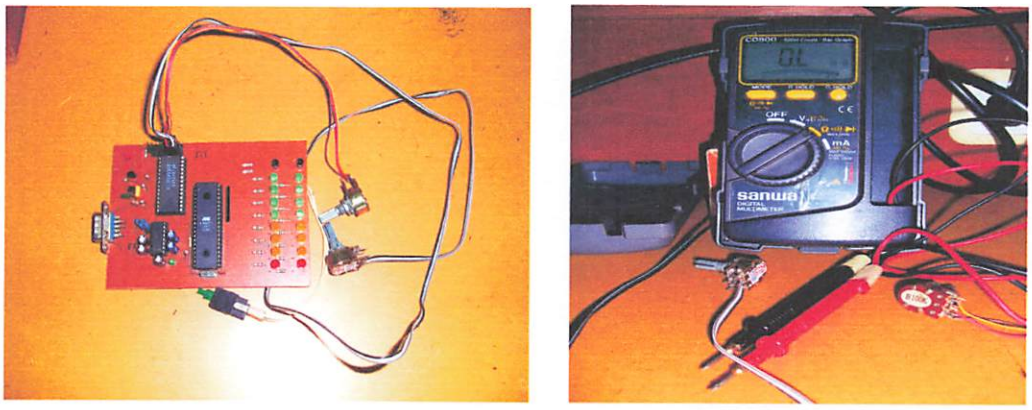
4.2.1. Tujuan

Pengujian ini bertujuan untuk mengetahui kemampuan sensor ketinggian level minyak untuk mendeteksi ketinggian level minyak dan mengirimkan data ini ke piranti berikutnya.

4.2.2. Alat dan Bahan

a. Rangkaian sensor ketinggian level minyak yaitu potensiometer 10KΩ
b. Power supply DC 5 Volt
c. SANWA Digital Multimeter CD800

4.2.3. Prosedur Pengujian



Gambar 4.1 Alat dan alat ukur

a. Menghubungkan power supply ke rangkaian potensiometer

b. Mengeset potensiometer menjadi 0 Ω

c. Mengukur tegangan yang dihasilkan saat tahanan 0 Ω

d. Mengulangi kegiatan a sampai c di atas dengan nilai tahanan berubah-ubah

e. Mencatat hasil pengujian pada Tabel 4.1.

4.2.4. Hasil Pengujian

Tabel 4.1 Hasil Pengujian Alat

No	Tahanan (KΩ)	Tegangan Output (V)
1	0	0
2	0,5	0,48
3	1	0,67
4	1,5	0,98
5	2	1,05
6	2,5	1,23
7	3	1,47
8	3,5	1,68
9	4	1,97
10	4,5	2,02
11	5	2,33
12	5,5	2,57
13	6	2,98
14	6,5	3,01
15	7	3,39
16	7,5	3,58
17	8	3,76
18	8,5	4,02
19	9	4,68
20	9,5	4,75
21	10	4,98

Tabel di atas menunjukkan kemampuan sensor mendeteksi ketinggian level minyak. Kenaikan tahanan pada potensiometer berbanding lurus dengan ketinggian level minyak dan berbanding terbalik dengan tegangan output yang dihasilkan. Tegangan yang dihasilkan dari 5 Volt sampai 0 Volt.

4.3. Pengujian Level Minyak dalam Tangki dengan Indikator

4.3.1. Tujuan

Pengujian ini bertujuan untuk mengetahui banyaknya minyak dalam tangki dengan melihat indikator pada rangkaian yang telah terhubung dengan pelampung dalam tangki.

4.3.2. Alat dan Bahan

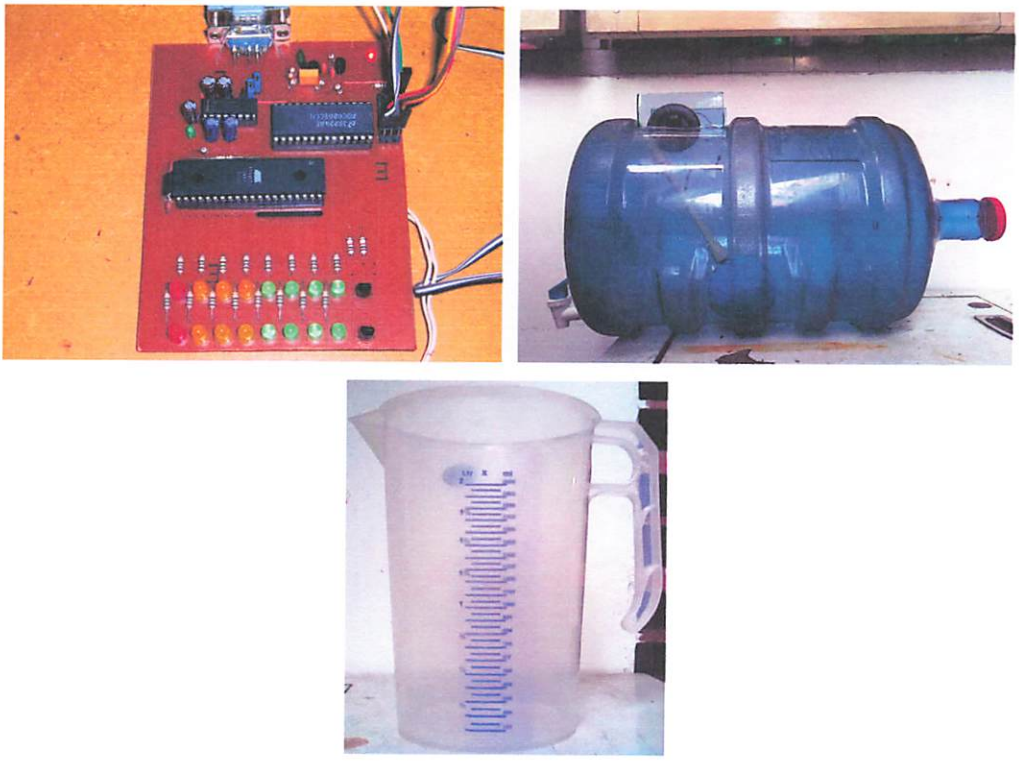
a. Tangki minyak berisi air

b. Perangkat pelampung dan gelas ukur

c. Rangkaian indikator

d. Power supply

4.3.3. Prosedur Pengujian



Gambar 4.2 Rangkaian Minimum Sistem Alat, Model Tangki dan Gelas Ukur

a. Tangki diisi penuh dengan air.

b. Pelampung diapungkan dalam tangki dan dihubungkan dengan rangkaian.

c. Mengaktifkan rangkaian indikator. Mencatatkan hasil pengamatan ke dalam tabel 4.2.

d. Mengurangi isi tangki, sampai terjadi perubahan pada indikator.

e. Mengukur tinggi air dalam tangki. Kemudian mencatatkan datanya ke dalam tabel 4.2.

f. Mengulangi langkah d dan e sampai kondisi indicator tidak berubah.

4.3.4. Hasil Pengujian

Tabel 4.2 Hasil Pengujian Indikator

NDIKATOR	LEVEL MINYAK (berkurang ...ml) Pengambilan Data ke:										
	1	2	3	4	5	6	7	8	9	10	Rata-rata
Hijau 1	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank	Full tank
Hijau 2	2600	2625	2575	2600	2550	2575	2550	2600	2625	2650	2595
Hijau 3	1300	1350	1290	1310	1275	1335	1350	1300	1350	1330	1319
Hijau 4	1350	1325	1315	1315	1380	1375	1300	1275	1340	1350	1332.5
Kuning 1	1250	1200	1320	1300	1250	1250	1320	1225	1260	1250	1262.5
Kuning 2	1450	1400	1500	1400	1475	1400	1350	1500	1425	1400	1430
Kuning 3	2500	2600	2450	2475	2590	2550	2520	2525	2475	2430	2511.5
Merah	5100	5050	5100	5150	5030	5050	5160	5125	5075	5140	5098

4.4. Pengujian ADC

4.4.1. Tujuan

Pengujian ini bertujuan untuk mengetahui kemampuan ADC menerima input dari sensor ketinggian level minyak dan mengubahnya menjadi data input untuk mikrokontroller.

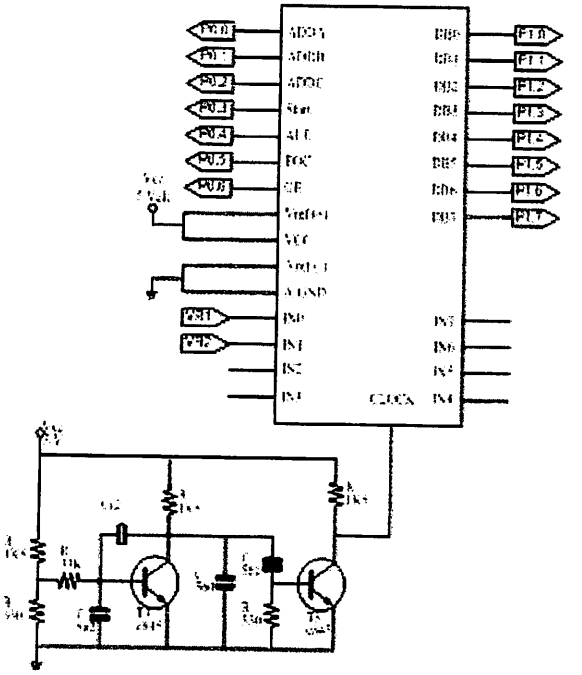
4.4.2. Alat dan Bahan

a. Rangkaian ADC dan LED

d. Power supply DC variabel dan DC 5 Volt

e. SANWA Digital Multimeter CD800

4.4.3. Prosedur Pengujian



Gambar 4.3 Diagram Blok Rangkaian ADC

a. Menyusun rangkaian sesuai dengan blok diagram yang ditunjukkan dalam gambar di atas.

b. Menaikkan tegangan input ADC dimulai dari tegangan 0 Volt sampai dengan 5 Volt.

c. Menghubungkan keluaran 8 bit ADC dengan LED peraga untuk mengetahui keluaran nilai biner dalam mengkonversikan masukan analog ke bentuk keluaran digital.

d. Mengamati hasil pengujian dengan memperhatikan nyala masing-masing LED peraga untuk tiap-tiap masukan analog yang berbeda.

e. Mencatat hasil pengujian pada tabel 4-2.

4.4.4. Hasil Pengujian

Tabel 4.3 Hasil Pengujian Alat

No	Tegangan Input (V)	Output ADC (biner)								HEX	DEC
		Db0	Db1	Db2	Db3	Db4	Db5	Db6	Db7		
1	0,0	0	0	0	0	0	0	0	0	00h	0
2	0,5	0	0	1	0	0	1	0	1	19h	25
3	1	0	1	0	1	0	0	0	0	32h	50
4	1,5	0	1	1	1	0	1	0	1	4Bh	75
5	2	0	1	1	0	1	0	1	1	6Bh	107
6	2,5	0	1	1	1	1	1	0	0	7Ch	124
7	3	1	0	0	1	0	1	1	0	96h	150
8	3,5	1	0	1	1	0	0	0	1	B1h	177
9	4	1	1	0	0	1	0	0	0	C8h	200
10	4,5	1	1	0	1	1	1	1	1	DFh	223
11	5	1	1	1	1	1	1	1	1	FFh	255

Dari tabel hasil pengujian dapat diketahui bahwa ADC 0809 mampu mengkonversi masukan analog menjadi keluaran biner. Hasil konversi tersebut

linier dengan bobot perbitnya yaitu ± 20mV. Dimana setiap perubahan 20mV maka pada output akan mengalami perubahan pula sebesar 1 bit.

Nilai kesalahan menunjukkan nilai penyimpangan data digital keluaran ADC dengan nilai yang sebenarnya. Nilai kesalahan maksimum berdasarkan data yang diberikan ke ADC 0809 adalah sebesar 1 LSB.

Bobot biner 1LSB adalah untuk sistem ini adalah:

$$\text{Resolusi 1Bit} = \frac{V_{ref}}{2^8 - 1} = \frac{5}{255} = 19{,}6mV$$

4.5. Pengujian Mikrokontroller AT89S52

4.5.1. Tujuan

Untuk mengetahui port-port paralel pada mikrokontroler yang digunakan dapat berjalan dengan baik atau tidak, maka dilakukan pengujian mikrokontroller sebagai input dan mikontroller sebagai output.

4.5.2. Alat dan bahan

a. Rangkaian mikrokontroller AT89S52

b. Power supply DC 5 Volt

4.5.3. Prosedur Pengujian

4.5.3.1.Prosedur Pengujian Mikrokontroller sebagai Ouput

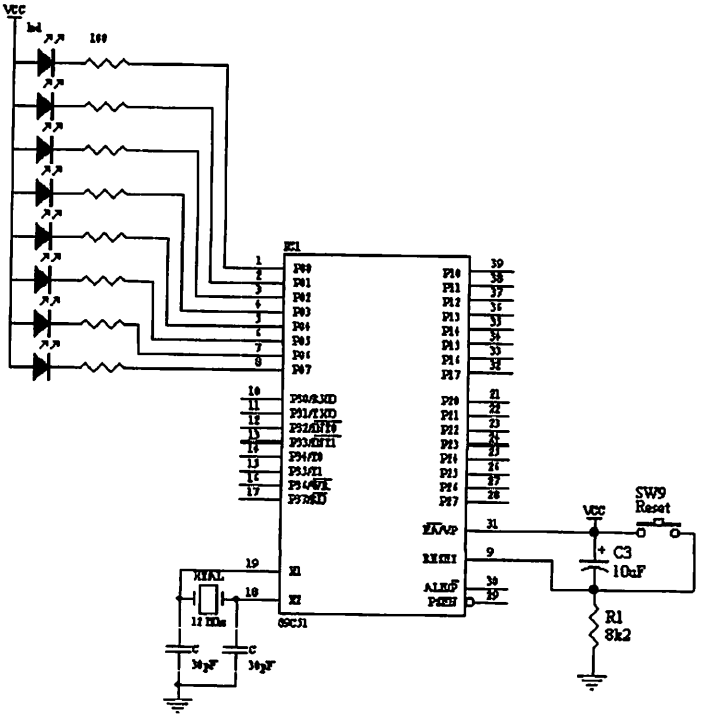
a. Mengisikan listing program berikut ke dalam mikrokontroller yang akan diuji menggunakan program Assembler melalui PC dan downloader.

Listing Program :

```
org 0
start:  mov p1,#00000000b
```

```
        call delay
        mov p1,#11111111b
        call delay
        jmp start
delay:  mov r4,#255
loop1:  mov r3,#255
loop2:  djnz r3,loop2
        djnz r4,loop1
        ret
end
```

b. Merangkai mikrokontroller dan LED seperti berikut



Gambar 4.4 Rangkaian Pengujian Mikrokontroler Sebagai Ouput

4.5.3.2.Prosedur Pengujian Mikrokontroller sebagai Input

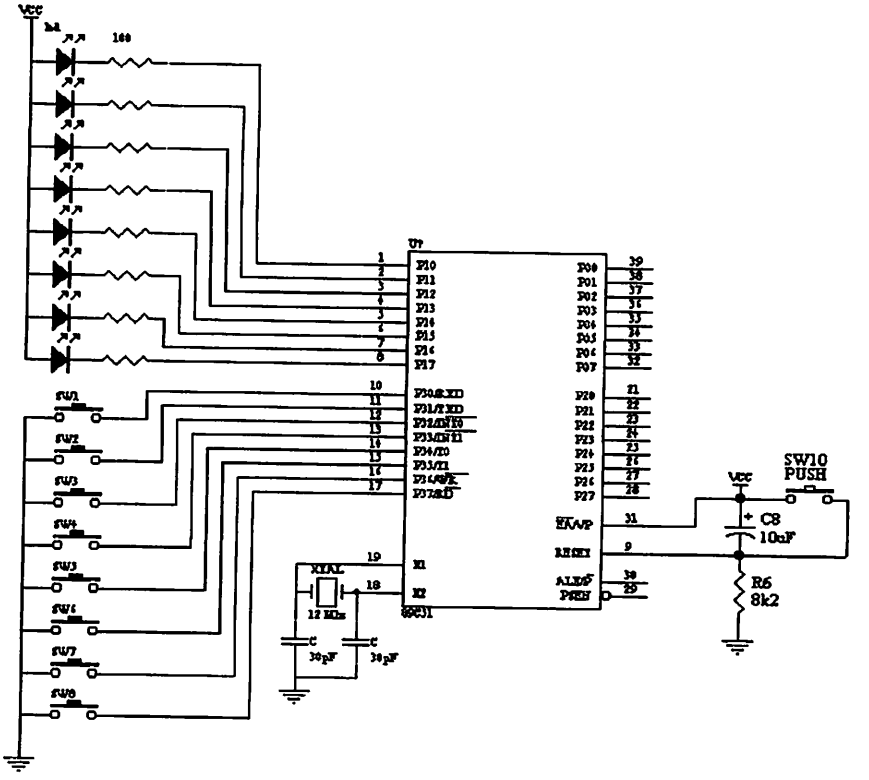
a. Mengisikan listing program berikut ke dalam mikrokontroller yang akan diuji menggunakan program Assembler melalui PC dan downloader.

Listing Program :

```
        org 0h
mulai:  jnb p3.0,nol
        jnb p3.1,satu
        jnb p3.2,dua
        jnb p3.3,tiga
        jnb p3.4,empat
        jnb p3.5,lima
        jnb p3.6,enam
        jnb p3.7,tujuh
        jmp mulai
nol:    mov p1,#11111111b
        jmp mulai
satu:   mov p1,#11111110b
        jmp mulai
dua:    mov p1,#11111101b
        jmp mulai
tiga:   mov p1,#11111100b
        jmp mulai
empat:  mov p1,#11111011b
```

```
        jmp mulai
lima:   mov p1,#11111010b
        jmp mulai
enam:   mov p1,#11111001b
        jmp mulai
tujuh:  mov p1,#11111000b
        jmp mulai
    end
```

b. Merangkai mikrokontroller dan LED seperti berikut



Gambar 4.5 Rangkaian Pengujian Mikrokontroler Sebagai Input

4.5.3.3.Hasil Pengujian Mikrokontroller AT89S52 sebagai Output

Tabel 4.4 Hasil Pengujian Mikrokontroler Sebagai Output

Waktu	Logika Pada P1	L7	L6	L5	L4	L3	L2	L1	L0
1	00000000B	H	H	H	H	H	H	H	H
2	11111111B	M	M	M	M	M	M	M	M

Keterangan : - H : Hidup
- M : Mati

4.5.3.4.Hasil Pengujian Mikrokontroller AT89S52 sebagai Input

Tabel 4.5 Pengujian Mikrokontroler Sebagai Input

Kondisi	Led
Tekan P3.0	11111111
Tekan P3.1	11111110
Tekan P3.2	11111101
Tekan P3.3	11111100
Tekan P3.4	11111011
Tekan P3.5	11111010
Tekan P3.6	11111001
Tekan P3.7	11111000

Keterangan : 1 = mati
0 = hidup

4.6. Pengujian Penerimaan Data

Pengujian ini sekaligus menguji kabel serial, line telepon, modem, dan PC. Pada tahap pengujian ini, dilakukan dua (2) macam metode penerimaan data, yaitu dengan menggunakan hyperteminal serta program yang dibuat.

4.6.1. Tujuan

Tujuan pengujian sistem secara menyeluruh ini adalah untuk mengetahui apakah sistem yang dirancang telah bekerja sesuai yang diinginkan.

4.6.2. Alat dan Bahan

a. Rangkaian hardware sistem

b. PC

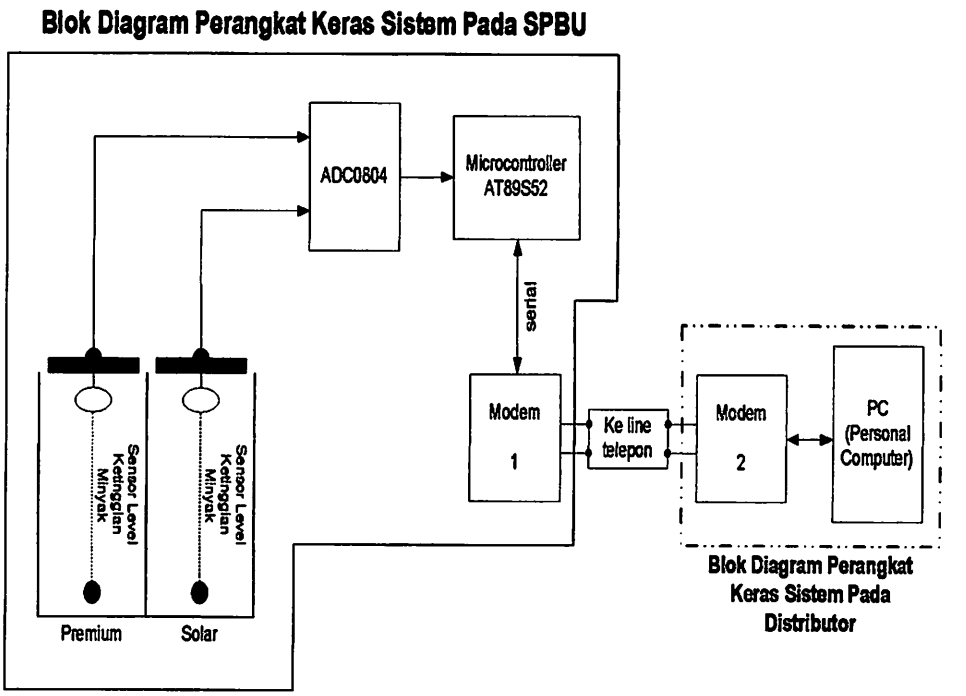
c. Modem

d. Kabel serial

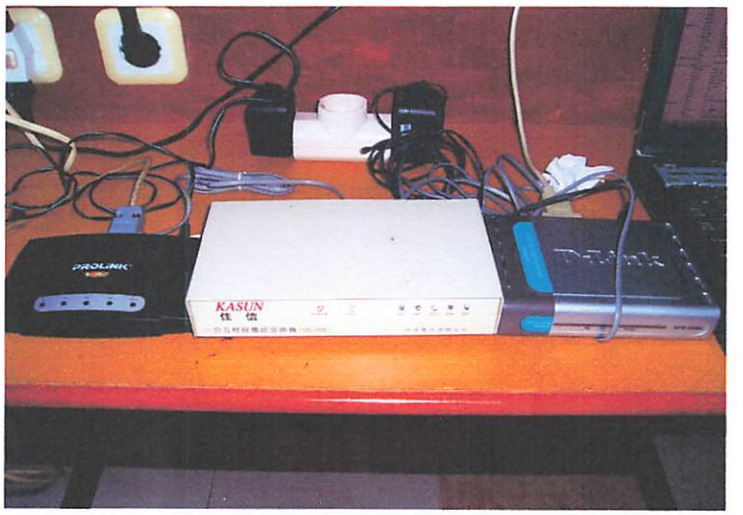
e. PABX (sebagai media pengganti line telepon)

4.6.3. Prosedur Pengujian

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



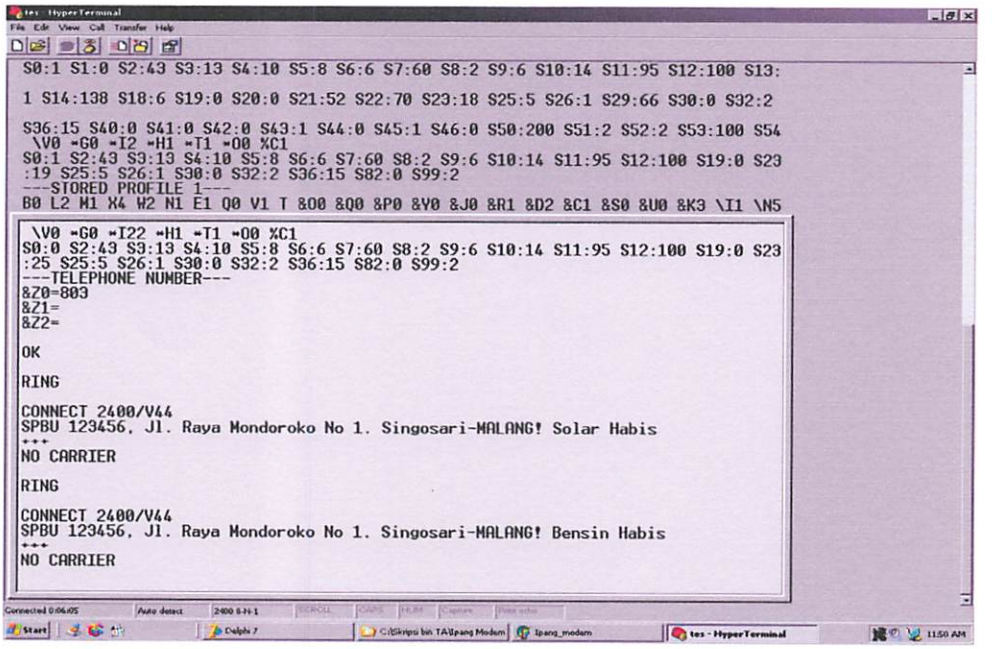
Gambar 4.6 Blok Diagram Keseluruhan Rangkaian



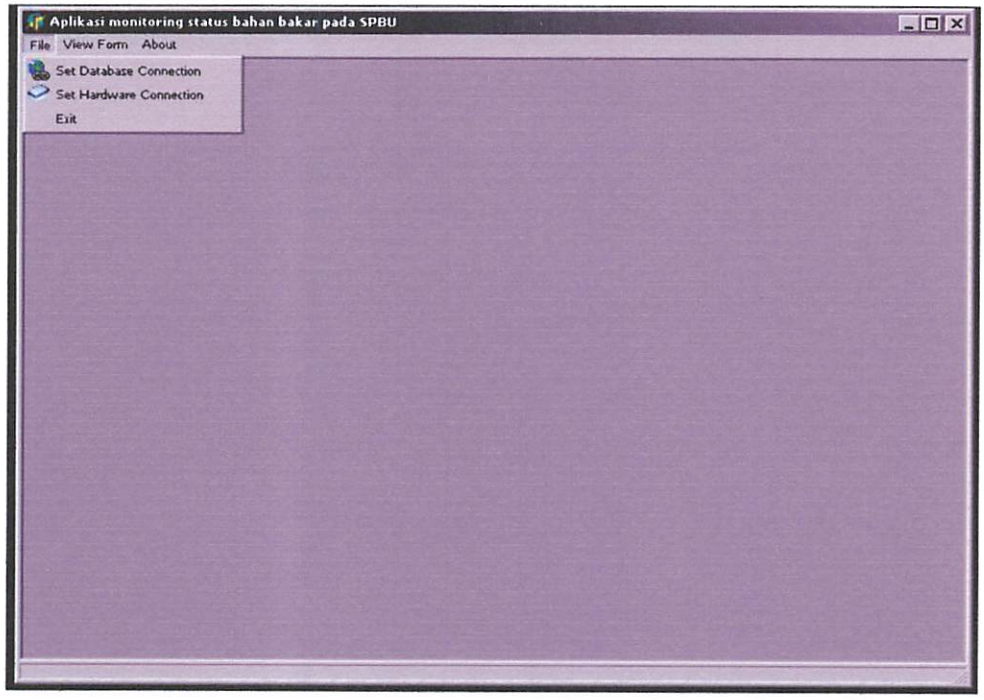
Gambar 4.7 Foto Modem Dan PABX Sesuai Blok Diagram

2. Proses kedua adalah dengan menjalankan program.
3. Setelah program Delphi dengan hardware siap digunakan maka Proses ketiga dilakukan pengujian keseluruhan sistem. Dengan melakukan proses identifikasi.

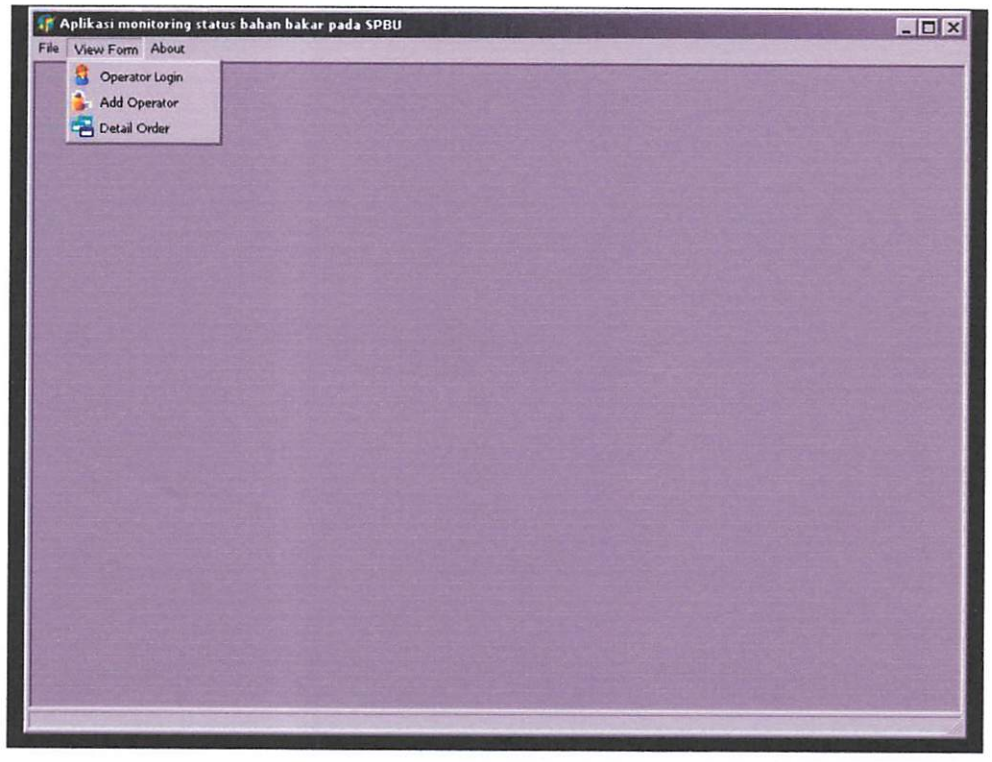
4.6.4. Hasil Pengujian



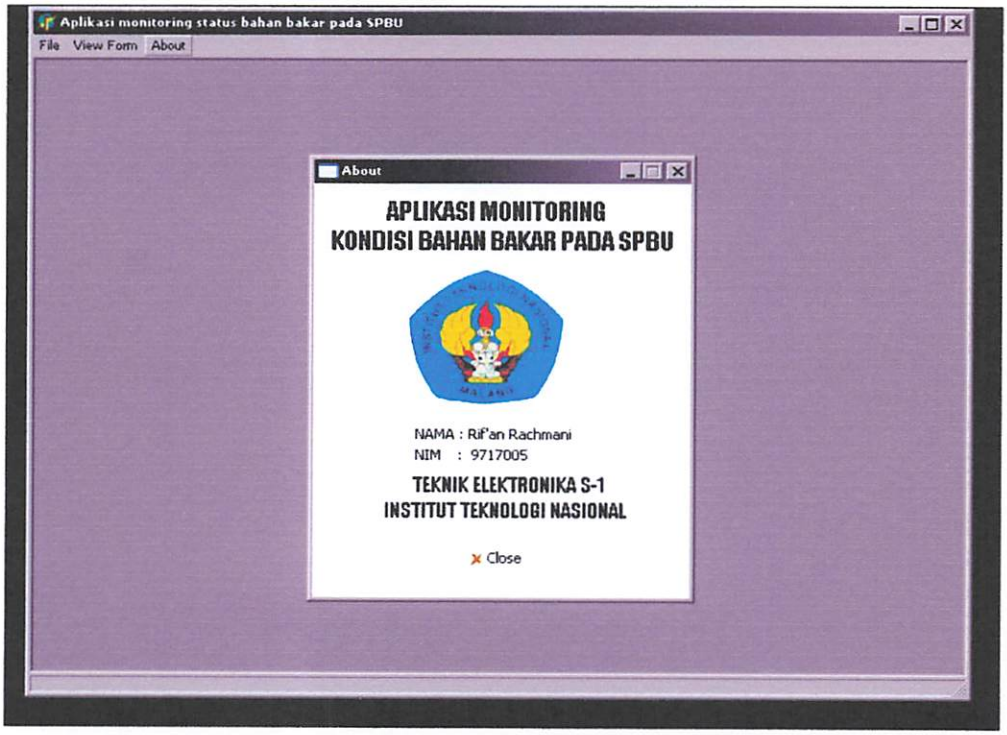
Gambar 4.8 Tampilan pada Hyperterminal



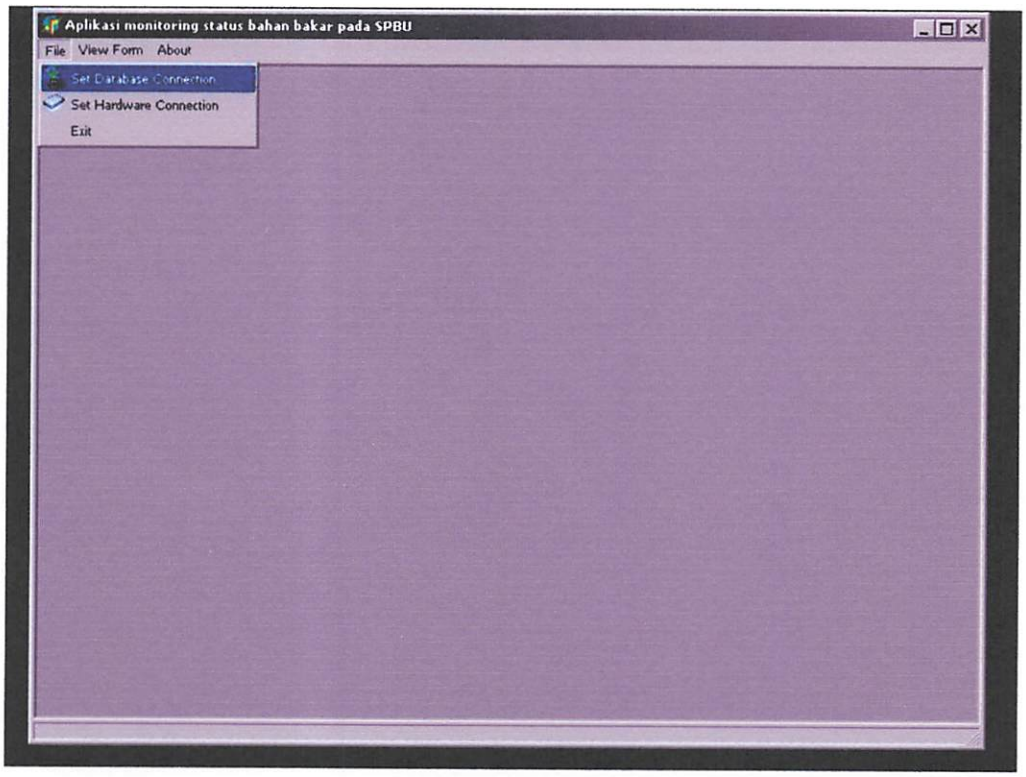
Gambar 4.9 Tampilan Menu "File"



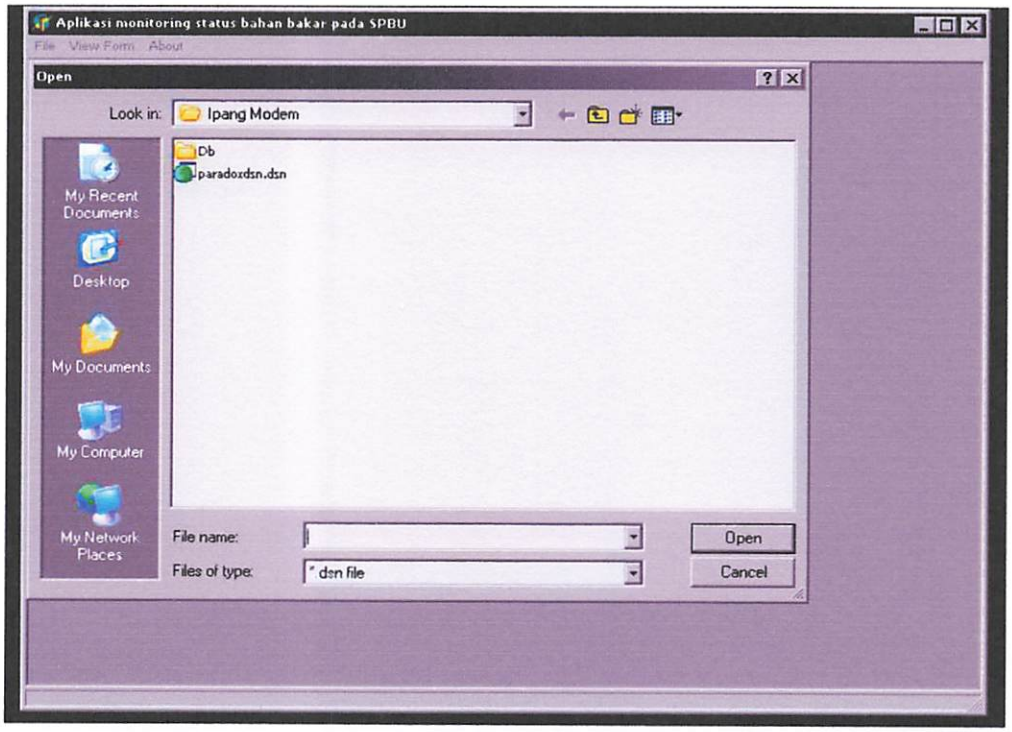
Gambar 4.10 Tampilan Menu "View Form"



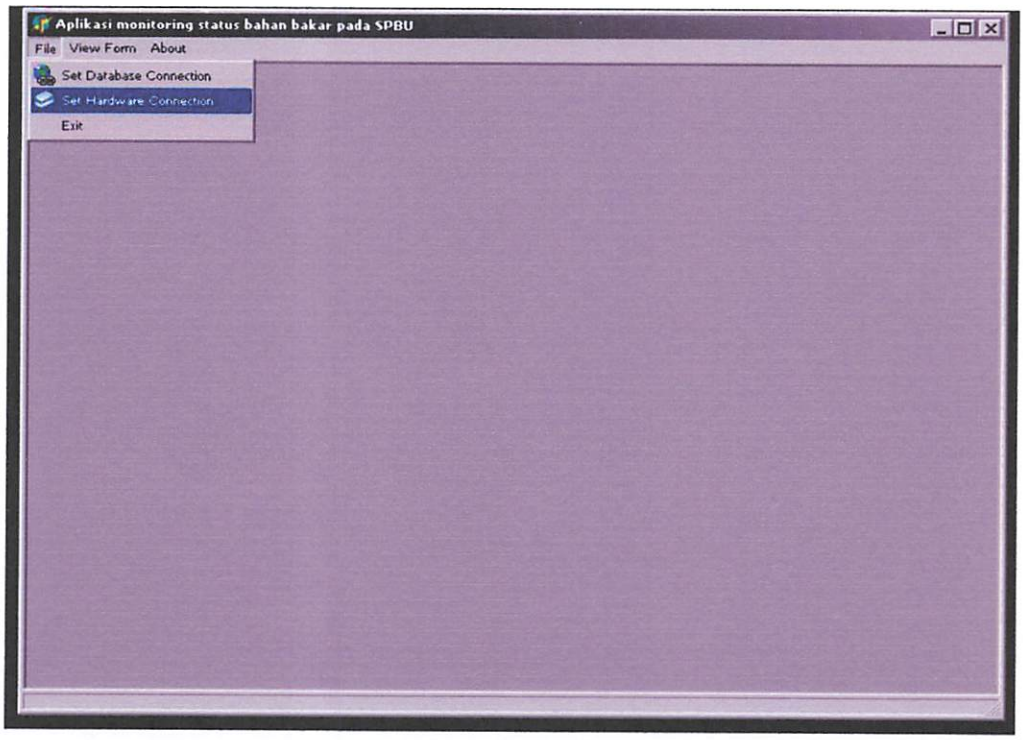
Gambar 4.11 Tampilan Menu "About"



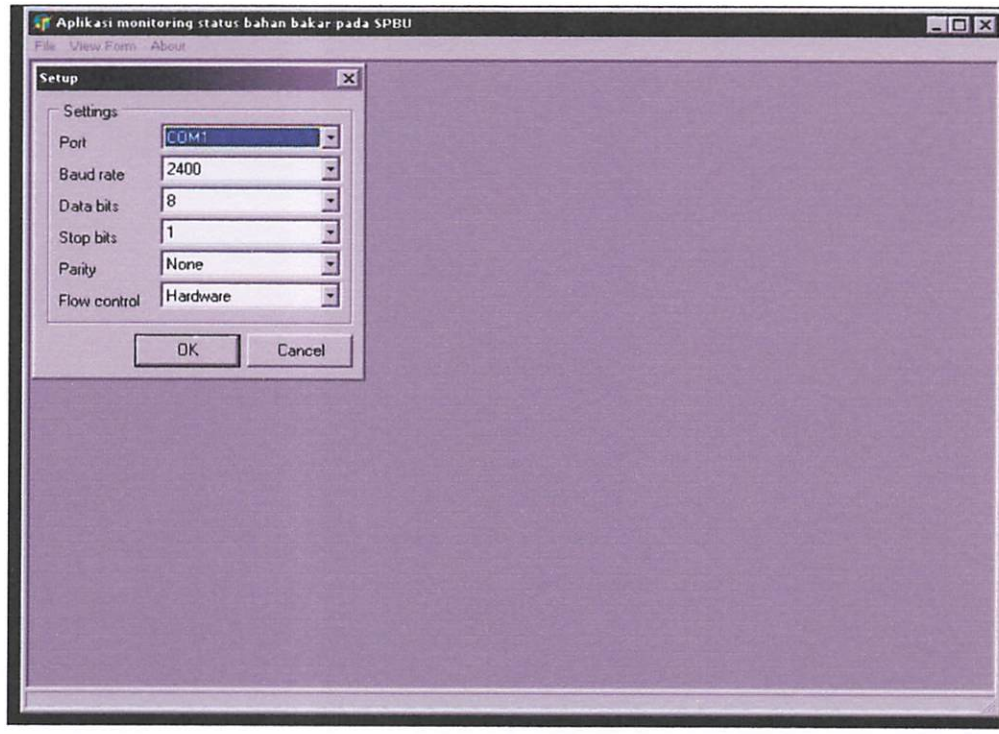
Gambar 4.12 Sub Menu "Set Database Connection"



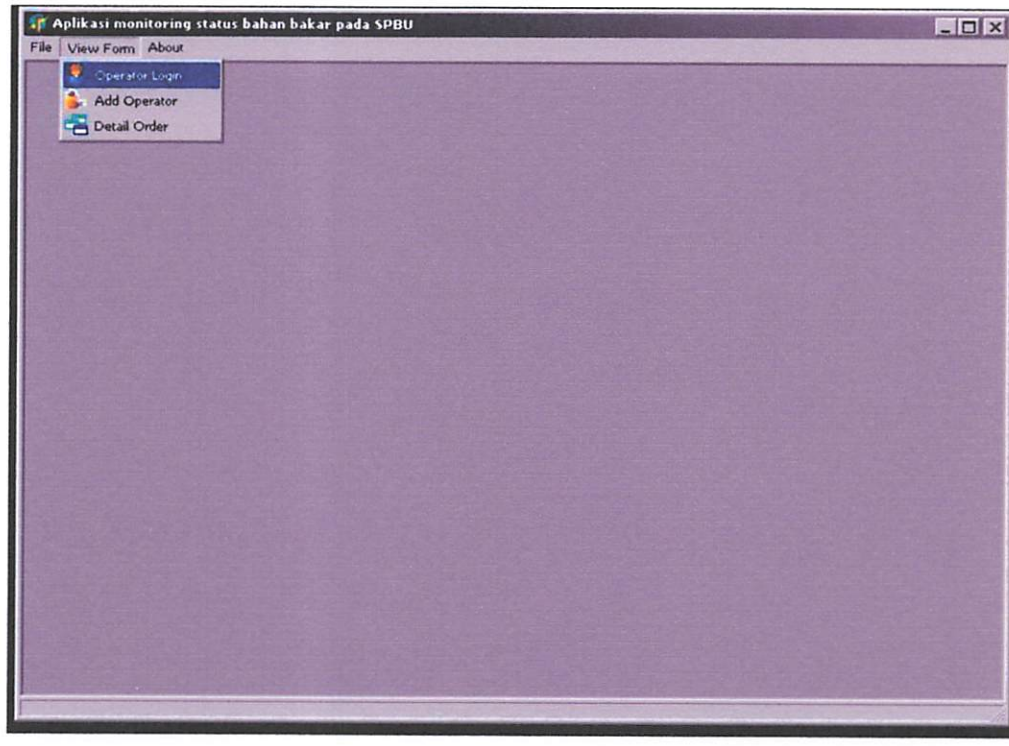
Gambar 4.13 Sub Menu "Open"



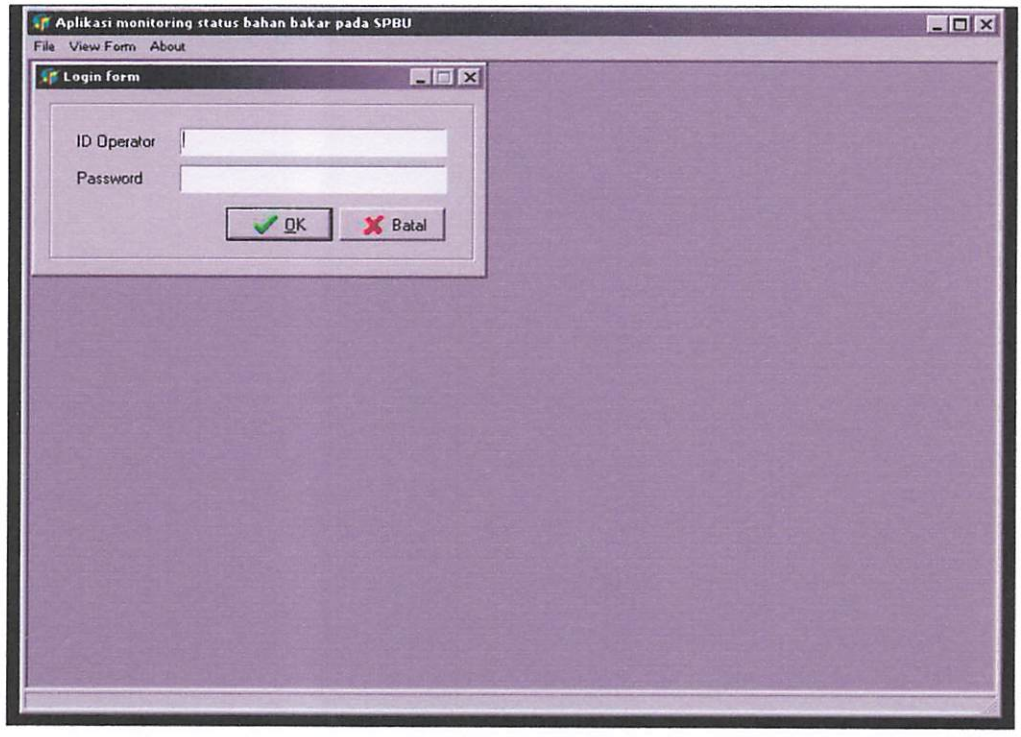
Gambar 4.14 Sub Menu "Set Hardware Connection"



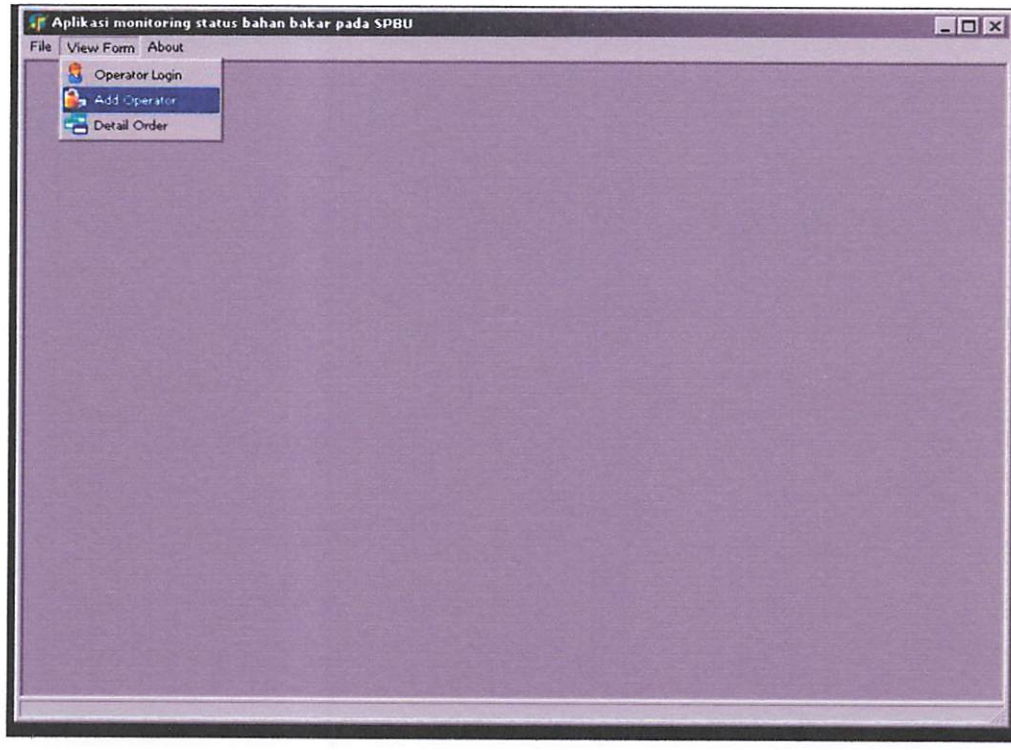
Gambar 4.15 Sub Menu "Setup"



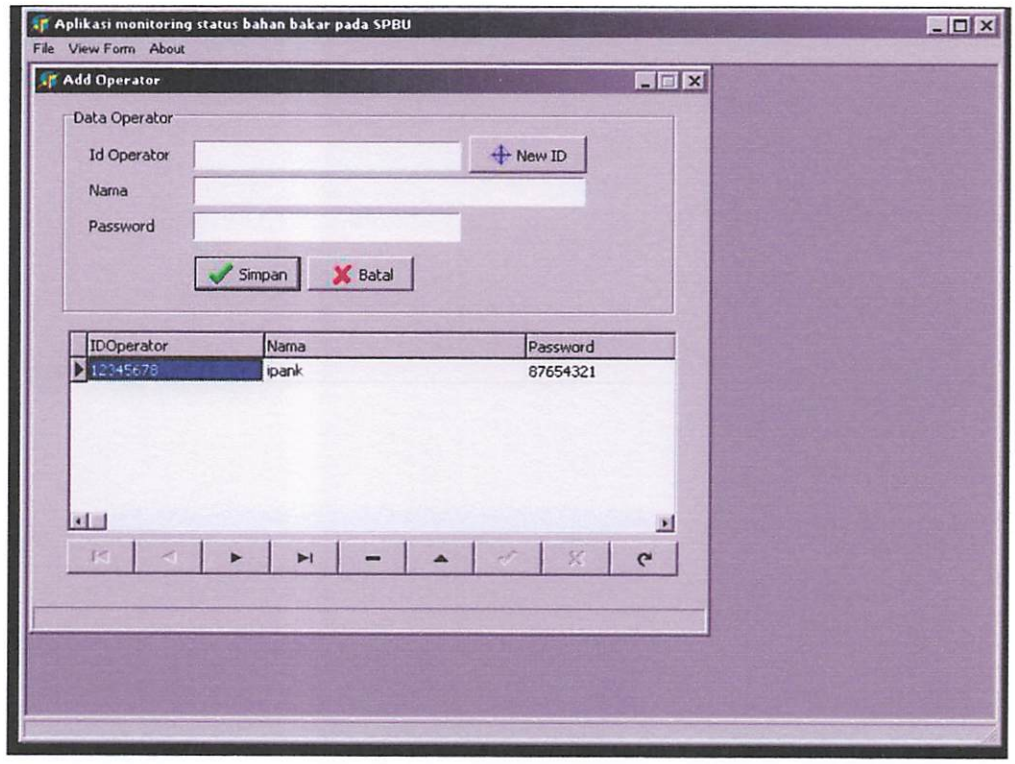
Gambar 4.16 Sub Menu "Operator Login"



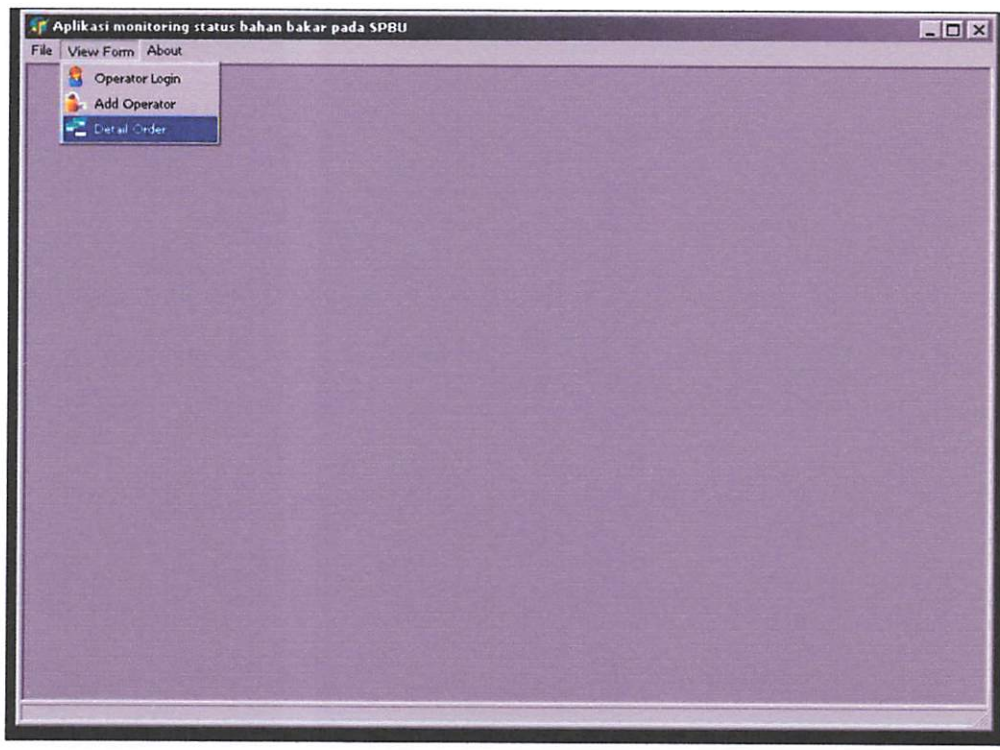
Gambar 4.17 Sub Menu "Login Form"



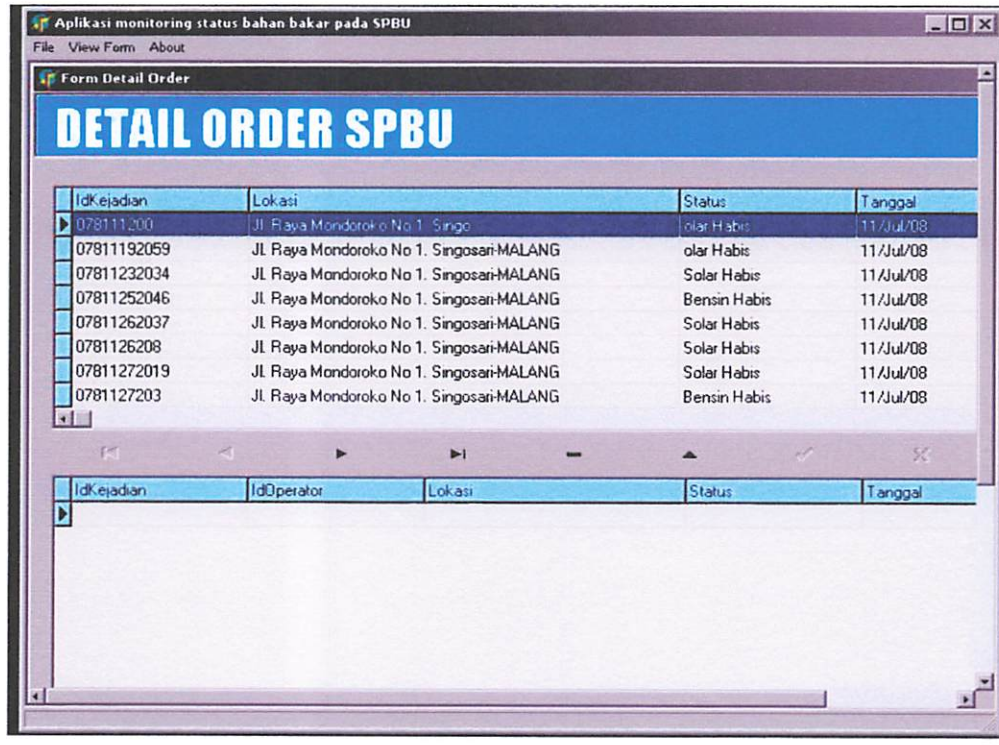
Gambar 4.18a Sub Menu "Add Operator"



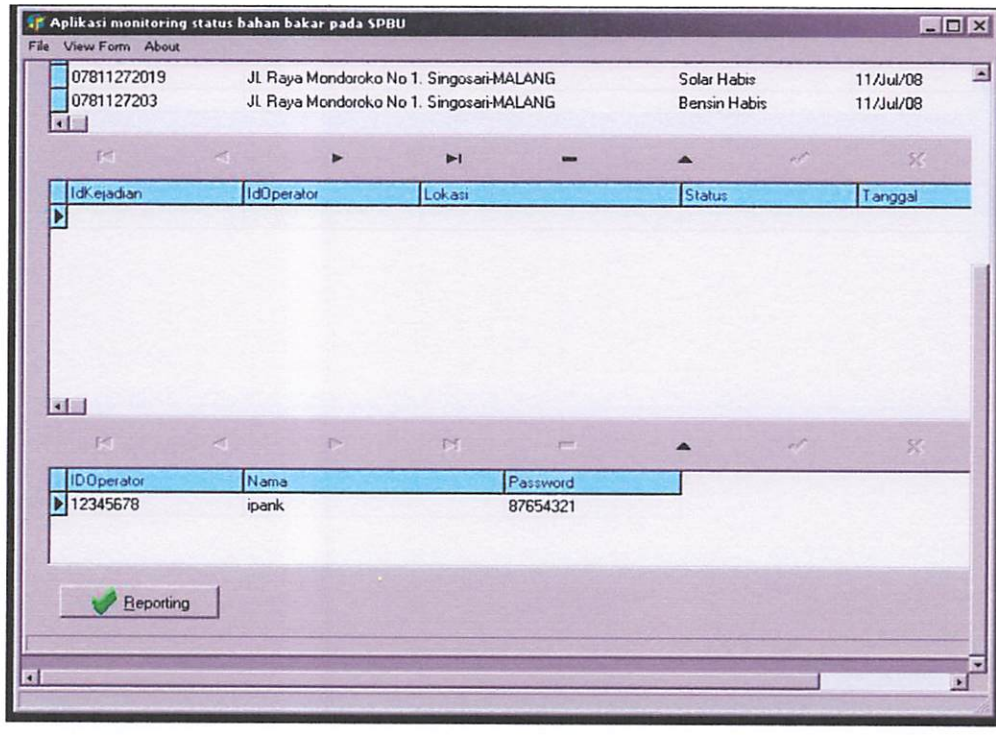
Gambar 4.18b Sub Menu "Add Operator 2"



Gambar 4.19 Sub Menu "Detail Order"

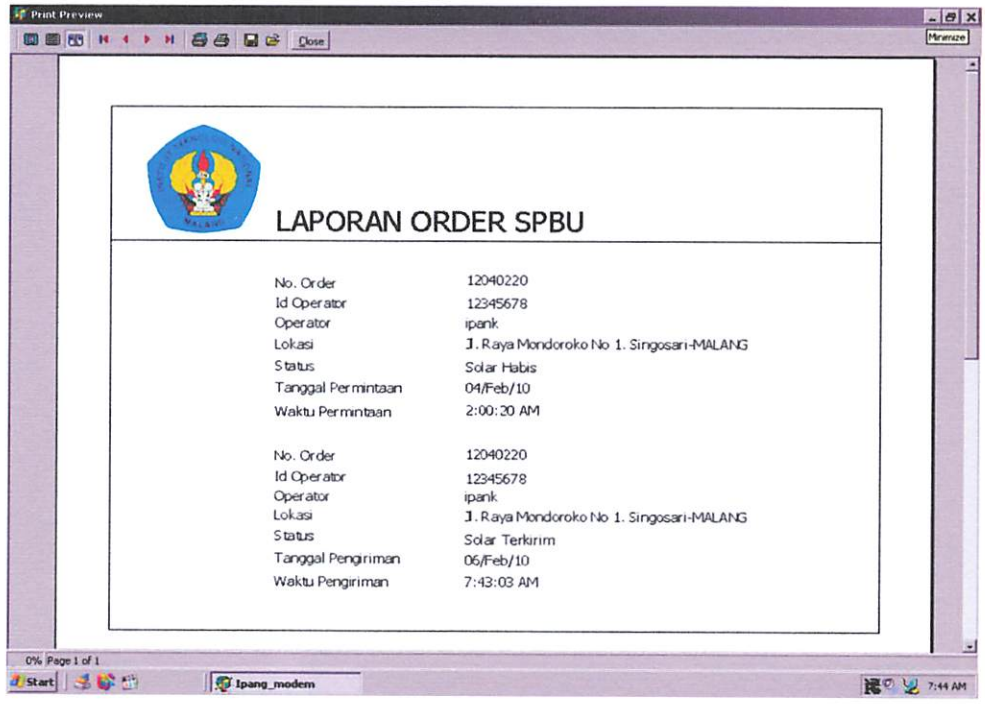


Gambar 4.20a Sub Menu "Detail Order SPBU"



Gambar 4.20b Sub Menu Sub Menu "Detail Order SPBU"

Jika user menghendaki data tersebut dicetak, maka user dapat mengklik Reporting. Dan hasilnya adalah preview di PC seperti berikut.



Gambar 4.21 Tampilan Laporan Order SPBU

4.7. Error Pengujian Penerimaan Data

Pengujian penerimaan data melalui hyperterminal maupun software seperti di atas dilakukan tidak hanya satu kali. Dari beberapa kali pengujian diperoleh data yang tidak selalu sama persis, sehingga timbul selisih yang disebut sebagai error.

4.7.1. Error Pada Hyperterminal

Tabel berikut adalah hasil pengujian penerimaan data sebanyak 20 kali. Data ini diperoleh dari penerimaan data memalui hyperterminal.

Tabel 4.6 Hasil Pengujian Dengan Hyperterminal

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Bensin	√	√	√	√	√	√	x	√	√	√	√	√	√	√	√	√	√	x	√	√
Solar	√	√	√	√	x	√	√	√	x	√	√	√	√	√	√	√	√	√	√	√
Keduanya	√	x	√	√	√	√	√	√	√	√	√	√	x	√	x	√	√	√	√	√

Keterangan : - √ : Data Di Terima
- X : Data Tidak Diterima Atau Salah

Dari tabel di atas, diketahui bahwa persentase kesalahan kemampuan alat yang dibuat ini adalah:

$$\frac{\text{Total Uji} - \text{Total Data Yang Diterima}}{\text{Total Uji}} \times 100\%$$

$$= \frac{60-53}{60} \times 100\% = \frac{7}{60} \times 100\% = 11.67\%$$

4.7.2. Error Pada Software

Tabel berikut adalah hasil pengujian penerimaan data dengan menggunakan software sebanyak 20 kali.

Tabel 4.7 Hasil Pengujian Dengan Software

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Bensin	√	√	√	√	√	√	√	√	√	√	√	√	x	√	√	x	√	√	√	√
Solar	√	√	√	√	√	√	√	x	√	√	x	√	√	√	√	√	√	x	√	√
Keduanya	√	√	√	√	x	√	√	√	√	√	√	√	√	√	√	x	√	√	√	√

Keterangan : - √ : Data Di Terima
- X : Data Tidak Diterima Atau Salah

Dari tabel di atas, diketahui bahwa persentase kesalahan kemampuan alat yang dibuat ini adalah:

$$\frac{\text{Total Uji} - \text{Total Data Yang Diterima}}{\text{Total Uji}} \times 100\%$$

$$= \frac{60-53}{60} \times 100\% = \frac{7}{60} \times 100\% = 11.67\%$$

BAB V

PENUTUP

5.1. Kesimpulan

Dari perancangan dan pengujian alat di atas, dapat disimpulkan bahwa:

1. Kecepatan baudrate yang digunakan untuk menghubungkan mikrokontroller dengan PC melalui modem adalah 2400Kbps.
2. Dari hasil pengujian, pemakaian pelampung yang terhubung dengan potensiometer sangat mempengaruhi data input alat dan hasil akhir pada PC.
3. Beberapa bahan pelampung yang akan diletakkan di dalam tangki perlu dites dan diamati lebih detil karena kondisi bahan bakar, suhu dan tekanan dalam tangki cukup mempengaruhi besarnya input bagi ADC.
4. Pada pengujian masih ditemukan kesalahan dalam penerimaan data melalui HyperTerminal dengan persentase sebesar 11,67%.
5. Pada pengujian penerimaan data melalui software juga ditemukan kesalahan dengan persentase sebesar 11,67%.
6. Dari pengujian yang dilakukan, kesalahan penerimaan data pada PC disebabkan oleh keterbatasan kemampuan alat komunikasi yang digunakan.

5.2. Saran

1. Jumlah tandon yang digunakan, masih dapat dikembangkan. Karena masih adanya port kosong pada ADC.
2. Kemampuan alat ini masih bisa dikembangkan lagi, yaitu mendeteksi level ketinggian minyak pada level-level tertentu (sesuai keinginan), misalnya pada level 25% atau 50%. Pengembangan ini bermanfaat untuk distributor dalam hal prioritas pengiriman bahan bakar.

DAFTAR PUSTAKA

[1] Agfianto Eko Putra, 2002, *Belajar Mikrokontroller AT89C51/52/55*

[2] S.Pd. Agus Sugiharto, 2002, *Penerapan Dasar Transducer dan Sensor*, Kanisius

[3] As Pratisto (2000) Beginilah Cara Kerja Modem, Chip Magazine, June 2000, pp 128. Cakrawala Gintings (2004) Apa Itu Modem dan Bagaimana Cara Kerjanya?, PC-Plus Tabloid, Volume 177/V, 25-31 May 2004, pp 5. Deny (2001) Kendala DSL Masih Seputar Infrastruktur yang Buruk, [http://www.infokomputer.com/actual/actual.php?id=2306] , acc: 28 July 2004 Gary Kessler and Walter Goralski (1997) An Overview of 56-kbps Modem Technology, [http://www23.brinkster.com/suseno/tampilpesan.asp?id=345&cari=modem], acc: 30 July 2004 Ron White (1999) How Computers Work, Millennium Edition, Que Corporation, Indianapolis, USA. Trezciak, J & Mackay, S E (1994) Study Skills for Academic Writing, Prentice Hall International , UK ………..(2001) Sudah Maksimalkah Kerja Modem Saat Ini?, [http://www.infokomputer.com/aktual/aktual.php?id=2089], acc: 28 July 2004 ………….(2002) Memilih Modem, [http://rahsia.20m.com/cgibin/ad/in line?page=modem.htm&Rtime=4777], acc: 28 July 2004. ………..(2002) Cara Kerja DSL, [http://www.infokomputer.com/inilah/inilah.php?id=2818] Acc: 30 July 2004.
::: SAFRI LUBIS WEBSITE ::: http://safri-lubis.info/file Powered by Joomla! Generated: 11 January, 2009, 22:33

[4] Datasheet ADC 0809 dan Datasheet ADC 0804

[5] Datasheet AT89S52, http://www.atmel.com

[6] Datasheet MAXIM RS-232, http://www.maxim-ic.com

[7] Delta Electronics, http://www.delta-electronic.com

[8] Malvino, 1999, *Dasar-dasar Elektronika*

[9] Muhammad Safri Lubis, 25 September 2004, Medan, *Surat Kabar Harian Analisa*

[10] Romy Budhi Widodo & Joseph Dedy Irawan, 2007, *Interfacing Paralel & Serial Menggunakan DELPHI*

[11] http://safri-lubis.info/file Powered by Joomla! Generated : 11 January, 2009, 22:33

[12] Wikipedia Foundation, http://en.wikipedia.org

[13] Yuwono, 1999, *Dasar-dasar Mikrokontroller*

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

FORMULIR PERBAIKAN SKRIPSI

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

Hari : Selasa

Tanggal : 09 – 02 – 2010

Telah dilakukan perbaikan Skripsi oleh mahasiswa,

Nama : RIF'AN RACHMANI

NIM : 97.17.005

Jurusan : T. ELEKTRO

Masa Bimbingan : 13 – 01 – 2010 s/d 13 – 07 – 2010

Judul : Perencanaan dan Pembuatan Alat Untuk Mendeteksi Volume Minyak Pada Tandon SPBU Yang Di Transmisikan Via Modem Berbasis Mikrokontroller AT89S52 Dan Terintegrasi Dengan PC

	Materi Perbaikan	Paraf
Penguji 1	1. Perbaikan Kesimpulan 2. Penambahan 2 (Dua) Kondisi 3. Pengelompokkan Listing Program Kedalam Lampiran	
Penguji 2	1. Kalimat Pengantar Pada Gambar Dan Tabel 2. Urut Abjad Untuk Daftar Pustaka 3. Data Shet ADC 4. Data Shet Mikrokontroller 5. Perencanaan Kristal 6. Perencanaan Driver LED	

Dosen Penguji I

Ir. Eko Nurcahyo
NIP.Y.1028700172

Dosen Penguji II

M. Ibrahim Ashari, ST. MT.
NIP.P.1030100358

Mengetahui,
Dosen Pembimbing

Ir. Yusuf Ismail Nakhoda, MT.
NIP.Y.1018800189

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

FORMULIR PERBAIKAN SKRIPSI

[illegible]

Dosen Penguji I

[illegible]

Dosen Penguji II

[illegible]

Mengetahui,
Dosen Pembimbing

[illegible]

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 : Rifan Rachmani
NIM : 9717005
Perbaikan meliputi :

1. Halaman Pengantar pd gbr dan tabel !
2. Urut abjad / Daftar Pustaka.
3. Data sheet ADC. → 606,06 KHz
   Data sheet micro
4. Rancangan lengkap ditambahkan.
5. Rancangan ~~dr~~ Driver Lcd ditambahkan

Malang, 9 Feb '10

(M. Ibrahim A, 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 : RIPAH RACHMATI
NIM : 9717005
Perbaikan meliputi :

- Perbaikan kesimpulan
- Ditambahkan 2 kondisi
- Listing program jadikan satu di lampiran.

Malang,

(_____)

INSTITUT TEKNOLOGI NASIONAL
JL. RAYA KARANGLO, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : RIF'AN RACHMANI
Nim : 97 17 005
Masa Bimbingan : 13 JANUARI 2010 s/d 13 JULI 2010
Judul Skripsi : PERENCANAAN DAN PEMBUATAN ALAT UNTUK MENDETEKSI VOLUME MINYAK PADA TANDON SPBU YANG DITRANSMISIKAN VIA MODEM BERBASIS MIKROKONTROLLER AT89S52 DAN TERINTEGRASI DENGAN PC

NO	TANGGAL	URAIAN	PARAF PEMBIMBING
1.	6 Januari 2010	BAB I : Pembenahan Sistematika Pembahasan	
2.	9 Januari 2010	BAB II : Perbaikan Urutan Pembahasan Teori	
3.	14 Januari 2010	BAB III : Perbaikan Blok Diagram	
4.	15 Januari 2010	BAB III : Perbaikan Flowchart	
5.	17 Januari 2010	BAB IV : Perbaikan Urutan Gambar Tampilan Software	
6.	18 Januari 2010	BAB V : Perbaikan Isi Kesimpulan Dan Saran	
7.	22 Januari 2010	Perbaikan Penulisan Daftar Pustaka	
8.	29 Januari 2010	BAB II : Perbaikan Urutan Nomer Saduran Dari Daftar Pustaka	
9.			
10.			

Malang,
Dosen Pembimbing

IR. YUSUF ISMAIL NAHKODA, MT
NIP.Y.1018800189

Form. S-4a

Lampiran 1

Listing Program pada Mikrokontroller:

```
ADC_A                   bit     P0.0
ADC_B                   bit     P0.1
ADC_C                   bit     P0.2
ADC_START       bit     P0.3
ADC_ALE         bit     p0.4
ADC_EOC                 bit     P0.5
ADC_OE                  bit     P0.6
b_data          equ     p1
udahkirim_bensin        bit     20h.0
udahkirim_solar         bit     20h.1
udahCONNECT             bit     20h.2
udahkirim_solarbensin   bit     20h.3
rtimerbensin    EQU     30H
rtimersolar     EQU     31H
rtimerbensinsolar       EQU     32H
Rtunggu1        equ     33h     ; untuk variabel time out
Rtunggu2        equ     34h
Rtunggu3        equ     35h
rulangcon       equ     36h
displaybensin   equ     62h
displaysolar    equ     63h
    Org     0h
      clr   udahkirim_bensin
      clr   udahkirim_solar
      clr   udahkirim_solarbensin
      clr   udahCONNECT
      JMP   MULAI
ORG 23H
```

```
        CLR ES
        PUSH ACC
        MOV  A,SBUF       ; MENERIMA DATA SERIAL
        JNB  RI,$
        CLR  RI
        CJNE A,#'T',LANJUT
        SETB udahCONNECT
LANJUT: POP  ACC
        SETB ES
        RETI

MULAI:  CALL inisialisasi_serial
        MOV  rulangcon,#1
        clr  ADC_OE
        clr  ADC_START
        clr  ADC_ale
Mulai1:
; BACA BENSIN
        CLR  ADC_A        ; chanel 0
        CLR  ADC_B
        CLR  ADC_C
        call ADC0809
        mov  60h,A ; hasil pebacaan adc
        mov  b,#4  ; konversi ke bar level
        div  ab
        INC  A
        MOV  R3,A
        mov  a,#11111111b
TMP1:   clr  c
        RLC  A
        DJNZ R3,TMP1
```

```
        MOV  displaybensin,A
 ; BACA SOLAR
        SETB  ADC_A  ; chanel 1
        CLR   ADC_B
        CLR   ADC_C
        call    ADC0809
        mov   61h,A  ; hasil pebacaan adc
        MOV  B,#4    ; konversi ke bar level
        DIV   AB
        INC   A
        MOV  R2,A
        mov   a,#11111111b
 TMP2:  clr    c
        RLC   A
        DJNZ  R2,TMP2
        MOV  displaysolar,A  ; akhir konversi
        MOV  A,displaybensin
        cjne   a,#11111110b,tdkkirim1                ; JIKA BENSIN KURANG
        MOV  A,displaysolar
        cjne   a,#11111110b,kirimbensinhabis                ; JIKA SOLAR KURANG
; INI KIRIM KEDUAANYA
        jb      udahkirim_solarbensin,tdkkirim3_JUGA     ; JIKA UDAH DIKIRM
        MOV  rtimerbensinsolar,#60
        setb    udahkirim_solarbensin
        call    kirimlaporan3
        JMP    tdkkirim3_JUGA
tdkkirim3:
        CLR    udahkirim_solarbensin
        JMP    TAMPILKANKONDISI
tdkkirim3_JUGA:
        DJNZ  rtimerbensinsolar,SSSSS
```

```
        CLR   udahkirim_solarbensin
SSSSS:  JMP   TAMPILKANKONDISI
; ini kirim bensin aja
kirimbensinhabis:
        jb    udahkirim_bensin,tdkkirim1_JUGA  ; JIKA UDAH DIKIRM
        MOV   rtimerbensin,#60
        setb  udahkirim_bensin
        call  kirimlaporan1
        JMP   tdkkirim1_JUGA
tdkkirim1:
        CLR   udahkirim_bensin
        clr   udahkirim_solarbensin
        JMP   UJISOLAR
tdkkirim1_JUGA:
        DJNZ  rtimerbensin,TAMPILKANKONDISI; UJISOLAR
        CLR   udahkirim_bensin
        clr   udahkirim_solarbensin
        JMP   TAMPILKANKONDISI
UJISOLAR:
        MOV   A,displaysolar
        cjne  a,#11111110b,tdkkirim2
        jb    udahkirim_solar,tdkkirim2_JUGA
        MOV   rtimersolar,#60
        setb  udahkirim_solar
        call  kirimlaporan2
        JMP   tdkkirim2_JUGA
tdkkirim2:
        CLR   udahkirim_solar
        clr   udahkirim_solarbensin
        JMP   TAMPILKANKONDISI
tdkkirim2_JUGA:
```

```
        clr     udahkirim_solarbensin
        DJNZ  rtimersolar,TAMPILKANKONDISI
        CLR   udahkirim_solar
        clr     udahkirim_solarbensin
 TAMPILKANKONDISI:
        cpl     p3.5
        MOV  R2,#255
 TAMPIL1:
        MOV  A,displaybensin
        MOV   P2,A
        SETB   P3.4
        clr    p3.3
      djnz   r3,$
        SETB   P3.3
        MOV  A,displaysolar
        MOV   P2,A
        SETB   P3.3
        clr    p3.4
      djnz   r3,$
        SETB   P3.4
        DJNZ  R2,TAMPIL1
        jmp    Mulai1          ;
Delaylama: Mov      79h,#090h     ;Isi RAM 79h dengan 0
DelayL:  djnz 7ah,$
        djnz   7ah,$
        Djnz  79h,DelayL    ;
        Ret                 ;Kembali ke langkah setelah perintah Acall
Delay: Mov    79h,#090h     ;Isi RAM 79h dengan 0
Delay1:
       Djnz   79h,Delay1    ;
       Ret                  ;Kembali ke langkah setelah perintah Acall
```

```
ADC0809:
next_sampling:
        setb    ADC_START
        setb    ADC_ale
        nop
        clr     ADC_START
        clr     ADC_ale
not_EOC:
        jnb     ADC_EOC,not_EOC
        djnz    R2,$
        djnz    R2,$
        djnz    R2,$
        setb    ADC_OE      ; Baca Data melalui P3
        djnz    R2,$
        mov     c,b_data.7
        rRc     a
        mov     c,b_data.6
        rRc     a
        mov     c,b_data.5
        rRc     a
        mov     c,b_data.4
        rRc     a
        mov     c,b_data.3
        rRc     a
        mov     c,b_data.2
        rRc     a
        mov     c,b_data.1
        rRc     a
        mov     c,b_data.0
        rRc     a
        ret
```

```
KIRIM_SERIAL:
        CLR   ES
        MOV   SBUF,A
        JNB   TI,$
        CLR   TI
        DJNZ   R7,$  ; TUNDA
        SETB  ES
        RET
inisialisasi_serial:
        setb    EA
        mov     87h,#0h
        mov     TMOD,#20h
;       mov     TH1,#0F4H    ; 2400
;       mov     TH1,#0FAH    ; 4800
        mov     TH1,#0FDH    ; 9600
        setb    TR1
        mov     SCON,#50h
        setb    es
RET
KE_RET:
        RET
kirimlaporan1:
;       DJNZ    rulangcon,KE_RET  ; UTUK DELAY PENGULANGAN
        MOV   rulangcon,#1
        clr     udahCONNECT
        mov     dptr,#nomortujuan
        call    kirimkalimat
        CALL DELAY
        CALL DELAY
        mov     rtunggu1,#255
        mov     rtunggu2,#255
```

```
        mov    rtunggu3,#100
TUNGGUCON1:
        djnz   Rtunggu1,tggc1
        djnz   Rtunggu2,tggc1
        djnz   Rtunggu3,tggc1
        mov    rulangcon,#50
        JMP    L_ATH1
tggc1:  JNB    udahCONNECT,TUNGGUCON1
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov    dptr,#lokasi
        call   kirimkalimat
        mov    dptr,#laporan1
        call   kirimkalimat
L_ATH1:      mov    dptr,#plus3     ; KIRIM ATH
        call   kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov    dptr,#ath
        call   kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
KE_RET1:ret
kirimlaporan2:
;       DJNZ   rulangcon,KE_RET1;  UTUK DELAY PENGULANGAN
        MOV  rulangcon,#1
        clr    udahCONNECT
        mov    dptr,#nomortujuan
        call   kirimkalimat
        CALL DELAY
        mov    rtunggu1,#255
```

```
        mov    rtunggu2,#255
        mov    rtunggu3,#100
TUNGGUCON2:
        djnz   Rtunggu1,tggc2
        djnz   Rtunggu2,tggc2
        djnz   Rtunggu3,tggc2
        mov    rulangcon,#100
        JMP    L_ATH2
tggc2:  JNB    udahCONNECT,TUNGGUCON2
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov    dptr,#lokasi
        call   kirimkalimat
        mov    dptr,#laporan2
        call   kirimkalimat
L_ATH2:     mov   dptr,#plus3
        call  kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov    dptr,#ath
        call  kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        ret
kirimlaporan3:
;       DJNZ   rulangcon,KE_RET1;  UTUK DELAY PENGULANGAN
        MOV  rulangcon,#1
        clr    udahCONNECT
        mov    dptr,#nomortujuan
        call  kirimkalimat
        CALL DELAY
```

```
        mov   rtunggu1,#255
        mov   rtunggu2,#255
        mov   rtunggu3,#100
TUNGGUCON3:
        djnz  Rtunggu1,tggc3
        djnz  Rtunggu2,tggc3
        djnz  Rtunggu3,tggc3
        mov   rulangcon,#100
        JMP   L_ATH3
tggc3:  JNB   udahCONNECT,TUNGGUCON3
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov   dptr,#lokasi
        call  kirimkalimat
        mov   dptr,#laporan1
        call  kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov   dptr,#lokasi
        call  kirimkalimat
        mov   dptr,#laporan2
        call  kirimkalimat
L_ATH3:     mov   dptr,#plus3
        call  kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        mov   dptr,#ath
        call  kirimkalimat
        CALL DELAYLAMA
        CALL DELAYLAMA
        ret
```

```
kirimkalimat:
        mov  a,#0
        movc  a,@a+dptr
        cjne a,#255,kirimterus
        ret
kirimterus:
        call KIRIM_SERIAL
        inc  dptr
        jmp kirimkalimat
NOMORTUJUAN: DB 'ATD 803',0Dh,255
LOKASI          : DB 'SPBU 123456, Jl. Raya Mondoroko No 1. Singosari-MALANG! ',255
Laporan1   : DB 'Bensin Habis',13,10,255
Laporan2   : DB 'Solar Habis',13,10,255
plus3      : db '+++',255
ath        : db 'ATH',0dh,255
END
```

Lampiran 2

Listing Program Software pada Delphi.

1. Listing program unit utama

```
unit UUtama;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, StdCtrls, Grids, DBGrids;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    SetDatabaseConnection1: TMenuItem;
    SetHardwareConnection1: TMenuItem;
    Exit1: TMenuItem;
    N1: TMenuItem;
    About1: TMenuItem;
    ViewForm1: TMenuItem;
    UserLogin1: TMenuItem;
    CreateReport1: TMenuItem;
    N2: TMenuItem;
    StatusBar1: TStatusBar;
    AddUser1: TMenuItem;
    procedure SetDatabaseConnection1Click(Sender: TObject);
    procedure AddUser1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure UserLogin1Click(Sender: TObject);
    procedure About1Click(Sender: TObject);
```

```
    procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
    procedure SetHardwareConnection1Click(Sender: TObject);
    procedure CreateReport1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

uses Umodule, Uaddope, Ulogin, Ubout, uREP;

{$R *.dfm}

procedure TForm1.SetDatabaseConnection1Click(Sender: TObject);
var od:TOpenDialog;
begin
od:=TOpenDialog.Create(nil);
od.Filter:='*.dsn file | *.dsn';
 if od.Execute then
  Begin
  DataModule1.ADOConnection1.Connected:=false;
  DataModule1.ADOConnection1.ConnectionString:='';
  //FILE NAME=E:\Data Delphi\Ipang Modem\paradoxdsn.dsn
  DataModule1.ADOConnection1.ConnectionString:='FILE NAME='+od.FileName;
  try
  DataModule1.ADOConnection1.Connected:=True;
```

```
  except
  ShowMessage('Koneksi gagal !!');
  end;
  end;
od.Free;
end;
procedure TForm1.AddUser1Click(Sender: TObject);
begin
try
Application.CreateForm(TForm2,form2);
except
ShowMessage('Gagal menciptakan window !!');
end;
end;
procedure TForm1.Exit1Click(Sender: TObject);
begin
close;
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
DataModule1.ADOConnection1.Connected:=false;
Application.Terminate;
end;
procedure TForm1.UserLogin1Click(Sender: TObject);
begin
Application.CreateForm(TForm3,Form3);
end;
procedure TForm1.About1Click(Sender: TObject);
begin
Application.CreateForm(TAboutBox,Aboutbox);
end;
```

```
procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
begin
if Application.MessageBox('Keluar dari aplikasi ??','Konfirmasi', Mb_YesNo or
MB_ICONQUESTION) = mryes then
CanClose:=true
else
CanClose:=false;
end;
procedure TForm1.SetHardwareConnection1Click(Sender: TObject);
begin
DataModule1.ComPort.ShowSetupDialog;
end;
procedure TForm1.CreateReport1Click(Sender: TObject);
begin
Application.CreateForm(TForm5,Form5);
end;
end.
```

2. Listing program tampilan laporan keadaan tangki

```
unit FMon;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, DBGrids, StdCtrls, ComCtrls, CPortCtl, CPort, Buttons,
  ExtCtrls, Unittyep,Mask, DBCtrls;

type
  TForm4 = class(TForm)
    GroupBox1: TGroupBox;
```

```
GroupBox2: TGroupBox;
h: TDBGrid;
StatusBar1: TStatusBar;
ComDataPacket2: TComDataPacket;
Timer1: TTimer;
Panel1: TPanel;
Panel2: TPanel;
DBText3: TDBText;
DBText2: TDBText;
Panel3: TPanel;
ComLed1: TComLed;
Label6: TLabel;
Label3: TLabel;
Panel4: TPanel;
Panel5: TPanel;
Label2: TLabel;
Label8: TLabel;
Label5: TLabel;
Label7: TLabel;
DBGrid2: TDBGrid;
GroupBox3: TGroupBox;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
ComLed2: TComLed;
ComLed3: TComLed;
Label1: TLabel;
Label4: TLabel;
ComDataPacket1: TComDataPacket;
DBNavigator1: TDBNavigator;
DBNavigator2: TDBNavigator;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
    procedure ComDataPacket1Packet(Sender: TObject; const Str: String);
    procedure ComDataPacket2Packet(Sender: TObject; const Str: String);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure hDblClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }

  Procedure InsertDataStatus2DB(datanya:TDataMateng);
  Procedure InsertDataPenangana2DB(IdKejadian,IdOPe,Lokasi,Status,waktu:string;Tgl:TDate);
  public
    { Public declarations }
  end;

const Linefeed=10;
      CarrReturn=13;
var
  Form4: TForm4;

implementation

uses Umodule, DB;

{$R *.dfm}

Procedure
TForm4.InsertDataPenangana2DB(IdKejadian,IdOPe,Lokasi,Status,waktu:string;Tgl:TDate);
Begin
  With DataModule1.ADOTable3 do
```

```
  Begin
  try
   Append;
    Fields[0].AsString:=IdKejadian;
    Fields[1].AsString:=IdOPe;
    Fields[2].AsString:=Lokasi;
    Fields[3].AsString:=Status;
    Fields[4].AsDateTime:=Date;
    Fields[5].AsString:=waktu;
   Post;
   Refresh;
  except
   ShowMessage('Error proses insert data !!!');
  end;
 end;
end;

Procedure TForm4.InsertDataStatus2DB(datanya:TDataMateng);
var Id,thn,idopertr:string;
   hari,bulan,tahun,jam,menit,detik,md:word;
   tgl:TTime;
   wktu:TDate;
begin
 DecodeDate(Date,tahun,bulan,hari);
 DecodeTime(now,jam,menit,detik,md);
 thn:=inttostr(tahun);
 id:=thn[Length(thn)-1]+inttostr(bulan)+thn[Length(thn)]+inttostr(hari)+
    inttostr(menit)+inttostr(jam)+inttostr(detik);
 wktu:=Now;
 tgl:=Now;
 idopertr:=DBText3.Caption;
```

```
  With DataModule1.ADOTable2 do
   Begin
   try
    Append;
     Fields[0].AsString:=id;
     Fields[1].AsString:=datanya.Alamat;
     Fields[2].AsString:=datanya.Status;
     Fields[3].AsDateTime:=tgl;
     Fields[4].AsString:=TimeToStr(now);
     Fields[5].AsString:=idopertr;
    Post;
    Refresh;
   except
    ShowMessage('Error proses insert data !!!');
   end;
  end;
end;

procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=cafree;
DataModule1.ComPort.Close;
end;

procedure TForm4.ComDataPacket1Packet(Sender: TObject; const Str: String);
var s:string;
   pesanmateng:TDataMateng;
   arrstring:TArry_str2;
   x:integer;
begin
s:=str;
```

```
s:=FClearLineFeed(s);
x:=LEngth(s);
 If x<=90 then begin
        pesanmateng:=FKonversiData(s);
        InsertDataStatus2DB(pesanmateng);
          end
          else
          Begin
       arrstring:=FPisah2Kata(s);
       pesanmateng:=FKonversiData(arrstring[1]);
       InsertDataStatus2DB(pesanmateng);
       pesanmateng:=FKonversiData(arrstring[2]);
       pesanmateng.Status:=pesanmateng.Status+'s';
       PDelay(500,500,100);
       InsertDataStatus2DB(pesanmateng);
         end;
end;

procedure TForm4.ComDataPacket2Packet(Sender: TObject; const Str: String);
begin
lABEL6.Caption:='OK';
timer1.Enabled:=true;
BitBtn1.Enabled:=true;
end;

procedure TForm4.Timer1Timer(Sender: TObject);
begin
Label7.Caption:=TimeTostr(Now);
Label8.Caption:=DateTostr(Now);
if ComLed1.State=lsOn then
ComLed1.State:=lsOff
```

```
else
ComLed1.State:=lsOn;
end;

procedure TForm4.BitBtn2Click(Sender: TObject);
begin
if DataModule1.ComPort.Connected then
 Begin
 DataModule1.ComPort.Connected:=false;
 BitBtn2.Caption:='Connect';
 BitBtn1.Enabled:=False;
 Timer1.Enabled:=false;
 Label6.Caption:='-';
 Label7.Caption:='-';
 Label8.Caption:='-';
 end
 else
 Begin
 BitBtn2.Caption:='Disconnect';
 DataModule1.ComPort.Connected:=true;
 BitBtn1.Enabled:=true;
 end;
end;

procedure TForm4.BitBtn1Click(Sender: TObject);
begin
with DataModule1 do begin
ComPort.WriteStr('AT&V'+#13#10); end;
BitBtn1.Enabled:=false;
Timer1.Enabled:=false;
Label6.Caption:='-';
```

```
 Label7.Caption:='-';
 Label8.Caption:='-';
 end;

 procedure TForm4.hDblClick(Sender: TObject);
 Var
 IdKejadian,IdOPe,Lokasi,Status,waktu:string;Tgl:TDate;
 i:integer;
 loop:boolean;
 c:char;
 begin
  With DataModule1.ADOTable2 do
   Begin
    IdKejadian:=Fields[0].AsString;
    IdOPe:=Fields[5].AsString;
    Lokasi:=Fields[1].AsString;
    Status:=Fields[2].AsString;
    Tgl:=Date;
    waktu:=Timetostr(Now);
   end;
  loop:=true;
  i:=Length(Status);
   While loop do
    Begin
     c:=status[i];
      if c=' 'then
      loop:=false;
     dec(i);
    end;
inc(i);
Delete(Status,i,6);
```

```
 Status:=status+' Terkirim';
 If not (DataModule1.ADOTable3.Locate('IdKejadian',IdKejadian,[lopartialkey])) then
 InsertDataPenangana2DB(IdKejadian,IdOPe,Lokasi,Status,waktu,Tgl)
 else
 Application.MessageBox('Maaf !!, Pengisian telah dilakukan !!','Informasi',Mb_Ok or
MB_ICONHAND);
end;

procedure TForm4.FormShow(Sender: TObject);
begin
 Try
    DataModule1.ADOTable2.Filtered:=False;
    DataModule1.ADOTable3.Filtered:=False;
 except
 ShowMessage('Error pada database !!!');
end;
end;
end.
```

3. Listing program Data Operator

```
unit Uaddope;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, Grids, DBGrids, ComCtrls, StdCtrls, Buttons, ExtCtrls, DBCtrls;

type
 TForm2 = class(TForm)
   DBGrid1: TDBGrid;
```

```
    GroupBox1: TGroupBox;
    StatusBar1: TStatusBar;
    Label1: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Label2: TLabel;
    Edit3: TEdit;
    Label3: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    DBNavigator1: TDBNavigator;
    BitBtn3: TBitBtn;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  Procedure PInsertOperator();
  end;

var
  Form2: TForm2;

implementation

uses Umodule, DB;
```

```
{$R *.dfm}

Procedure TForm2.PInsertOperator();
Begin
 With DataModule1.ADOTable1 do
 Begin
  try
   Append;
    fields[0].AsString:=Edit1.Text;
    fields[1].AsString:=Edit2.Text;
    fields[2].AsString:=Edit3.Text;
   Post;
    Edit1.Clear;
    Edit2.Clear;
    Edit3.Clear;
  except
   Cancel;
   ShowMessage('Error pada proses entri data !!'+#13+'Cek kembali ID Operator !!');
  end;
 end;
end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
action:=caFree;
end;

procedure TForm2.BitBtn1Click(Sender: TObject);
begin
If (Edit1.Text<>'') and (Edit2.Text<>'') and (Edit3.Text<>'') then
 PInsertOperator
```

```
  else
  Application.MessageBox('Data tidak boleh dikosongkan !!!','Peringatan',mb_Ok or
MB_ICONHAND);
end;

procedure TForm2.BitBtn2Click(Sender: TObject);
begin
close;
end;

procedure TForm2.BitBtn3Click(Sender: TObject);
var Id,thn:string;
   hari,bulan,tahun,jam,menit,detik,md:word;
begin
 DecodeDate(Date,tahun,bulan,hari);
 DecodeTime(now,jam,menit,detik,md);
 thn:=inttostr(tahun);
 id:=thn[Length(thn)-1]+thn[Length(thn)]+inttostr(bulan)+inttostr(hari)+
    inttostr(jam)+inttostr(menit);
 Edit1.Text:=Id;
end;

procedure TForm2.FormShow(Sender: TObject);
begin
try
 DataModule1.ADOTable1.Filtered:=False;
except
ShowMessage('Error pada database !!!');
end;
end;
end.
```

4. Listing program informasi tentang perancang program

```
unit Ubout;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,
  Buttons, ExtCtrls, jpeg;

type
  TAboutBox = class(TForm)
    Label1: TLabel;
    SpeedButton1: TSpeedButton;
    Label2: TLabel;
    Label3: TLabel;
    Image1: TImage;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure OKButtonClick(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AboutBox: TAboutBox;

implementation
```

```
{$R *.dfm}

procedure TAboutBox.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Action:=cafree;
end;

procedure TAboutBox.OKButtonClick(Sender: TObject);
begin
close;
end;

procedure TAboutBox.SpeedButton1Click(Sender: TObject);
begin
close;
end;
end.
```

5. Listing program Login User

```
unit Ulogin;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, Buttons,DB;

type
 TForm3 = class(TForm)
  GroupBox1: TGroupBox;
```

```
    Label1: TLabel;
    Label3: TLabel;
    Edit1: TEdit;
    Edit3: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  Procedure PLogin();
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses FMon, Umodule, ADODB;

{$R *.dfm}

Procedure TForm3.PLogin;
Begin
try
Application.CreateForm(TForm4,Form4);
except
ShowMessage('Gagal menciptakan form !!');
```

```
 end;
 close;
 end;

 procedure TForm3.BitBtn1Click(Sender: TObject);
 var id,psw:string;
    ketemu:boolean;
 begin
 id:=Edit1.Text;
 psw:=Edit3.Text;
 With DataModule1.ADOTable1 do
  Begin
   ketemu:=Locate('IDOperator;Password', VarArrayOf([id,psw]), [loCaseInsensitive]);
   if ketemu then
    PLogin
   else
   Application.MessageBox('Masukan nama dan password secara benar !!','Peringatan',Mb_OK
or MB_ICONHAND);
  end;
end;

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Action:=cafree;
end;

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
close;
end;
end.
```

6. Listing program modul

```
unit Umodule;

interface

uses
  SysUtils, Classes, DB, ADODB,Dialogs, CPort, ImgList, Controls;

type
  TDataModule1 = class(TDataModule)
    DataSource1: TDataSource;
    ADOConnection1: TADOConnection;
    ADOTable1: TADOTable;
    ADOTable2: TADOTable;
    DataSource2: TDataSource;
    ComPort: TComPort;
    ADOTable3: TADOTable;
    DataSource3: TDataSource;
    PopupMenuImages: TImageList;
    procedure ADOConnection1AfterConnect(Sender: TObject);
    procedure ADOConnection1AfterDisconnect(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  DataModule1: TDataModule1;

implementation
```

```
uses FMon, UUtama;

{$R *.dfm}

procedure TDataModule1.ADOConnection1AfterConnect(Sender: TObject);
begin
try
ADOTable1.TableName:='Operator';
ADOTable1.Open;
ADOTable2.TableName:='Kejadian';
ADOTable2.Open;
ADOTable3.TableName:='Kejadian1';
ADOTable3.Open;
 with Form1 do
  Begin
   UserLogin1.Enabled:=true;
   AddUser1.Enabled:=true;
   CreateReport1.Enabled:=true;
  end;
except
ADOTable1.close;
ADOTable2.close;
ADOTable3.close;
 with Form1 do
  Begin
   UserLogin1.Enabled:=false;
   AddUser1.Enabled:=false;
   CreateReport1.Enabled:=false;
  end;
ShowMessage('Tabel tidak ditemukan !!');
```

```
 end;
 end;

 procedure TDataModule1.ADOConnection1AfterDisconnect(Sender: TObject);
 begin
 try
 ADOTable1.Close;
 ADOTable2.Close;
 ADOTable3.Close;
  with Form1 do
   Begin
    UserLogin1.Enabled:=False;
    AddUser1.Enabled:=False;
    CreateReport1.Enabled:=False;
   end;
 except
 ShowMessage('Terjadi eksepsi !! !!');
 end;
 end;
 end.
```

7. Listing program Report

```
unit Unitrep;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, QuickRpt, QRCtrls, ExtCtrls, jpeg;
type
  TForm6 = class(TForm)
```

QuickRep1: TQuickRep;
TitleBand1: TQRBand;
SummaryBand1: TQRBand;
QRLabel1: TQRLabel;
QRLabel2: TQRLabel;
QRLabel3: TQRLabel;
QRLabel4: TQRLabel;
QRLabel5: TQRLabel;
QRLabel10: TQRLabel;
QRLabel6: TQRLabel;
QRLabel7: TQRLabel;
QRLabel8: TQRLabel;
QRLabel9: TQRLabel;
QRLabel11: TQRLabel;
QRLabel12: TQRLabel;
QRLabel13: TQRLabel;
QRDBText1: TQRDBText;
QRDBText2: TQRDBText;
QRDBText3: TQRDBText;
QRDBText4: TQRDBText;
QRDBText5: TQRDBText;
QRDBText6: TQRDBText;
QRDBText7: TQRDBText;
QRDBText8: TQRDBText;
QRDBText9: TQRDBText;
QRDBText10: TQRDBText;
QRDBText11: TQRDBText;
QRDBText12: TQRDBText;
QRImage1: TQRImage;
QRLabel14: TQRLabel;
QRDBText13: TQRDBText;

```
    QRLabel15: TQRLabel;
    QRDBText14: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

implementation

uses Umodule;

{$R *.dfm}

end.
```

8. Listing program Laporan data SPBU dan status tangki

```
unit Unittyep;

interface
uses
  SysUtils, Classes, DB, ADODB,Dialogs, CPort;

Type   TArry_str2=array [1..2] of String;
      TDataMateng = record
      NamaSPBU,Alamat,Status:string;
      end;
```

```
Function FClearLineFeed(pesan:string):string;
Function FKonversiData(Pesan:String):TDataMateng;
Function FPisah2Kata(pesandouble:string):TArry_str2;
Procedure PDelay(x,y:integer;pengali:integer);

implementation

Procedure PDelay(x,y:integer;pengali:integer);
var i,j,k:integer;
Begin
 For i:=0 to x*pengali do
  For j:=0 to y*pengali do
end;

Function FPisah2Kata(pesandouble:string):TArry_str2;
var i:integer;
   ordc:byte;
   loop:boolean;
Begin
Result[1]:='';
Result[2]:='';
   i:=0;
   loop:=true;
   While loop do
    Begin
    inc(i);
    ordc:=ord(pesandouble[i]);
     if ordc=10 then
      loop:=false
     else
     Result[1]:=Result[1]+pesandouble[i];
    end;
```

```
    loop:=true;
//   inc(i);
    While loop do
    Begin
    inc(i);
    ordc:=ord(pesandouble[i]);
     if ordc=10 then
      loop:=false
     else
     Result[2]:=Result[2]+pesandouble[i];
    end;
  end;

Function FClearLineFeed(pesan:string):string;
var i:integer;
   c:char;
Begin
 {
 Result:='';
 For i:=1 to Length(Pesan) do
  Begin
   c:=Pesan[i];
   if (ord(c)<>CarrReturn) and (ord(c)<>Linefeed) then
   Result:=Result+Pesan[i];
 end;
 }
Result:=Pesan;
Delete(Result,1,2);
Delete(Result,Length(Result)-1,1);
end;
```

```
Function FKonversiData(Pesan:String):TDataMateng;
Var i:integer;
   loop:boolean;
Begin
 Result.Alamat:='';
 Result.NamaSPBU:='';
 Result.Status:='';
 loop:=true;
 i:=0;
 While loop do
  Begin
  inc(i);
   if Pesan[i]<>',' then
   Result.NamaSPBU:=Result.NamaSPBU+Pesan[i]
   else
   loop:=false;
 end;
 loop:=true;
 While loop do
 Begin
 inc(i);
  if Pesan[i]='!' then
    loop:=false
 else
    Result.Alamat:=Result.Alamat+Pesan[i];
end;
delete(Result.Alamat,1,1);
loop:=true;
While loop do
Begin
inc(i);
```

```
    if i<>length(pesan) then
    Result.Status:=Result.Status+Pesan[i]
    else
    loop:=false;
   end;
   delete(Result.Status,1,1);
end;
end.
```

9. Listing program Detail Data SPBU

```
unit uREP;

interface

uses
 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, ComCtrls, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids, Buttons;

type
 TForm5 = class(TForm)
  DBGrid1: TDBGrid;
  DBGrid2: TDBGrid;
  DBNavigator1: TDBNavigator;
  DBNavigator2: TDBNavigator;
  Panel1: TPanel;
  Label1: TLabel;
  StatusBar1: TStatusBar;
  DBGrid3: TDBGrid;
  BitBtn1: TBitBtn;
  procedure DBGrid1DblClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
```

```
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

uses Umodule, DB, ADODB, Unitrep;

{$R *.dfm}

procedure TForm5.DBGrid1DblClick(Sender: TObject);
VAR idk,idop:string;
begin
 with DataModule1.ADOTable2 do
  idk:=Fields[0].AsString;
 with DataModule1.ADOTable3 do
 Begin
// Locate('IdKejadian',idk,[loCaseInsensitive]);
 Filtered:=true;
 Filter:='IdKejadian = '+QuotedStr(idk);
 idop:=Fields[1].AsString;

 end;
 with DataModule1.ADOTable1 do
 Begin
// Locate('IdOperator',idop,[loCaseInsensitive]);
```

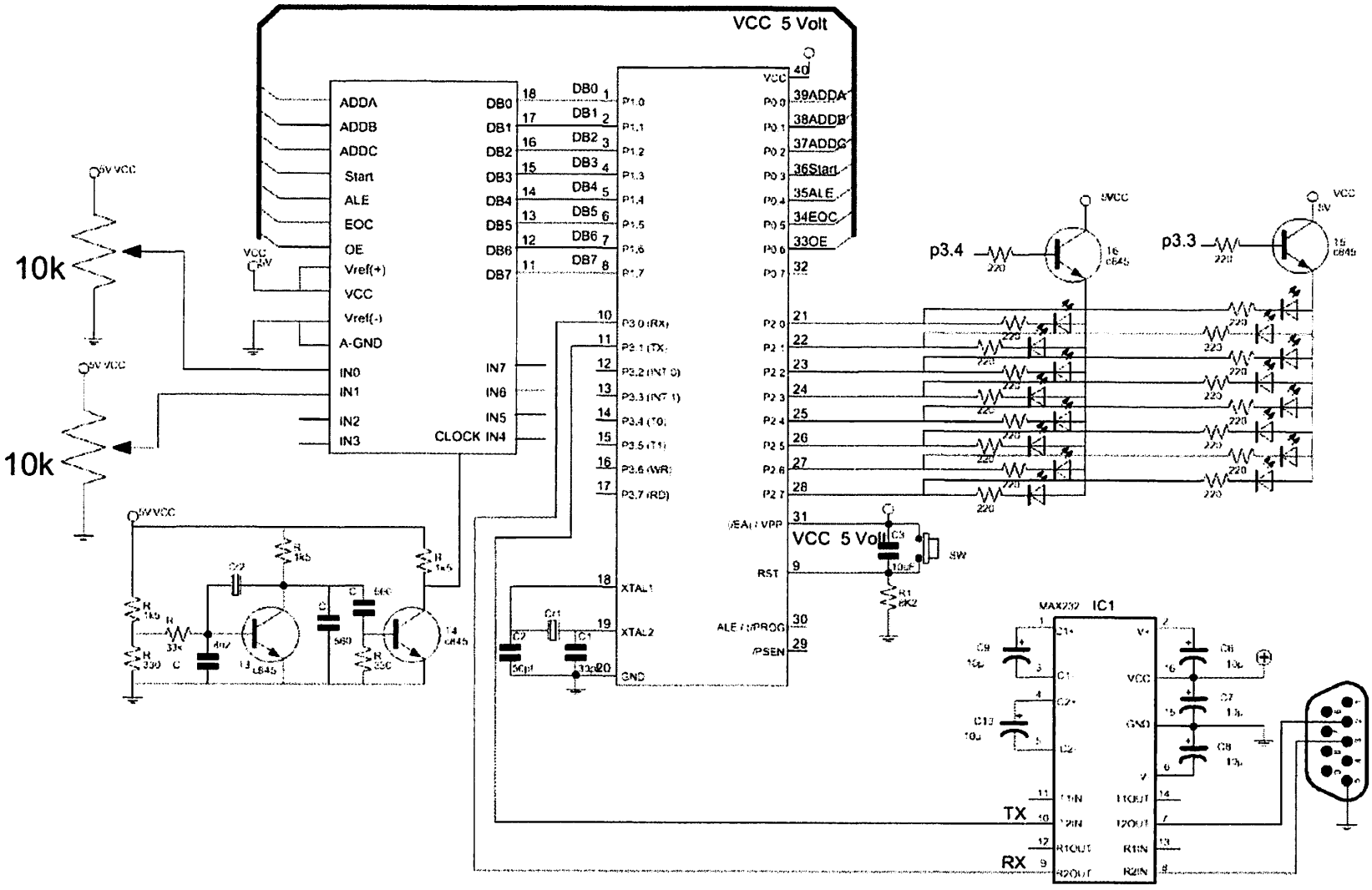
```
  Filtered:=true;
  Filter:='IdOperator = '+QuotedStr(idop);
  idop:=Fields[0].AsString;
 end;
end;

procedure TForm5.FormClose(Sender: TObject; var Action: TCloseAction);
begin
Action:=cafree;
end;

procedure TForm5.BitBtn1Click(Sender: TObject);
begin
Form6.QuickRep1.Preview;
end;
end.
```

10. Listing program Database

```
[ODBC]
DRIVER=Driver do Microsoft Paradox (*.db )
ParadoxUserName=admin
ParadoxNetStyle=4.x
ParadoxNetPath=C:\WINDOWS\SYSTEM
PageTimeout=600
MaxBufferSize=2048
FIL=Paradox 5.X
DriverId=538
DefaultDir=C:\SKRIPSI BIN TA\IPANG MODEM\DB
DBQ=C:\SKRIPSI BIN TA\IPANG MODEM\DB
CollatingSequence=ASCII
```


VCC 5 Volt
10k
10k
ADDA
ADDB
ADDC
Start
ALE
EOC
OE
Vref(+)
VCC
Vref(-)
A-GND
IN0
IN1
IN2
IN3
IN7
IN6
IN5
CLOCK IN4
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
39ADDA
38ADDB
37ADDC
36Start
35ALE
34EOC
33OE
p3.4
p3.3
VCC 5 Volt
SW
XTAL1
XTAL2
RST
/PSEN
ALE /PROG
MAX232 IC1
TX
RX

atures

ompatible with MCS-51® Products
K Bytes of In-System Programmable (ISP) Flash Memory
– Endurance: 1000 Write/Erase Cycles
.0V to 5.5V Operating Range
ully Static Operation: 0 Hz to 33 MHz
hree-level Program Memory Lock
56 x 8-bit Internal RAM
2 Programmable I/O Lines
hree 16-bit Timer/Counters
ight Interrupt Sources
ull Duplex UART Serial Channel
ow-power Idle and Power-down Modes
nterrupt Recovery from Power-down Mode
Watchdog Timer
ual Data Pointer
ower-off Flag
ast Programming Time
lexible ISP Programming (Byte and Page Mode)

escription

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

AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes AM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, clock circuitry. In addition, the AT89S52 is designed with static logic for operation vn to zero frequency and supports two software selectable power saving modes. Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and rrupt system to continue functioning. The Power-down mode saves the RAM con- s but freezes the oscillator, disabling all other chip functions until the next interrupt ardware reset.

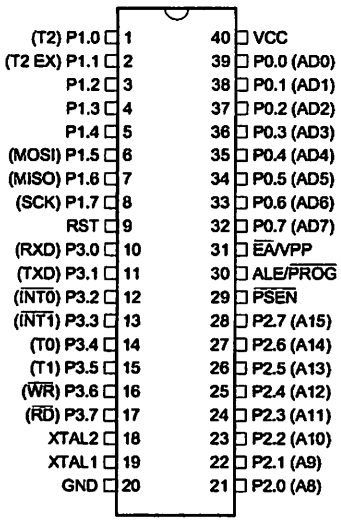
8-bit Microcontroller with 8K Bytes In-System Programmable Flash

AT89S52

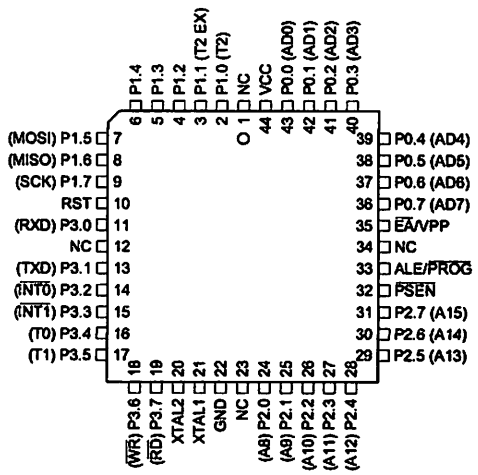
1919B–MICRO–11/03

n Configurations

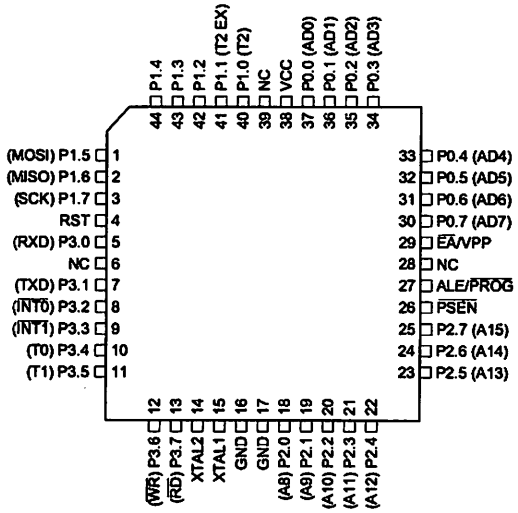
PDIP



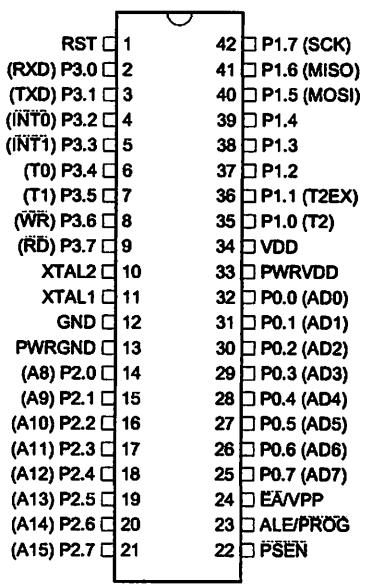
PLCC



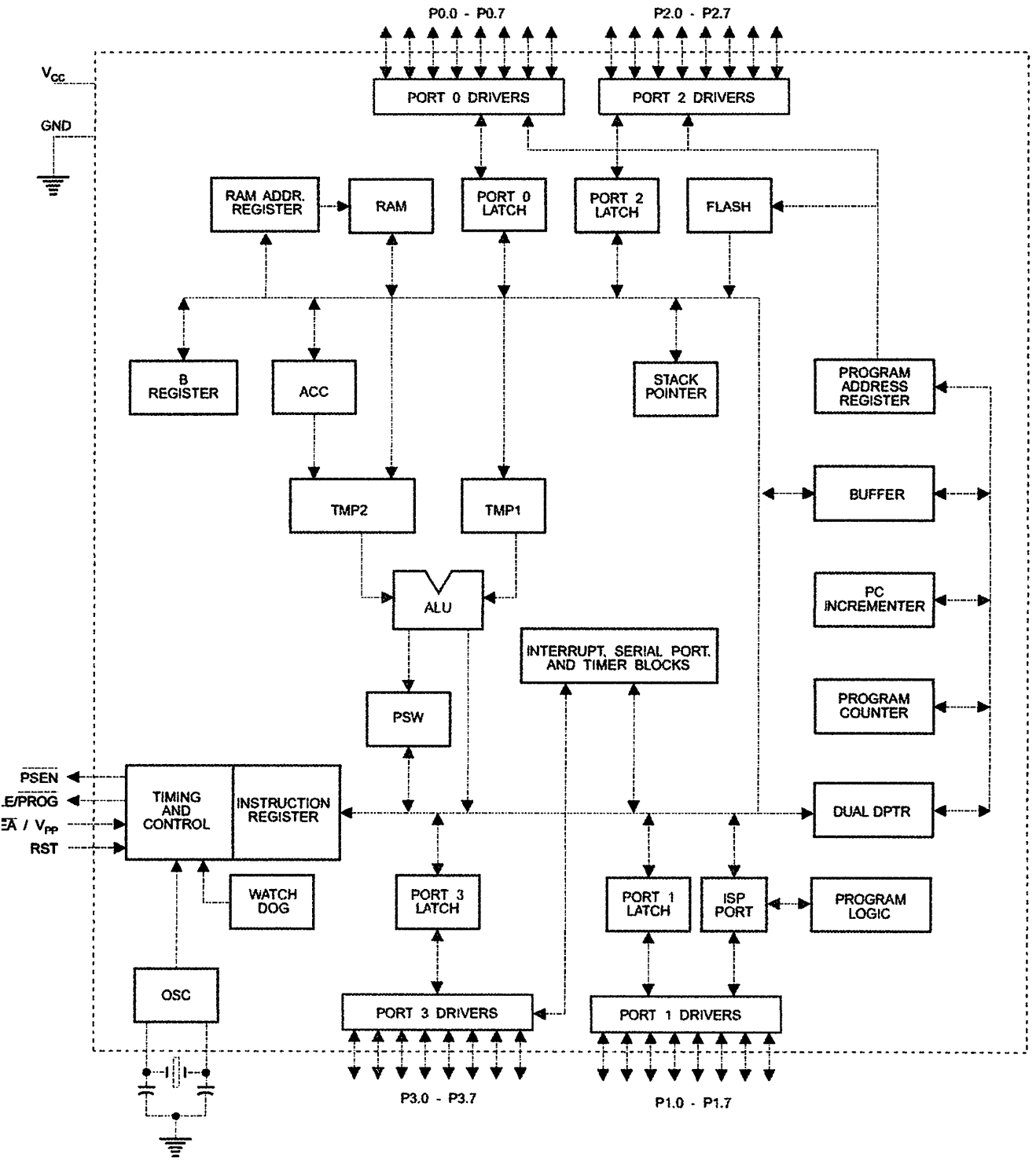
TQFP



PDIP



ock Diagram


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

n Description

C Supply voltage.

ID Ground.

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

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

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

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

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

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

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.5	MOSI (used for In-System Programming)
P1.6	MISO (used for In-System Programming)
P1.7	SCK (used for In-System Programming)

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

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

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

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

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

r

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

:/$\overline{PROG}$

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

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

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

$\overline{N}$

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

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

/PP

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

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

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

-1

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

-2

Output from the inverting oscillator amplifier.

ıecial Function ɨgisters

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

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

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

Timer 2 Registers: Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 6) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

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

ıle 1. AT89S52 SFR Map and Reset Values

ƆF8H								0FFH
ƆF0H	B 00000000							0F7H
ƆE8H								0EFH
ƆE0H	ACC 00000000							0E7H
ƆD8H								0DFH
ƆD0H	PSW 00000000							0D7H
ƆC8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
ƆC0H								0C7H
ƆB8H	IP XX000000							0BFH
ƆB0H	P3 11111111							0B7H
ƆA8H	IE 0X000000							0AFH
ƆA0H	P2 11111111		AUXR1 XXXXXXX0				WDTRST XXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX00XX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H

ıle 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

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

ymbol	Function
F2	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.
XF2	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).
CLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
CLK	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.
XEN2	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.
R2	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).
P/$\overline{RL2}$	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.

ıle 3. AUXR: Auxiliary Register

JXR — Address = 8EH — Reset Value = XXX00XX0B

Not Bit Addressable

	–	–	–	WDIDLE	DISRTO	–	–	DISALE
Bit	7	6	5	4	3	2	1	0

Reserved for future expansion

SALE — Disable/Enable ALE

DISALE	Operating Mode
0	ALE is emitted at a constant rate of 1/6 the oscillator frequency
1	ALE is active only during a MOVX or MOVC instruction

SRTO — Disable/Enable Reset out

DISRTO	
0	Reset pin is driven High after WDT times out
1	Reset pin is input only

'DIDLE — Disable/Enable WDT in IDLE mode

WDIDLE	
0	WDT continues to count in IDLE mode
1	WDT halts counting in IDLE mode

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

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

ıle 4. AUXR1: Auxiliary Register 1

JXR1 — Address = A2H — Reset Value = XXXXXXX0B

Not Bit Addressable

	–	–	–	–	–	–	–	DPS
Bit	7	6	5	4	3	2	1	0

Reserved for future expansion

PS — Data Pointer Register Select

DPS	
0	Selects DPTR Registers DP0L, DP0H
1	Selects DPTR Registers DP1L, DP1H

emory Organization

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

ogram Memory

If the $\overline{EA}$ pin is connected to GND, all program fetches are directed to external memory.

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

ıta Memory

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

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

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

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

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

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

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

atchdog Timer ne-time Enabled th Reset-out)

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

ıing the WDT

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

DT During Power-own and Idle

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

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

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

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

ART

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

ner 0 and 1

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

ner 2

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

Table 5. Timer 2 Operating Modes

RCLK +TCLK	CP/$\overline{RL2}$	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)

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.

ıpture Mode

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

ıto-reload (Up or Down ıunter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 6). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

ure 1. Timer in Capture Mode

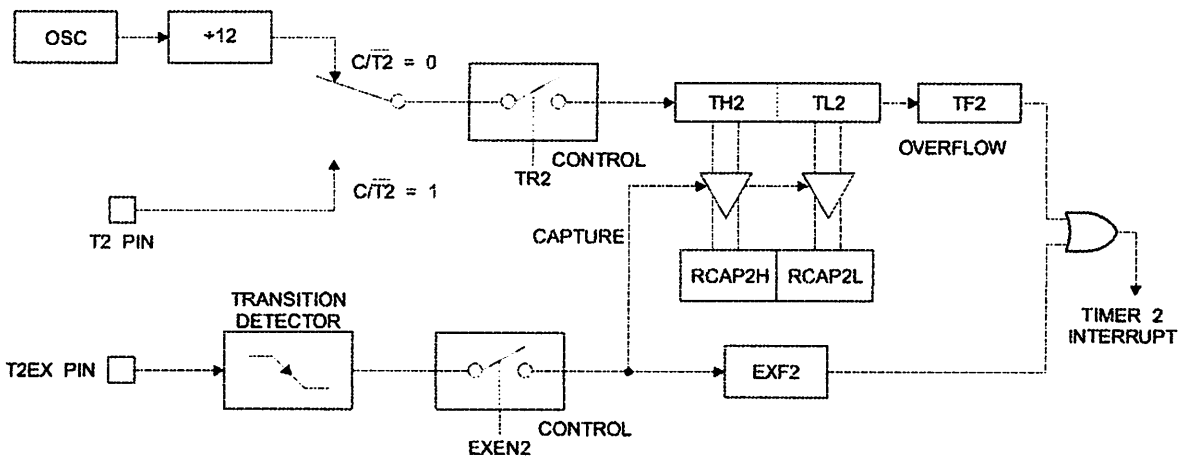


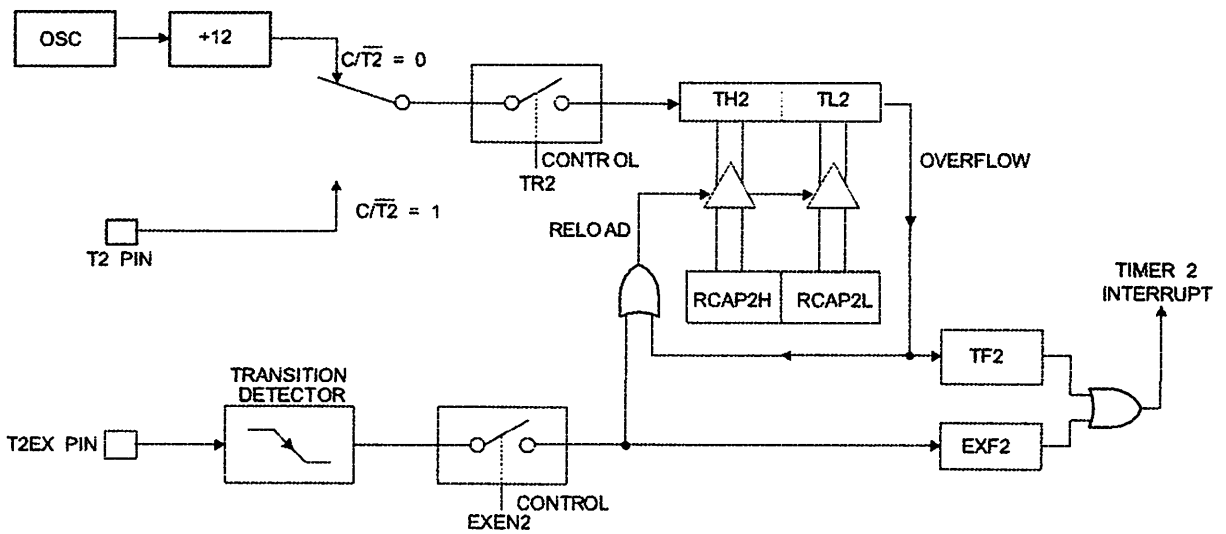
Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture ModeRCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

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

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

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

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



ble 6. T2MOD – Timer 2 Mode Control Register

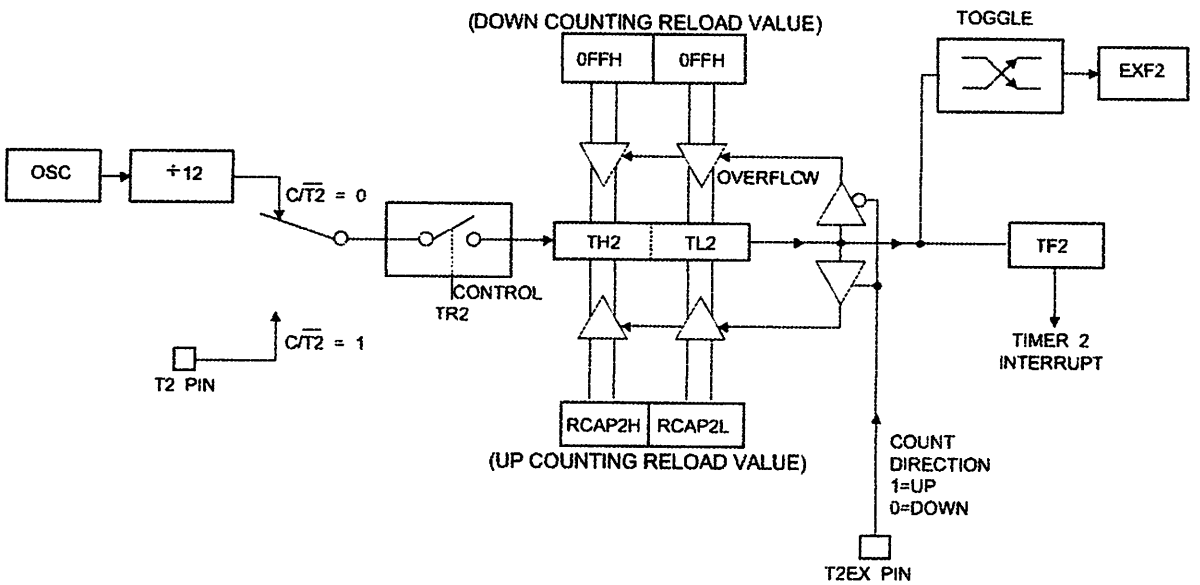
T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

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

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

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



aud Rate Generator

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

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

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

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

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

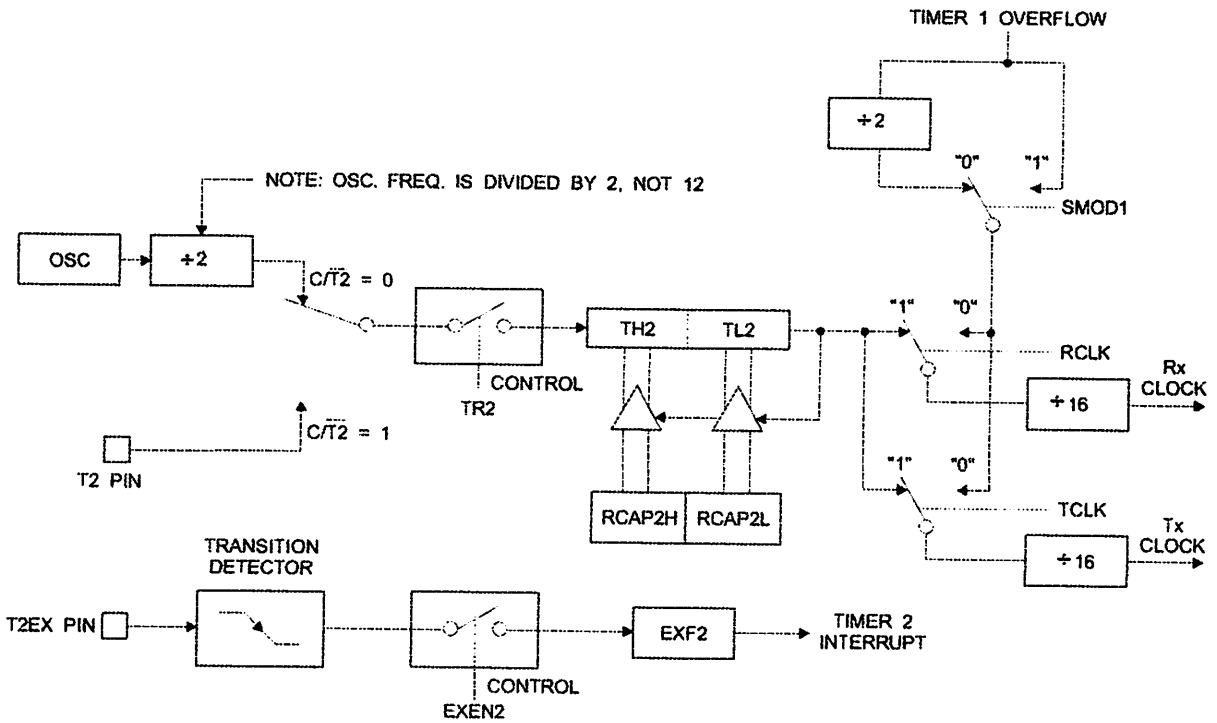
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536\text{-RCAP2H,RCAP2L})]}$$

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 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.

ıure 4. Timer 2 in Baud Rate Generator Mode



ɔgrammable ɔck Out

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

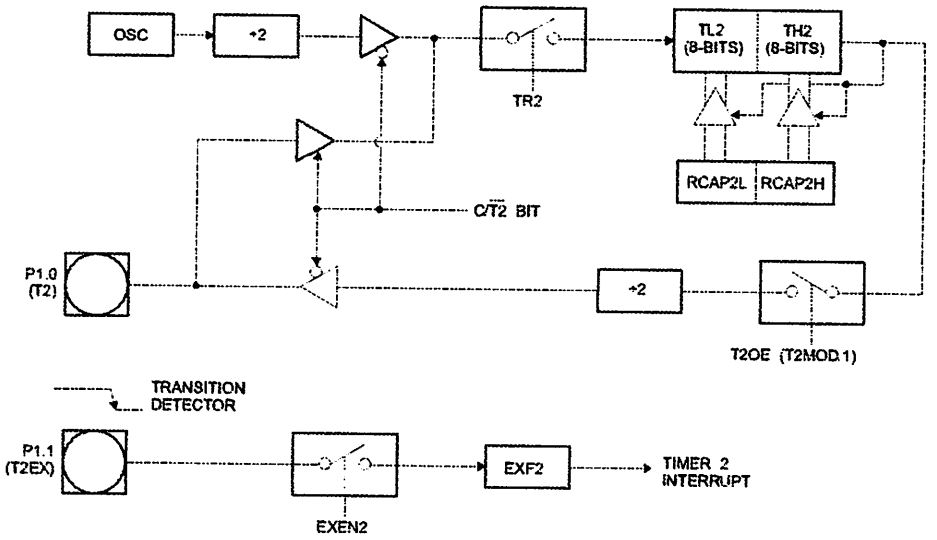
To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

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

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

ure 5. Timer 2 in Clock-Out Mode



errupts

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

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

Note that Table 5 shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it may be used in future AT89 products.

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

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

ble 7. Interrupt Enable (IE) Register

(MSB)					(LSB)		
EA	–	ET2	ES	ET1	EX1	ET0	EX0

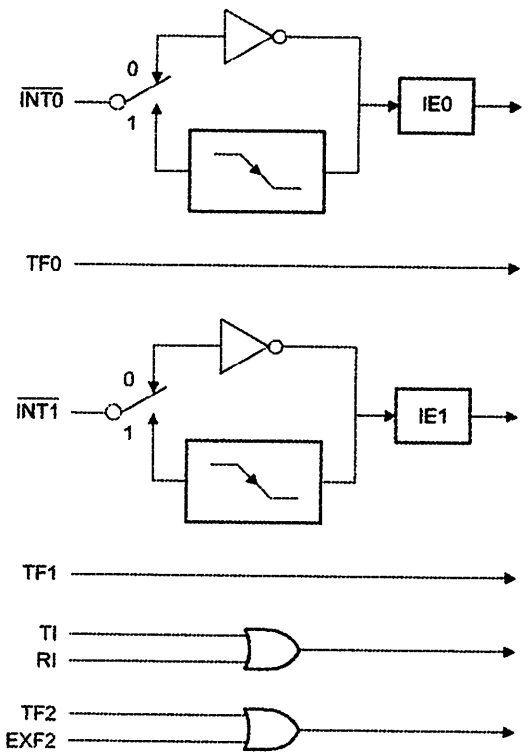
Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

ymbol	Position	Function
A	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.
T2	IE.5	Timer 2 interrupt enable bit.
S	IE.4	Serial Port interrupt enable bit.
T1	IE.3	Timer 1 interrupt enable bit.
X1	IE.2	External interrupt 1 enable bit.
T0	IE.1	Timer 0 interrupt enable bit.
X0	IE.0	External interrupt 0 enable bit.

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

ure 6. Interrupt Sources



scillator haracteristics

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 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

le Mode

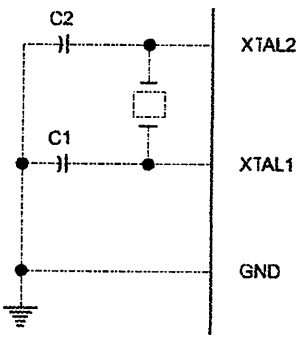
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

ıwer-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by 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.

Figure 7. Oscillator Connections



Note: 1. C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 8. External Clock Drive Configuration

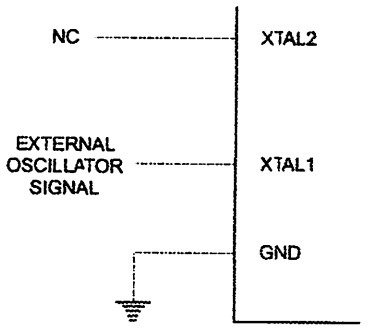


Table 8. Status of External Pins During Idle and Power-down Modes

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

ogram Memory ck Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 9. Lock Bit Protection Modes

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

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

rogramming the ash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, take the following steps:

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

$\overline{Data}$ Polling: The AT89S52 features $\overline{Data}$ Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. 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.

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

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

(000H) = 1EH indicates manufactured by Atmel
(100H) = 52H indicates AT89S52
(200H) = 06H

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

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

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

rogramming the ash – Serial Mode

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

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

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

rial Programming gorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:

   Apply power between VCC and GND pins.

   Set RST pin to "H".

   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction 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 device operation.

Power-off sequence (if needed):

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

Set RST to "L".

Turn V_{CC} power off.

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

erial Programming struction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 11.

ogramming terface – Parallel ode

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

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

ble 10. Flash Programming Modes

ode	V_{CC}	RST	$\overline{PSEN}$	ALE/ $\overline{PROG}$	$\overline{EA}$/ V_{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.4-0 Address	P1.7-0 Address
rite Code Data	5V	H	L	(2)	12V	L	H	H	H	H	D_{IN}	A12-8	A7-0
ad Code Data	5V	H	L	H	H	L	L	L	H	H	D_{OUT}	A12-8	A7-0
rite Lock Bit 1	5V	H	L	(3)	12V	H	H	H	H	H	X	X	X
rite Lock Bit 2	5V	H	L	(3)	12V	H	H	H	L	L	X	X	X
rite Lock Bit 3	5V	H	L	(3)	12V	H	L	H	H	L	X	X	X
ad Lock Bits 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
ip Erase	5V	H	L	(1)	12V	H	L	H	L	L	X	X	X
ad Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	X 0000	00H
ad Device ID	5V	H	L	H	H	L	L	L	L	L	52H	X 0001	00H
ad Device ID	5V	H	L	H	H	L	L	L	L	L	06H	X 0010	00H

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

Figure 9. Programming the Flash Memory (Parallel Mode)

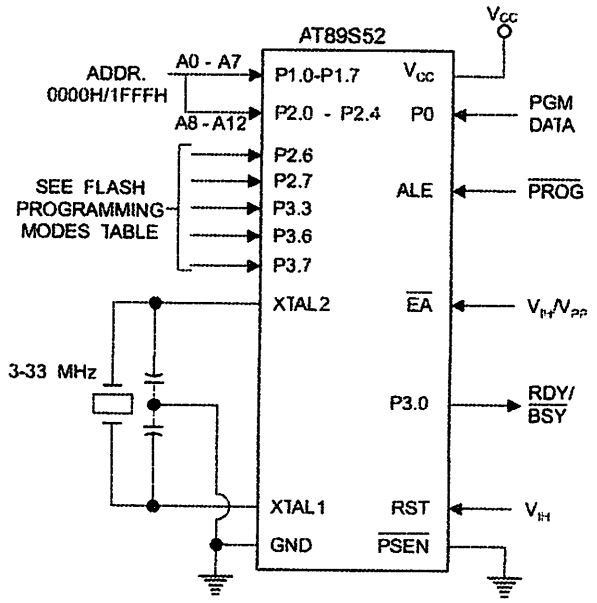
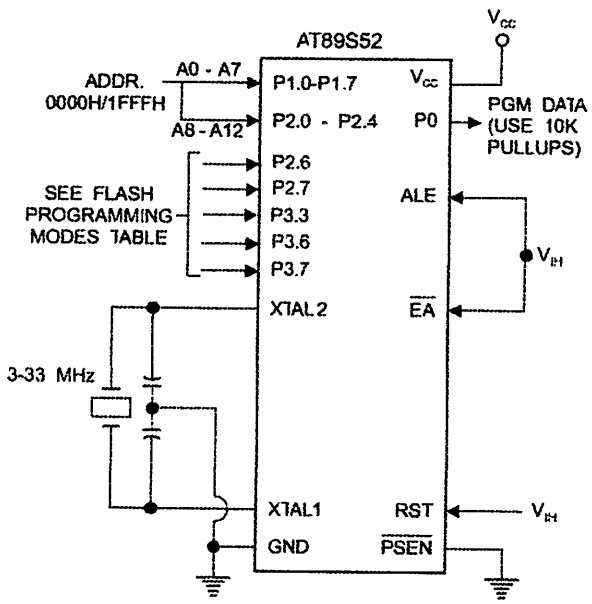


Figure 10. Verifying the Flash Memory (Parallel Mode)



ash Programming and Verification Characteristics (Parallel Mode)

= 20°C to 30°C, V_{CC} = 4.5 to 5.5V

ymbol	Parameter	Min	Max	Units
PP	Programming Supply Voltage	11.5	12.5	V
P	Programming Supply Current		10	mA
C	V_{CC} Supply Current		30	mA
t_{CLCL}	Oscillator Frequency	3	33	MHz
VGL	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
HAX	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
VGL	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
HDX	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
HSH	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
HGL	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
HSL	V_{PP} Hold After $\overline{PROG}$	10		µs
LGH	$\overline{PROG}$ Width	0.2	1	µs
/QV	Address to Data Valid		$48t_{CLCL}$	
LQV	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
HQZ	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
HBL	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
C	Byte Write Cycle Time		50	µs

ure 11. Flash Programming and Verification Waveforms – Parallel Mode

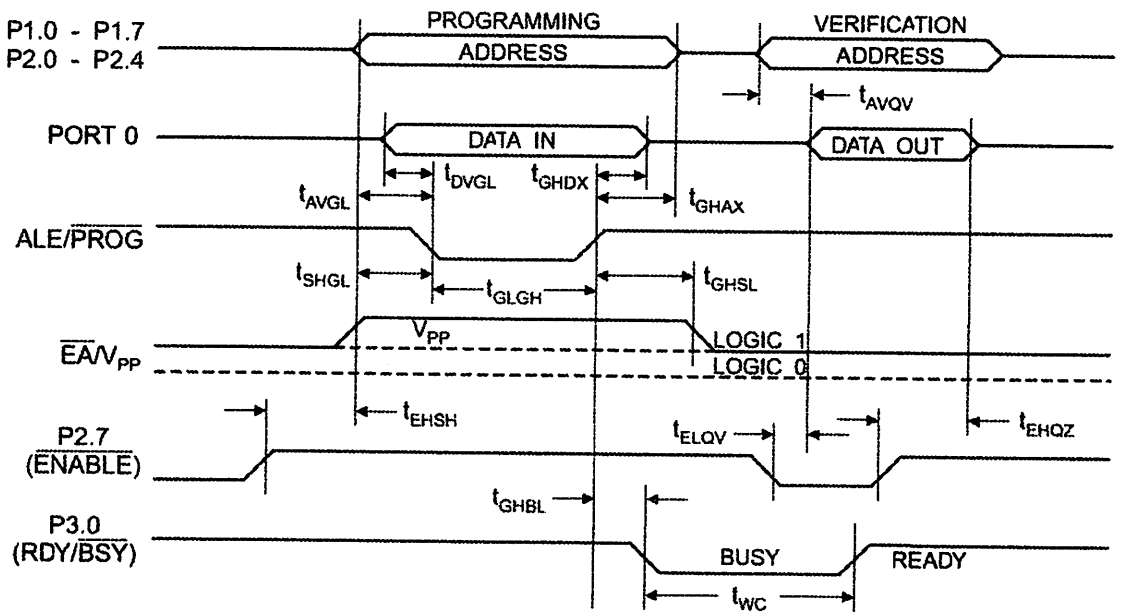
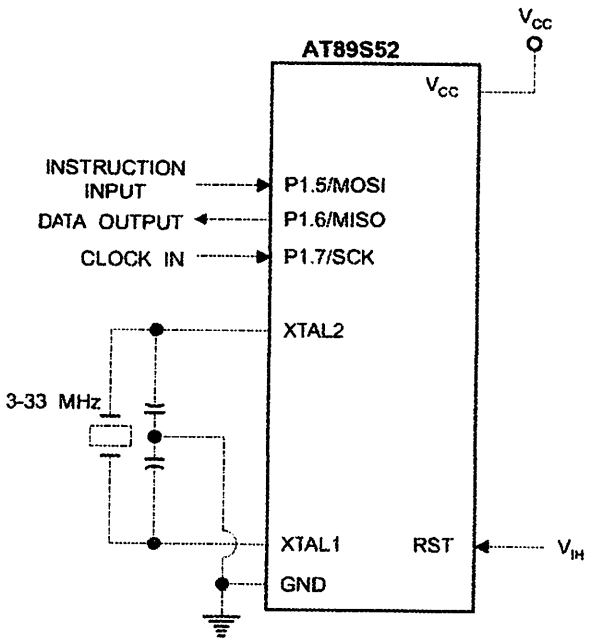
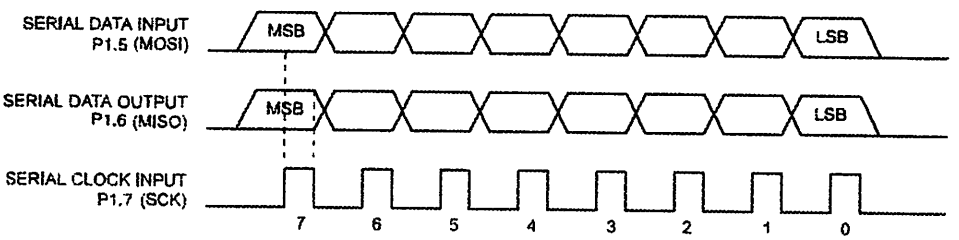


Figure 12. Flash Memory Serial Downloading



ash Programming d Verification aveforms – Serial ode

Figure 13. Serial Programming Waveforms



ble 11. Serial Programming Instruction Set

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

e: 1. B1 = 0, B2 = 0 —> Mode 1, no lock protection
B1 = 0, B2 = 1 —> Mode 2, lock bit 1 activated
B1 = 1, B2 = 0 —> Mode 3, lock bit 2 activated
B1 = 1, B2 = 1 —> Mode 4, lock bit 3 activated

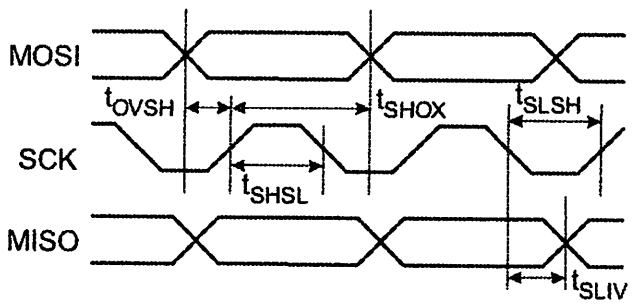
} Each of the lock bit modes needs to be activated sequentially before Mode 4 can be executed.

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

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

ırial Programming Characteristics

ıure 14. Serial Programming Timing



ıle 12. Serial Programming Characteristics, T_A = -40° C to 85° C, V_{CC} = 4.0 - 5.5V (Unless Otherwise Noted)

ımbol	Parameter	Min	Typ	Max	Units
CLCL	Oscillator Frequency	3		33	MHz
CL	Oscillator Period	30			ns
ISL	SCK Pulse Width High	8 t_{CLCL}			ns
SH	SCK Pulse Width Low	8 t_{CLCL}			ns
SH	MOSI Setup to SCK High	t_{CLCL}			ns
OX	MOSI Hold after SCK High	2 t_{CLCL}			ns
IV	SCK Low to MISO Valid	10	16	32	ns
ASE	Chip Erase Instruction Cycle Time			500	ms
VC	Serial Byte Write Cycle Time			64 t_{CLCL} + 400	µs

bsolute Maximum Ratings*

perating Temperature	-55°C to +125°C
torage Temperature	-65°C to +150°C
oltage on Any Pin ith Respect to Ground	-1.0V to +7.0V
laximum Operating Voltage	6.6V
C Output Current	15.0 mA

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

C Characteristics

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

ymbol	Parameter	Condition	Min	Max	Units
L	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
L1	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
H	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
H1	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
OL	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
OL1	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
OH	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
OH1	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
ST	Reset Pulldown Resistor		50	300	KΩ
	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode(1)	V_{CC} = 5.5V		50	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:
Port 0: 26 mA Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

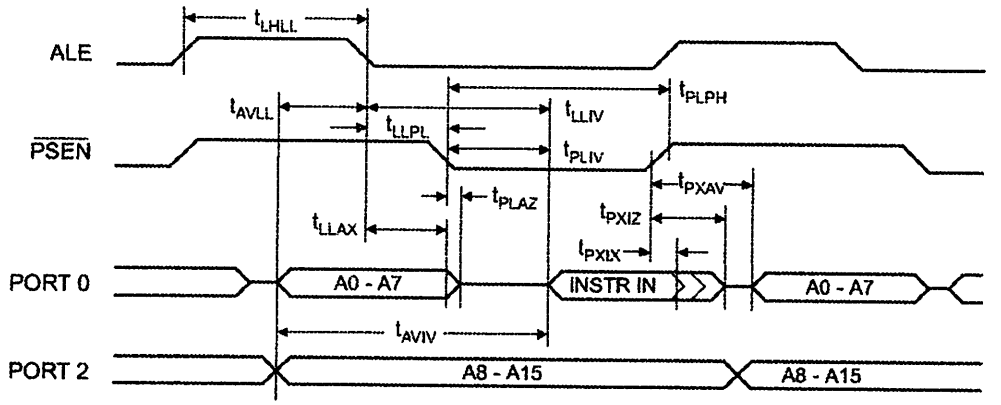
C Characteristics

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

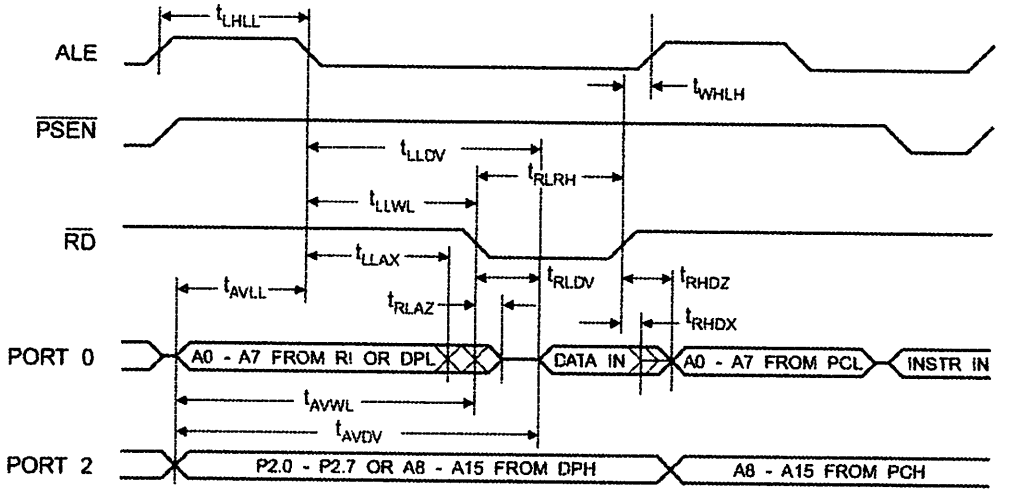
xternal Program and Data Memory Characteristics

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

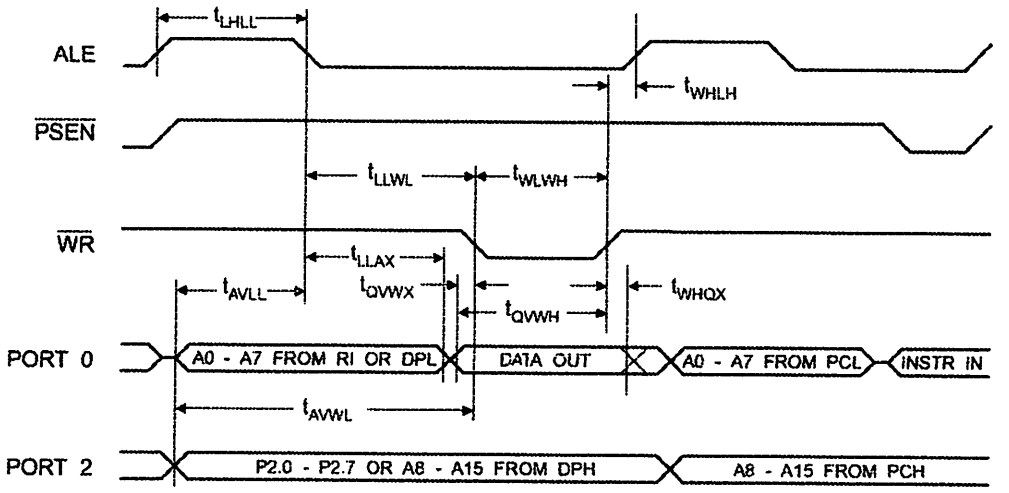
xternal Program Memory Read Cycle



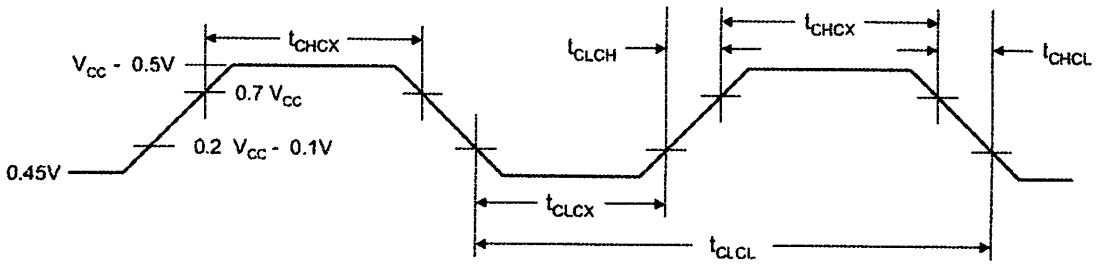
xternal Data Memory Read Cycle



xternal Data Memory Write Cycle



xternal Clock Drive Waveforms



xternal Clock Drive

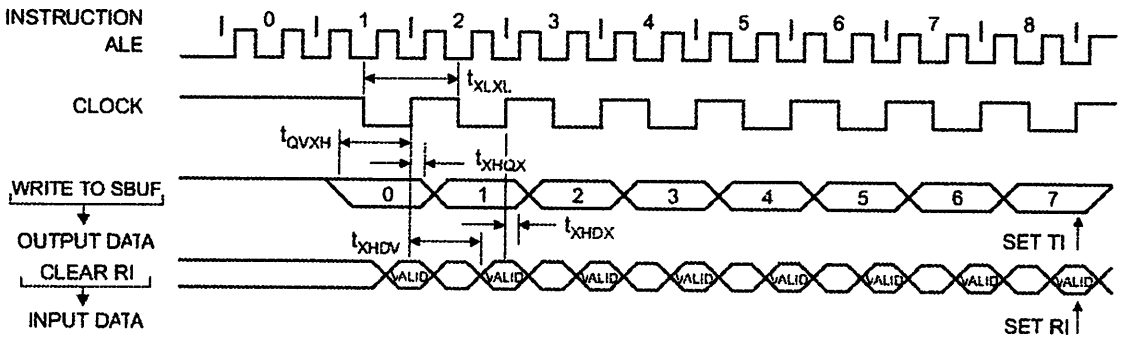
ymbol	Parameter	Min	Max	Units
CLCL	Oscillator Frequency	0	33	MHz
CL	Clock Period	30		ns
CX	High Time	12		ns
CX	Low Time	12		ns
CH	Rise Time		5	ns
CL	Fall Time		5	ns

erial Port Timing: Shift Register Mode Test Conditions

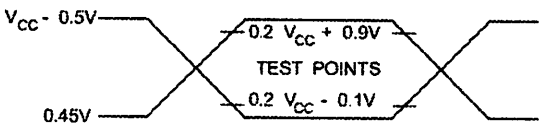
e values in this table are valid for V_{CC} = 4.0V to 5.5V and Load Capacitance = 80 pF.

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

ıift Register Mode Timing Waveforms

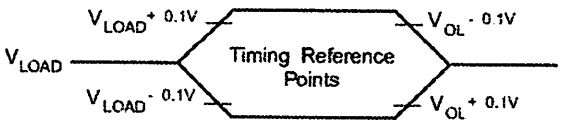


: Testing Input/Output Waveforms(1)



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.

ɔat Waveforms(1)



ı: 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.

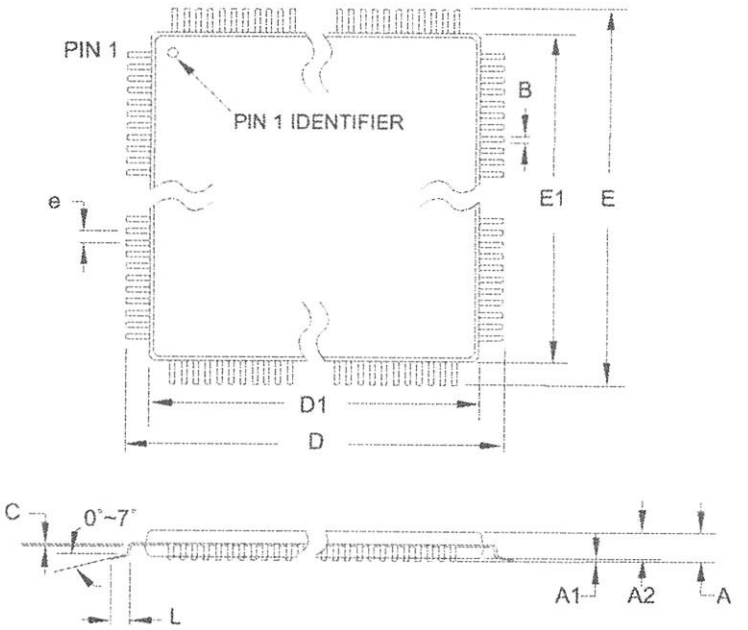
rdering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC AT89S52-24JC AT89S52-24PC AT89S52-24SC	44A 44J 40P6 42PS6	Commercial (0° C to 70° C)
		AT89S52-24AI AT89S52-24JI AT89S52-24PI AT89S52-24SI	44A 44J 40P6 42PS6	Industrial (-40° C to 85° C)
33	4.5V to 5.5V	AT89S52-33AC AT89S52-33JC AT89S52-33PC AT89S52-33SC	44A 44J 40P6 42PS6	Commercial (0° C to 70° C)

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

ackaging Information

4A – TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

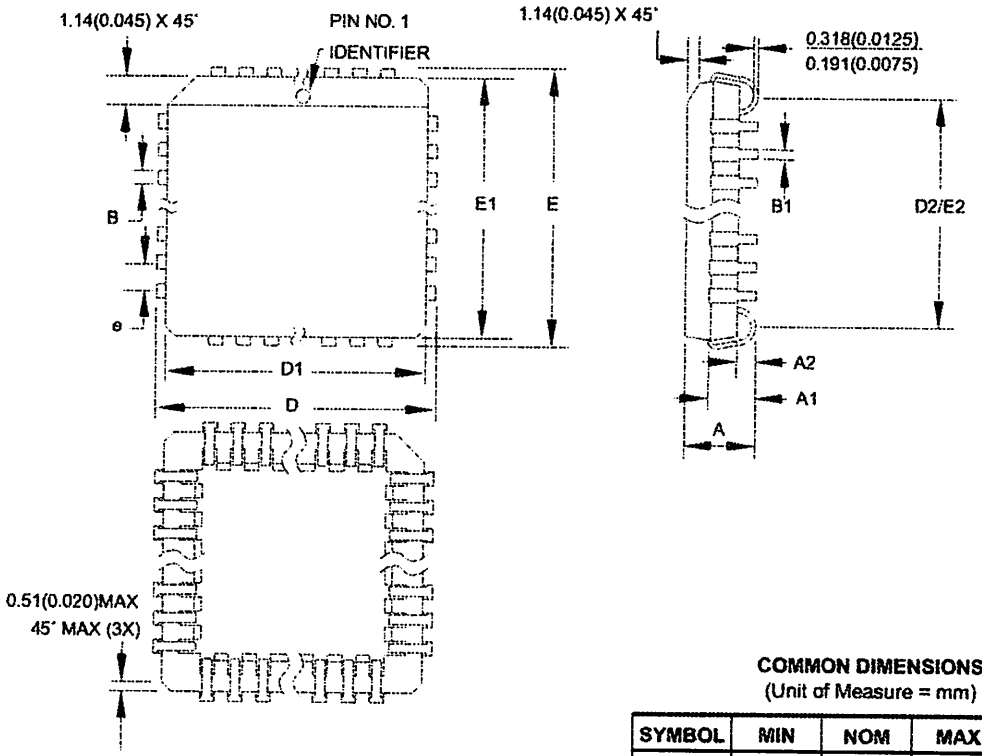
SYMBOL	MIN	NOM	MAX	NOTE
A	--	--	1.20	
A1	0.05	--	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	--	0.45	
C	0.09	--	0.20	
L	0.45	--	0.75	
e	0.80 TYP			

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

JJ – PLCC



COMMON DIMENSIONS
(Unit of Measure = mm)

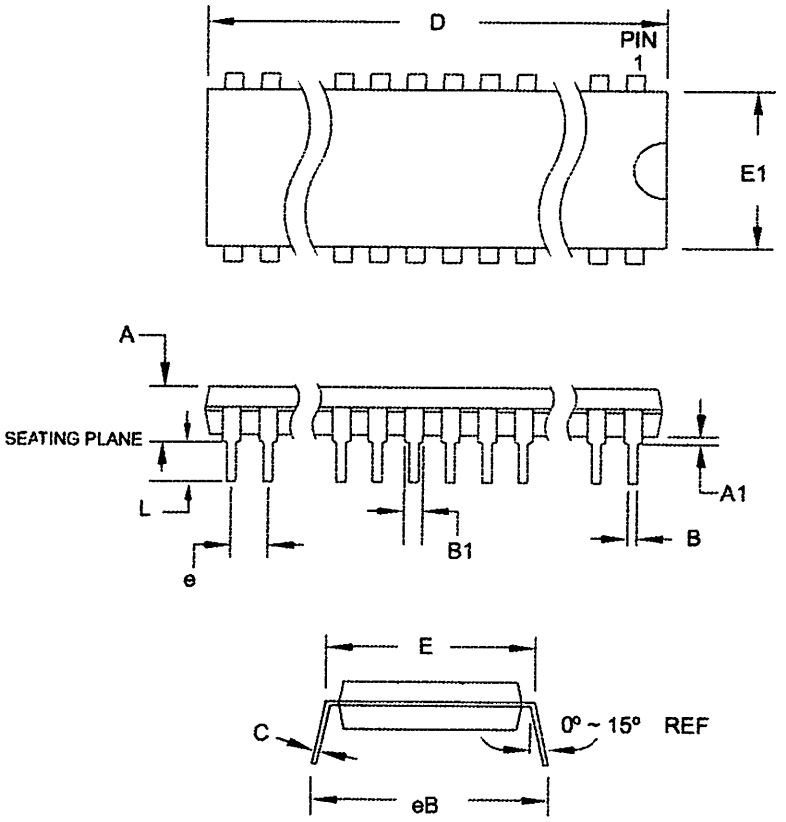
SYMBOL	MIN	NOM	MAX	NOTE
A	4.191	–	4.572	
A1	2.286	–	3.048	
A2	0.508	–	–	
D	17.399	–	17.653	
D1	16.510	–	16.662	Note 2
E	17.399	–	17.653	
E1	16.510	–	16.662	Note 2
D2/E2	14.986	–	16.002	
B	0.660	–	0.813	
B1	0.330	–	0.533	
e	1.270 TYP			

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

P6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

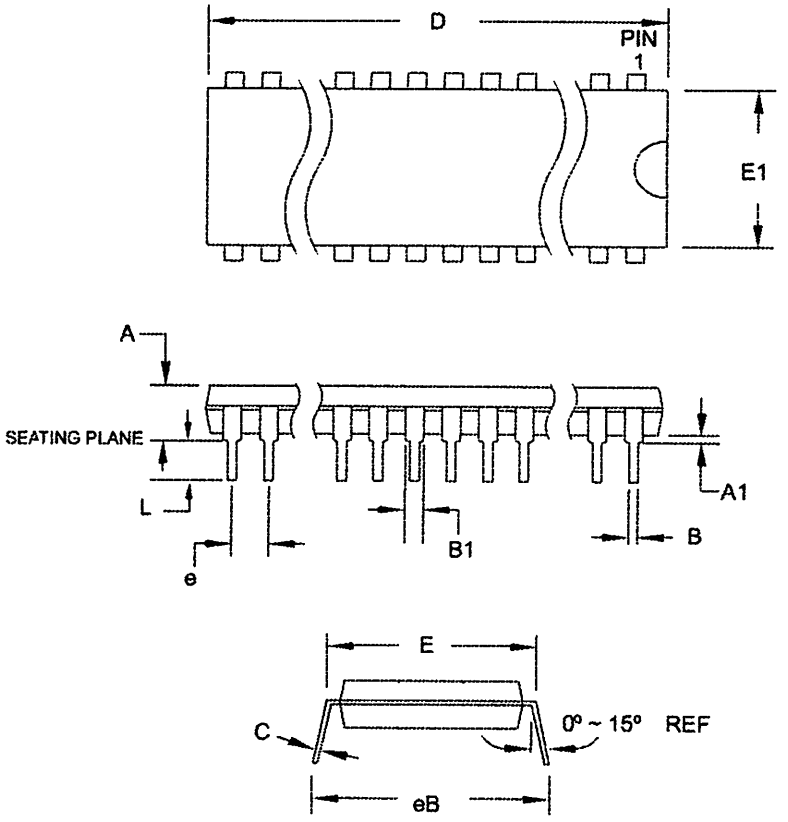
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

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

!PS6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.83	
A1	0.51	–	–	
D	36.70	–	36.96	Note 2
E	15.24	–	15.88	
E1	13.46	–	13.97	Note 2
B	0.38	–	0.56	
B1	0.76	–	1.27	
L	3.05	–	3.43	
C	0.20	–	0.30	
eB	–	–	18.55	
e	1.78 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").

11/6/03

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**42PS6**, 42-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	42PS6	A

National Semiconductor

October 2002

ADC0808/ADC0809
8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

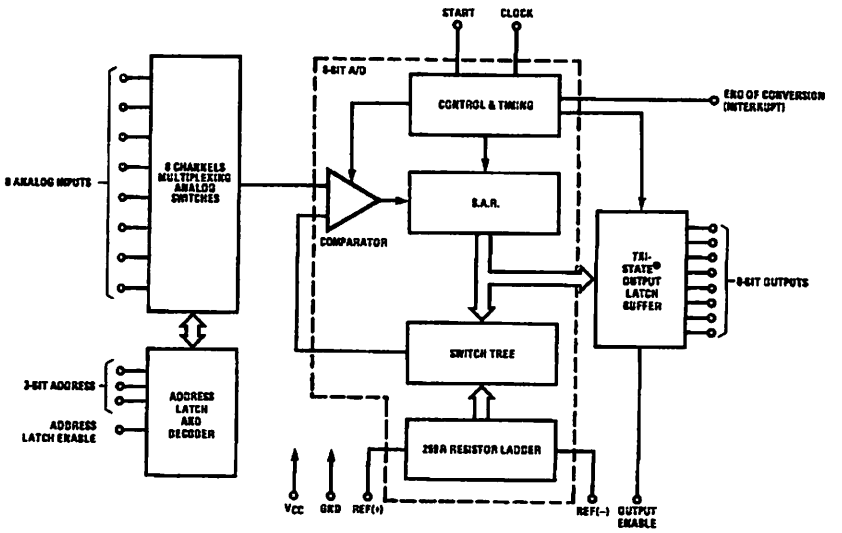
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- ADC0808 equivalent to MM74C949
- ADC0809 equivalent to MM74C949-1

Key Specifications

■ Resolution	8 Bits
■ Total Unadjusted Error	±½ LSB and ±1 LSB
■ Single Supply	5 V_{DC}
■ Low Power	15 mW
■ Conversion Time	100 µs

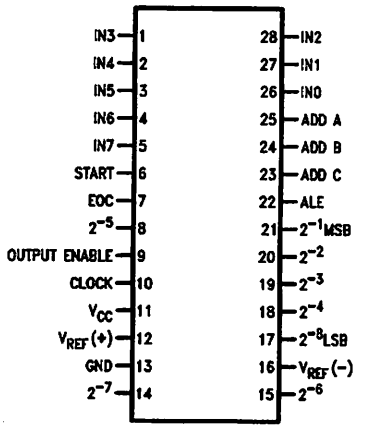
Block Diagram



See Ordering Information

Connection Diagrams

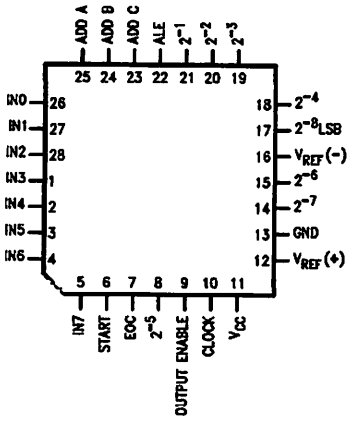
Dual-In-Line Package



00567211

Order Number ADC0808CCN or ADC0809CCN
See NS Package J28A or N28A

Molded Chip Carrier Package



00567212

Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Ordering Information

TEMPERATURE RANGE		-40°C to +85°C	
Error	±½ LSB Unadjusted	ADC0808CCN	ADC0808CCV
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier

Absolute Maximum Ratings (Notes 2, 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage at Any Pin	−0.3V to (V_{CC}+0.3V)
Except Control Inputs	
Voltage at Control Inputs	−0.3V to +15V
(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)	
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A=25°C	875 mW
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Molded Chip Carrier Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
ESD Susceptibility (Note 8)	400V

Operating Conditions (Notes 1, 2)

Temperature Range (Note 1)	$T_{MIN} \le T_A \le T_{MAX}$
ADC0808CCN,ADC0809CCN	−40°C≤T_A≤+85°C
ADC0808CCV, ADC0809CCV	−40°C≤T_A≤+85°C
Range of V_{CC} (Note 1)	4.5 V_{DC} to 6.0 V_{DC}

Electrical Characteristics

Converter Specifications: V_{CC}=5 V_{DC}=V_{REF+}, $V_{REF(-)}$=GND, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK}=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error	25°C			±½	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±¾	LSB
	ADC0809					
	Total Unadjusted Error	0°C to 70°C			±1	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±1¼	LSB
	Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
REF(+)	Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
$\frac{V_{REF(+)}+V_{REF(-)}}{2}$	Voltage, Center of Ladder		V_{CC}/2-0.1	V_{CC}/2	V_{CC}/2+0.1	V
REF(−)	Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
I	Comparator Input Current	f_c=640 kHz, (Note 6)	−2	±0.5	2	µA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75≤V_{CC}≤5.25V, −40°C≤T_A≤+85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
NALOG MULTIPLEXER						
F(+)	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=5V,				
		T_A=25°C		10	200	nA
		T_{MIN} to T_{MAX}			1.0	µA
F(−)	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=0,				
		T_A=25°C	−200	−10		nA
		T_{MIN} to T_{MAX}	−1.0			µA
ONTROL INPUTS						
(1)	Logical "1" Input Voltage		V_{CC}−1.5			V
(0)	Logical "0" Input Voltage				1.5	V
(1)	Logical "1" Input Current (The Control Inputs)	V_{IN}=15V			1.0	µA
(0)	Logical "0" Input Current (The Control Inputs)	V_{IN}=0	−1.0			µA
	Supply Current	f_{CLK}=640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, $4.75 \le V_{CC} \le 5.25V$, $-40^{\circ}C \le T_A \le +85^{\circ}C$ unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
DATA OUTPUTS AND EOC (INTERRUPT)						
$V_{OUT(1)}$	Logical "1" Output Voltage	V_{CC} = 4.75V I_{OUT} = −360µA I_{OUT} = −10µA		 2.4 4.5		 V(min) V(min)
$V_{OUT(0)}$	Logical "0" Output Voltage	I_O=1.6 mA			0.45	V
$V_{OUT(0)}$	Logical "0" Output Voltage EOC	I_O=1.2 mA			0.45	V
I_{OUT}	TRI-STATE Output Current	V_O=5V V_O=0	 −3		3	µA µA

Electrical Characteristics

Timing Specifications $V_{CC}=V_{REF(+)}=5V$, $V_{REF(-)}=GND$, $t_r=t_f=20$ ns and $T_A=25^{\circ}C$ unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(*Figure 5*)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(*Figure 5*)		100	200	ns
t_S	Minimum Address Set-Up Time	(*Figure 5*)		25	50	ns
t_H	Minimum Address Hold Time	(*Figure 5*)		25	50	ns
t_D	Analog MUX Delay Time From ALE	R_S=0Ω (*Figure 5*)		1	2.5	µs
t_{H1}, t_{H0}	OE Control to Q Logic State	C_L=50 pF, R_L=10k (*Figure 8*)		125	250	ns
t_{1H}, t_{0H}	OE Control to Hi-Z	C_L=10 pF, R_L=10k (*Figure 8*)		125	250	ns
t_c	Conversion Time	f_c=640 kHz, (*Figure 5*) (Note 7)	90	100	116	µs
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(*Figure 5*)	0		8+2 µS	Clock Periods
C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{DC}.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC}n supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute $0V_{DC}$ to $5V_{DC}$ input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See *Figure 3*. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See *Figure 13*.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (*Figure 6*). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. *Table 1* shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (*Figure 1*) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in *Figure 1* are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +½ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. *Figure 2* shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion start pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)

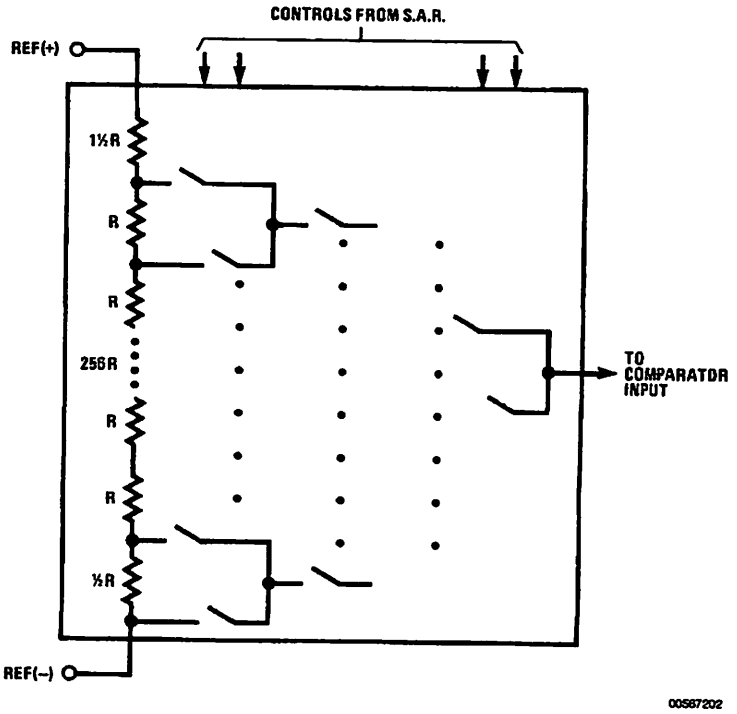


FIGURE 1. Resistor Ladder and Switch Tree

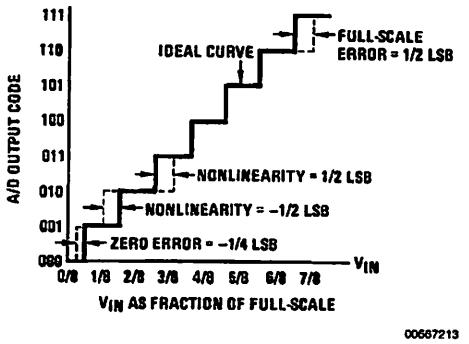


FIGURE 2. 3-Bit A/D Transfer Curve

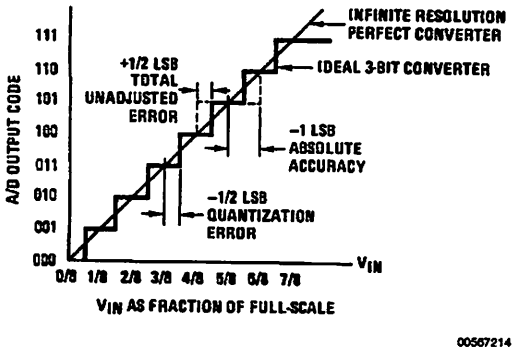


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

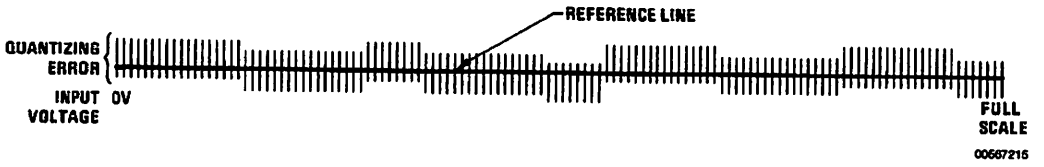


FIGURE 4. Typical Error Curve

Timing Diagram

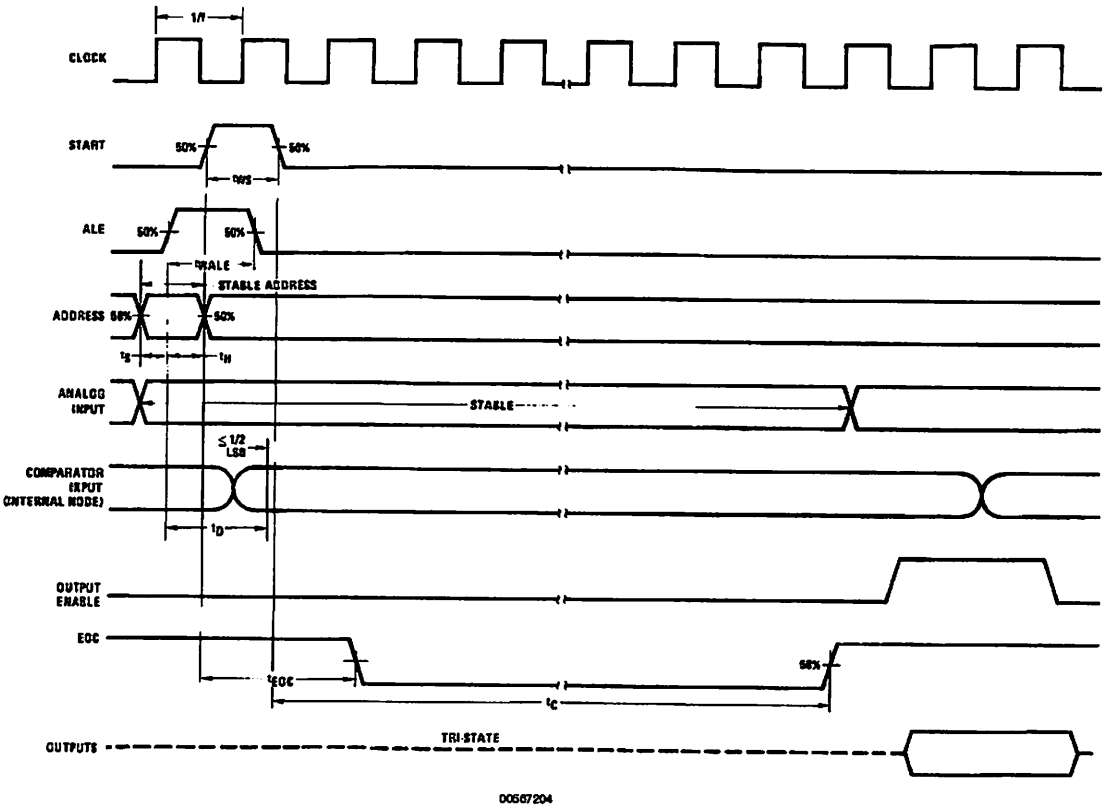


FIGURE 5.

Typical Performance Characteristics

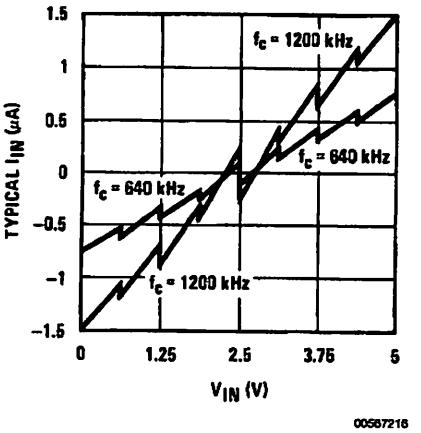


FIGURE 6. Comparator I_{IN} vs V_{IN} (V_{CC}=V_{REF}=5V)

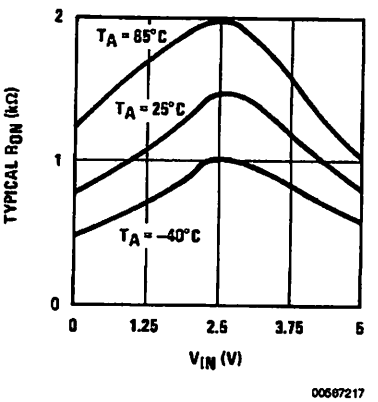


FIGURE 7. Multiplexer R_{ON} vs V_{IN} (V_{CC}=V_{REF}=5V)

TRI-STATE Test Circuits and Timing Diagrams

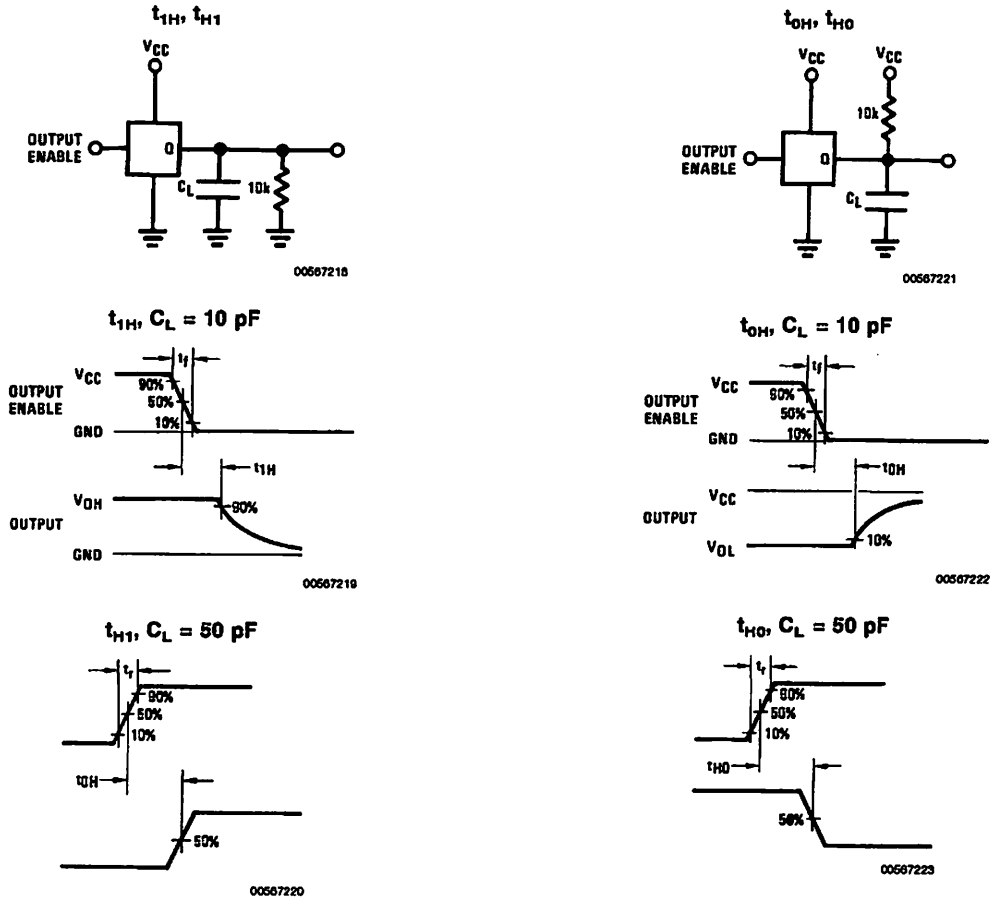


FIGURE 8.

Applications Information

OPERATION

1.0 RATIOMETRIC CONVERSION

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratiometric conversion systems. In ratiometric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN}}{V_{fs}-V_Z}=\frac{D_X}{D_{MAX}-D_{MIN}} \qquad (1)$$

V_{IN}=Input voltage into the ADC0808

V_{fs}=Full-scale voltage

V_Z=Zero voltage

D_X=Data point being measured

D_{MAX}=Maximum data limit

D_{MIN}=Minimum data limit

A good example of a ratiometric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, (*Figure 9*).

Ratiometric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if $V_{CC}=V_{REF}=5.12V$, then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

Applications Information (Continued)

2.0 RESISTOR LADDER LIMITATIONS

The voltages from the resistor ladder are compared to the selected into 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(−), should not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.

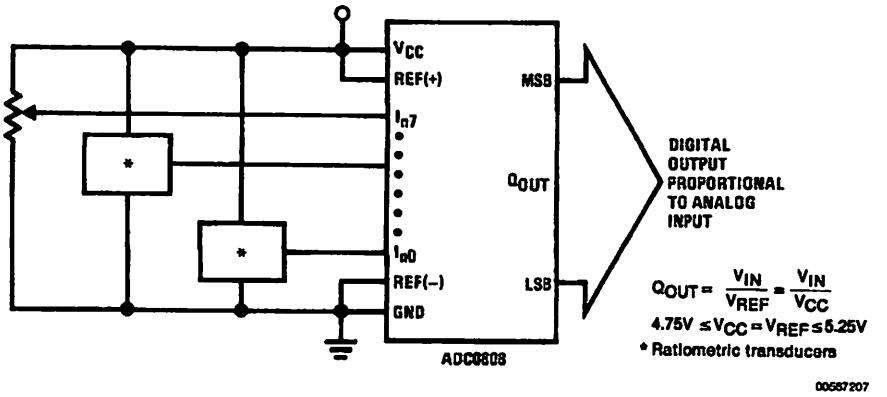
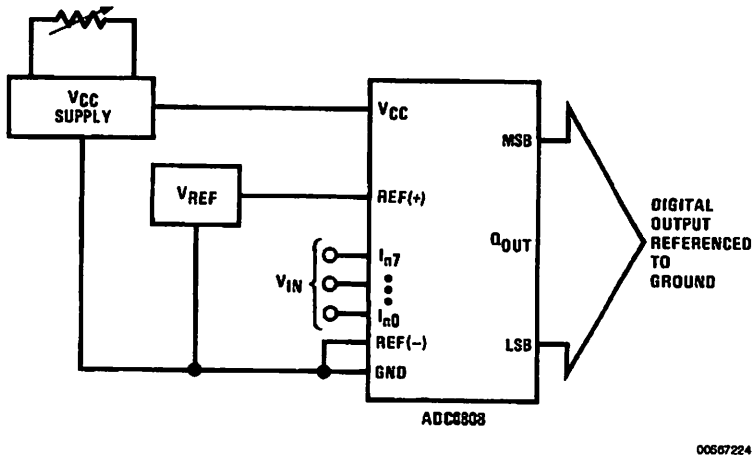


FIGURE 9. Ratiometric Conversion System

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In *Figure 11* a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in *Figure 12*. The LM301 is overcompensated to insure stability when loaded by the 10 µF output capacitor.

The top and bottom ladder voltages cannot exceed V_{CC} and ground, respectively, but they can be symmetrically less than V_{CC} and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In *Figure 13*, a 2.5V reference is symmetrically centered about $V_{CC}/2$ since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.

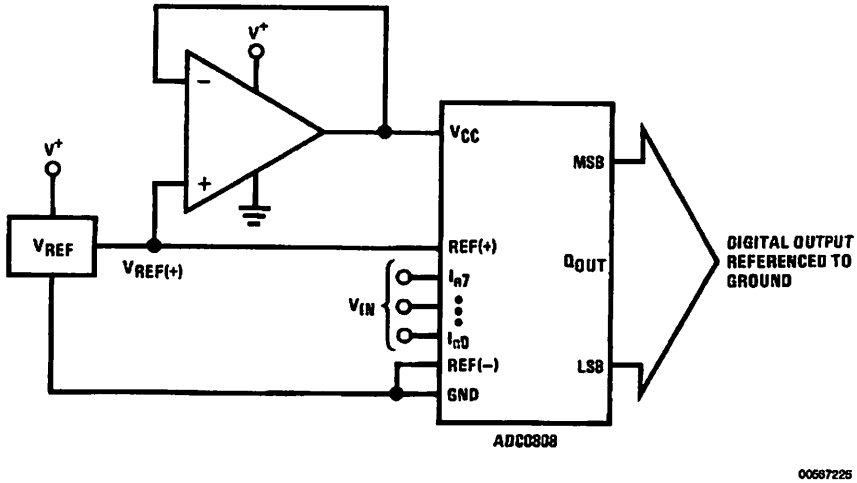
Applications Information (Continued)



$$D_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$4.75V \le V_{CC} = V_{REF} \le 5.25V$

FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply



$$D_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$4.75V \le V_{CC} = V_{REF} \le 5.25V$

FIGURE 11. Ground Referenced Conversion System with Reference Generating V_{CC} Supply

Applications Information (Continued)

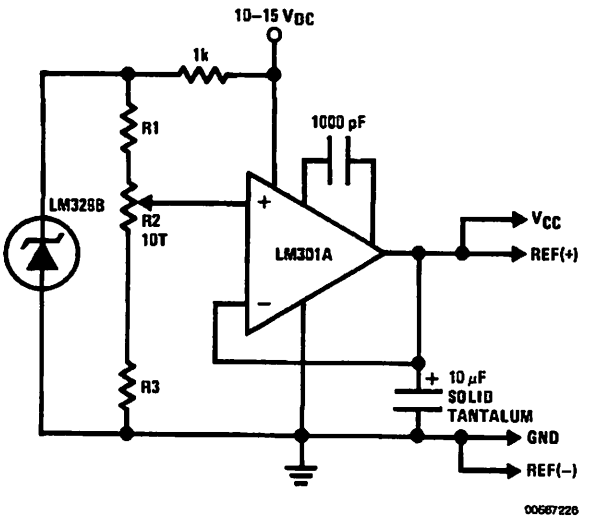
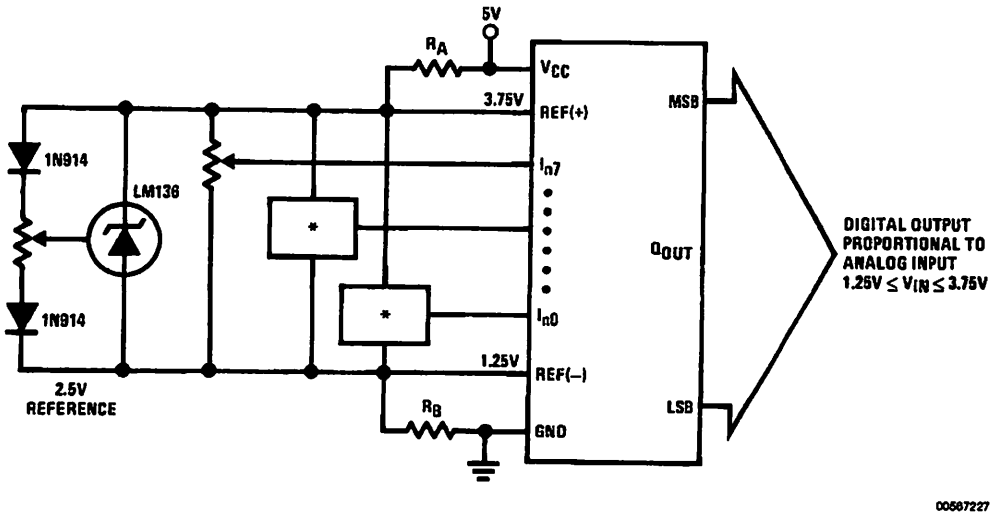


FIGURE 12. Typical Reference and Supply Circuit



$R_A = R_B$

*Ratiometric transducers

FIGURE 13. Symmetrically Centered Reference

3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left\{(V_{REF(+)} - V_{REF(-)})\left[\frac{N}{256} + \frac{1}{512}\right] \pm V_{TUE}\right\} + V_{REF(-)} \quad (2)$$

The center of an output code N is given by:

$$V_{IN}\left\{(V_{REF(+)} - V_{REF(-)})\left[\frac{N}{256}\right] \pm V_{TUE}\right\} + V_{REF(-)} \quad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{Absolute Accuracy} \quad (4)$$

Where: V_{IN}=Voltage at comparator input

$V_{REF(+)}$=Voltage at Ref(+)

$V_{REF(-)}$=Voltage at Ref(–)

V_{TUE}=Total unadjusted error voltage (typically $V_{REF(+)} \div 512$)

Applications Information (Continued)

4.0 ANALOG COMPARATOR INPUTS

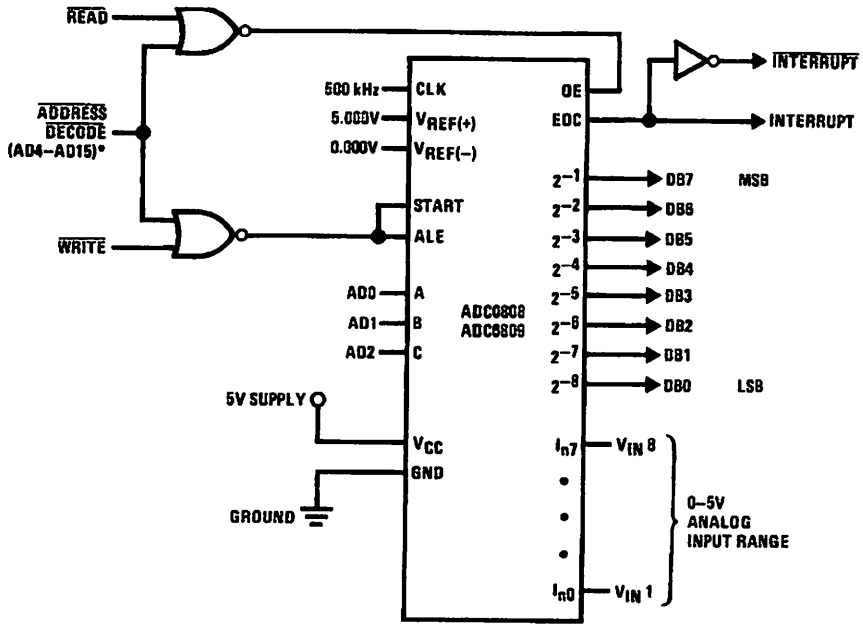
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in *Figure 6.*

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

Typical Application



00567210

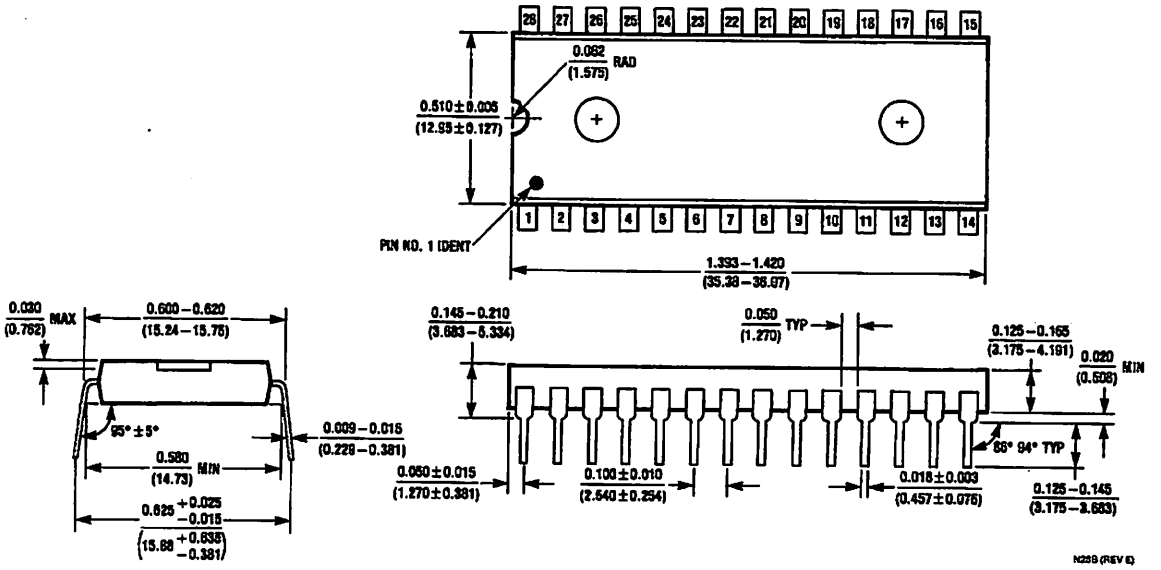
*Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

TABLE 2. Microprocessor Interface Table

PROCESSOR	$\overline{\text{READ}}$	$\overline{\text{WRITE}}$	INTERRUPT (COMMENT)
8080	$\overline{\text{MEMR}}$	$\overline{\text{MEMW}}$	INTR (Thru RST Circuit)
8085	$\overline{\text{RD}}$	$\overline{\text{WR}}$	INTR (Thru RST Circuit)
Z-80	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{INT}}$ (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA•φ2•R/W	VMA•φ•$\overline{\text{R/W}}$	$\overline{\text{IRQA}}$ or $\overline{\text{IRQB}}$ (Thru PIA)

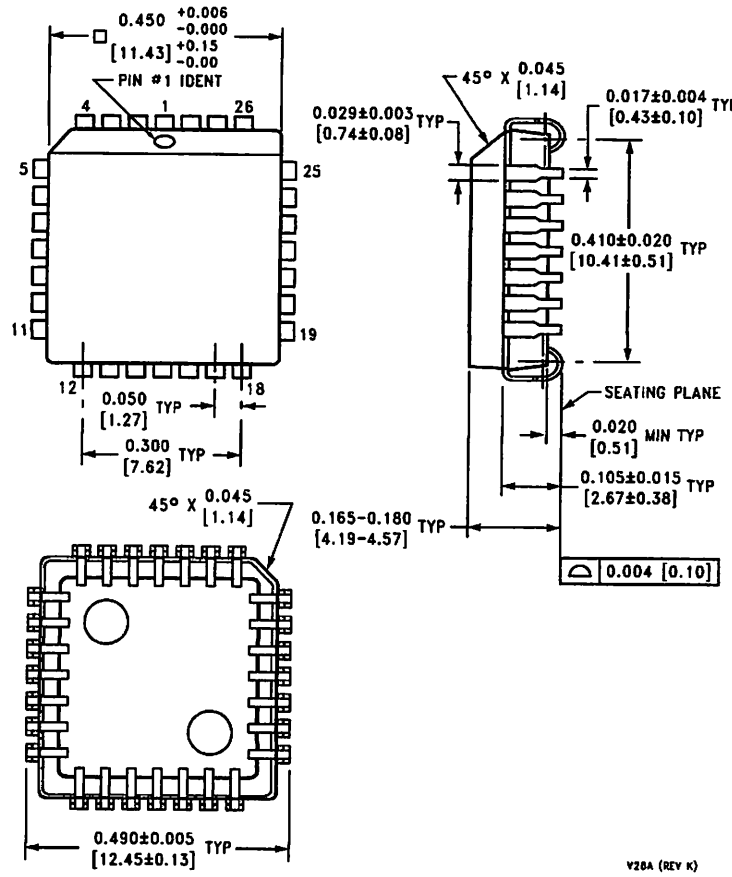
Physical Dimensions inches (millimeters)

unless otherwise noted



Molded Dual-In-Line Package (N)
Order Number ADC0808CCN or ADC0809CCN
NS Package Number N28B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Chip Carrier (V)
Order Number ADC0808CCV or ADC0809CCV
NS Package Number V28A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Email: support@nsc.com

www.national.com

National Semiconductor Europe
Fax: +49 (0) 180-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 69 9508 6208
English Tel: +44 (0) 870 24 0 2171
Français Tel: +33 (0) 1 41 91 8790

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: ap.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

National Semiconductor

December 1994

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit μP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE® output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

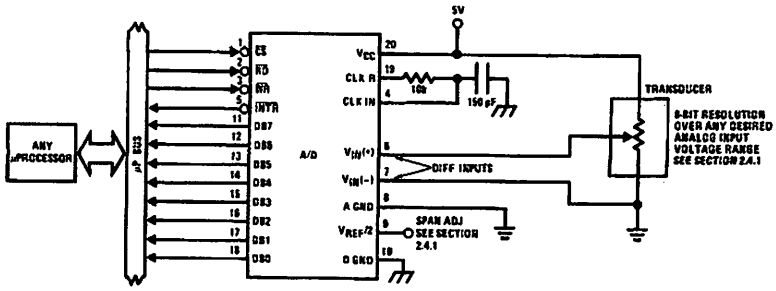
Features

- Compatible with 8080 μP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

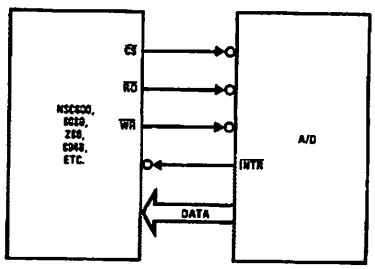
- Resolution: 8 bits
- Total error: ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time: 100 μs

Typical Applications



TL/H/5671–1

8080 Interface



TL/H/5671–31

Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF}/2 = 2.500\ V_{DC}$ (No Adjustments)	$V_{REF}/2$ = No Connection (No Adjustments)
ADC0801	±¼ LSB		
ADC0802		±½ LSB	
ADC0803	±½ LSB		
ADC0804		±1 LSB	
ADC0805			±1 LSB

TRI-STATE® is a registered trademark of National Semiconductor Corp.
Z-80® is a registered trademark of Zilog Corp.

RRD-B30M115/Printed in U. S. A.

Absolute Maximum Ratings (Notes 1 & 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage	
Logic Control Inputs	−0.3V to +18V
At Other Input and Outputs	−0.3V to (V_{CC} + 0.3V)
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A = 25°C	875 mW
ESD Susceptibility (Note 10)	800V

Operating Ratings (Notes 1 & 2)

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
ADC0801/02LJ, ADC0802LJ/883	−55°C ≤ T_A ≤ +125°C
ADC0801/02/03/04LCJ	−40°C ≤ T_A ≤ +85°C
ADC0801/02/03/05LCN	−40°C ≤ T_A ≤ +85°C
ADC0804LCN	0°C ≤ T_A ≤ +70°C
ADC0802/03/04LCV	0°C ≤ T_A ≤ +70°C
ADC0802/03/04LCWM	0°C ≤ T_A ≤ +70°C
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

The following specifications apply for V_{CC} = 5 V_{DC}, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK} = 640 kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±¼	LSB
ADC0802: Total Unadjusted Error (Note 8)	$V_{REF}/2$ = 2.500 V_{DC}			±½	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±½	LSB
ADC0804: Total Unadjusted Error (Note 8)	$V_{REF}/2$ = 2.500 V_{DC}			±1	LSB
ADC0805: Total Unadjusted Error (Note 8)	$V_{REF}/2$-No Connection			±1	LSB
$V_{REF}/2$ Input Resistance (Pin 9)	ADC0801/02/03/05 ADC0804 (Note 9)	2.5 0.75	8.0 1.1		kΩ kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	Gnd−0.05		V_{CC} + 0.05	V_{DC}
DC Common-Mode Error	Over Analog Input Voltage Range		±1/16	±⅛	LSB
Power Supply Sensitivity	V_{CC} = 5 V_{DC} ±10% Over Allowed V_{IN}(+) and V_{IN}(−) Voltage Range (Note 4)		±1/16	±⅛	LSB

AC Electrical Characteristics

The following specifications apply for V_{CC} = 5 V_{DC} and T_A = 25°C unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_C	Conversion Time	f_{CLK} = 640 kHz (Note 6)	103		114	μs
T_C	Conversion Time	(Note 5, 6)	66		73	$1/f_{CLK}$
f_{CLK}	Clock Frequency Clock Duty Cycle	V_{CC} = 5V, (Note 5) (Note 5)	100 40	640	1460 60	kHz %
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$ = 0 V_{DC}, f_{CLK} = 640 kHz	8770		9708	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}$ = 0 V_{DC} (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	C_L = 100 pF		135	200	ns
t_{1H}, t_{0H}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	C_L = 10 pF, R_L = 10k (See TRI-STATE Test Circuits)		125	200	ns
t_{WI}, t_{RI}	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (1)	Logical "1" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 5.25 V_{DC}	2.0		15	V_{DC}

AC Electrical Characteristics (Continued)

The following specifications apply for V_{CC} = 5V_{DC} and $T_{MIN} \le T_A \le T_{MAX}$, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	V_{CC} = 4.75 V_{DC}			0.8	V_{DC}
I_{IN} (1)	Logical "1" Input Current (All Inputs)	V_{IN} = 5 V_{DC}		0.005	1	μA_{DC}
I_{IN} (0)	Logical "0" Input Current (All Inputs)	V_{IN} = 0 V_{DC}	−1	−0.005		μA_{DC}
CLOCK IN AND CLOCK R						
V_T+	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_T-	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis $(V_T+)-(V_T-)$		0.6	1.3	2.0	V_{DC}
V_{OUT} (0)	Logical "0" CLK R Output Voltage	I_O = 360 μA V_{CC} = 4.75 V_{DC}			0.4	V_{DC}
V_{OUT} (1)	Logical "1" CLK R Output Voltage	I_O = −360 μA V_{CC} = 4.75 V_{DC}	2.4			V_{DC}
DATA OUTPUTS AND $\overline{INTR}$						
V_{OUT} (0)	Logical "0" Output Voltage Data Outputs $\overline{INTR}$ Output	 I_{OUT} = 1.6 mA, V_{CC} = 4.75 V_{DC} I_{OUT} = 1.0 mA, V_{CC} = 4.75 V_{DC}			 0.4 0.4	 V_{DC} V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O = −360 μA, V_{CC} = 4.75 V_{DC}	2.4			V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O = −10 μA, V_{CC} = 4.75 V_{DC}	4.5			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	V_{OUT} = 0 V_{DC} V_{OUT} = 5 V_{DC}	−3		 3	μA_{DC} μA_{DC}
I_{SOURCE}		V_{OUT} Short to Gnd, T_A = 25°C	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC}, T_A = 25°C	9.0	16		mA_{DC}
POWER SUPPLY						
I_{CC}	Supply Current (Includes Ladder Current) ADC0801/02/03/04LCJ/05 ADC0804LCN/LCV/LCWM	f_{CLK} = 640 kHz, $V_{REF}/2$ = NC, T_A = 25°C and $\overline{CS}$ = 5V		 1.1 1.9	 1.8 2.5	 mA mA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{DC}.

Note 4: For $V_{IN}(-) \ge V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct—especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Accuracy is guaranteed at f_{CLK} = 640 kHz. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see *Figure 2* and section 2.0.

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the convertor in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

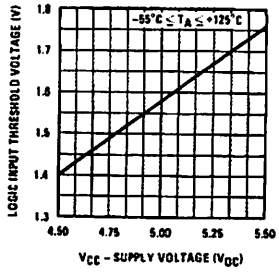
Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and *Figure 5*.

Note 9: The $V_{REF}/2$ pin is the center point of a two-resistor divider connected from V_{CC} to ground. In all versions of the ADC0801, ADC0802, ADC0803, and ADC0805, and in the ADC0804LCJ, each resistor is typically 16 kΩ. In all versions of the ADC0804 except the ADC0804LCJ, each resistor is typically 2.2 kΩ.

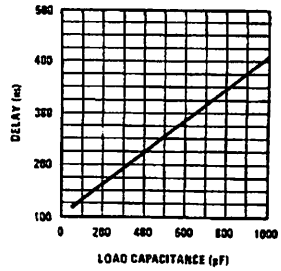
Note 10: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Typical Performance Characteristics

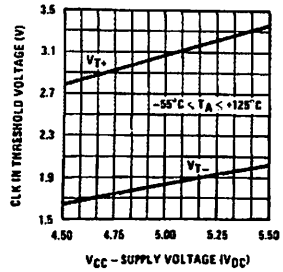
Logic Input Threshold Voltage vs. Supply Voltage



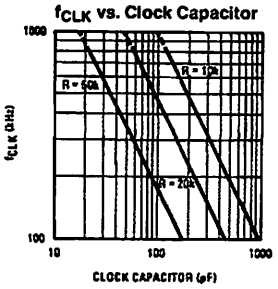
Delay From Falling Edge of $\overline{RD}$ to Output Data Valid vs. Load Capacitance



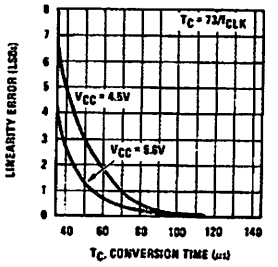
CLK IN Schmitt Trip Levels vs. Supply Voltage



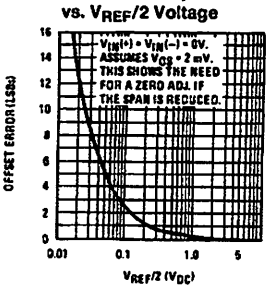
f_{CLK} vs. Clock Capacitor



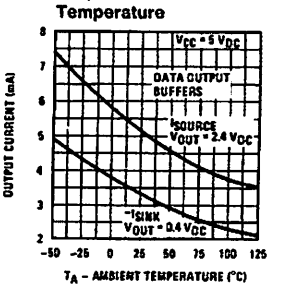
Full-Scale Error vs Conversion Time



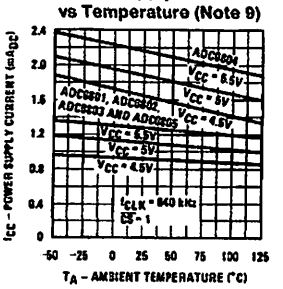
Effect of Unadjusted Offset Error vs. $V_{REF}/2$ Voltage



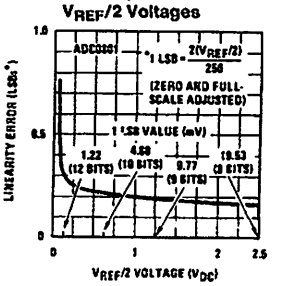
Output Current vs Temperature



Power Supply Current vs Temperature (Note 9)

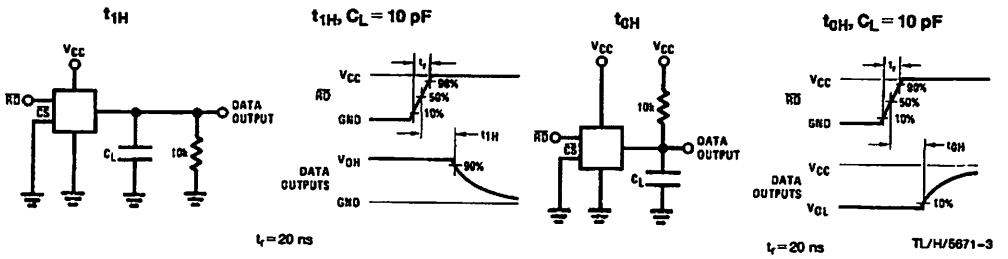


Linearity Error at Low $V_{REF}/2$ Voltages



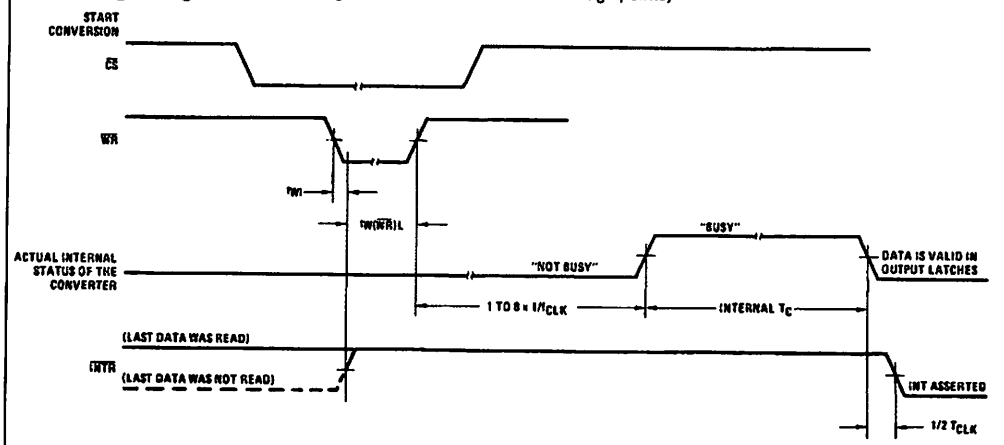
TL/H/5671–2

TRI-STATE Test Circuits and Waveforms

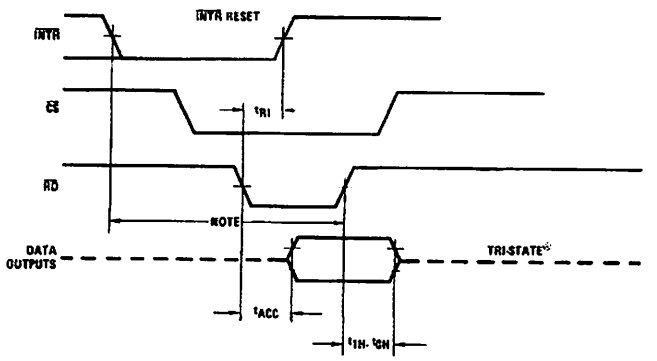


TL/H/5671–3

Timing Diagrams (All timing is measured from the 50% voltage points)



Output Enable and Reset $\overline{INTR}$

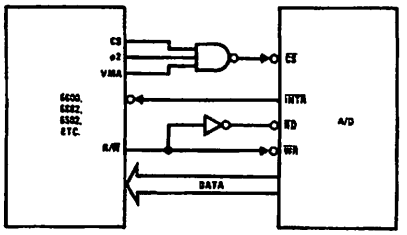


Note: Read strobe must occur 8 clock periods ($8/f_{CLK}$) after assertion of interrupt to guarantee reset of $\overline{INTR}$.

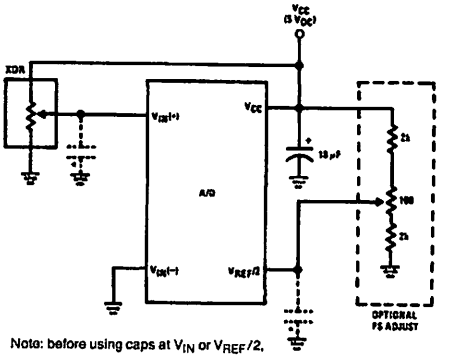
TL/H/5671–4

Typical Applications (Continued)

6800 Interface

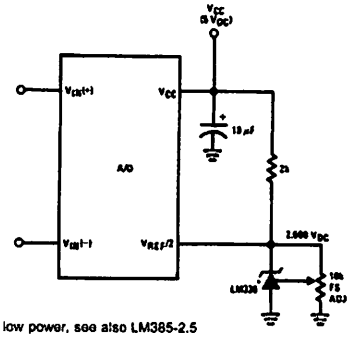


Ratiometric with Full-Scale Adjust



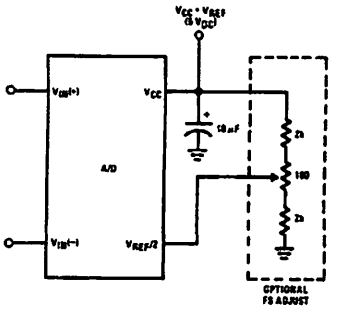
Note: before using caps at V_{IN} or $V_{REF}/2$, see section 2.3.2 Input Bypass Capacitors.

Absolute with a 2.500V Reference

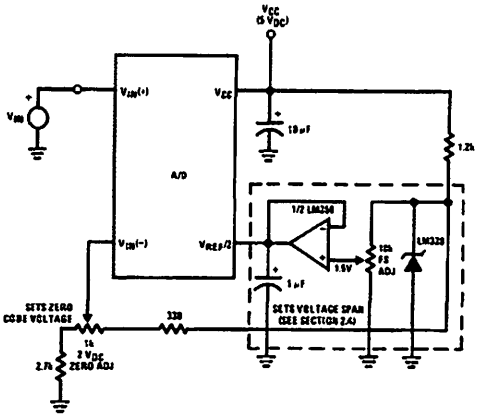


*For low power, see also LM385-2.5

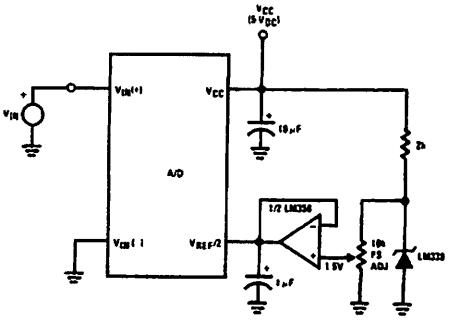
Absolute with a 5V Reference



Zero-Shift and Span Adjust: $2V \leq V_{IN} \leq 5V$



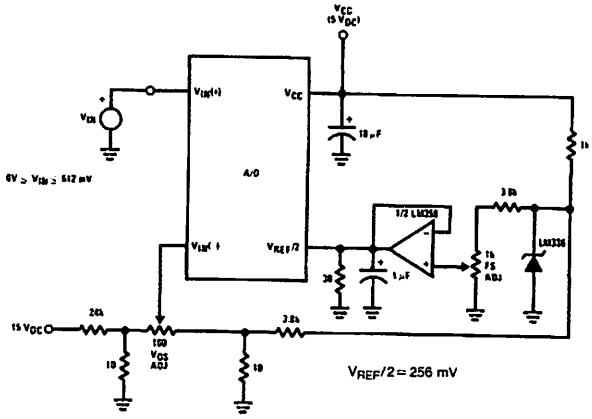
Span Adjust: $0V \leq V_{IN} \leq 3V$



TL/H/5671–5

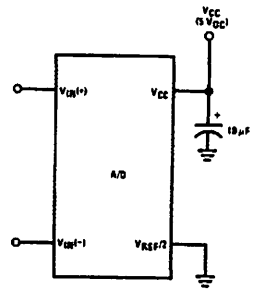
Typical Applications (Continued)

Directly Converting a Low-Level Signal



$V_{REF}/2 = 256$ mV

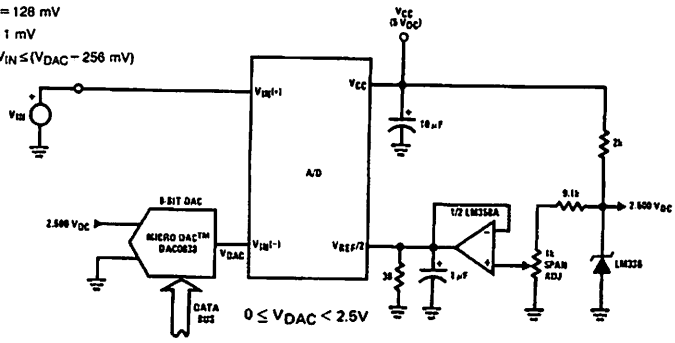
A µP Interfaced Comparator



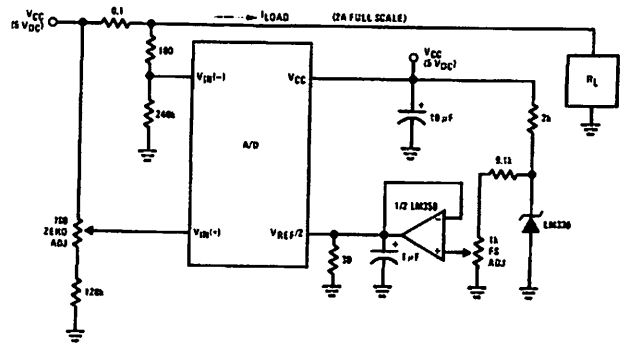
For: $V_{IN}(+) > V_{IN}(-)$
Output = FF_{HEX}
For: $V_{IN}(+) < V_{IN}(-)$
Output = 00_{HEX}

1 mV Resolution with µP Controlled Range

$V_{REF}/2 = 128$ mV
1 LSB = 1 mV
$V_{DAC} \leq V_{IN} \leq (V_{DAC} - 256$ mV)



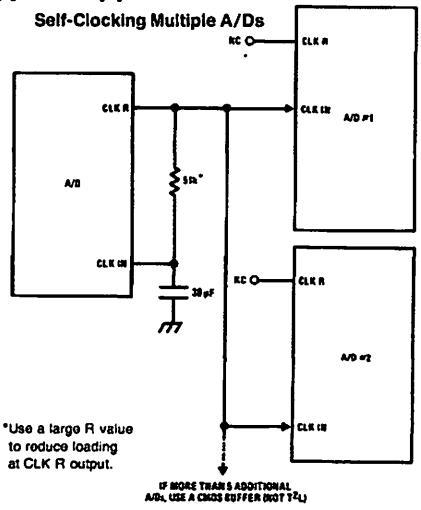
Digitizing a Current Flow



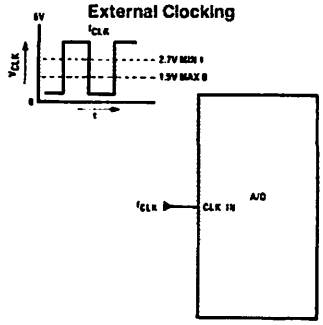
TL/H/5671–6

Typical Applications (Continued)

Self-Clocking Multiple A/Ds

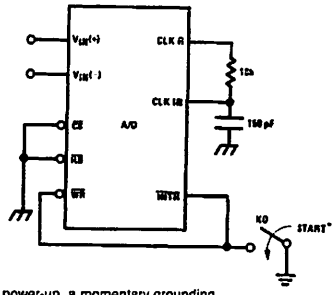


External Clocking



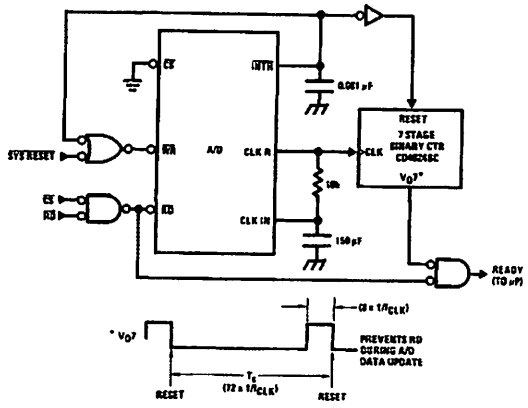
100 kHz ≤ f_{CLK} ≤ 1460 kHz

Self-Clocking in Free-Running Mode

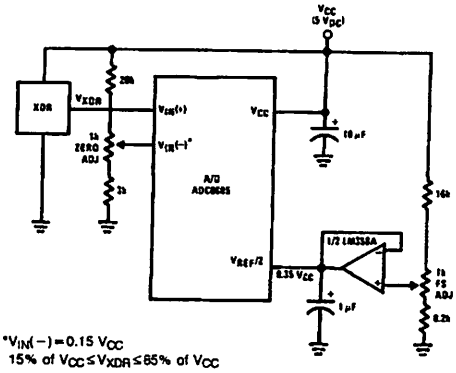


*After power-up, a momentary grounding of the $\overline{WR}$ input is needed to guarantee operation.

µP Interface for Free-Running A/D

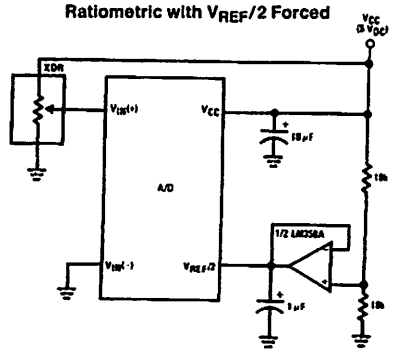


Operating with "Automotive" Ratiometric Transducers



*$V_{IN}(-) = 0.15\ V_{CC}$
15% of $V_{CC} \leq V_{XDR} \leq 85\%$ of V_{CC}

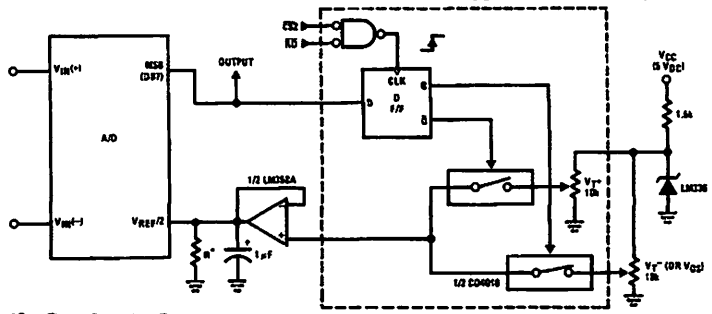
Ratiometric with $V_{REF}/2$ Forced



TL/H/5671–7

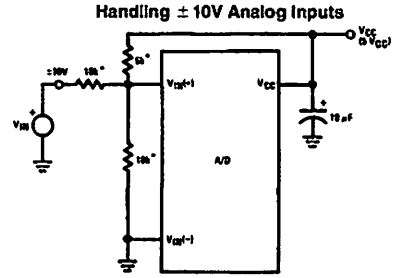
Typical Applications (Continued)

µP Compatible Differential-Input Comparator with Pre-Set V_{OS} (with or without Hysteresis)



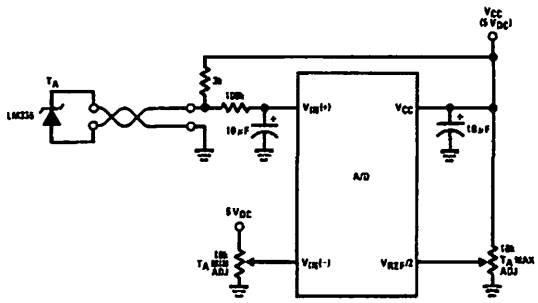
*See *Figure 5* to select R value
DB7 = "1" for $V_{IN}(+) > V_{IN}(-) - (V_{REF}/2)$
Omit circuitry within the dotted area if hysteresis is not needed

Handling ± 10V Analog Inputs

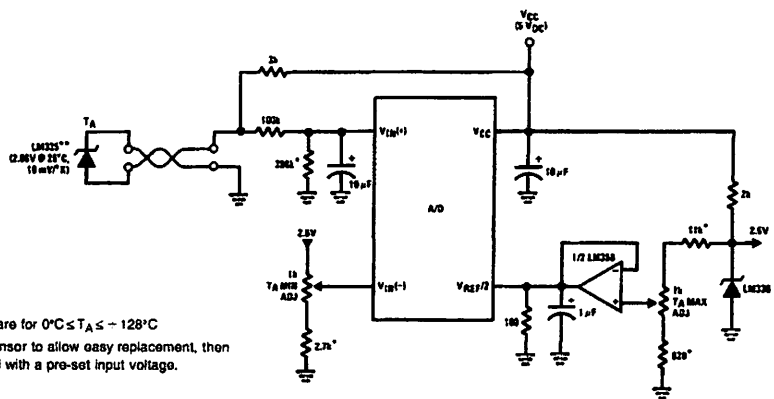


*Beckman Instruments #694-3-R10K resistor array

Low-Cost, µP Interfaced, Temperature-to-Digital Converter



µP Interfaced Temperature-to-Digital Converter

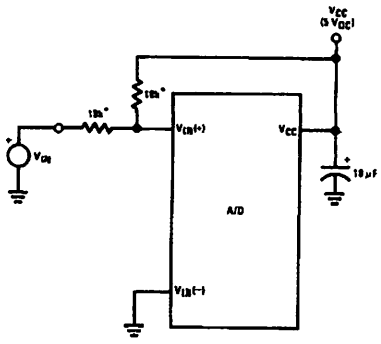


*Circuit values shown are for $0°C \leq T_A \leq +128°C$
**Can calibrate each sensor to allow easy replacement, then A/D can be calibrated with a pre-set input voltage.

TL/H/5871–6

Typical Applications (Continued)

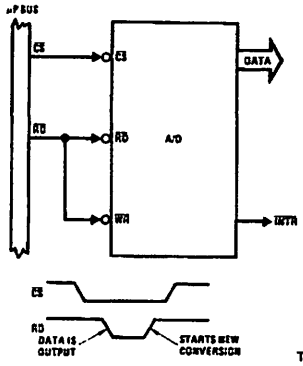
Handling ±5V Analog Inputs



TL/H/5671–33

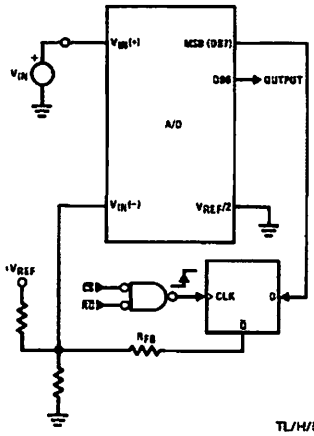
*Beckman Instruments #694-3-R10K resistor array

Read-Only Interface



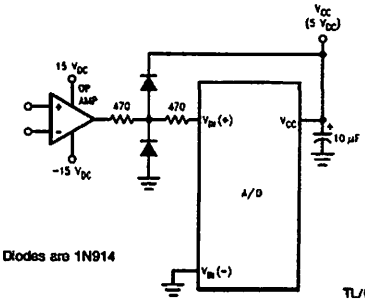
TL/H/5671–34

µP Interfaced Comparator with Hysteresis



TL/H/5671–35

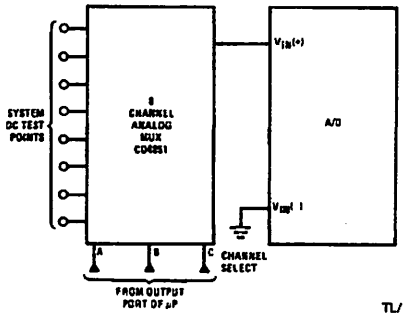
Protecting the Input



Diodes are 1N914

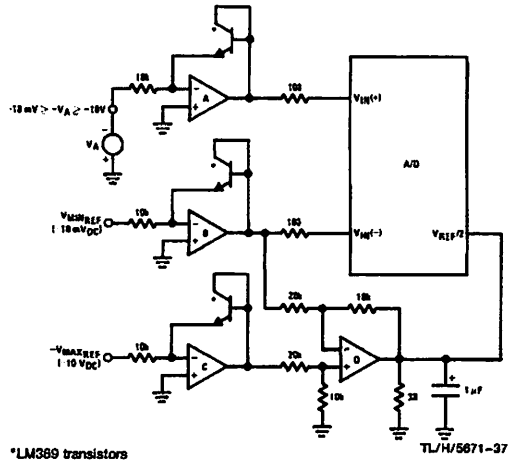
TL/H/5671–9

Analog Self-Test for a System



TL/H/5671–36

A Low-Cost, 3-Decade Logarithmic Converter



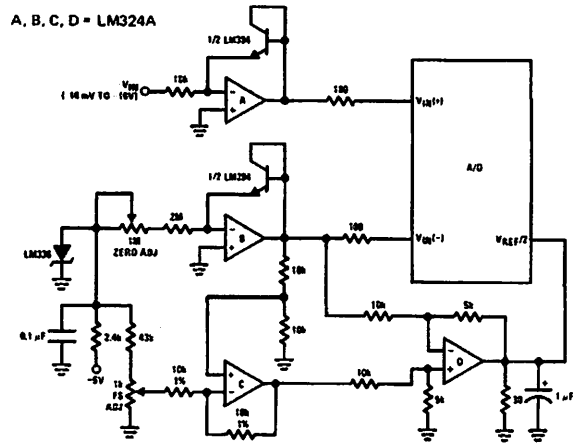
TL/H/5671–37

*LM389 transistors

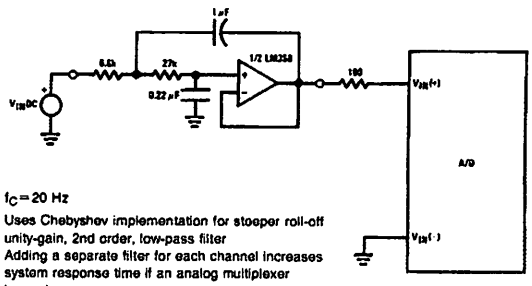
A, B, C, D = LM324A quad op amp

Typical Applications (Continued)

3-Decade Logarithmic A/D Converter



Noise Filtering the Analog Input

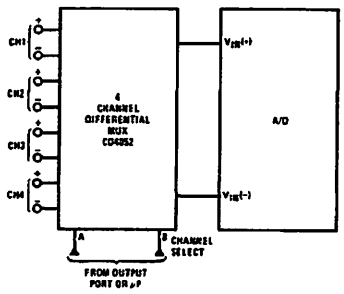


$f_C = 20$ Hz

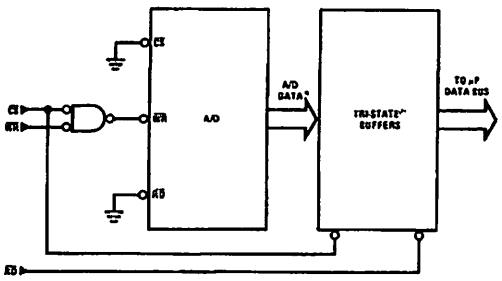
Uses Chebyshev implementation for steeper roll-off unity-gain, 2nd order, low-pass filter

Adding a separate filter for each channel increases system response time if an analog multiplexer is used

Multiplexing Differential Inputs

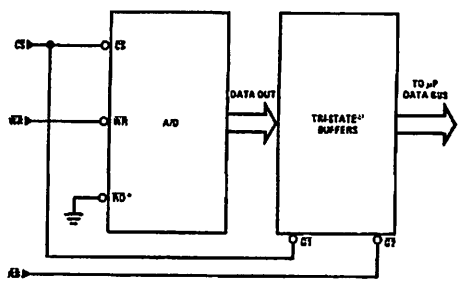


Output Buffers with A/D Data Enabled



*A/D output data is updated 1 CLK period prior to assertion of $\overline{INTR}$

Increasing Bus Drive and/or Reducing Time on Bus

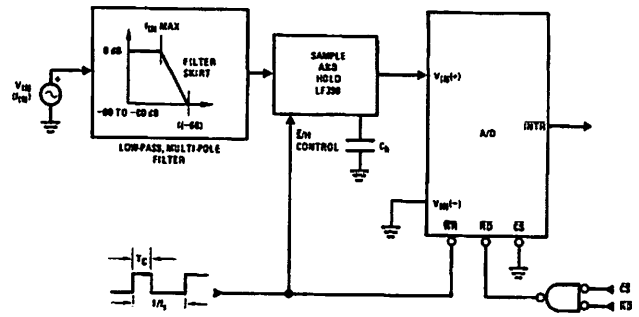


TL/H/5671–10

*Allows output data to set-up at falling edge of $\overline{CS}$

Typical Applications (Continued)

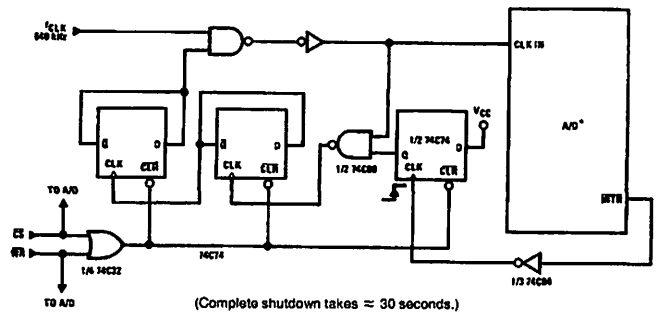
Sampling an AC Input Signal



Note 1: Oversample whenever possible (keep $f_S > 2f(-60)$) to eliminate input frequency folding (aliasing) and to allow for the skirt response of the filter.

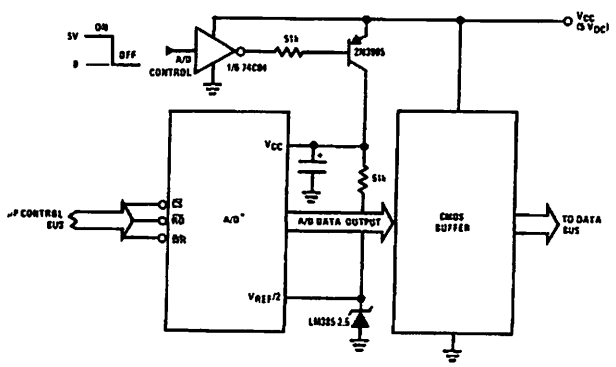
Note 2: Consider the amplitude errors which are introduced within the passband of the filter.

70% Power Savings by Clock Gating



(Complete shutdown takes ≈ 30 seconds.)

Power Savings by A/D and V_{REF} Shutdown



TL/H/5671–11

*Use ADC0801, 02, 03 or 05 for lowest power consumption.

Note: Logic inputs can be driven to V_{CC} with A/D supply at zero volts.

Buffer prevents data bus from overdriving output of A/D when in shutdown mode.

Functional Description

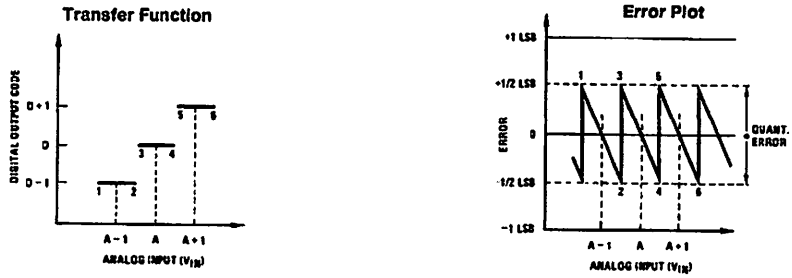
1.0 UNDERSTANDING A/D ERROR SPECS

A perfect A/D transfer characteristic (staircase waveform) is shown in *Figure 1a*. The horizontal scale is analog input voltage and the particular points labeled are in steps of 1 LSB (19.53 mV with 2.5V tied to the $V_{REF}/2$ pin). The digital output codes that correspond to these inputs are shown as D−1, D, and D+1. For the perfect A/D, not only will center-value (A−1, A, A+1, ...) analog inputs produce the correct output ditigal codes, but also each riser (the transitions between adjacent output codes) will be located ±½ LSB away from each center-value. As shown, the risers are ideal and have no width. Correct digital output codes will be provided for a range of analog input voltages that extend ±½ LSB from the ideal center-values. Each tread (the range of analog input voltage that provides the same digital output code) is therefore 1 LSB wide.

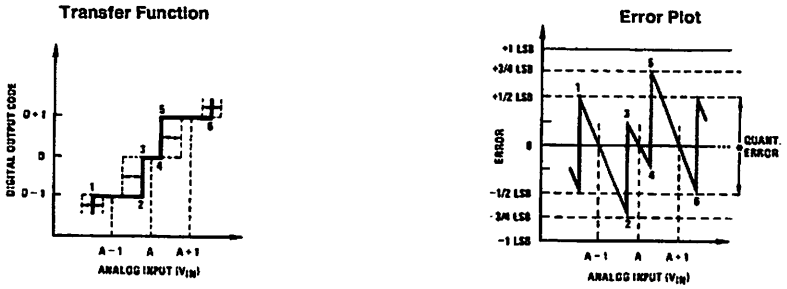
Figure 1b shows a worst case error plot for the ADC0801. All center-valued inputs are guaranteed to produce the correct output codes and the adjacent risers are guaranteed to be no closer to the center-value points than ±¼ LSB. In other words, if we apply an analog input equal to the center-value ±¼ LSB, *we guarantee* that the A/D will produce the correct digital code. The maximum range of the position of the code transition is indicated by the horizontal arrow and it is guaranteed to be no more than ½ LSB.

The error curve of *Figure 1c* shows a worst case error plot for the ADC0802. Here we guarantee that if we apply an analog input equal to the LSB analog voltage center-value the A/D will produce the correct digital code.

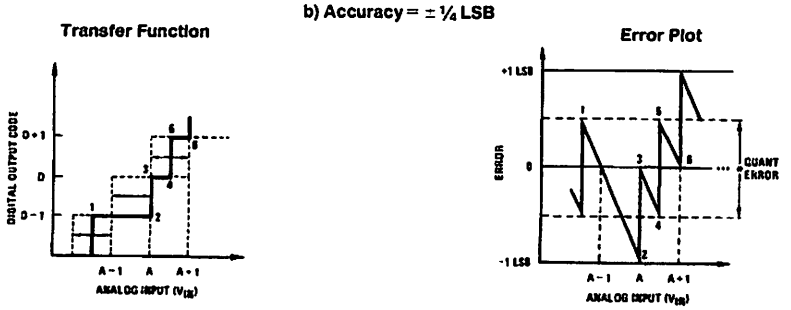
Next to each transfer function is shown the corresponding error plot. Many people may be more familiar with error plots than transfer functions. The analog input voltage to the A/D is provided by either a linear ramp or by the discrete output steps of a high resolution DAC. Notice that the error is continuously displayed and includes the quantization uncertainty of the A/D. For example the error at point 1 of *Figure 1a* is +½ LSB because the digital code appeared ½ LSB in advance of the center-value of the tread. The error plots always have a constant negative slope and the abrupt upside steps are always 1 LSB in magnitude.



a) Accuracy = ±0 LSB: A Perfect A/D



b) Accuracy = ±¼ LSB



c) Accuracy = ±½ LSB

TL/H/5671–12

FIGURE 1. Clarifying the Error Specs of an A/D Converter

Functional Description (Continued)

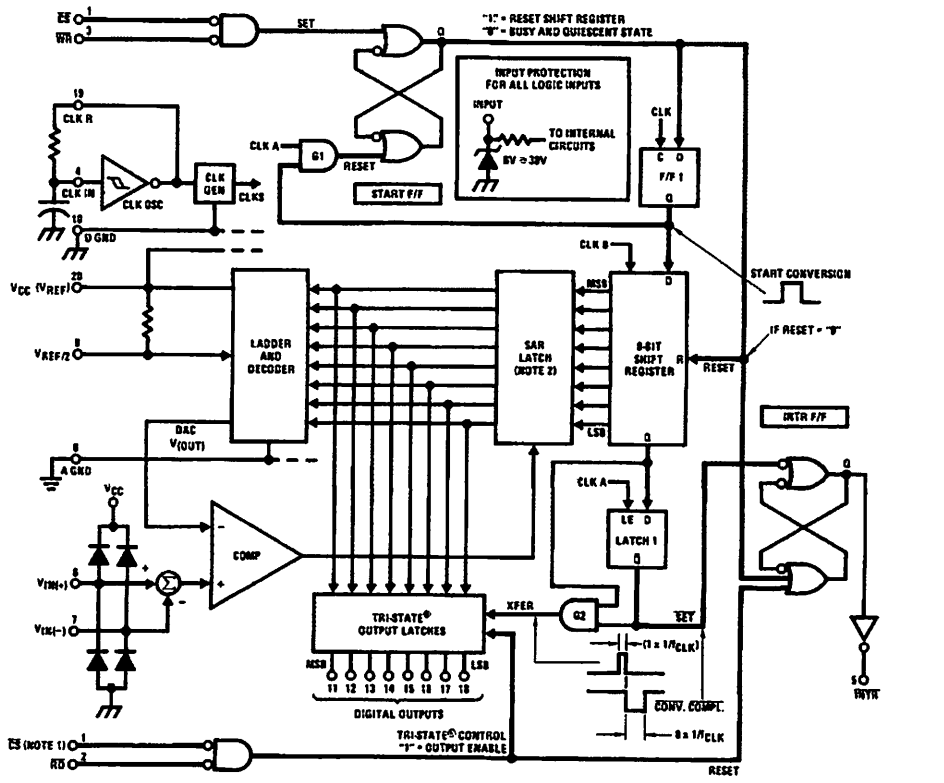
2.0 FUNCTIONAL DESCRIPTION

The ADC0801 series contains a circuit equivalent of the 256R network. Analog switches are sequenced by successive approximation logic to match the analog difference input voltage [$V_{IN}(+) - V_{IN}(-)$] to a corresponding tap on the R network. The most significant bit is tested first and after 8 comparisons (64 clock cycles) a digital 8-bit binary code (1111 1111 = full-scale) is transferred to an output latch and then an interrupt is asserted ($\overline{INTR}$ makes a high-to-low transition). A conversion in process can be interrupted by issuing a second start command. The device may be operated in the free-running mode by connecting $\overline{INTR}$ to the $\overline{WR}$ input with $\overline{CS}=0$. To ensure start-up under all possible conditions, an external $\overline{WR}$ pulse is required during the first power-up cycle.

On the high-to-low transition of the $\overline{WR}$ input the internal SAR latches and the shift register stages are reset. As long as the $\overline{CS}$ input and $\overline{WR}$ input remain low, the A/D will remain in a reset state. *Conversion will start from 1 to 8 clock periods after at least one of these inputs makes a low-to-high transition.*

A functional diagram of the A/D converter is shown in *Figure 2*. All of the package pinouts are shown and the major logic control paths are drawn in heavier weight lines.

The converter is started by having $\overline{CS}$ and $\overline{WR}$ simultaneously low. This sets the start flip-flop (F/F) and the resulting "1" level resets the 8-bit shift register, resets the Interrupt ($\overline{INTR}$) F/F and inputs a "1" to the D flop, F/F1, which is at the input end of the 8-bit shift register. Internal clock signals then transfer this "1" to the Q output of F/F1. The AND gate, G1, combines this "1" output with a clock signal to provide a reset signal to the start F/F. If the set signal is no longer present (either $\overline{WR}$ or $\overline{CS}$ is a "1") the start F/F is reset and the 8-bit shift register then can have the "1" clocked in, which starts the conversion process. If the set signal were to still be present, this reset pulse would have no effect (both outputs of the start F/F would momentarily be at a "1" level) and the 8-bit shift register would continue to be held in the reset mode. This logic therefore allows for wide $\overline{CS}$ and $\overline{WR}$ signals and the converter will start after at least one of these signals returns high and the internal clocks again provide a reset signal for the start F/F.



TL/H/5671–13

Note 1: $\overline{CS}$ shown twice for clarity.

Note 2: SAR = Successive Approximation Register.

FIGURE 2. Block Diagram

Functional Description (Continued)

After the "1" is clocked through the 8-bit shift register (which completes the SAR search) it appears as the input to the D-type latch, LATCH 1. As soon as this "1" is output from the shift register, the AND gate, G2, causes the new digital word to transfer to the TRI-STATE output latches. When LATCH 1 is subsequently enabled, the Q output makes a high-to-low transition which causes the INTR F/F to set. An inverting buffer then supplies the $\overline{INTR}$ input signal.

Note that this $\overline{SET}$ control of the INTR F/F remains low for 8 of the external clock periods (as the internal clocks run at 1/8 of the frequency of the external clock). If the data output is continuously enabled ($\overline{CS}$ and $\overline{RD}$ both held low), the $\overline{INTR}$ output will still signal the end of conversion (by a high-to-low transition), because the $\overline{SET}$ input can control the Q output of the INTR F/F even though the RESET input is constantly at a "1" level in this operating mode. This $\overline{INTR}$ output will therefore stay low for the duration of the $\overline{SET}$ signal, which is 8 periods of the external clock frequency (assuming the A/D is not started during this interval).

When operating in the free-running or continuous conversion mode ($\overline{INTR}$ pin tied to $\overline{WR}$ and $\overline{CS}$ wired low—see also section 2.6), the START F/F is SET by the high-to-low transition of the $\overline{INTR}$ signal. This resets the SHIFT REGISTER which causes the input to the D-type latch, LATCH 1, to go low. As the latch enable input is still present, the $\overline{Q}$ output will go high, which then allows the INTR F/F to be RESET. This reduces the width of the resulting $\overline{INTR}$ output pulse to only a few propagation delays (approximately 300 ns).

When data is to be read, the combination of both $\overline{CS}$ and $\overline{RD}$ being low will cause the INTR F/F to be reset and the TRI-STATE output latches will be enabled to provide the 8-bit digital outputs.

2.1 Digital Control Inputs

The digital control inputs ($\overline{CS}$, $\overline{RD}$, and $\overline{WR}$) meet standard T^2L logic voltage levels. These signals have been renamed when compared to the standard A/D Start and Output Enable labels. In addition, these inputs are active low to allow an easy interface to microprocessor control busses. For non-microprocessor based applications, the $\overline{CS}$ input (pin 1) can be grounded and the standard A/D Start function is obtained by an active low pulse applied at the $\overline{WR}$ input (pin 3) and the Output Enable function is caused by an active low pulse at the $\overline{RD}$ input (pin 2).

2.2 Analog Differential Voltage Inputs and Common-Mode Rejection

This A/D has additional applications flexibility due to the analog differential voltage input. The $V_{IN}(-)$ input (pin 7) can be used to automatically subtract a fixed voltage value from the input reading (tare correction). This is also useful in 4 mA–20 mA current loop conversion. In addition, common-mode noise can be reduced by use of the differential input.

The time interval between sampling $V_{IN}(+)$ and $V_{IN}(-)$ is 4-1/2 clock periods. The maximum error voltage due to this slight time difference between the input voltage samples is given by:

$$\Delta V_e(MAX) = (V_P)(2\pi f_{cm})\left(\frac{4.5}{f_{CLK}}\right),$$

where:

ΔV_e is the error voltage due to sampling delay

V_P is the peak value of the common-mode voltage

f_{cm} is the common-mode frequency

As an example, to keep this error to 1/4 LSB (~5 mV) when operating with a 60 Hz common-mode frequency, f_{cm}, and using a 640 kHz A/D clock, f_{CLK}, would allow a peak value of the common-mode voltage, V_P, which is given by:

$$V_P = \frac{[\Delta V_{e(MAX)}(f_{CLK})]}{(2\pi f_{cm})(4.5)}$$

or

$$V_P = \frac{(5 \times 10^{-3})(640 \times 10^3)}{(6.28)(60)(4.5)}$$

which gives

$V_P \cong 1.9V$.

The allowed range of analog input voltages usually places more severe restrictions on input common-mode noise levels.

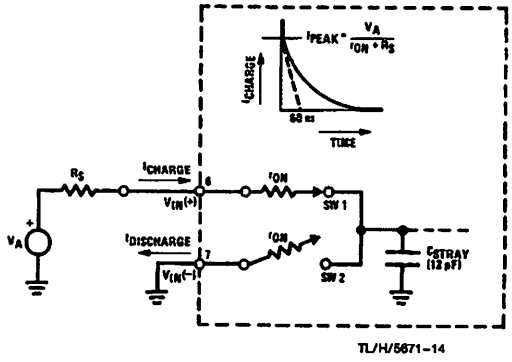
An analog input voltage with a reduced span and a relatively large zero offset can be handled easily by making use of the differential input (see section 2.4 Reference Voltage).

2.3 Analog Inputs

2.3.1 Input Current

Normal Mode

Due to the internal switching action, displacement currents will flow at the analog inputs. This is due to on-chip stray capacitance to ground as shown in *Figure 3*.



TL/H/5671–14

r_{ON} of SW 1 and SW 2 = 5 kΩ

$\tau = r_{ON} C_{STRAY} = 5\ k\Omega \times 12\ pF = 60\ ns$

FIGURE 3. Analog Input Impedance

Functional Description (Continued)

The voltage on this capacitance is switched and will result in currents entering the $V_{IN}(+)$ input pin and leaving the $V_{IN}(-)$ input which will depend on the analog differential input voltage levels. These current transients occur at the leading edge of the internal clocks. They rapidly decay and *do not cause errors* as the on-chip comparator is strobed at the end of the clock period.

Fault Mode

If the voltage source applied to the $V_{IN}(+)$ or $V_{IN}(-)$ pin exceeds the allowed operating range of $V_{CC}+50$ mV, large input currents can flow through a parasitic diode to the V_{CC} pin. If these currents can exceed the 1 mA max allowed spec, an external diode (1N914) should be added to bypass this current to the V_{CC} pin (with the current bypassed with this diode, the voltage at the $V_{IN}(+)$ pin can exceed the V_{CC} voltage by the forward voltage of this diode).

2.3.2 Input Bypass Capacitors

Bypass capacitors at the inputs will average these charges and cause a DC current to flow through the output resistances of the analog signal sources. This charge pumping action is worse for continuous conversions with the $V_{IN}(+)$ input voltage at full-scale. For continuous conversions with a 640 kHz clock frequency with the $V_{IN}(+)$ input at 5V, this DC current is at a maximum of approximately 5 μA. Therefore, *bypass capacitors should not be used at the analog inputs or the $V_{REF}/2$ pin* for high resistance sources (> 1 kΩ). If input bypass capacitors are necessary for noise filtering and high source resistance is desirable to minimize capacitor size, the detrimental effects of the voltage drop across this input resistance, which is due to the average value of the input current, can be eliminated with a full-scale adjustment while the given source resistor and input bypass capacitor are both in place. This is possible because the average value of the input current is a precise linear function of the differential input voltage.

2.3.3 Input Source Resistance

Large values of source resistance where an input bypass capacitor is not used, *will not cause errors* as the input currents settle out prior to the comparison time. If a low pass filter is required in the system, use a low valued series resistor (≤ 1 kΩ) for a passive RC section or add an op amp RC active low pass filter. For low source resistance applications, (≤ 1 kΩ), a 0.1 μF bypass capacitor at the inputs will prevent noise pickup due to series lead inductance of a long wire. A 100Ω series resistor can be used to isolate this capacitor—both the R and C are placed outside the feedback loop—from the output of an op amp, if used.

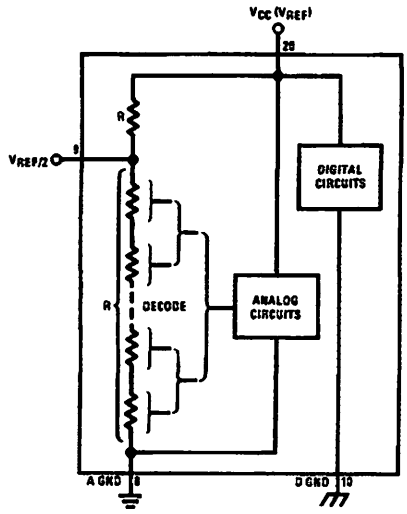
2.3.4 Noise

The leads to the analog inputs (pin 6 and 7) should be kept as short as possible to minimize input noise coupling. Both noise and undesired digital clock coupling to these inputs can cause system errors. The source resistance for these inputs should, in general, be kept below 5 kΩ. Larger values of source resistance can cause undesired system noise pickup. Input bypass capacitors, placed from the analog inputs to ground, will eliminate system noise pickup but can create analog scale errors as these capacitors will average the transient input switching currents of the A/D (see section 2.3.1.). This scale error depends on both a large source resistance and the use of an input bypass capacitor. This error can be eliminated by doing a full-scale adjustment of the A/D (adjust $V_{REF}/2$ for a proper full-scale reading—see section 2.5.2 on Full-Scale Adjustment) with the source resistance and input bypass capacitor in place.

2.4 Reference Voltage

2.4.1 Span Adjust

For maximum applications flexibility, these A/Ds have been designed to accommodate a 5 V_{DC}, 2.5 V_{DC} or an adjusted voltage reference. This has been achieved in the design of the IC as shown in *Figure 4*.



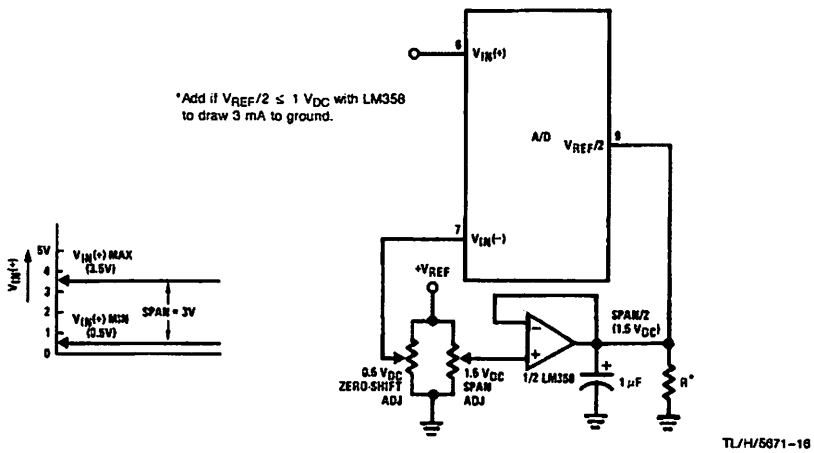
TL/H/5671–15

FIGURE 4. The $V_{REFERENCE}$ Design on the IC

Notice that the reference voltage for the IC is either ½ of the voltage applied to the V_{CC} supply pin, or is equal to the voltage that is externally forced at the $V_{REF}/2$ pin. This allows for a ratiometric voltage reference using the V_{CC} supply, a 5 V_{DC} reference voltage can be used for the V_{CC} supply or a voltage less than 2.5 V_{DC} can be applied to the $V_{REF}/2$ input for increased application flexibility. The internal gain to the $V_{REF}/2$ input is 2, making the full-scale differential input voltage twice the voltage at pin 9.

An example of the use of an adjusted reference voltage is to accommodate a reduced span—or dynamic voltage range of the analog input voltage. If the analog input voltage were to range from 0.5 V_{DC} to 3.5 V_{DC}, instead of 0V to 5 V_{DC}, the span would be 3V as shown in *Figure 5*. With 0.5 V_{DC} applied to the $V_{IN}(-)$ pin to absorb the offset, the reference voltage can be made equal to ½ of the 3V span or 1.5 V_{DC}. The A/D now will encode the $V_{IN}(+)$ signal from 0.5V to 3.5 V with the 0.5V input corresponding to zero and the 3.5 V_{DC} input corresponding to full-scale. The full 8 bits of resolution are therefore applied over this reduced analog input voltage range.

Functional Description (Continued)



TL/H/5671–16

a) Analog Input Signal Example

b) Accommodating an Analog Input from 0.5V (Digital Out = = 00_{HEX}) to 3.5V (Digital Out = FF_{HEX})

FIGURE 5. Adapting the A/D Analog Input Voltages to Match an Arbitrary Input Signal Range

2.4.2 Reference Accuracy Requirements

The converter can be operated in a ratiometric mode or an absolute mode. In ratiometric converter applications, the magnitude of the reference voltage is a factor in both the output of the source transducer and the output of the A/D converter and therefore cancels out in the final digital output code. The ADC0805 is specified particularly for use in ratiometric applications with no adjustments required. In absolute conversion applications, both the initial value and the temperature stability of the reference voltage are important factors in the accuracy of the A/D converter. For $V_{REF}/2$ voltages of 2.4 V_{DC} nominal value, initial errors of ±10 mV_{DC} will cause conversion errors of ±1 LSB due to the gain of 2 of the $V_{REF}/2$ input. In reduced span applications, the initial value and the stability of the $V_{REF}/2$ input voltage become even more important. For example, if the span is reduced to 2.5V, the analog input LSB voltage value is correspondingly reduced from 20 mV (5V span) to 10 mV and 1 LSB at the $V_{REF}/2$ input becomes 5 mV. As can be seen, this reduces the allowed initial tolerance of the reference voltage and requires correspondingly less absolute change with temperature variations. Note that spans smaller than 2.5V place even tighter requirements on the initial accuracy and stability of the reference source.

In general, the magnitude of the reference voltage will require an initial adjustment. Errors due to an improper value of reference voltage appear as full-scale errors in the A/D transfer function. IC voltage regulators may be used for references if the ambient temperature changes are not excessive. The LM336B 2.5V IC reference diode (from National Semiconductor) has a temperature stability of 1.8 mV typ (6 mV max) over $0°C \leq T_A \leq +70°C$. Other temperature range parts are also available.

2.5 Errors and Reference Voltage Adjustments

2.5.1 Zero Error

The zero of the A/D does not require adjustment. If the minimum analog input voltage value, $V_{IN(MIN)}$, is not ground, a zero offset can be done. The converter can be made to output 0000 0000 digital code for this minimum input voltage by biasing the A/D $V_{IN}(-)$ input at this $V_{IN(MIN)}$ value (see Applications section). This utilizes the differential mode operation of the A/D.

The zero error of the A/D converter relates to the location of the first riser of the transfer function and can be measured by grounding the V_{IN} (−) input and applying a small magnitude positive voltage to the V_{IN} (+) input. Zero error is the difference between the actual DC input voltage that is necessary to just cause an output digital code transition from 0000 0000 to 0000 0001 and the ideal ½ LSB value (½ LSB = 9.8 mV for $V_{REF}/2 = 2.500\ V_{DC}$).

2.5.2 Full-Scale

The full-scale adjustment can be made by applying a differential input voltage that is 1½ LSB less than the desired analog full-scale voltage range and then adjusting the magnitude of the $V_{REF}/2$ input (pin 9 or the V_{CC} supply if pin 9 is not used) for a digital output code that is just changing from 1111 1110 to 1111 1111.

Functional Description (Continued)

2.5.3 Adjusting for an Arbitrary Analog Input Voltage Range

If the analog zero voltage of the A/D is shifted away from ground (for example, to accommodate an analog input signal that does not go to ground) this new zero reference should be properly adjusted first. A $V_{IN}(+)$ voltage that equals this desired zero reference plus $\frac{1}{2}$ LSB (where the LSB is calculated for the desired analog span, 1 LSB = analog span/256) is applied to pin 6 and the zero reference voltage at pin 7 should then be adjusted to just obtain the 00_{HEX} to 01_{HEX} code transition.

The full-scale adjustment should then be made (with the proper $V_{IN}(-)$ voltage applied) by forcing a voltage to the $V_{IN}(+)$ input which is given by:

$$V_{IN}(+) \text{ fs adj} = V_{MAX} - 1.5\left[\frac{(V_{MAX} - V_{MIN})}{256}\right],$$

where:

V_{MAX} = The high end of the analog input range

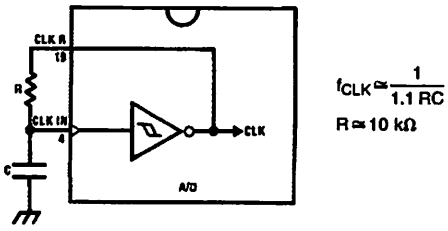
and

V_{MIN} = the low end (the offset zero) of the analog range. (Both are ground referenced.)

The $V_{REF}/2$ (or V_{CC}) voltage is then adjusted to provide a code change from FE_{HEX} to FF_{HEX}. This completes the adjustment procedure.

2.6 Clocking Option

The clock for the A/D can be derived from the CPU clock or an external RC can be added to provide self-clocking. The CLK IN (pin 4) makes use of a Schmitt trigger as shown in *Figure 6*.



$$f_{CLK} \simeq \frac{1}{1.1\,RC}$$

$$R \simeq 10\ k\Omega$$

TL/H/5671–17

FIGURE 6. Self-Clocking the A/D

Heavy capacitive or DC loading of the clock R pin should be avoided as this will disturb normal converter operation. Loads less than 50 pF, such as driving up to 7 A/D converter clock inputs from a single clock R pin of 1 converter, are allowed. For larger clock line loading, a CMOS or low power TTL buffer or PNP input logic should be used to minimize the loading on the clock R pin (do not use a standard TTL buffer).

2.7 Restart During a Conversion

If the A/D is restarted ($\overline{CS}$ and $\overline{WR}$ go low and return high) during a conversion, the converter is reset and a new conversion is started. The output data latch is not updated if the conversion in process is not allowed to be completed, therefore the data of the previous conversion remains in this latch. The $\overline{INTR}$ output simply remains at the "1" level.

2.8 Continuous Conversions

For operation in the free-running mode an initializing pulse should be used, following power-up, to ensure circuit operation. In this application, the $\overline{CS}$ input is grounded and the $\overline{WR}$ input is tied to the $\overline{INTR}$ output. This $\overline{WR}$ and $\overline{INTR}$ node should be momentarily forced to logic low following a power-up cycle to guarantee operation.

2.9 Driving the Data Bus

This MOS A/D, like MOS microprocessors and memories, will require a bus driver when the total capacitance of the data bus gets large. Other circuitry, which is tied to the data bus, will add to the total capacitive loading, even in TRI-STATE (high impedance mode). Backplane bussing also greatly adds to the stray capacitance of the data bus.

There are some alternatives available to the designer to handle this problem. Basically, the capacitive loading of the data bus slows down the response time, even though DC specifications are still met. For systems operating with a relatively slow CPU clock frequency, more time is available in which to establish proper logic levels on the bus and therefore higher capacitive loads can be driven (see typical characteristics curves).

At higher CPU clock frequencies time can be extended for I/O reads (and/or writes) by inserting wait states (8080) or using clock extending circuits (6800).

Finally, if time is short and capacitive loading is high, external bus drivers must be used. These can be TRI-STATE buffers (low power Schottky such as the DM74LS240 series is recommended) or special higher drive current products which are designed as bus drivers. High current bipolar bus drivers with PNP inputs are recommended.

2.10 Power Supplies

Noise spikes on the V_{CC} supply line can cause conversion errors as the comparator will respond to this noise. A low inductance tantalum filter capacitor should be used close to the converter V_{CC} pin and values of 1 μF or greater are recommended. If an unregulated voltage is available in the system, a separate LM340LAZ-5.0, TO-92, 5V voltage regulator for the converter (and other analog circuitry) will greatly reduce digital noise on the V_{CC} supply.

2.11 Wiring and Hook-Up Precautions

Standard digital wire wrap sockets are not satisfactory for breadboarding this A/D converter. Sockets on PC boards can be used and all logic signal wires and leads should be grouped and kept as far away as possible from the analog signal leads. Exposed leads to the analog inputs can cause undesired digital noise and hum pickup, therefore shielded leads may be necessary in many applications.

Functional Description (Continued)

A single point analog ground that is separate from the logic ground points should be used. The power supply bypass capacitor and the self-clocking capacitor (if used) should both be returned to digital ground. Any $V_{REF}/2$ bypass capacitors, analog input filter capacitors, or input signal shielding should be returned to the analog ground point. A test for proper grounding is to measure the zero error of the A/D converter. Zero errors in excess of ¼ LSB can usually be traced to improper board layout and wiring (see section 2.5.1 for measuring the zero error).

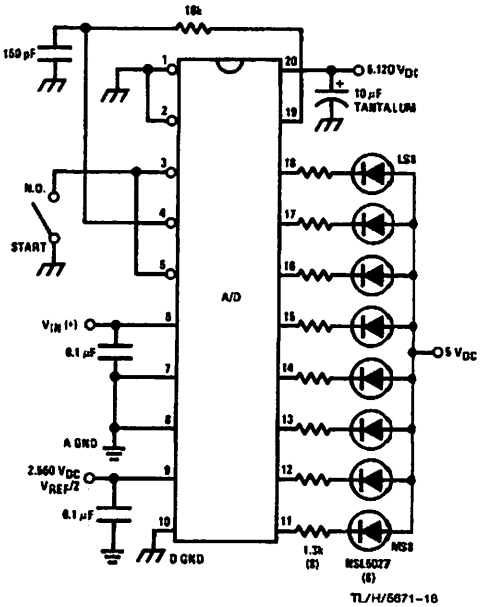
3.0 TESTING THE A/D CONVERTER

There are many degrees of complexity associated with testing an A/D converter. One of the simplest tests is to apply a known analog input voltage to the converter and use LEDs to display the resulting digital output code as shown in *Figure 7*.

For ease of testing, the $V_{REF}/2$ (pin 9) should be supplied with 2.560 V_{DC} and a V_{CC} supply voltage of 5.12 V_{DC} should be used. This provides an LSB value of 20 mV.

If a full-scale adjustment is to be made, an analog input voltage of 5.090 V_{DC} (5.120−1½ LSB) should be applied to the $V_{IN}(+)$ pin with the $V_{IN}(-)$ pin grounded. The value of the $V_{REF}/2$ input voltage should then be adjusted until the digital output code is just changing from 1111 1110 to 1111 1111. This value of $V_{REF}/2$ should then be used for all the tests.

The digital output LED display can be decoded by dividing the 8 bits into 2 hex characters, the 4 most significant (MS) and the 4 least significant (LS). Table I shows the fractional binary equivalent of these two 4-bit groups. By adding the voltages obtained from the "VMS" and "VLS" columns in Table I, the nominal value of the digital display (when $V_{REF}/2$ = 2.560V) can be determined. For example, for an output LED display of 1011 0110 or B6 (in hex), the voltage values from the table are 3.520 + 0.120 or 3.640 V_{DC}. These voltage values represent the center-values of a perfect A/D converter. The effects of quantization error have to be accounted for in the interpretation of the test results.



TL/H/5671–16

FIGURE 7. Basic A/D Tester

For a higher speed test system, or to obtain plotted data, a digital-to-analog converter is needed for the test set-up. An accurate 10-bit DAC can serve as the precision voltage source for the A/D. Errors of the A/D under test can be expressed as either analog voltages or differences in 2 digital words.

A basic A/D tester that uses a DAC and provides the error as an analog output voltage is shown in *Figure 8*. The 2 op amps can be eliminated if a lab DVM with a numerical subtraction feature is available to read the difference voltage, "A–C", directly. The analog input voltage can be supplied by a low frequency ramp generator and an X-Y plotter can be used to provide analog error (Y axis) versus analog input (X axis).

For operation with a microprocessor or a computer-based test system, it is more convenient to present the errors digitally. This can be done with the circuit of *Figure 9*, where the output code transitions can be detected as the 10-bit DAC is incremented. This provides ¼ LSB steps for the 8-bit A/D under test. If the results of this test are automatically plotted with the analog input on the X axis and the error (in LSB's) as the Y axis, a useful transfer function of the A/D under test results. For acceptance testing, the plot is not necessary and the testing speed can be increased by establishing internal limits on the allowed error for each code.

4.0 MICROPROCESSOR INTERFACING

To dicuss the interface with 6080A and 6800 microprocessors, a common sample subroutine structure is used. The microprocessor starts the A/D, reads and stores the results of 16 successive conversions, then returns to the user's program. The 16 data bytes are stored in 16 successive memory locations. All Data and Addresses will be given in hexadecimal form. Software and hardware details are provided separately for each type of microprocessor.

4.1 Interfacing 8080 Microprocessor Derivatives (8048, 8085)

This converter has been designed to directly interface with derivatives of the 8080 microprocessor. The A/D can be mapped into memory space (using standard memory address decoding for $\overline{CS}$ and the $\overline{MEMR}$ and $\overline{MEMW}$ strobes) or it can be controlled as an I/O device by using the $\overline{I/O\ R}$ and $\overline{I/O\ W}$ strobes and decoding the address bits A0 → A7 (or address bits A8 → A15 as they will contain the same 8-bit address information) to obtain the $\overline{CS}$ input. Using the I/O space provides 256 additional addresses and may allow a simpler 8-bit address decoder but the data can only be input to the accumulator. To make use of the additional memory reference instructions, the A/D should be mapped into memory space. An example of an A/D in I/O space is shown in *Figure 10*.

Functional Description (Continued)

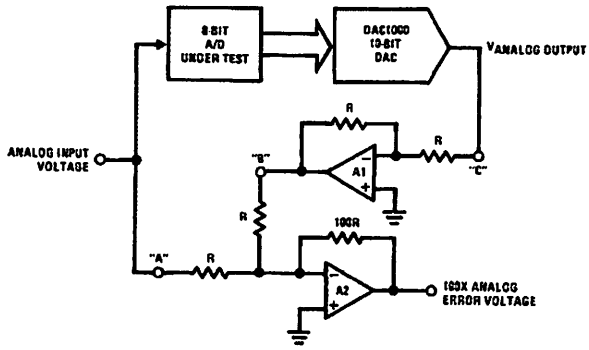
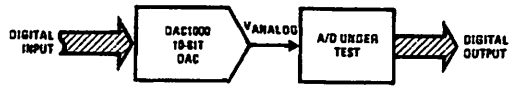


FIGURE 8. A/D Tester with Analog Error Output



TL/H/5671–19

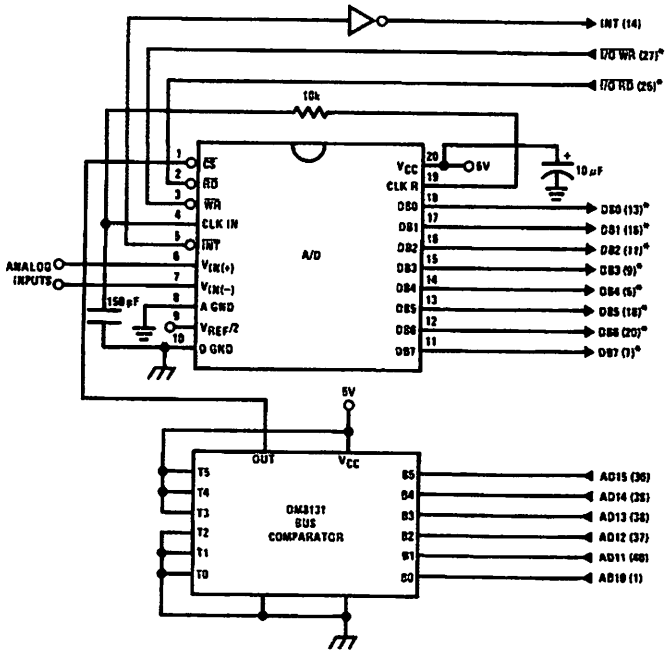
FIGURE 9. Basic "Digital" A/D Tester

TABLE I. DECODING THE DIGITAL OUTPUT LEDs

HEX	BINARY				FRACTIONAL BINARY VALUE FOR		OUTPUT VOLTAGE CENTER VALUES WITH $V_{REF}/2 = 2.560\ V_{DC}$	
					MS GROUP	LS GROUP	VMS GROUP*	VLS GROUP*
F	1	1	1	1	15/16	15/256	4.800	0.300
E	1	1	1	0	7/8	7/128	4.480	0.280
D	1	1	0	1	13/16	13/256	4.160	0.260
C	1	1	0	0	3/4	3/64	3.840	0.240
B	1	0	1	1	11/16	11/256	3.520	0.220
A	1	0	1	0	5/8	5/128	3.200	0.200
9	1	0	0	1	9/16	9/256	2/880	0.180
8	1	0	0	0	1/2	1/32	2/560	0.160
7	0	1	1	1	7/16	7/256	2.240	0.140
6	0	1	1	0	3/8	3/128	1.920	0.120
5	0	1	0	1	5/16	2/256	1.600	0.100
4	0	1	0	0	1/4	1/64	1/280	0.080
3	0	0	1	1	3/16	3/256	0.960	0.060
2	0	0	1	0	1/8	1/128	0.640	0.040
1	0	0	0	1	1/16	1/256	0.320	0.020
0	0	0	0	0			0	0

*Display Output = VMS Group + VLS Group

Functional Description (Continued)



TL/H/5871–20

Note 1: *Pin numbers for the DP8228 system controller, others are INS8080A.

Note 2: Pin 23 of the INS8228 must be tied to +12V through a 1 kΩ resistor to generate the RST 7 instruction when an interrupt is acknowledged as required by the accompanying sample program.

FIGURE 10. ADC0801–INS8080A CPU Interface

SAMPLE PROGRAM FOR *FIGURE 10* ADC0801–INS8080A CPU INTERFACE

```
0038    C3 00 03    RST 7:      JMP     LD DATA
  •         •           •
  •         •           •
0100    21 00 02    START:      LXI H 0200H     ; HL pair will point to
                                                ; data storage locations
0103    31 00 04    RETURN:     LXI SP 0400H    ; Initialize stack pointer (Note 1)
0106    7D                      MOV A, L        ; Test # of bytes entered
0107    FE 0F                   CPI 0F H        ; If # = 16. JMP to
0109    CA 13 01                JZ CONT         ; user program
010C    D3 E0                   OUT E0 H        ; Start A/D
010E    FB                      EI              ; Enable interrupt
010F    00          LOOP:       NOP             ; Loop until end of
0110    C3 0F 01                JMP LOOP        ; conversion
0113      •         CONT:         •
  •       •           •           •
  •       •         (User program to  •
  •       •         process data)     •
  •       •           •               •
  •       •           •               •
0300    DB E0       LD DATA:    IN E0 H         ; Load data into accumulator
0302    77                      MOV M, A        ; Store data
0303    23                      INX H           ; Increment storage pointer
0304    C3 03 01                JMP RETURN
```

Note 1: The stack pointer must be dimensioned because a RST 7 instruction pushes the PC onto the stack.

Note 2: All address used were arbitrarily chosen.

Functional Description (Continued)

The standard control bus signals of the 8080 $\overline{CS}$, $\overline{RD}$ and $\overline{WR}$) can be directly wired to the digital control inputs of the A/D and the bus timing requirements are met to allow both starting the converter and outputting the data onto the data bus. A bus driver should be used for larger microprocessor systems where the data bus leaves the PC board and/or must drive capacitive loads larger than 100 pF.

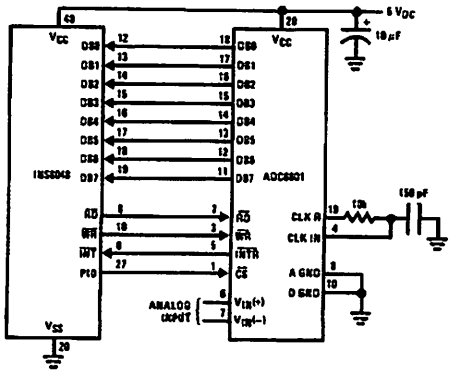
4.1.1 Sample 8080A CPU Interfacing Circuitry and Program

The following sample program and associated hardware shown in *Figure 10* may be used to input data from the converter to the INS8080A CPU chip set (comprised of the INS8080A microprocessor, the INS8228 system controller and the INS8224 clock generator). For simplicity, the A/D is controlled as an I/O device, specifically an 8-bit bi-directional port located at an arbitrarily chosen port address, E0. The TRI-STATE output capability of the A/D eliminates the need for a peripheral interface device, however address decoding is still required to generate the appropriate $\overline{CS}$ for the converter.

It is important to note that in systems where the A/D converter is 1-of-8 or less I/O mapped devices, no address decoding circuitry is necessary. Each of the 8 address bits (A0 to A7) can be directly used as $\overline{CS}$ inputs—one for each I/O device.

4.1.2 INS8048 Interface

The INS8048 interface technique with the ADC0801 series (see *Figure 11*) is simpler than the 8080A CPU interface. There are 24 I/O lines and three test input lines in the 8048. With these extra I/O lines available, one of the I/O lines (bit 0 of port 1) is used as the chip select signal to the A/D, thus eliminating the use of an external address decoder. Bus control signals $\overline{RD}$, $\overline{WR}$ and $\overline{INT}$ of the 8048 are tied directly to the A/D. The 16 converted data words are stored at on-chip RAM locations from 20 to 2F (Hex). The $\overline{RD}$ and $\overline{WR}$ signals are generated by reading from and writing into a dummy address, respectively. A sample interface program is shown below.



TL/H/5671–21

FIGURE 11. INS8048 Interface

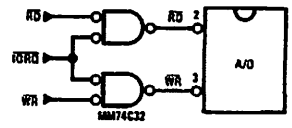
SAMPLE PROGRAM FOR *FIGURE 11* INS8048 INTERFACE

```
04 10                   JMP     10H             ;Program starts at addr 10
                        ORG     3H
04 50                   JMP     50H             ;Interrupt jump vector
                        ORG     10H             ;Main program
99 FE                   ANL     P1, #0FEH       ;Chip select
81                      MOVX    A, @R1          ;Read in the 1st data
                                                ;to reset the intr
89 01           START:  ORL     P1, #1          ;Set port pin high
B8 20                   MOV     R0, #20H        ;Data address
B9 FF                   MOV     R1, #0FFH       ;Dummy address
BA 10                   MOV     R2, #10H        ;Counter for 16 bytes
23 FF           AGAIN:  MOV     A, #0FFH        ;Set ACC for intr loop
99 FE                   ANL     P1, #0FEH       ;Send CS (bit 0 of P1)
91                      MOVX    @R1, A          ;Send WR out
05                      EN      I               ;Enable interrupt
96 21           LOOP:   JNZ     LOOP            ;Wait for interrupt
EA 1B                   DJNZ    R2, AGAIN       ;If 16 bytes are read
00                      NOP                     ;go to user's program
00                      NOP
                        ORG     50H
81              INDATA: MOVX    A, @R1          ;Input data, CS still low
A0                      MOV     @R0, A          ;Store in memory
18                      INC     R0              ;Increment storage counter
89 01                   ORL     P1, #1          ;Reset CS signal
27                      CLR     A               ;Clear ACC to get out of
93                      RETR                    ;the interrupt loop
```

Functional Description (Continued)

4.2 Interfacing the Z-80

The Z-80 control bus is slightly different from that of the 8080. General $\overline{RD}$ and $\overline{WR}$ strobes are provided and separate memory request, $\overline{MREQ}$, and I/O request, $\overline{IORQ}$, signals are used which have to be combined with the generalized strobes to provide the equivalent 8080 signals. An advantage of operating the A/D in I/O space with the Z-80 is that the CPU will automatically insert one wait state (the $\overline{RD}$ and $\overline{WR}$ strobes are extended one clock period) to allow more time for the I/O devices to respond. Logic to map the A/D in I/O space is shown in *Figure 13*.



TL/H/5671–23

FIGURE 13. Mapping the A/D as an I/O Device for Use with the Z-80 CPU

Additional I/O advantages exist as software DMA routines are available and use can be made of the output data transfer which exists on the upper 8 address lines (A8 to A15) during I/O input instructions. For example, MUX channel selection for the A/D can be accomplished with this operating mode.

4.3 Interfacing 6800 Microprocessor Derivatives (6502, etc.)

The control bus for the 6800 microprocessor derivatives does not use the $\overline{RD}$ and $\overline{WR}$ strobe signals. Instead it employs a single R/$\overline{W}$ line and additional timing, if needed, can be derived fom the $\phi 2$ clock. All I/O devices are memory mapped in the 6800 system, and a special signal, VMA, indicates that the current address is valid. *Figure 14* shows an interface schematic where the A/D is memory mapped in the 6800 system. For simplicity, the $\overline{CS}$ decoding is shown using ½ DM8092. Note that in many 6800 systems, an already decoded $\overline{475}$ line is brought out to the common bus at pin 21. This can be tied directly to the $\overline{CS}$ pin of the A/D, provided that no other devices are addressed at HX ADDR: 4XXX or 5XXX.

The following subroutine performs essentially the same function as in the case of the 8080A interface and it can be called from anywhere in the user's program.

In *Figure 15* the ADC0801 series is interfaced to the M6800 microprocessor through (the arbitrarily chosen) Port B of the MC6820 or MC6821 Peripheral Interface Adapter, (PIA). Here the $\overline{CS}$ pin of the A/D is grounded since the PIA is already memory mapped in the M6800 system and no $\overline{CS}$ decoding is necessary. Also notice that the A/D output data lines are connected to the microprocessor bus under program control through the PIA and therefore the A/D $\overline{RD}$ pin can be grounded.

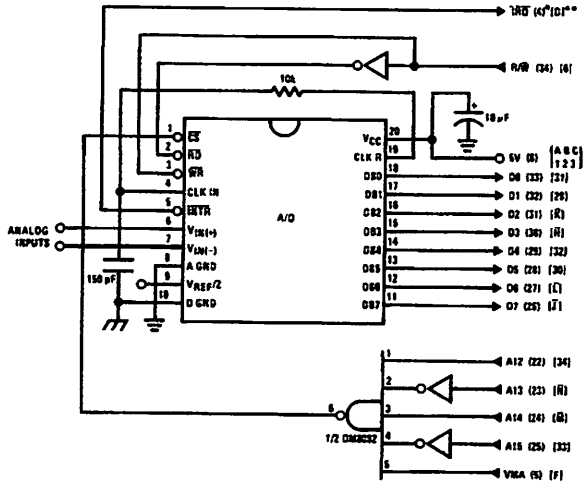
A sample interface program equivalent to the previous one is shown below *Figure 15*. The PIA Data and Control Registers of Port B are located at HEX addresses 8006 and 8007, respectively.

5.0 GENERAL APPLICATIONS

The following applications show some interesting uses for the A/D. The fact that one particular microprocessor is used is not meant to be restrictive. Each of these application circuits would have its counterpart using any microprocessor that is desired.

5.1 Multiple ADC0801 Series to MC6800 CPU Interface

To transfer analog data from several channels to a single microprocessor system, a multiple converter scheme presents several advantages over the conventional multiplexer single-converter approach. With the ADC0801 series, the differential inputs allow individual span adjustment for each channel. Furthermore, all analog input channels are sensed simultaneously, which essentially divides the microprocessor's total system servicing time by the number of channels, since all conversions occur simultaneously. This scheme is shown in *Figure 16*.



Note 1: Numbers in paronthoses refer to MC6800 CPU pin out.
Note 2: Number or letters in brackets refer to standard M6800 system common bus code.

TL/H/5671–24

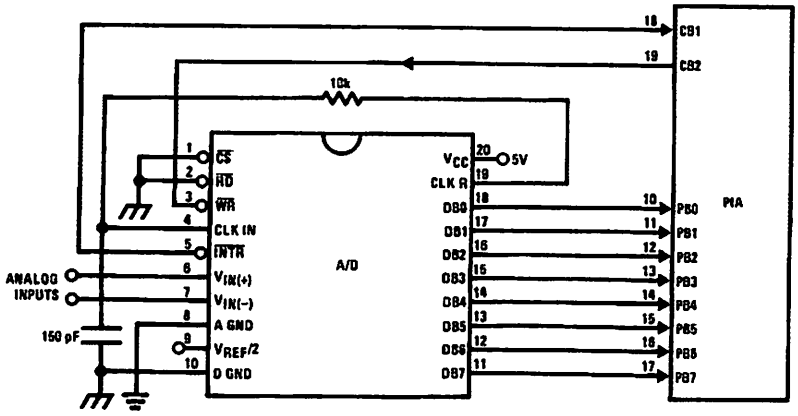
FIGURE 14. ADC0801-MC6800 CPU Interface

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 14* ADC0801-MC6800 CPU INTERFACE

```
0010  DF 36     DATAIN   STX    TEMP2    ; Save contents of X
0012  CE 00 2C           LDX    #$002C   ; Upon IRQ low CPU
0015  FF FF F8           STX    $FFF8    ; jumps to 002C
0018  B7 50 00           STAA   $5000    ; Start ADC0801
001B  0E                 CLI
001C  3E       CONVRT    WAI             ; Wait for interrupt
001D  DE 34              LDX    TEMP1
001F  8C 02 0F           CPX    #$020F   ; Is final data stored?
0022  27 14              BEQ    ENDP
0024  B7 50 00           STAA   $5000    ; Restarts ADC0801
0027  08                 INX
0028  DF 34              STX    TEMP1
002A  20 F0              BRA    CONVRT
002C  DE 34    INTRPT    LDX    TEMP1
002E  B6 50 00           LDAA   $5000    ; Read data
0031  A7 00              STAA   X        ; Store it at X
0033  3B                 RTI
0034  02 00    TEMP1     FDB    $0200    ; Starting address for
                                         ; data storage
0036  00 00    TEMP2     FDB    $0000
0038  CE 02 00 ENDP      LDX    #$0200   ; Reinitialize TEMP1
003B  DF 34              STX    TEMP1
003D  DE 36              LDX    TEMP2
003F  39                 RTS             ; Return from subroutine
                                         ; To user's program
```

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.



TL/H/5671–25

FIGURE 15. ADC0801–MC6820 PIA Interface

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 15* ADC0801–MC6820 PIA INTERFACE

```
0010  CE 00 38   DATAIN   LDX   #$0038    ; Upon IRQ low CPU
0013  FF FF F8           STX   $FFF8      ; jumps to 0038
0016  B6 80 06           LDAA  PIAORB     ; Clear possible IRQ flags
0019  4F                 CLRA
001A  B7 80 07           STAA  PIACRB
001D  B7 80 06           STAA  PIAORB     ; Set Port B as input
0020  0E                 CLI
0021  C6 34              LDAB  #$34
0023  86 3D              LDAA  #$3D
0025  F7 80 07   CONVRT  STAB  PIACRB     ; Starts ADC0801
0028  B7 80 07           STAA  PIACRB
002B  3E                 WAI               ; Wait for interrupt
002C  DE 40              LDX   TEMP1
002E  8C 02 0F           CPX   #$020F      ; Is final data stored?
0031  27 0F              BEQ   ENDP
0033  08                 INX
0034  DF 40              STX   TEMP1
0036  20 ED              BRA   CONVRT
0038  DE 40      INTRPT  LDX   TEMP1
003A  B6 80 06           LDAA  PIAORB     ; Read data in
003D  A7 00              STAA  X           ; Store it at X
003F  3B                 RTI
0040  02 00      TEMP1   FDB   $0200      ; Starting address for
                                          ; data storage
0042  CE 02 00   ENDP    LDX   #$0200     ; Reinitialize TEMP1
0045  DF 40              STX   TEMP1
0047  39                 RTS               ; Return from subroutine
              PIAORB  EQU   $8006      ; To user's program
              PIACRB  EQU   $8007
```

The following schematic and sample subroutine (DATA IN) may be used to interface (up to) 8 ADC0801's directly to the MC6800 CPU. This scheme can easily be extended to allow the interface of more converters. In this configuration the converters are (arbitrarily) located at HEX address 5000 in the MC6800 memory space. To save components, the clock signal is derived from just one RC pair on the first converter. This output drives the other A/Ds.

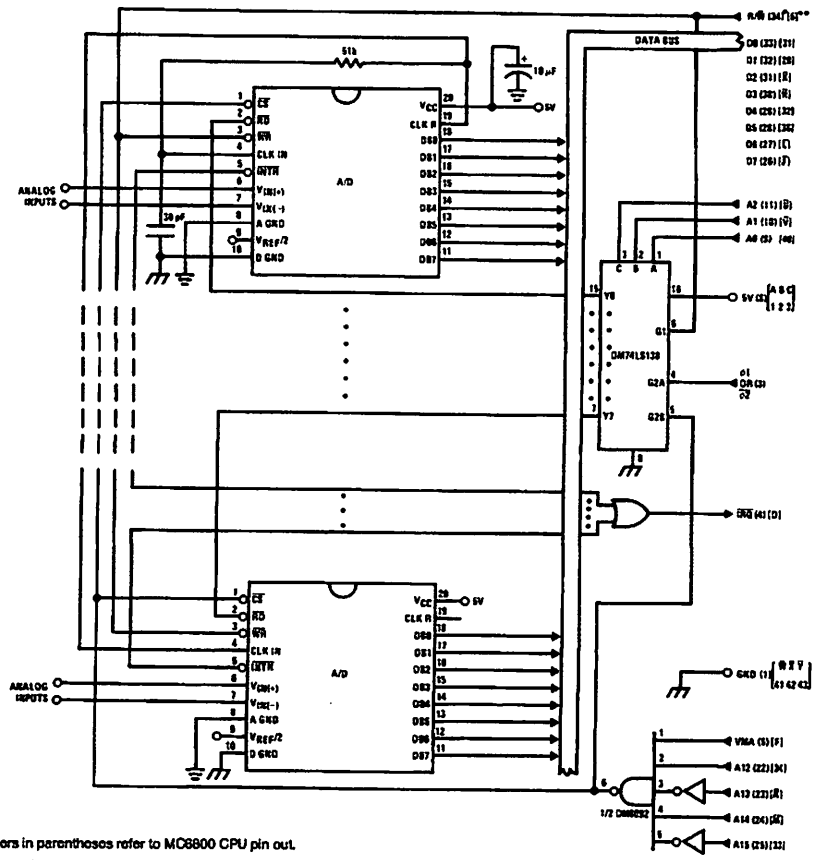
All the converters are started simultaneously with a STORE instruction at HEX address 5000. Note that any other HEX address of the form 5XXX will be decoded by the circuit, pulling all the $\overline{CS}$ inputs low. This can easily be avoided by using a more definitive address decoding scheme. All the interrupts are ORed together to insure that all A/Ds have completed their conversion before the microprocessor is interrupted.

The subroutine, DATA IN, may be called from anywhere in the user's program. Once called, this routine initializes the CPU, starts all the converters simultaneously and waits for the interrupt signal. Upon receiving the interrupt, it reads the converters (from HEX addresses 5000 through 5007) and stores the data successively at (arbitrarily chosen) HEX addresses 0200 to 0207, before returning to the user's program. All CPU registers then recover the original data they had before servicing DATA IN.

5.2 Auto-Zeroed Differential Transducer Amplifier and A/D Converter

The differential inputs of the ADC0801 series eliminate the need to perform a differential to single ended conversion for a differential transducer. Thus, one op amp can be eliminated since the differential to single ended conversion is provided by the differential input of the ADC0801 series. In general, a transducer preamp is required to take advantage of the full A/D converter input dynamic range.

Functional Description (Continued)



Note 1: Numbers in parentheses refer to MC6800 CPU pin out.

Note 2: Numbers of letters in brackets refer to standard M6800 system common bus code.

TL/H/5671–26

FIGURE 16. Interfacing Multiple A/Ds in an MC6800 System

SAMPLE PROGRAM FOR *FIGURE 16* INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE	MNEMONICS			COMMENTS
0010	DF 44	DATAIN	STX	TEMP	; Save Contents of X
0012	CE 00 2A		LDX	#$002A	; Upon $\overline{IRQ}$ LOW CPU
0015	FF FF F8		STX	$FFF8	; Jumps to 002A
0018	B7 50 00		STAA	$5000	; Starts all A/D's
001B	0E		CLI		
001C	3E		WAI		; Wait for interrupt
001D	CE 50 00		LDX	#$5000	
0020	DF 40		STX	INDEX1	; Reset both INDEX
0022	CE 02 00		LDX	#$0200	; 1 and 2 to starting
0025	DF 42		STX	INDEX2	; addresses
0027	DE 44		LDX	TEMP	
0029	39		RTS		; Return from subroutine
002A	DE 40	INTRPT	LDX	INDEX1	; INDEX1 → X
002C	A6 00		LDAA	X	; Read data in from A/D at X
002E	08		INX		; Increment X by one
002F	DF 40		STX	INDEX1	; X → INDEX1
0031	DE 42		LDX	INDEX2	; INDEX2 → X

Functional Description (Continued)

SAMPLE PROGRAM FOR *FIGURE 16* INTERFACING MULTIPLE A/Ds IN AN MC6800 SYSTEM

ADDRESS	HEX CODE		MNEMONICS		COMMENTS
0033	A7 00		STAA	X	; Store data at X
0035	8C 02 07		CPX	#$0207	; Have all A/D's been read?
0038	27 05		BEQ	RETURN	; Yes: branch to RETURN
003A	08		INX		; No: increment X by one
003B	DF 42		STX	INDEX2	; X → INDEX2
003D	20 EB		BRA	INTRPT	; Branch to 002A
003F	3B	RETURN	RTI		
0040	50 00	INDEX1	FDB	$5000	; Starting address for A/D
0042	02 00	INDEX2	FDB	$0200	; Starting address for data storage
0044	00 00	TEMP	FDB	$0000	

Note 1: In order for the microprocessor to service subroutines and interrupts, the stack pointer must be dimensioned in the user's program.

For amplification of DC input signals, a major system error is the input offset voltage of the amplifiers used for the preamp. *Figure 17* is a gain of 100 differential preamp whose offset voltage errors will be cancelled by a zeroing subroutine which is performed by the INS8080A microprocessor system. The total allowable input offset voltage error for this preamp is only 50 μV for ¼ LSB error. This would obviously require very precise amplifiers. The expression for the differential output voltage of the preamp is:

$$V_O = \underbrace{[V_{IN}(+) - V_{IN}(-)]}_{\text{SIGNAL}} \underbrace{\left[1 + \frac{2R2}{R1}\right]}_{\text{GAIN}} +$$

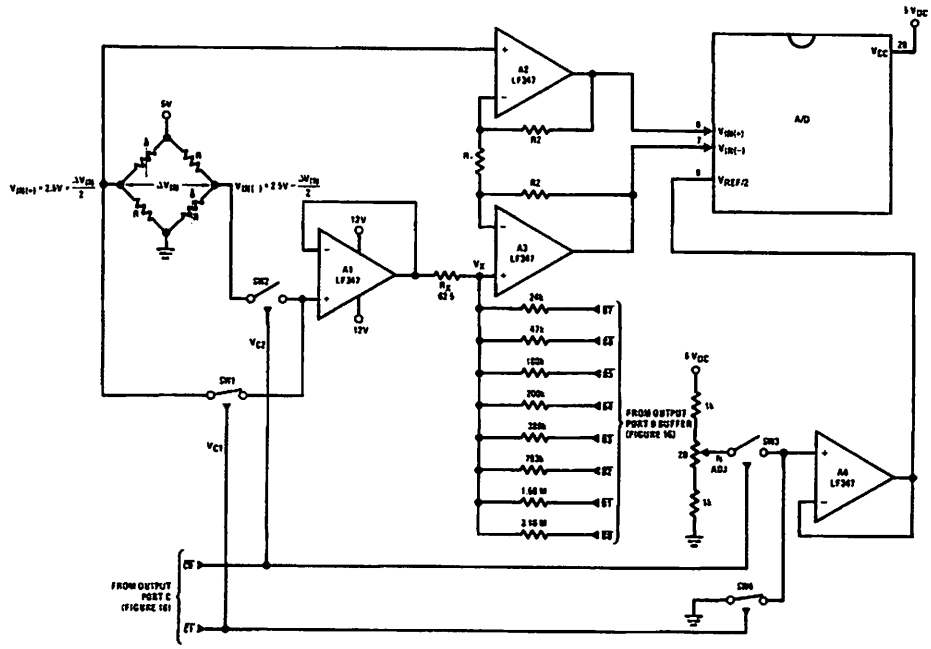
$$\underbrace{(V_{OS_2} - V_{OS_1} - V_{OS_3} \pm I_X R_X)}_{\text{DC ERROR TERM}} \underbrace{\left(1 + \frac{2R2}{R1}\right)}_{\text{GAIN}}$$

where I_X is the current through resistor R_X. All of the offset error terms can be cancelled by making $\pm I_X R_X = V_{OS1} + V_{OS3} - V_{OS2}$. This is the principle of this auto-zeroing scheme.

The INS8080A uses the 3 I/O ports of an INS8255 Programable Peripheral Interface (PPI) to control the auto zeroing and input data from the ADC0801 as shown in *Figure 18*. The PPI is programmed for basic I/O operation (mode 0) with Port A being an input port and Ports B and C being output ports. Two bits of Port C are used to alternately open or close the 2 switches at the input of the preamp. Switch SW1 is closed to force the preamp's differential input to be zero during the zeroing subroutine and then opened and SW2 is then closed for conversion of the actual differential input signal. Using 2 switches in this manner eliminates concern for the ON resistance of the switches as they must conduct only the input bias current of the input amplifiers.

Output Port B is used as a successive approximation register by the 8080 and the binary scaled resistors in series with each output bit create a D/A converter. During the zeroing subroutine, the voltage at V_x increases or decreases as required to make the differential output voltage equal to zero. This is accomplished by ensuring that the voltage at the output of A1 is approximately 2.5V so that a logic "1" (5V) on any output of Port B will source current into node V_X thus raising the voltage at V_X and making the output differential more negative. Conversely, a logic "0" (0V) will pull current out of node V_X and decrease the voltage, causing the differential output to become more positive. For the resistor values shown, V_X can move ±12 mV with a resolution of 50 μV, which will null the offset error term to ¼ LSB of full-scale for the ADC0801. It is important that the voltage levels that drive the auto-zero resistors be constant. Also, for symmetry, a logic swing of 0V to 5V is convenient. To achieve this, a CMOS buffer is used for the logic output signals of Port B and this CMOS package is powered with a stable 5V source. Buffer amplifier A1 is necessary so that it can source or sink the D/A output current.

Functional Description (Continued)

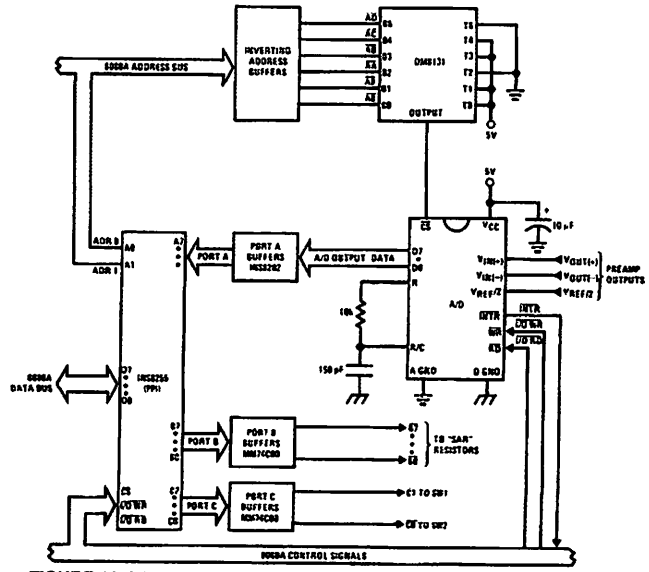


Note 1: R2 = 49.5 R1

Note 2: Switches are LMC13334 CMOS analog switches.

Note 3: The 9 resistors used in the auto-zero section can be ± 5% tolerance.

FIGURE 17. Gain of 100 Differential Transducer Preamp



TL/H/5671–27

FIGURE 18. Microprocessor Interface Circuitry for Differential Preamp

A flow chart for the zeroing subroutine is shown in *Figure 19*. It must be noted that the ADC0801 series will output an all zero code when it converts a negative input [$V_{IN}(-) \geq V_{IN}(+)$]. Also, a logic inversion exists as all of the I/O ports are buffered with inverting gates.

Basically, if the data read is zero, the differential output voltage is negative, so a bit in Port B is cleared to pull V_X more negative which will make the output more positive for the next conversion. If the data read is not zero, the output voltage is positive so a bit in Port B is set to make V_X more positive and the output more negative. This continues for 8 approximations and the differential output eventually converges to within 5 mV of zero.

The actual program is given in *Figure 20*. All addresses used are compatible with the BLC 80/10 microcomputer system. In particular:

Port A and the ADC0801 are at port address E4

Port B is at port address E5

Port C is at port address E6

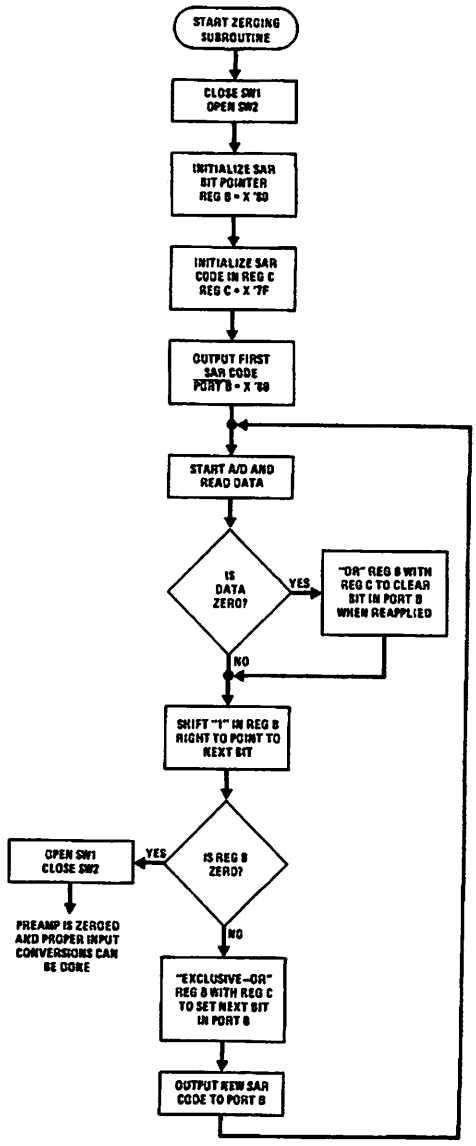
PPI control word port is at port address E7

Program Counter automatically goes to ADDR:3C3D upon acknowledgement of an interrupt from the ADC0801

5.3 Multiple A/D Converters in a Z-80 Interrupt Driven Mode

In data acquisition systems where more than one A/D converter (or other peripheral device) will be interrupting program execution of a microprocessor, there is obviously a need for the CPU to determine which device requires servicing. *Figure 21* and the accompanying software is a method of determining which of 7 ADC0801 converters has completed a conversion ($\overline{INTR}$ asserted) and is requesting an interrupt. This circuit allows starting the A/D converters in any sequence, but will input and store valid data from the converters with a priority sequence of A/D 1 being read first, A/D 2 second, etc., through A/D 7 which would have the lowest priority for data being read. Only the converters whose INT is asserted will be read.

The key to decoding circuitry is the DM74LS373, 8-bit D type flip-flop. When the Z-80 acknowledges the interrupt, the program is vectored to a data input Z-80 subroutine. This subroutine will read a peripheral status word from the DM74LS373 which contains the logic state of the $\overline{INTR}$ outputs of all the converters. Each converter which initiates an interrupt will place a logic "0" in a unique bit position in the status word and the subroutine will determine the identity of the converter and execute a data read. An identifier word (which indicates which A/D the data came from) is stored in the next sequential memory location above the location of the data so the program can keep track of the identity of the data entered.



TL/H/5671–28

FIGURE 19. Flow Chart for Auto-Zero Routine

```
3D00   3E90     MVI 90
3D02   D3E7     Out Control Port                    ; Program PPI
3D04   2601     MVI H 01          Auto-Zero Subroutine
3D06   7C       MOV A,H
3D07   D3E6     OUT C                              ; Close SW1 open SW2
3D09   0680     MVI B 80                           ; Initialize SAR bit pointer
3D0B   3E7F     MVI A 7F                           ; Initialize SAR code
3D0D   4F       MOV C,A           Return
3D0E   D3E5     OUT B                              ; Port B = SAR code
3D10   31AA3D   LXI SP 3DAA       Start            ; Dimension stack pointer
3D13   D3E4     OUT A                              ; Start A/D
3D15   FB       IE
3D16   00       NOP               Loop             ; Loop until INT asserted
3D17   C3163D   JMP Loop
3D1A   7A       MOV A,D           Auto-Zero
3D1B   C600     ADI 00
3D1D   CA2D3D   JZ Set C                           ; Test A/D output data for zero
3D20   78       MOV A,B           Shift B
3D21   F600     ORI 00                             ; Clear carry
3D23   1F       RAR                                ; Shift "1" in B right one place
3D24   FE00     CPI 00                             ; Is B zero? If yes last
3D26   CA373D   JZ Done                            ; approximation has been made
3D29   47       MOV B,A
3D2A   C3333D   JMP New C
3D2D   79       MOV A,C           Set C
3D2E   B0       ORA B                              ; Set bit in C that is in same
3D2F   4F       MOV C,A                            ; position as "1" in B
3D30   C3203D   JMP Shift B
3D33   A9       XRA C             New C            ; Clear bit in C that is in
3D34   C30D3D   JMP Return                         ; same position as "1" in B
3D37   47       MOV B,A           Done             ; then output new SAR code.
3D38   7C       MOV A,H                            ; Open SW1, close SW2 then
3D39   EE03     XRI 03                             ; proceed with program. Preamp
3D3B   D3E6     OUT C                              ; is now zeroed.
3D3D            •                 Normal
                •
                •
                Program for processing
                proper data values
3C3D   DBE4     IN A              Read A/D Subroutine  ; Read A/D data
3C3F   EEFF     XRI FF                             ; Invert data
3C41   57       MOV D,A
3C42   78       MOV A,B                            ; Is B Reg = 0? If not stay
3C43   E6FF     ANI FF                             ; in auto zero subroutine
3C45   C21A3D   JNZ Auto-Zero
3C48   C33D3D   JMP Normal
```

Note: All numerical values are hexadecimal representations.

FIGURE 20. Software for Auto-Zeroed Differential A/D

5.3 Multiple A/D Converters in a Z-80® Interrupt Driven Mode (Continued)

The following notes apply:

1) It is assumed that the CPU automatically performs a RST 7 instruction when a valid interrupt is acknowledged (CPU is in interrupt mode 1). Hence, the subroutine starting address of X0038.

2) The address bus from the Z-80 and the data bus to the Z-80 are assumed to be inverted by bus drivers.

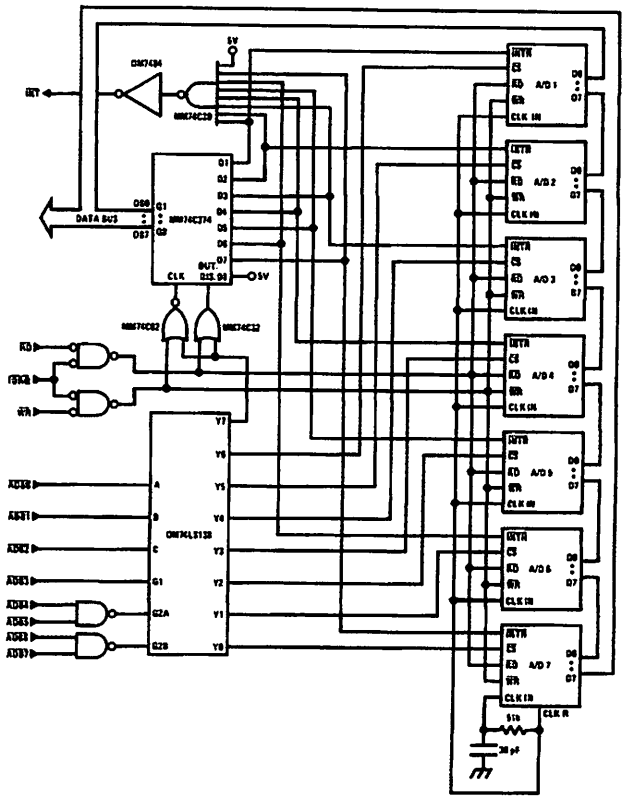
3) A/D data and identifying words will be stored in sequential memory locations starting at the arbitrarily chosen address X 3E00.

4) The stack pointer must be dimensioned in the main program as the RST 7 instruction automatically pushes the PC onto the stack and the subroutine uses an additional 6 stack addresses.

5) The peripherals of concern are mapped into I/O space with the following port assignments:

HEX PORT ADDRESS	PERIPHERAL
00	MM74C374 8-bit flip-flop
01	A/D 1
02	A/D 2
03	A/D 3
04	A/D 4
05	A/D 5
06	A/D 6
07	A/D 7

This port address also serves as the A/D identifying word in the program.



TL/H/5671-29

FIGURE 21. Multiple A/Ds with Z-80 Type Microprocessor

INTERRUPT SERVICING SUBROUTINE

LOC	OBJ CODE	SOURCE STATEMENT		COMMENT
0038	E5		PUSH HL	; Save contents of all registers affected by
0039	C5		PUSH BC	; this subroutine.
003A	F5		PUSH AF	; Assumed INT mode 1 earlier set.
003B	21 00 3E		LD (HL), X3E00	; Initialize memory pointer where data will be stored.
003E	0E 01		LD C, X01	; C register will be port ADDR of A/D converters.
0040	D300		OUT X00, A	; Load peripheral status word into 8-bit latch.
0042	DB00		IN A, X00	; Load status word into accumulator.
0044	47		LD B, A	; Save the status word.
0045	79	TEST	LD A, C	; Test to see if the status of all A/D's have
0046	FE 08		CP, X08	; been checked. If so, exit subroutine
0048	CA 60 00		JPZ, DONE	
004B	78		LD A, B	; Test a single bit in status word by looking for
004C	1F		RRA	; a "1" to be rotated into the CARRY (an $\overline{INT}$
004D	47		LD B, A	; is loaded as a "1"). If CARRY is set then load
004E	DA 5500		JPC, LOAD	; contents of A/D at port ADDR in C register.
0051	0C	NEXT	INC C	; If CARRY is not set, increment C register to point
0052	C3 4500		JP, TEST	; to next A/D, then test next bit in status word.
0055	ED 78	LOAD	IN A, (C)	; Read data from interrupting A/D and invert
0057	EE FF		XOR FF	; the data.
0059	77		LD (HL), A	; Store the data
005A	2C		INC L	
005B	71		LD (HL), C	; Store A/D identifier (A/D port ADDR).
005C	2C		INC L	
005D	C3 51 00		JP, NEXT	; Test next bit in status word.
0060	F1	DONE	POP AF	; Re-establish all registers as they were
0061	C1		POP BC	; before the interrupt.
0062	E1		POP HL	
0063	C9		RET	; Return to original program

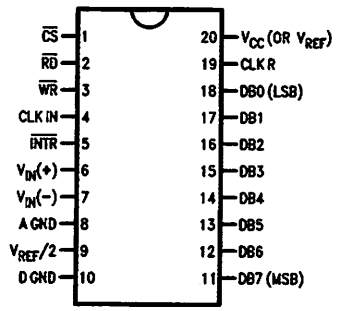
Ordering Information

TEMP RANGE		0°C TO 70°C	0°C TO 70°C	0°C TO 70°C	−40°C TO +85°C
ERROR	±¼ Bit Adjusted				ADC0801LCN
	±½ Bit Unadjusted	ADC0802LCWM	ADC0802LCV		ADC0802LCN
	±½ Bit Adjusted	ADC0803LCWM	ADC0803LCV		ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCV	ADC0804LCN	ADC0805LCN
PACKAGE OUTLINE		M20B—Small Outline	V20A—Chip Carrier	N20A—Molded DIP	

TEMP RANGE		−40°C TO +85°C	−55°C TO +125°C
ERROR	±¼ Bit Adjusted	ADC0801LCJ	ADC0801LJ
	±½ Bit Unadjusted	ADC0802LCJ	ADC0802LJ,
	±½ Bit Adjusted	ADC0803LCJ	ADC0802LJ/883
	±1Bit Unadjusted	ADC0804LCJ	
PACKAGE OUTLINE		J20A—Cavity DIP	J20A—Cavity DIP

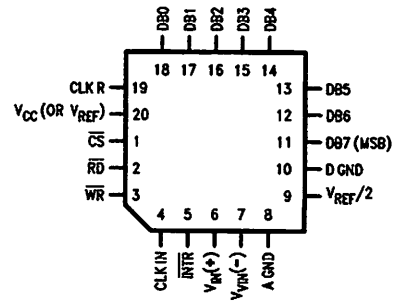
Connection Diagrams

ADC080X
Dual-In-Line and Small Outline (SO) Packages



TL/H/5671–30

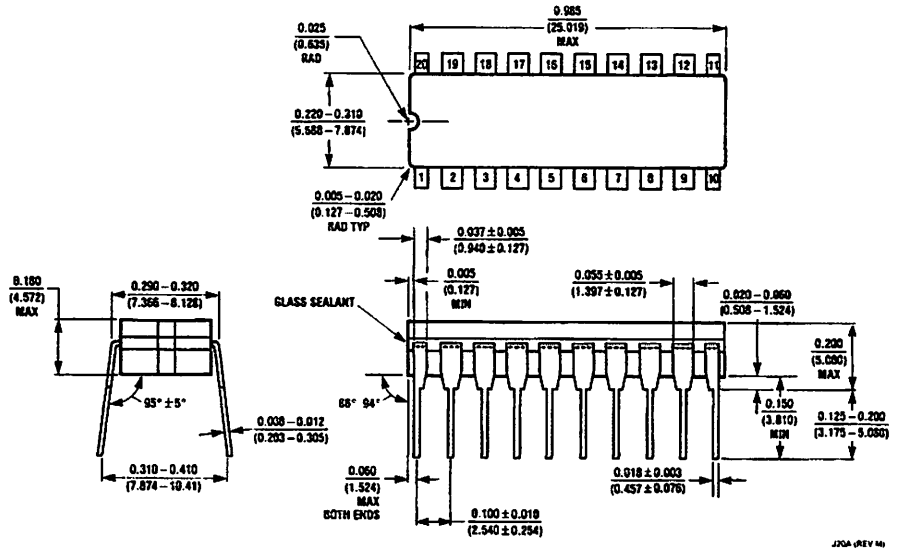
ADC080X
Molded Chip Carrier (PCC) Package



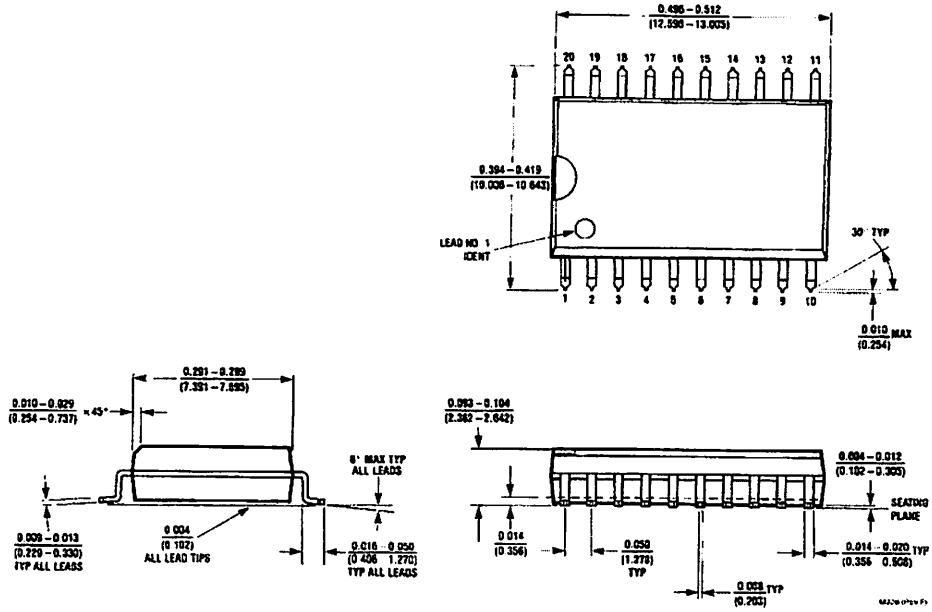
TL/H/5671–32

See Ordering Information

Physical Dimensions inches (millimeters)

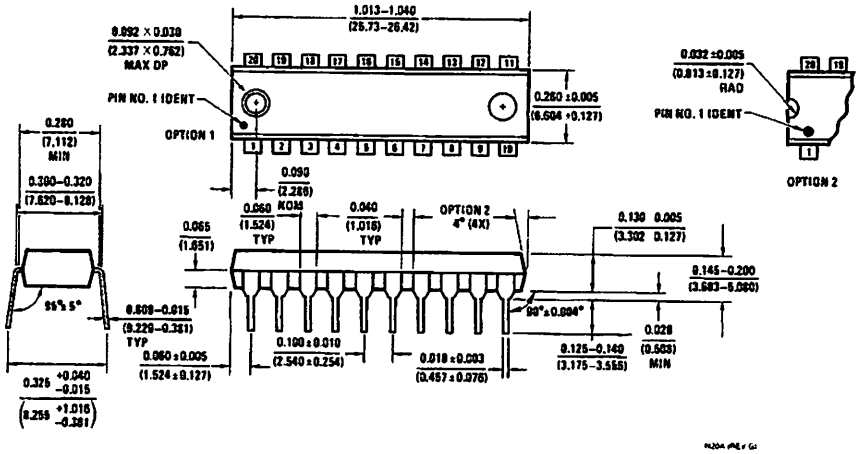


Dual-In-Line Package (J)
Order Number ADC0801LJ, ADC0802LJ, ADC0801LCJ, ADC0802LCJ, ADC0803LCJ or ADC0804LCJ
ADC0802LJ/883 or 5962-9096601MRA
NS Package Number J20A



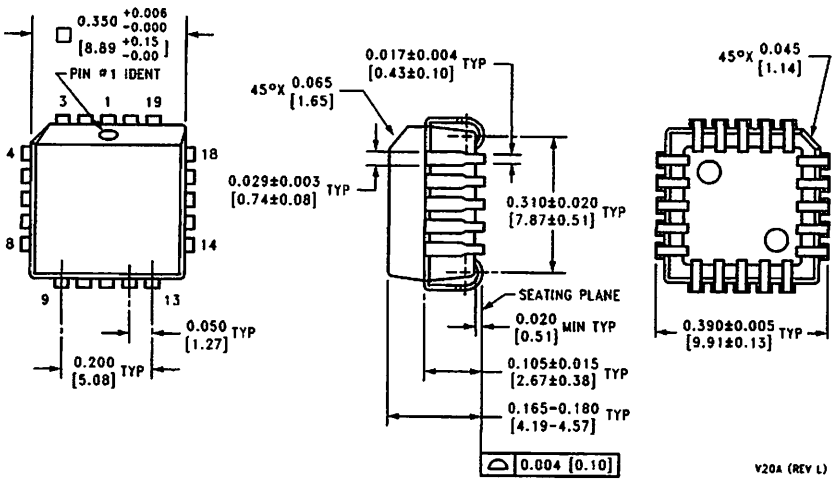
SO Package (M)
Order Number ADC0802LCWM, ADC0803LCWM or ADC0804LCWM
NS Package Number M20B

Physical Dimensions inches (millimeters) (Continued)



Molded Dual-In-Line Package (N)
Order Number ADC0801LCN, ADC0802LCN,
ADC0803LCN, ADC0804LCN or ADC0805LCN
NS Package Number N20A

Physical Dimensions inches (millimeters) (Continued)



Molded Chip Carrier Package (V)
Order Number ADC0802LCV, ADC0803LCV or ADC0804LCV
NS Package Number V20A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

19-4323; Rev 13; 4/04

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

AutoShutdown and UCSP are trademarks of Maxim Integrated Products, Inc.

Next-Generation Device Features

- **For Low-Voltage, Integrated ESD Applications: MAX3222E/MAX3232E/MAX3237E/MAX3241E/ MAX3246E: +3.0V to +5.5V, Low-Power, Up to 1Mbps, True RS-232 Transceivers Using Four 0.1µF External Capacitors (MAX3246E Available in a UCSP™ Package)**
- **For Low-Cost Applications: MAX221E: ±15kV ESD-Protected, +5V, 1µA, Single RS-232 Transceiver with AutoShutdown™**

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	0.047/0.33	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, 10s) ..... +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	
		MAX220, V_{OUT} = ±12V			±25	
Data Rate				200	116	kbps
Transmitter Output Resistance	V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V_{OUT} = 0V	V_{OUT} = 0V	±7	±22		mA
		MAX220			±60	
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
		MAX220			±25	
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
	T_A = +25°C (MAX220)		3	5	7	

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 Voltage Low	I_{OUT} = 3.2mA			0.2	0.4	V
	I_{OUT} = 1.6mA (MAX220)				0.4	
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	$\overline{SHDN}$ = 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

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

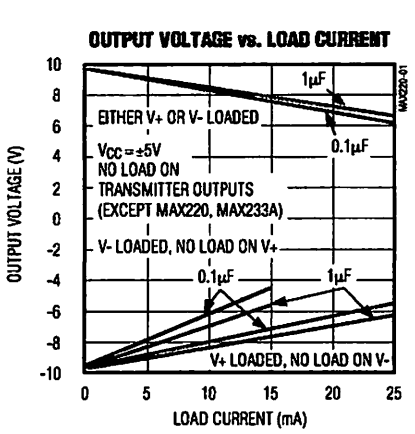
ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)

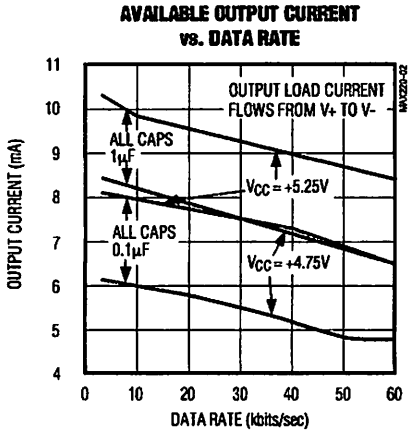
(V_{CC} = +5V ±10%, C1–C4 = 0.1µF, MAX220, C1 = 0.047µF, C2–C4 = 0.33µF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

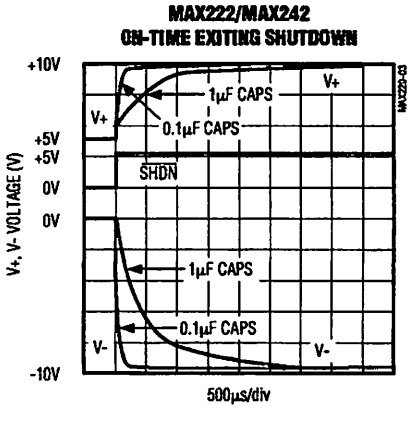
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
Transmitter-Output Enable Time ($\overline{SHDN}$ Goes High), Figure 4	t_{ET}	MAX222/242, 0.1µF caps (includes charge-pump start-up)		250		µs
Transmitter-Output Disable Time ($\overline{SHDN}$ Goes Low), Figure 4	t_{DT}	MAX222/242, 0.1µF caps		600		ns
Transmitter + 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		

Typical Operating Characteristics

MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243







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, 10s) ..... +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)

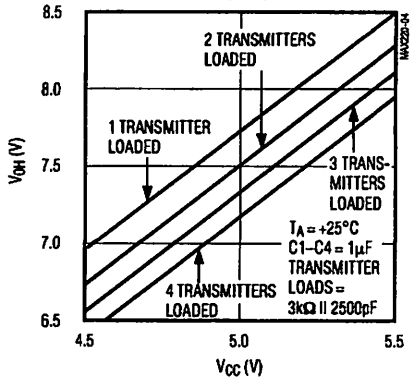
(MAX223/230/232/234/236/237/238/240/241, V_{CC} = +5V ±10; MAX233/MAX235, V_{CC} = 5V ±5%, C1–C4 = 1.0µF; MAX231/MAX239, V_{CC} = 5V ±10%; V+ = 7.5V to 13.2V; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.)

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

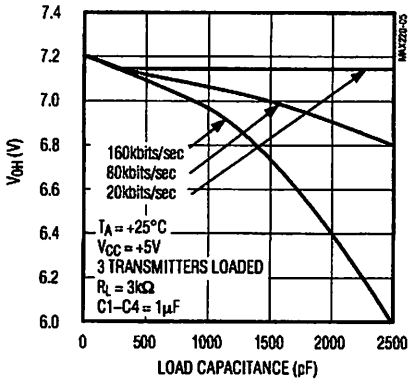
Typical Operating Characteristics

MAX223/MAX230–MAX241

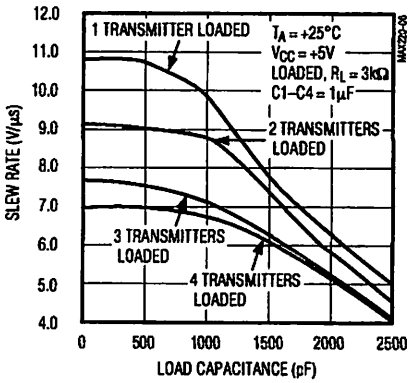
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. V_{CC}



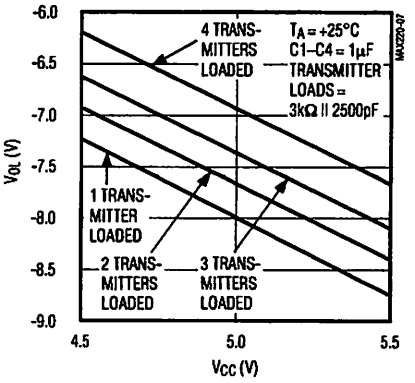
TRANSMITTER OUTPUT VOLTAGE (V_{OH}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



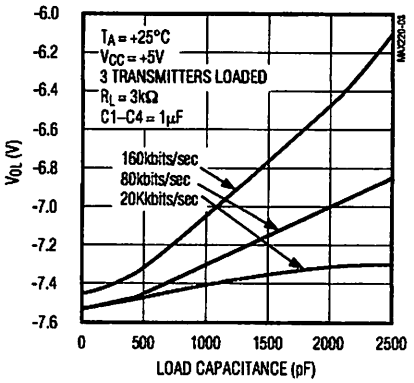
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



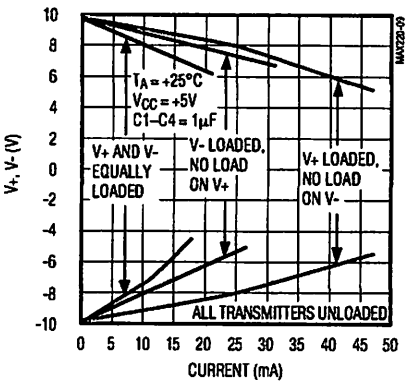
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. V_{CC}



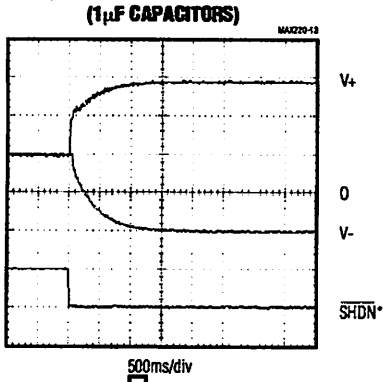
TRANSMITTER OUTPUT VOLTAGE (V_{OL}) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



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
Input 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, 10s) ..... +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	kbps
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.)

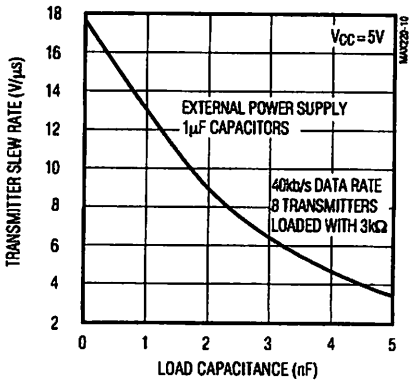
PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
POWER SUPPLY AND CONTROL LOGIC						
Operating Supply Voltage		MAX225	4.75		5.25	V
		MAX244–MAX249	4.5		5.5	
V_{CC} Supply Current (Normal Operation)	No load	MAX225		10	20	mA
		MAX244–MAX249		11	30	
	3kΩ loads on all outputs	MAX225		40		
		MAX244–MAX249		57		
Shutdown 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 startup)		5		µs
		MAX225/MAX245–MAX249 (includes charge-pump startup)		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.

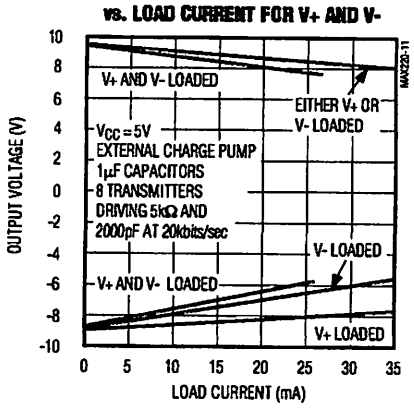
Typical Operating Characteristics

MAX225/MAX244–MAX249

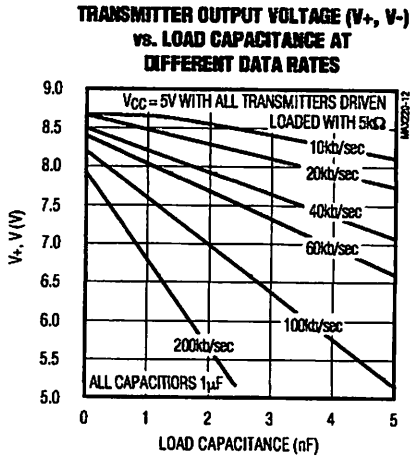
TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



OUTPUT VOLTAGE vs. LOAD CURRENT FOR V+ AND V-



TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES



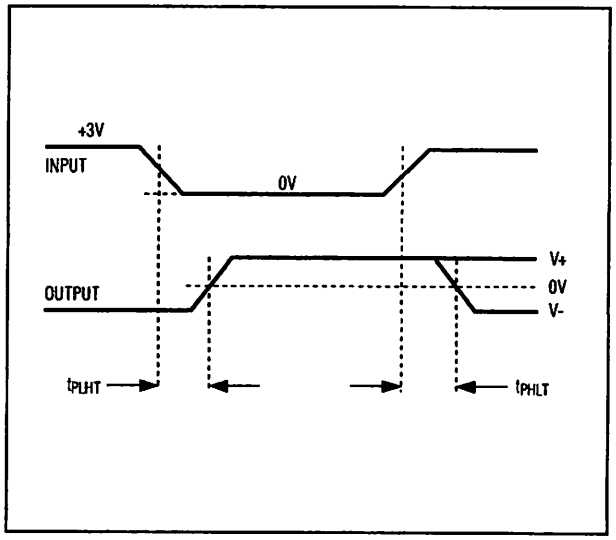


Figure 1. Transmitter Propagation-Delay Timing

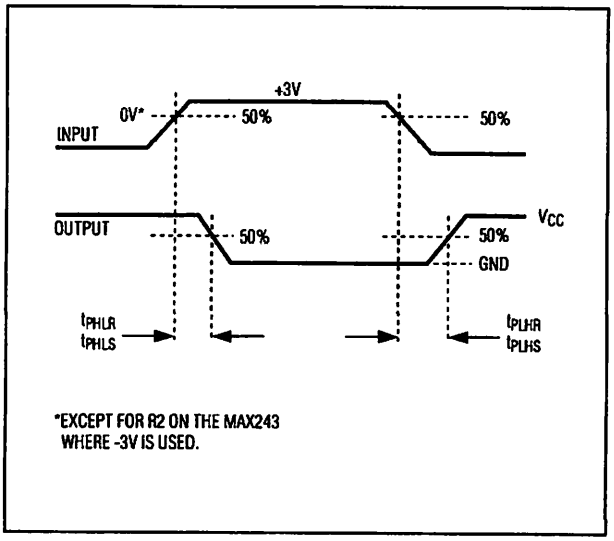


Figure 2. Receiver Propagation-Delay Timing

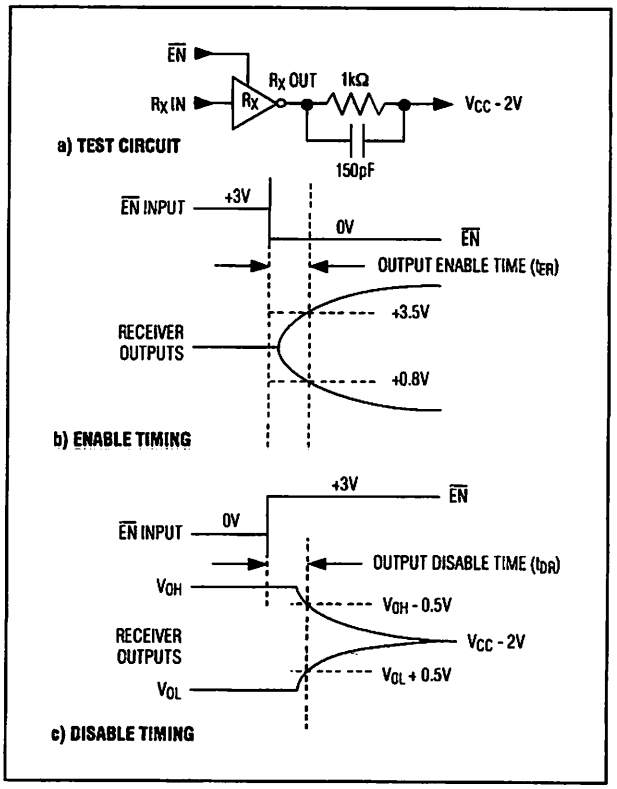


Figure 3. Receiver-Output Enable and Disable Timing

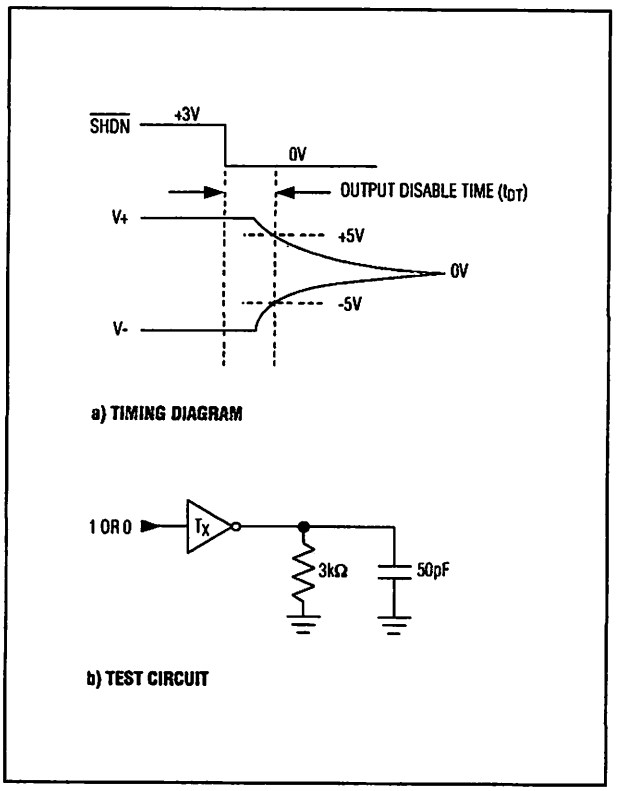


Figure 4. Transmitter-Output Disable Timing

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).

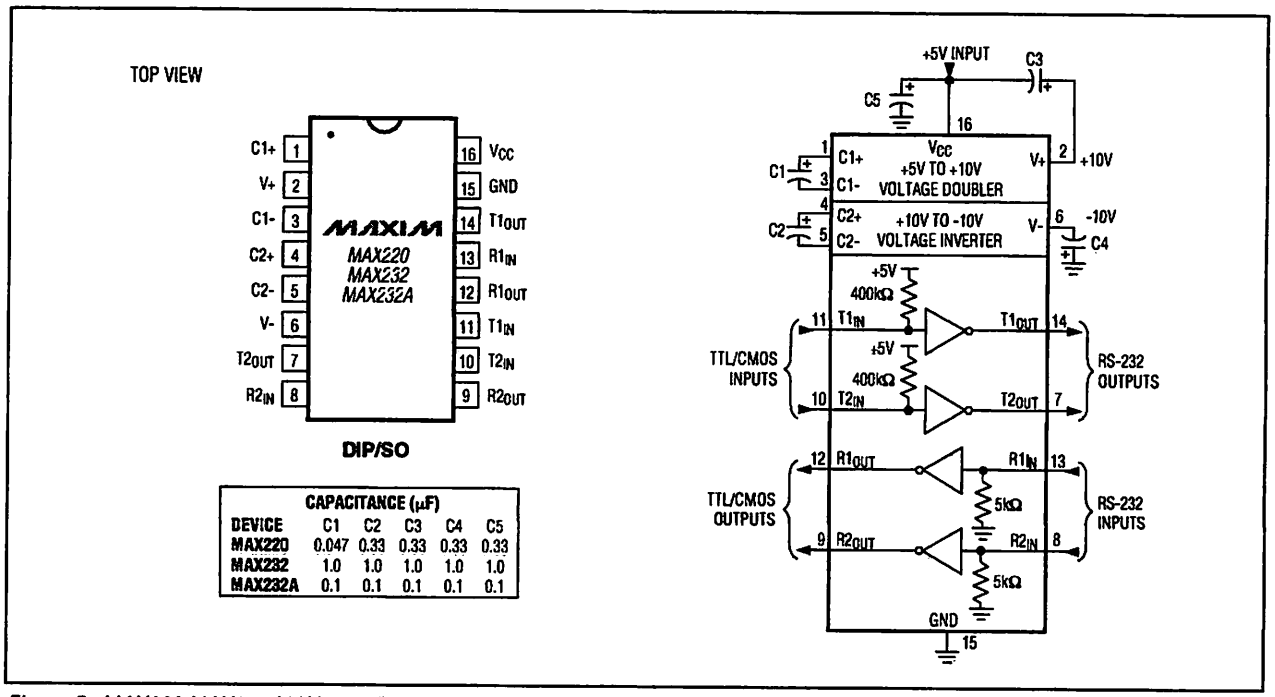
The MAX247 provides nine receivers and eight drivers with four control pins. The $\overline{ENRA}$ and $\overline{ENRB}$ receiver enable inputs each control four receiver outputs. The $\overline{ENTA}$ and $\overline{ENTB}$ transmitter enable inputs each control four drivers. The ninth receiver (RB5) 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	0.047	0.33	0.33	0.33	0.33
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

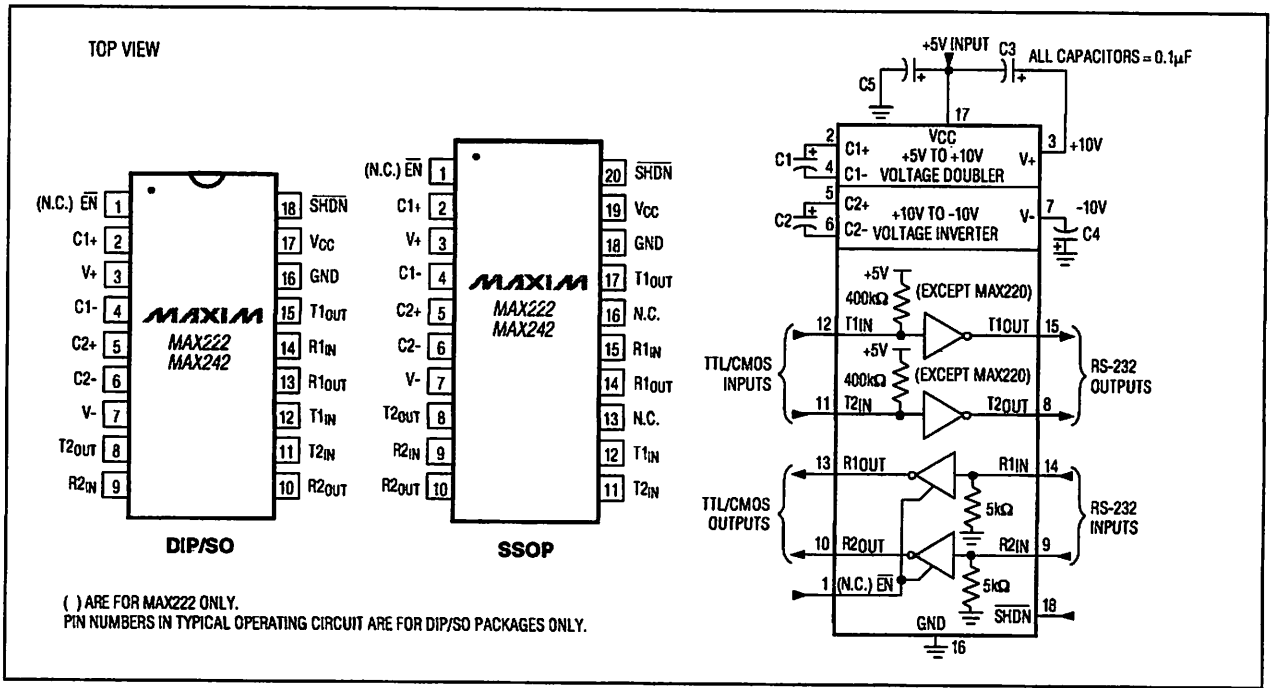
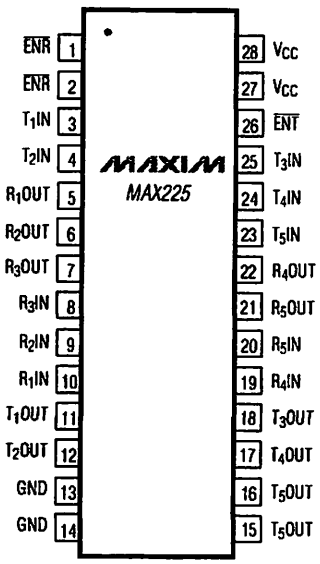


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

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.

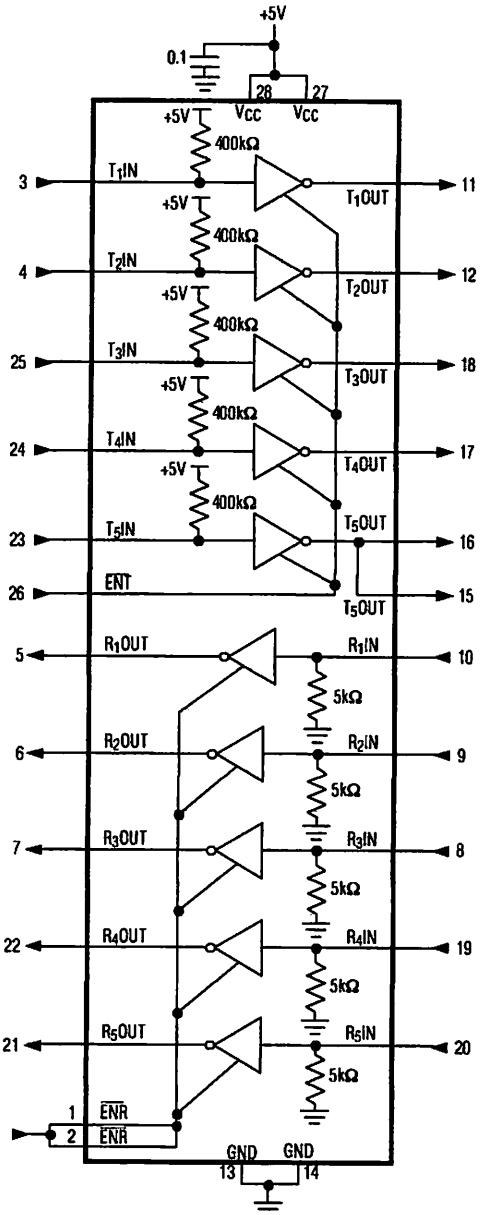


Figure 7. MAX225 Pin Configuration and Typical Operating Circuit

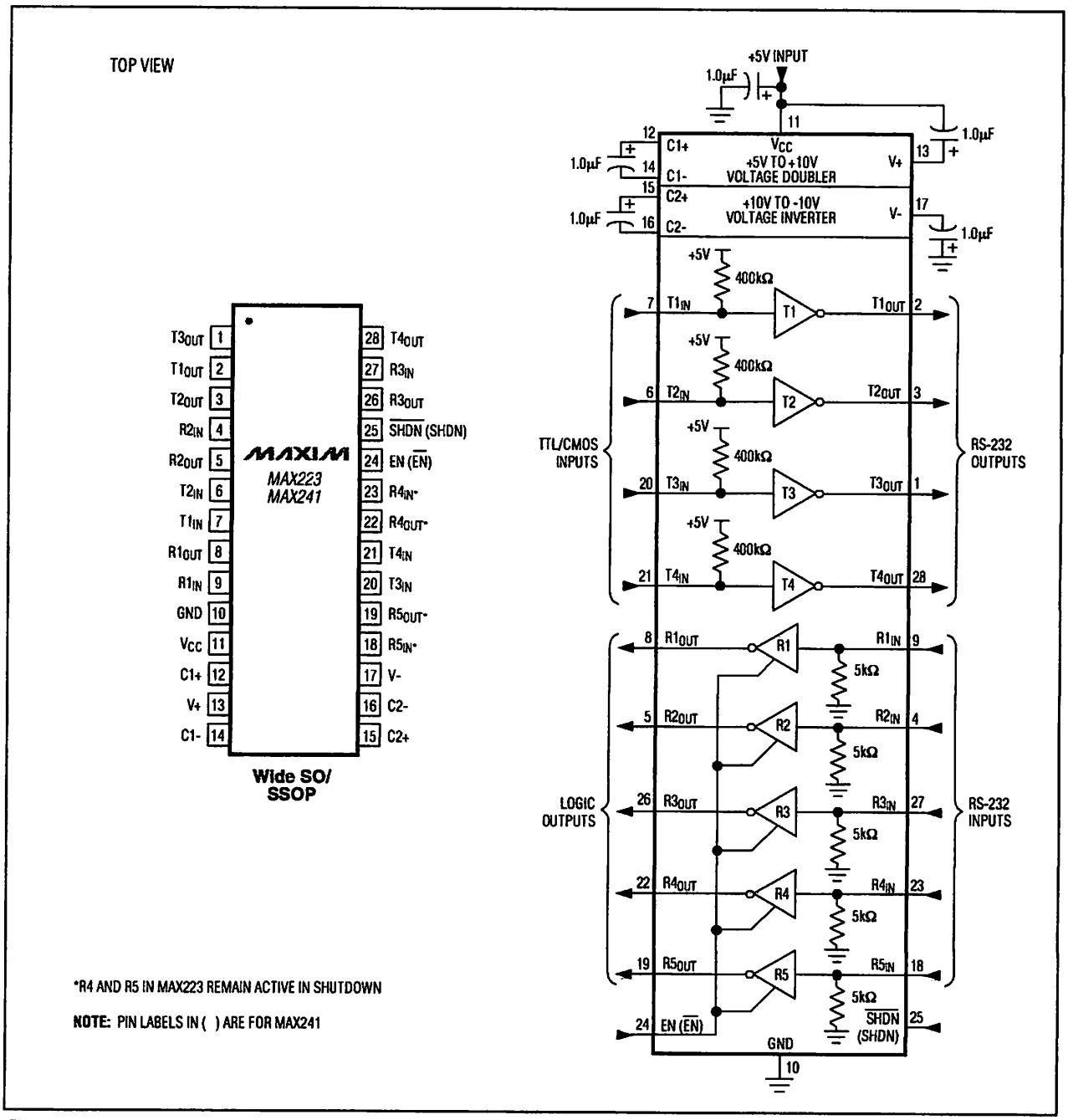


Figure 8. MAX223/MAX241 Pin Configuration and Typical Operating Circuit

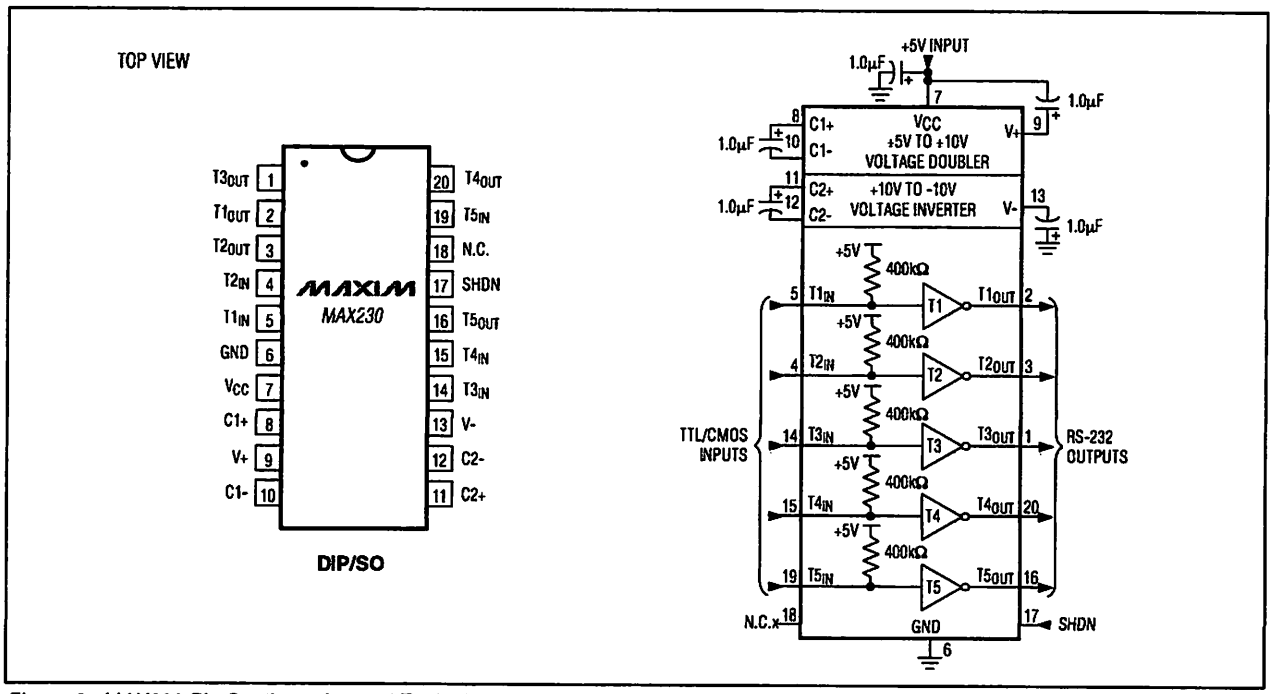


Figure 9. MAX230 Pin Configuration and Typical Operating Circuit

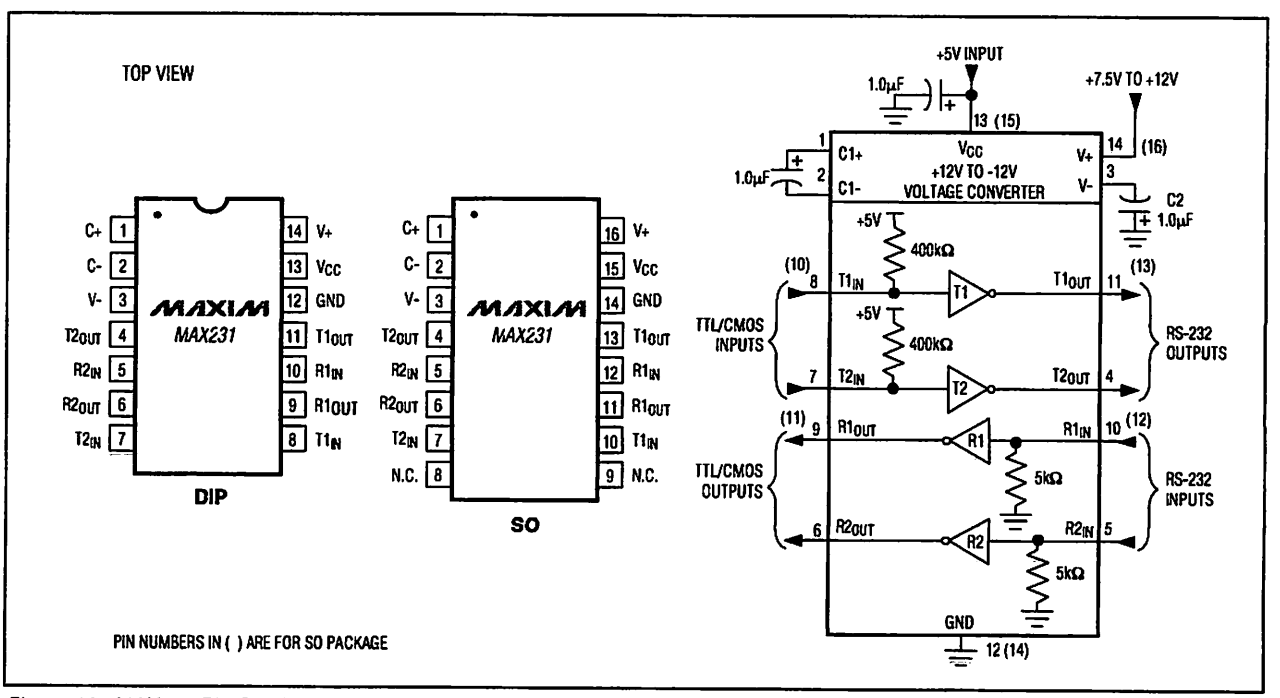


Figure 10. MAX231 Pin Configurations and Typical Operating Circuit

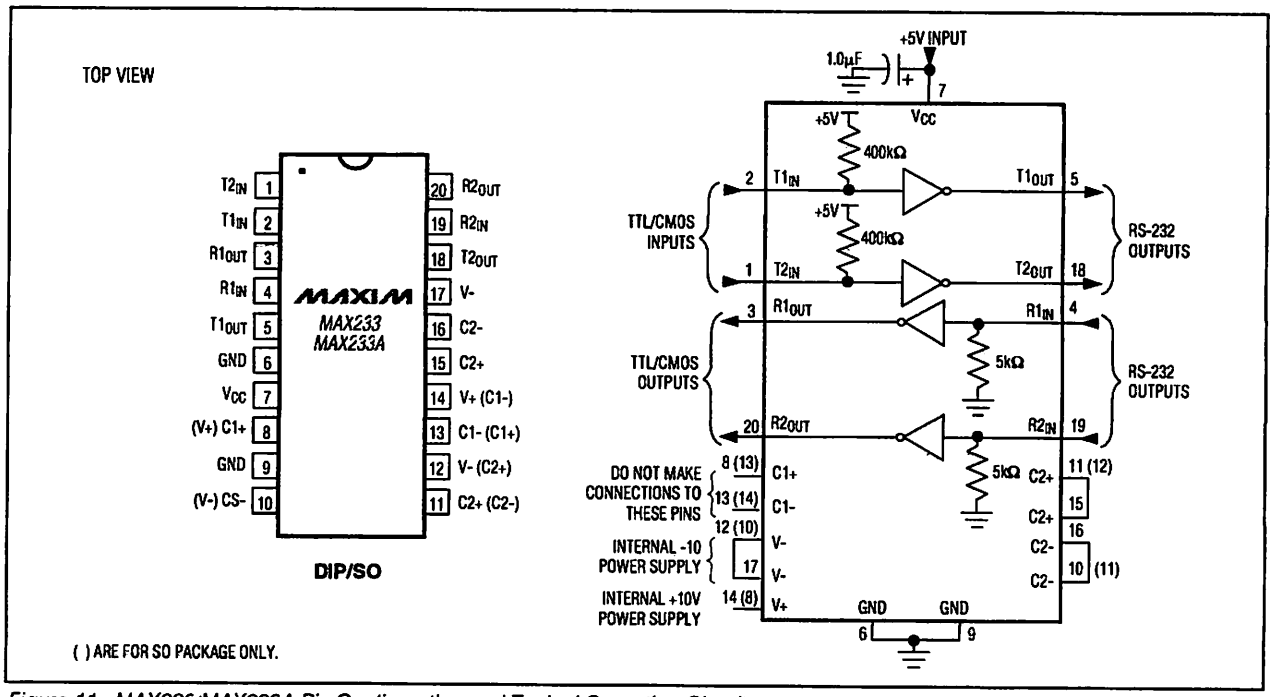


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

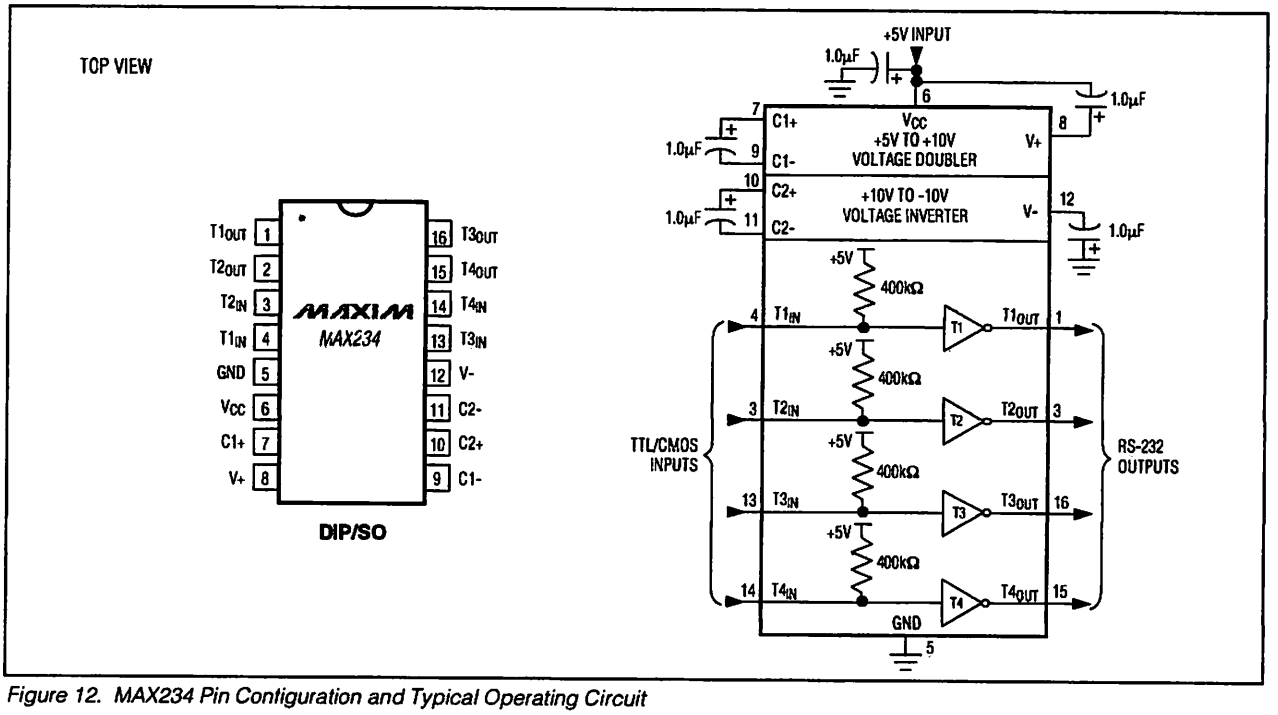


Figure 12. MAX234 Pin Configuration and Typical Operating Circuit

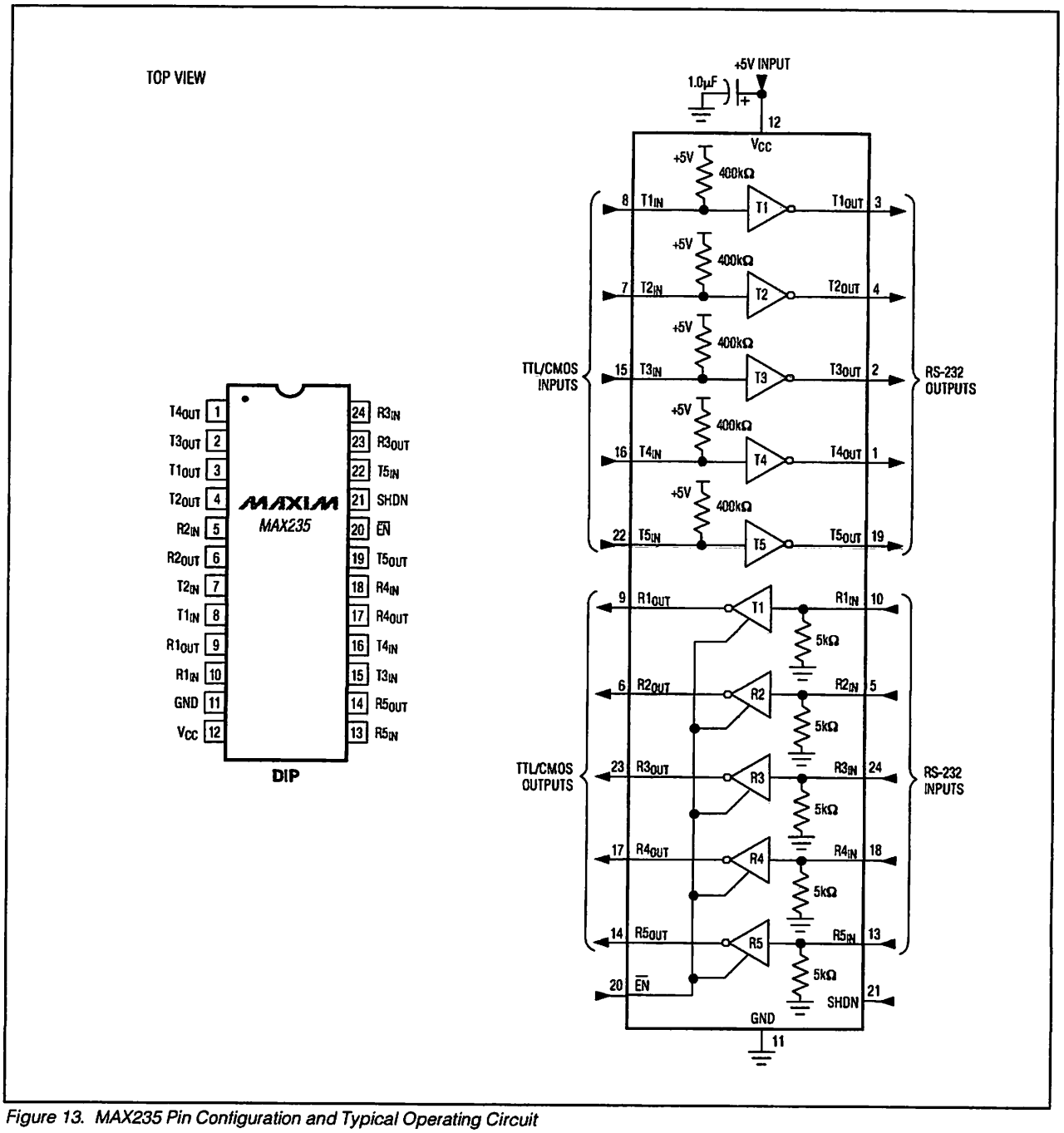


Figure 13. MAX235 Pin Configuration and Typical Operating Circuit

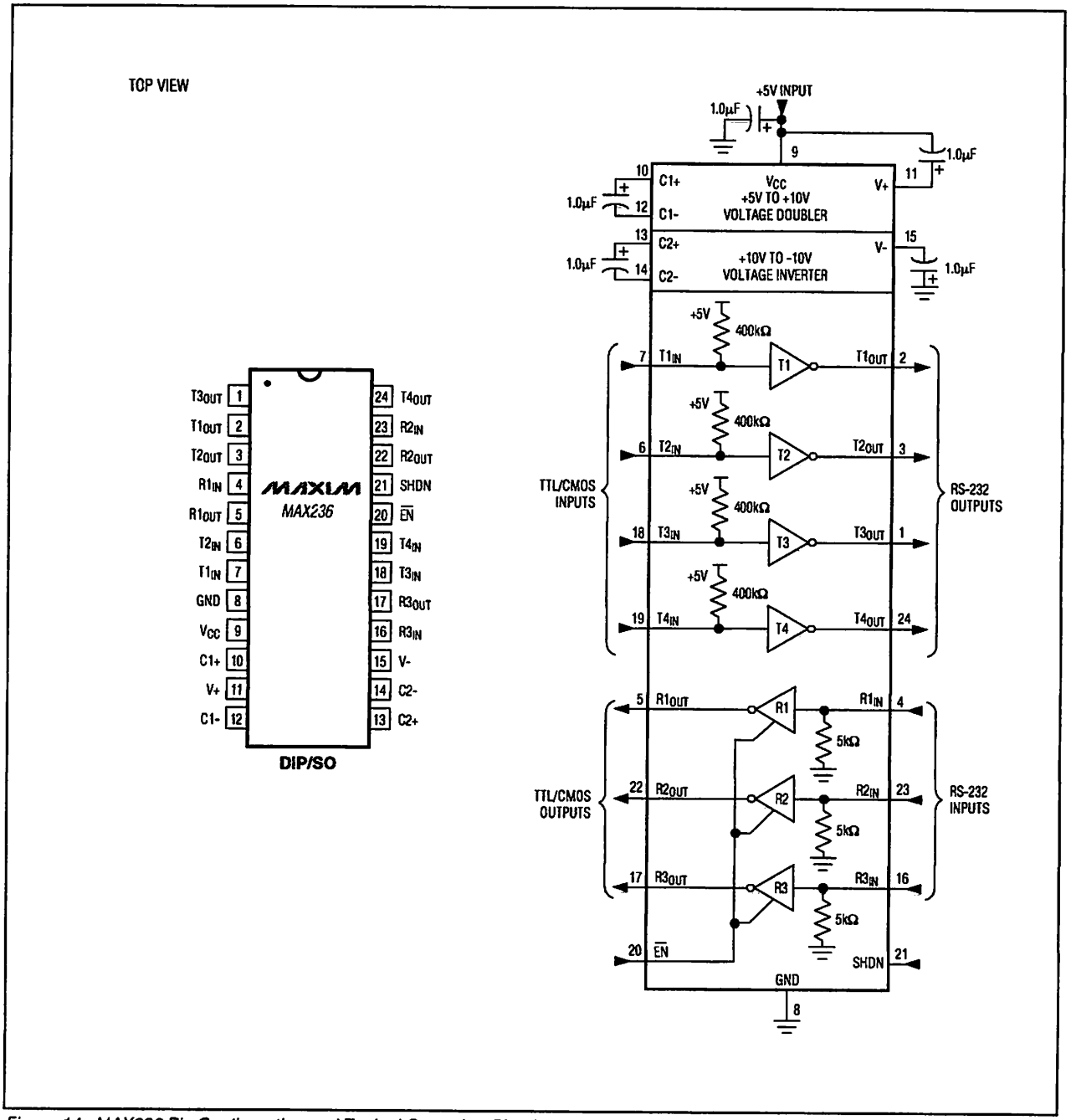


Figure 14. MAX236 Pin Configuration and Typical Operating Circuit

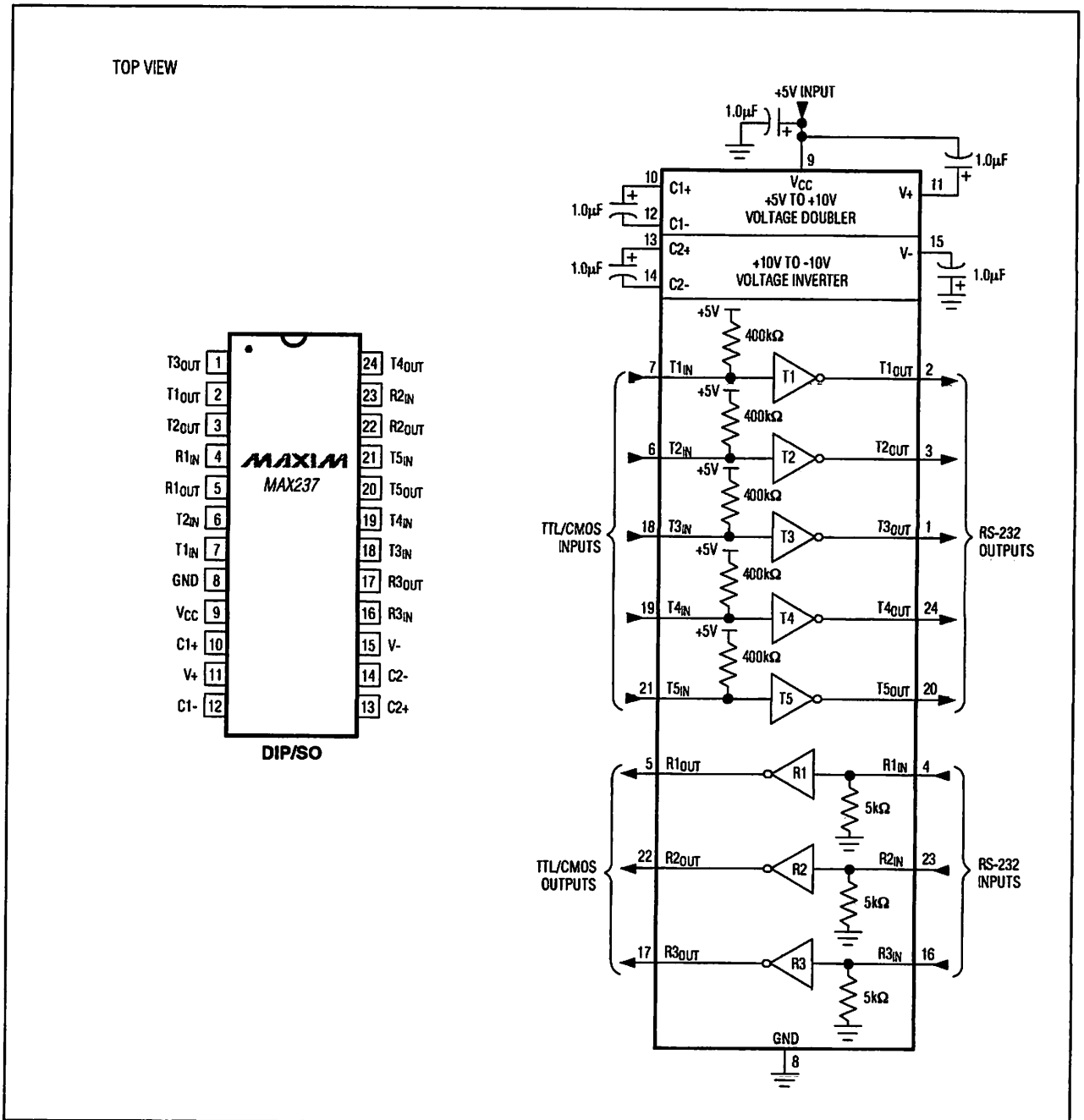


Figure 15. MAX237 Pin Configuration and Typical Operating Circuit

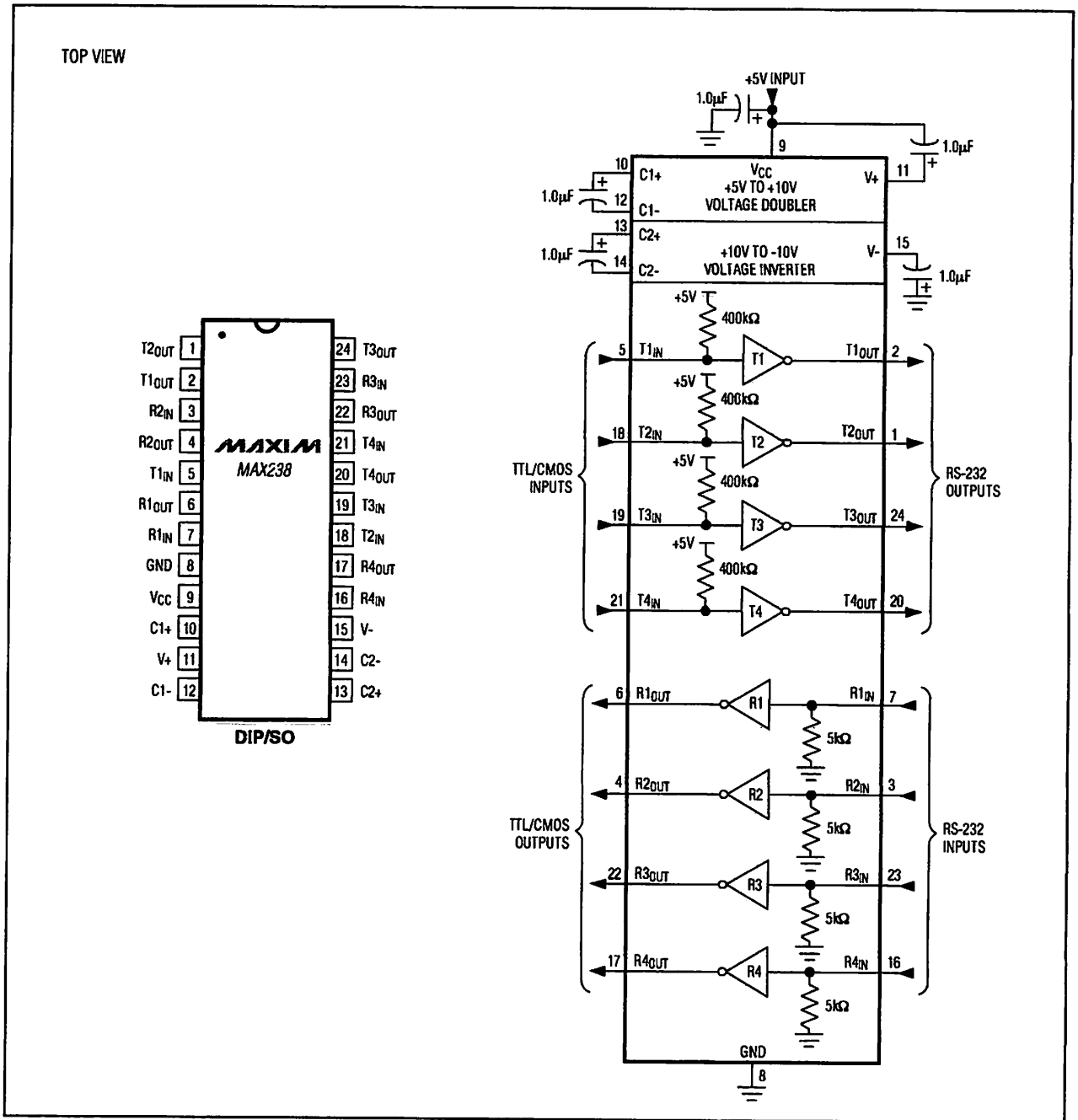
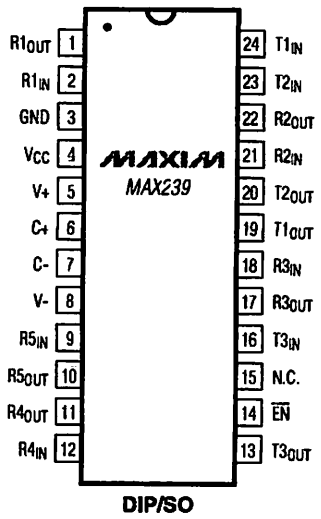


Figure 16. MAX238 Pin Configuration and Typical Operating Circuit



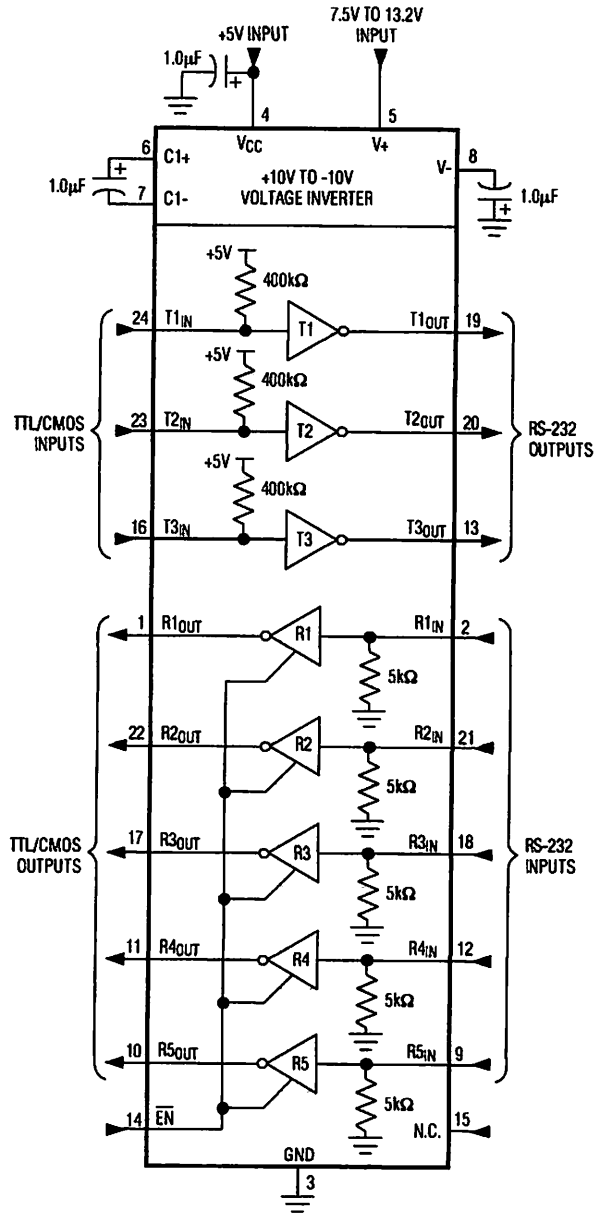


Figure 17. MAX239 Pin Configuration and Typical Operating Circuit

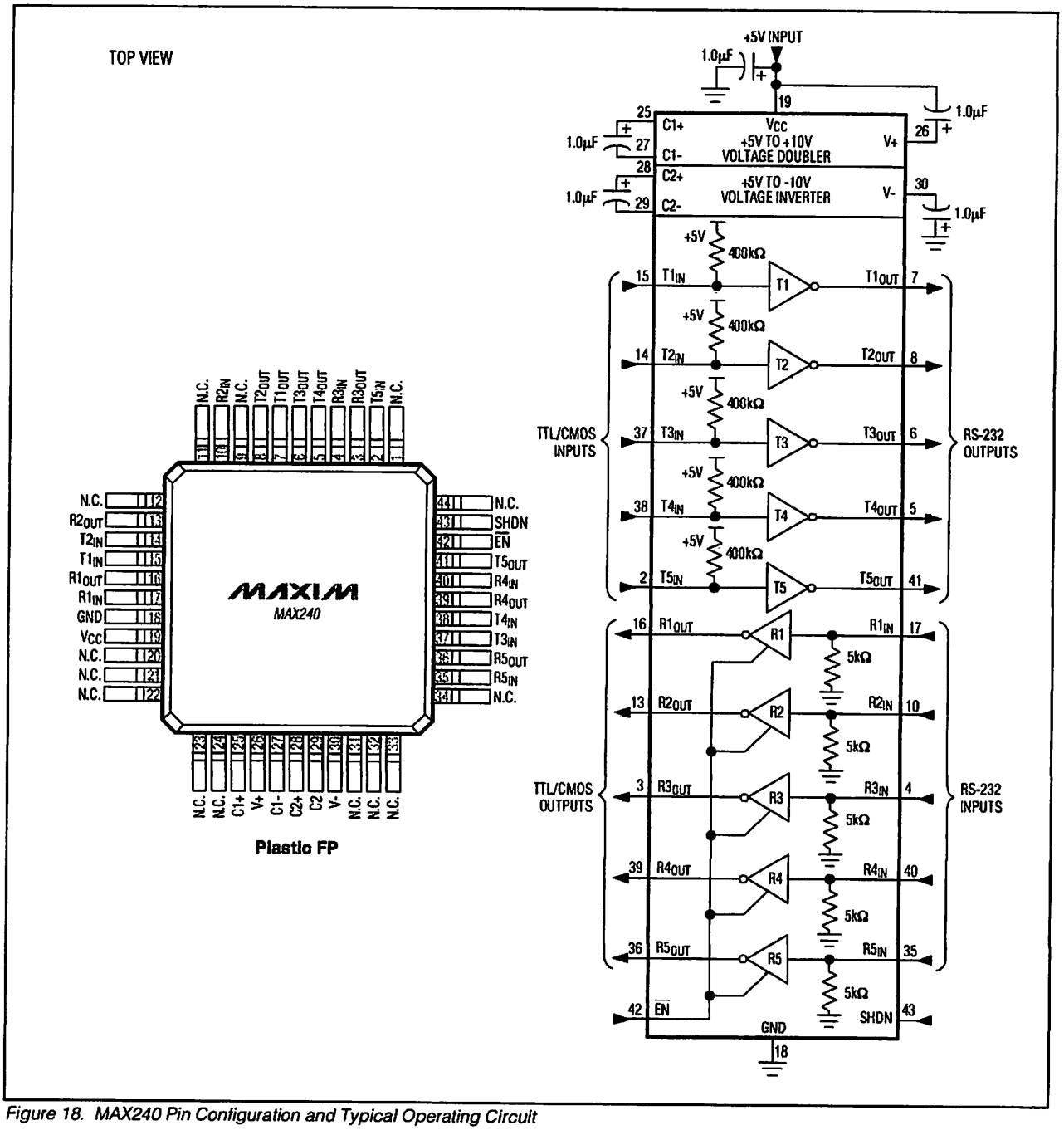
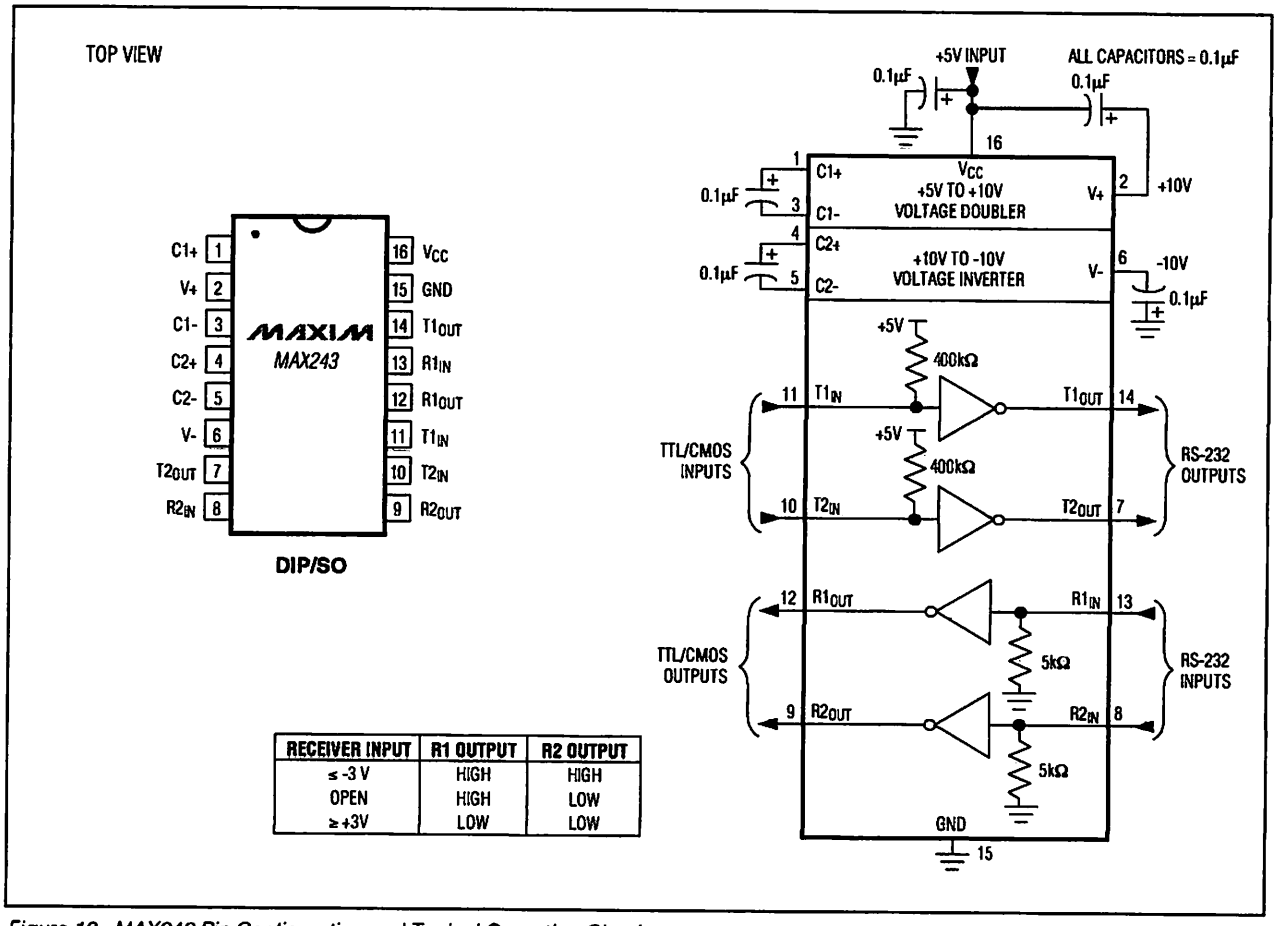
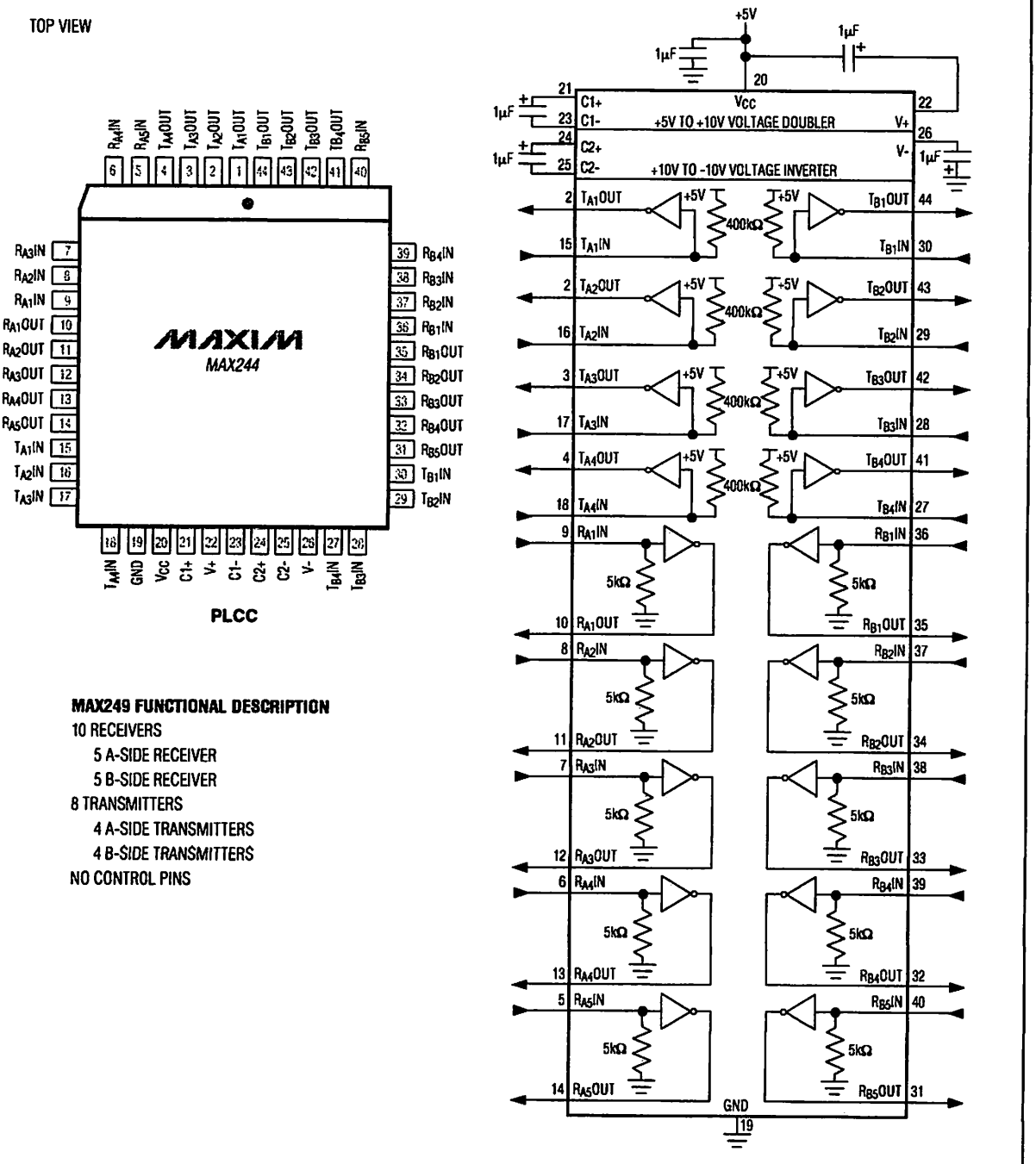


Figure 18. MAX240 Pin Configuration and Typical Operating Circuit



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

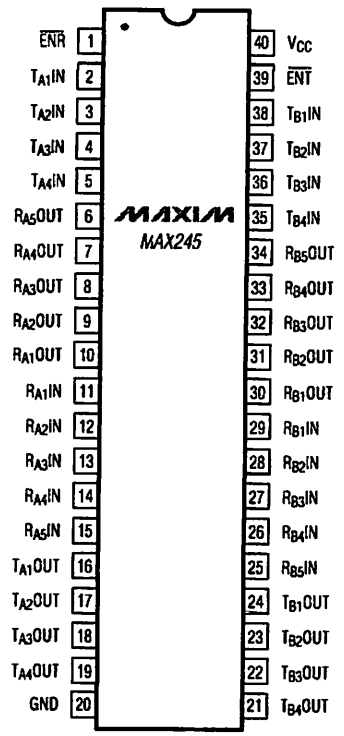
10 RECEIVERS
- 5 A-SIDE RECEIVER
- 5 B-SIDE RECEIVER

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

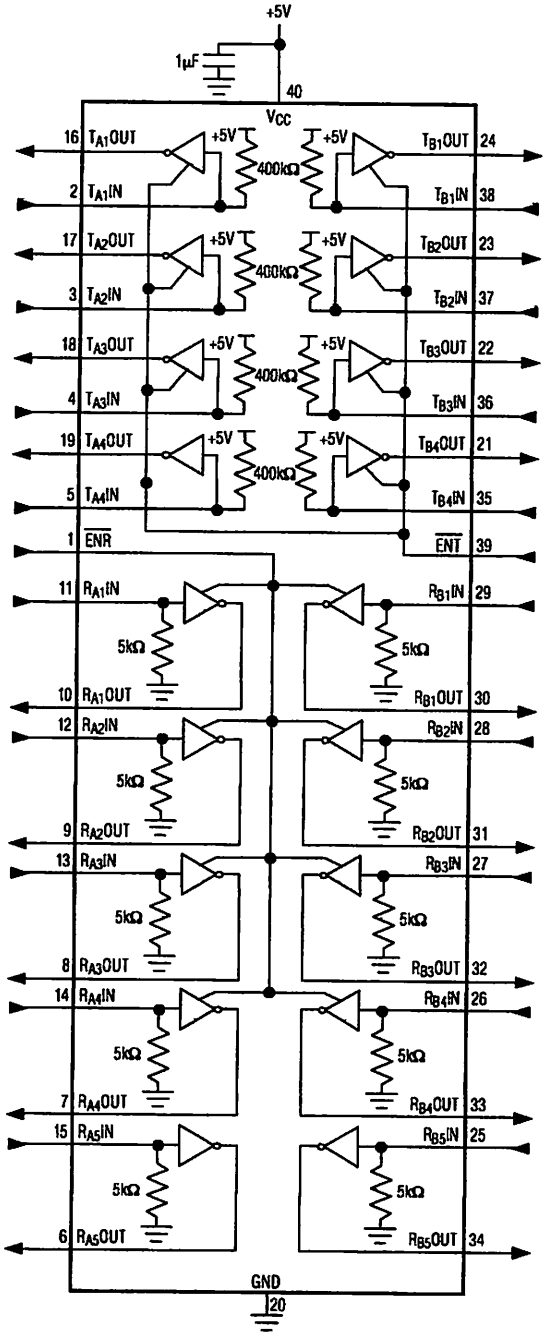
NO CONTROL PINS

Figure 20. MAX244 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP



MAX245 FUNCTIONAL DESCRIPTION

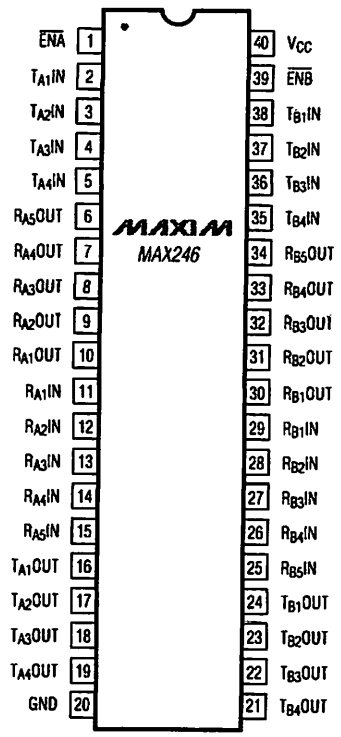
10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

8 TRANSMITTTERS
- 4 A-SIDE TRANSMITTERS

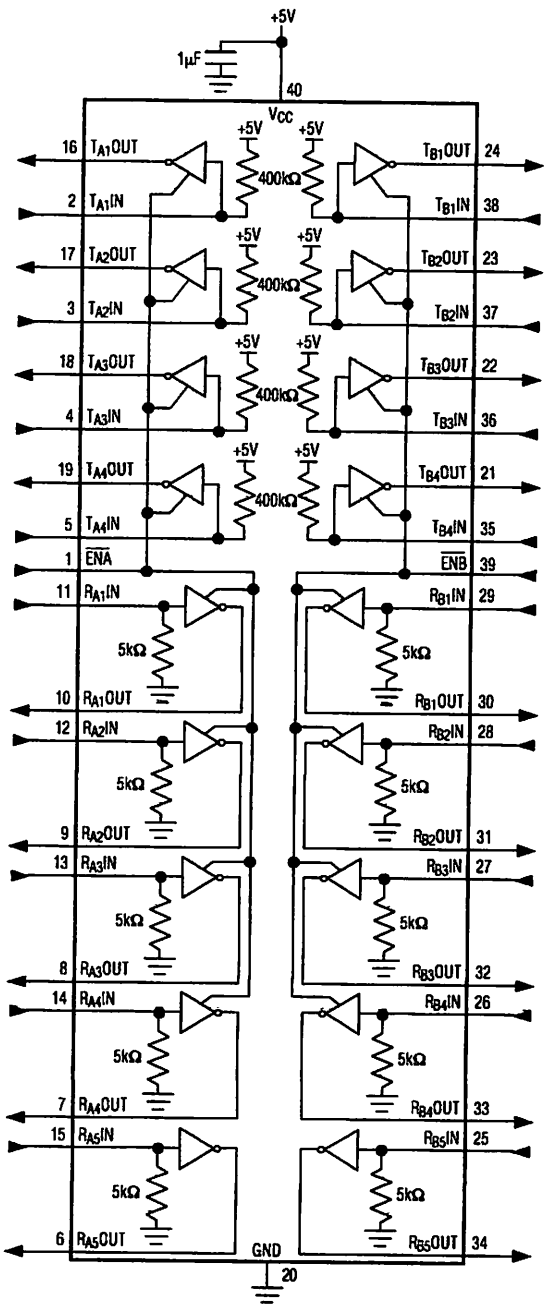
2 CONTROL PINS
- 1 RECEIVER ENABLE ($\overline{ENR}$)
- 1 TRANSMITTER ENABLE ($\overline{ENT}$)

Figure 21. MAX245 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP



MAX246 FUNCTIONAL DESCRIPTION

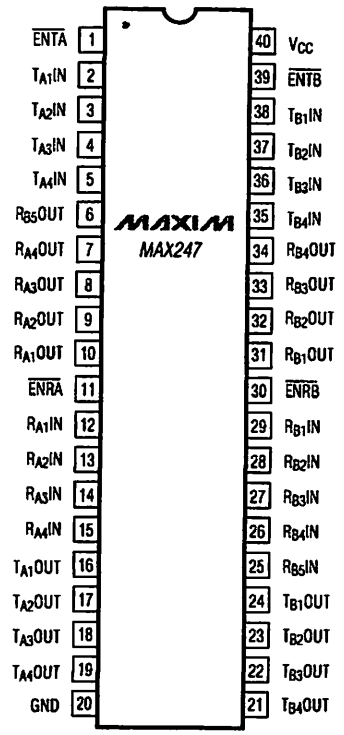
10 RECEIVERS
- 5 A-SIDE RECEIVERS (R_{A5} ALWAYS ACTIVE)
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

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

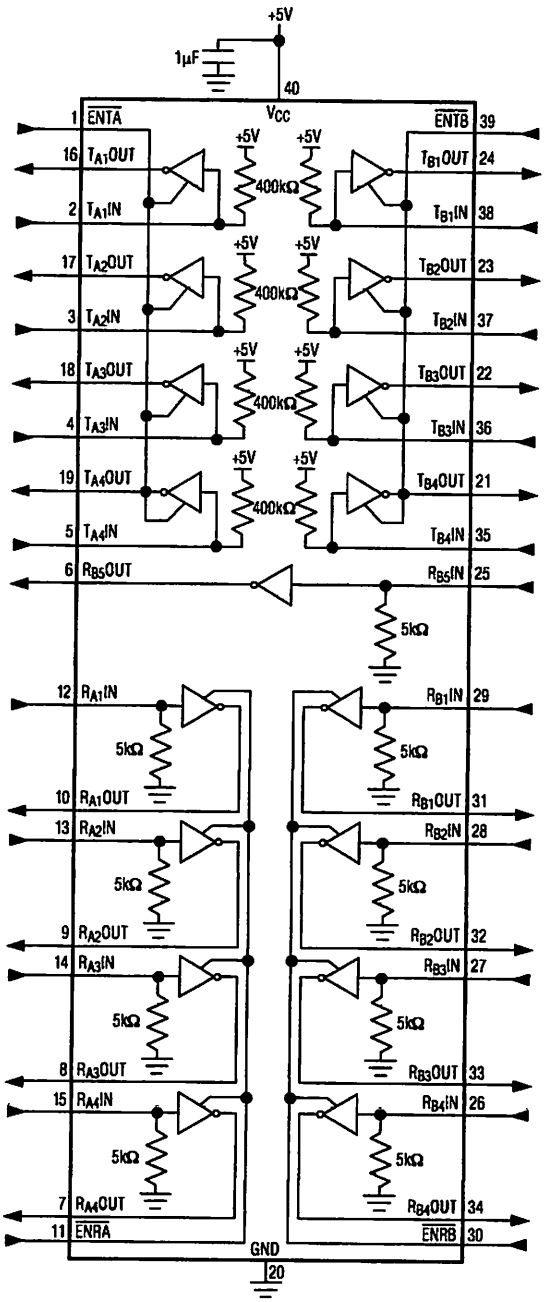
2 CONTROL PINS
- ENABLE A-SIDE ($\overline{ENA}$)
- ENABLE B-SIDE ($\overline{ENB}$)

Figure 22. MAX246 Pin Configuration and Typical Operating Circuit

TOP VIEW



DIP



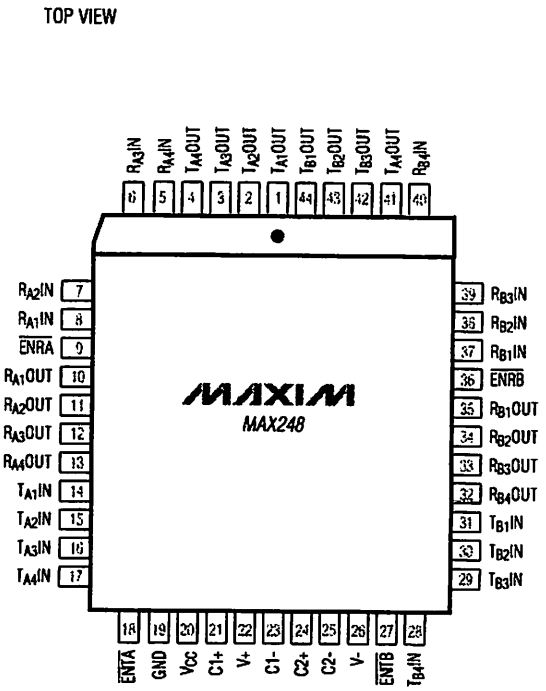
MAX247 FUNCTIONAL DESCRIPTION

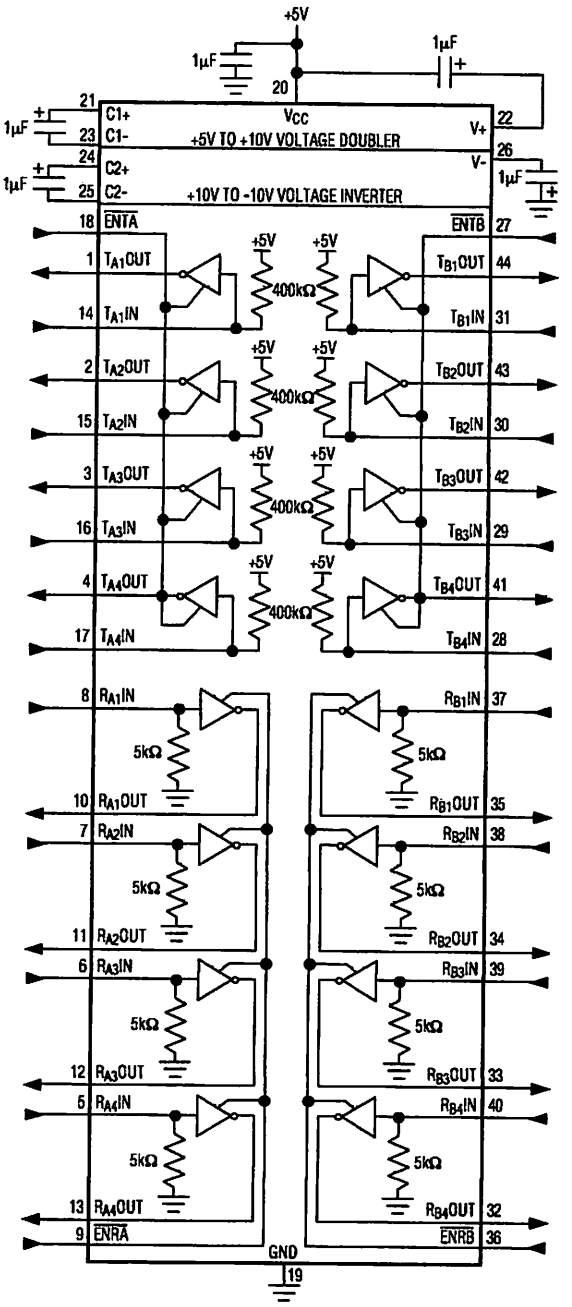
9 RECEIVERS
- 4 A-SIDE RECEIVERS
- 5 B-SIDE RECEIVERS (R_{B5} ALWAYS ACTIVE)

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





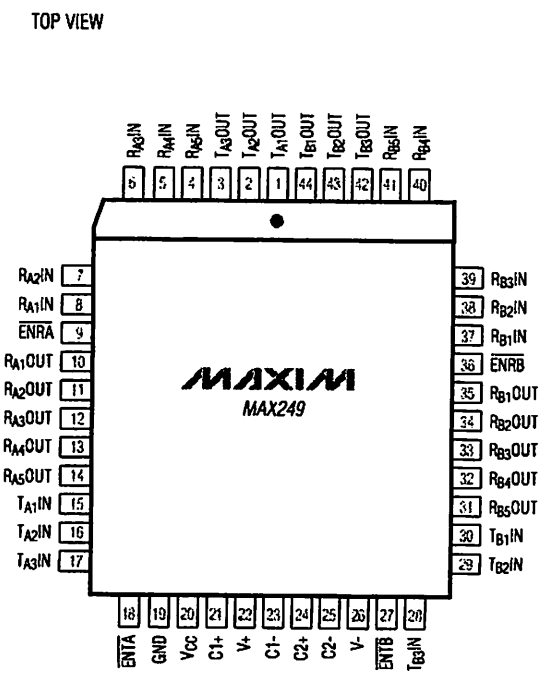
MAX248 FUNCTIONAL DESCRIPTION

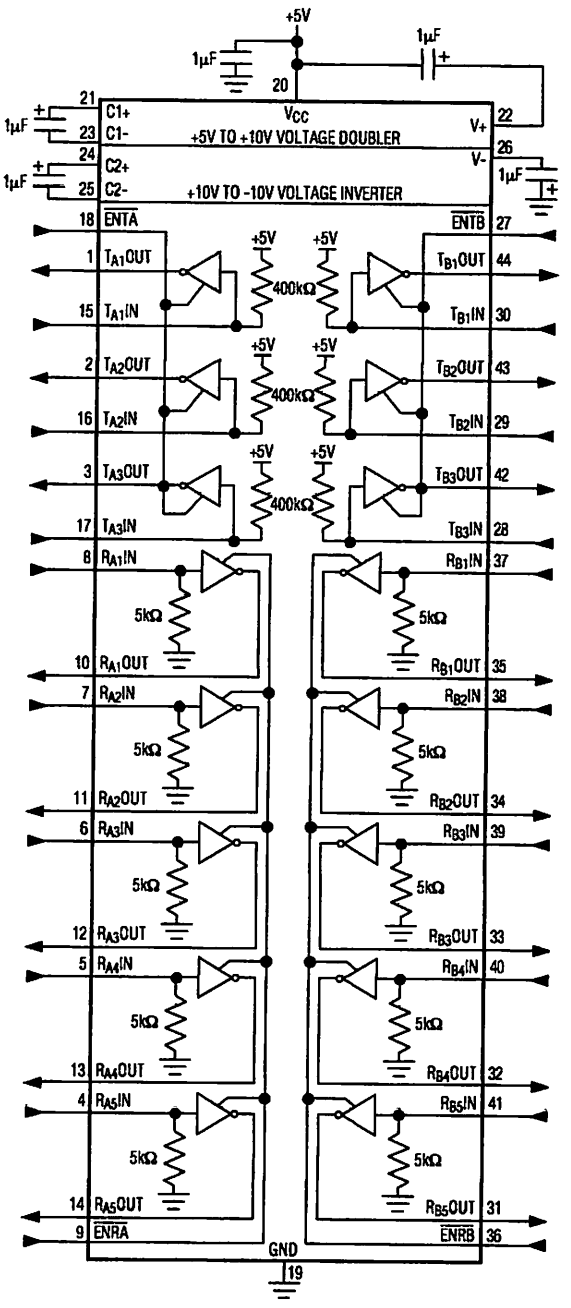
8 RECEIVERS
- 4 A-SIDE RECEIVERS
- 4 B-SIDE RECEIVERS

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

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

PART	TEMP RANGE	PIN-PACKAGE
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

PART	TEMP RANGE	PIN-PACKAGE
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.*

Package Information

For the latest package outline information, go to www.maxim-ic.com/packages.

TP560i
56Kbps Data/Fax/Voice Modem Chip Set
PCI Bus Interface

User's Manual
AT Command Set

Introduction

This is a high performance ITU-V.90 56Kbps modem; Computer can make connection with remote terminal, send a fax and be a telephone answer machine by this modem. In data mode, this modem provides ITU V.90, V.34, V.32bis, V.32, V.22bis, V.22, V.23, V.21 and bell212A/103 connection protocol speed from 300bps to 56Kbps. This modem also provides MNP class 5 and V.42bis compression method during data transmission. In Fax mode, this modem can send and receive Fax in 14.4kbps that is compatible to ITU-V.17 standard, Fax function is follow ITU Group 3 fax standard. More detail information, please refer to user's manual of communication software and AT command set.

Packing information

1. User's manual
2. Communication Software
3. RJ-11 telephone cable
4. 3.5 inch Disk (Win95/98 Driver & INF files)
5. Ear Phone and Mic Phone

Hardware Installation

Starting the Installation

1. Turn off your computer and unplug its power cable.
2. Unplug your computer before removing the top or you could hurt yourself and damage the equipment.
3. Remove the computer cover according to the directions in the computer manual.
4. Choose a PCI slot.
5. Remove the rear panel metal bracket that corresponds to the slot you plan to use.
6. Hold the modem over the expansion slot. Place the modem into the slot and push it down so that it is firmly seated.
7. Replace the computer cover.
8. Plug one end of the supplied phone cable into line jack on the modem metal bracket, the other into a modular phone wall jack.
9. You can connect a phone to your modem by plugging the phone modular plug into the modem's phone jack. With this arrangement you can use your telephone normally when the modem isn't being used.
11. Turn on the computer.

Windows 95/98 installation

If your computer uses Windows 95, plug and play? Both Windows 95 and your computer support plug & Play. Plug and Play lets you plug in a device such as your modem and then use it immediately.

To finish your Windows 95 installation, follow these steps

1. Turn on your computer if it not already on. If the New Hardware Found dialog box appears, you need to install a modem driver file, as explained in steps 2. If the New Hardware Found box does NOT appear, Windows 95 already has a driver for your Modem. Skip ahead to step 6.
2. Insert the Disk that is provided by hardware manufacture into floppy drive A (or B).
3. Chose Driver source is in floppy drive A (or B).
4. Click on OK. Windows 95 automatically copies the modem driver to your hard disk and uses it.
5. Eject the disk and store it in a safe place.
6. Your modem is now installed. Now install the fax and data software that came with your modem, as explained in your modem manual.

AT Commands Reference

AT commands are issued to DTE to control the modem's operation. AT commands can only be entered while the modem is in command mode. Except for the A/ command and the +++ escape command, all commands must be prefixed with the attention code AT. For instance, the "A" command (below) would be entered as: "ATA<CR>". Without the AT prefix, the command line cannot be executed. For example: The format for entering AT commands is "ATXn" where X is the AT command, and n is the specific value for that command.

More than one command can be placed on a single line and (if desired) separated with spaces for readability. Once the carriage return (Enter) key is pressed, the command line is executed. A line with no carriage return is ignored. Any command issued is acknowledged with a response in either text or numeric values known as result codes. In the following, all commands, command-values and result code recognized by the modem are shown; any entries other than those shown cause the ERROR result code. The modem accepts either upper or lower case characters in the command line and ignores any spaces within or between commands. Typing errors can be corrected with the Backspace key. Exceptions are noted in the description of specific commands.

+++AT Escape Sequence

The escape sequence allows the modem to exit data mode and enter online command mode. While in online command mode, you may communicate directly to your modem using AT commands. You may return to data mode by issuing the ATO command.

Section 1. Data Mode Command

A/ Executes Last Command

When modem receives this command, modem executes the last command string entered. Do not conclude it by pressing "Enter".

A Answer Command

This command instructs the modem to go off-hook and answer an incoming call.

Bn Communication Standard Setting

This command determines ITU-T vs. Bell standard.

B0 Selects ITU-T V.22 mode when the modem is at 1200 bits/s.

B1 Selects Bell 212A when the modem is at 1200 bits/s (default).

Result Codes:

OK : n = 0, 1

ERROR : Otherwise

Dn Dial
This command instructs the modem to begin the dialing sequence. A dial string can be up to 40 characters long. Any digit or symbol (0-9, *, #, A, B, C, D) could be dialed as touch-tone digits. The following may be used as dial string modifiers:

P Pulse dialing.
T Touch tone dialing (default).
W Wait for second dial tone.
@ Wait for quiet answer. Wait for five seconds of silence after dialing the number.
! Hook flash.
, Pause during dialing.
; Return to command mode.
^ Enable data calling tone transmission.
S = n Dial a telephone number previously stored using the &Zn = x command (see the &Zn = x command for further information). The range of n is 0,1,2.

En Echo Command
ATEn command controls whether or not the characters entered from your computer keyboard are echoed back to your monitor while the modem is in command mode.
E0 Disables echo to the computer.
E1* Enables echo to the computer (default).
Result Codes:
OK: n = 0, 1
ERROR: Otherwise

Hn Hook Control
This command instructs the modem to go on-hook to disconnect a call, or off-hook to make the phone line busy.
H0 Modem goes on-hook (default).
H1 Modem goes off-hook.
Result Codes:
OK: n = 0, 1
ERROR: Otherwise

In Request Identification Information
This command displays specific product information about the modem.
I0 Returns product ID code.
I1 Display Firmware version and checksum on the DTE.
I2 Customer Used.
I3 Returns fix ID information for application software identification. "TP560 Data/Fax/Voice 56K Modem"
I4 Returns firmware version for data pump.
I5 Returns country code.
I6 Blacklist times
I9 Return Software Speakerphone or Hardware speakerphone
I10 Checksum

Result Codes:
n = 0, 1, 2, 3, 4, 5,6,10
ERROR: Otherwise

Ln Monitor Speaker Volume
This command sets speaker volume to low, medium, or high.
L0 Select low volume.
L1 Select low volume.
L2* Select medium volume.
L3 Select high volume.
Result Codes:
OK n = 0, 1, 2, 3
ERROR Otherwise

Mn Monitor Speaker Mode
This command turns the speaker on or off.
M0 The speaker is off.
M1* The speaker is on until the modem detects the carrier signal.
M2 The speaker is always on when modem is off-hook.
M3 The speaker is on until the carrier is detected, except while dialing.
Result Codes:
OK n = 0, 1, 2, 3
ERROR Otherwise

Nn Modulation Selection
This command controls whether or not the local modem performs a negotiated handshake at connection time with the remote modem when the communication speed of the two modems is different.
N0 When originating or answering, this is for handshake only at the communication standard specified by AT*In.
N1* During handshake, Highest speed is specified by AT*In. Depend on line quality fallback to a lower speed may occur.
Result Codes:
OK n = 0, 1
ERROR Otherwise

On Return Online to Data Mode
O0* Exit online command mode and return to data mode (see AT Escape Sequence, +++AT).
O1 This command issues a retrain before returning to online data mode.
O2 This command issues a rate re-negotiation before returning to online data mode.
Result Codes:
OK n = 0, 1, 2
ERROR Otherwise

P Select Pulse Dialing
This command configures the modem for pulse (non-touch-tone) dialing.

Result Codes: OK

Qn Result Code Control
Result codes are informational messages sent from the modem and displayed on your monitor.
Q0* Enables modem to send result codes to the DTE.
Q1 Disables modem from sending result codes.
Result Codes:
OK n = 0, 1
ERROR Otherwise

T Select Tone Dialing
This command instructs the modem to send DTMF tones while dialing.

Result Codes: OK

Vn DCE Response Format
This command controls whether result codes are displayed as words or their numeric equivalents.
V0 Displays result codes as numeric.
V1* Displays result codes as text.
Result Codes:
OK n = 0, 1
ERROR Otherwise

Wn Result Code Option

W0 Display "CONNECT DCE speed" without V42/MNP extended.
W1 Display "CONNECT DTE speed" without V42/MNP extended.
W2* Display "CONNECT DCE speed" with V42/MNP extended.
W3 Display "CONNECT DTE speed" with V42/MNP extended.

Result Codes:
OK n = 0, 1, 2
ERROR Otherwise

Xn Result Code Selection and Call Progress detection
This command enables tone detection options during dialing procedure. As these functions are chosen, the modem chip set result codes are also affected. Therefore, this command is frequently used to control the modem chip set responses and dial tone detection.

X0/X Disables monitoring of busy tones unless forced otherwise by country requirements; send only OK, CONNECT, RING, NO CARRIER, ERROR, and NO ANSWER result codes. Blind dialing is enabled/disabled by country parameters. If busy tone detection is enforced and busy tone is detected, NO CARRIER will be reported. If dial tone detection is enforced or selected and dial tone is not detected, NO CARRIER will be reported instead of NO DIAL TONE. (Default)

X1 Disables monitoring of busy tones unless forced otherwise by country requirements; send only OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER, and CONNECT XXXX(XXXX=rate). Blind dialing is enabled/disabled by country parameters. If busy tone detection is enforced and busy tone is detected, NO CARRIER will be reported instead of BUSY. If dial tone detection is enforced or selected and dial tone is not elected, NO CARRIER will be reported instead of NO DIAL TONE.

X2 Disables monitoring of busy tones unless forced otherwise by country requirements; send only OK, CONNECT, RING, NO CARRIER, ERROR, NO DIAL TONE, NO ANSWER, and CONNECT XXXX. If busy tone detection is enforced and busy tone is detected, NO CARRIER will be reported instead of BUSY.

X3 Enables monitoring of busy tones; send only OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER, and CONNECT XXXX. Blind dialing is enabled/disabled by country parameters. If dial tone detection is enforced and dial tone is not detected, NO CARRIER will be reported.

X4 Enables monitoring of busy tones; send all messages.

Result Codes:
OK n = 0, 1, 2, 3, 4
ERROR Otherwise

Zn Recall Stored Profile
This command instructs the modem chip set to go on-hook and restore the profile saved by the last &W command.
Z0 Reset modem and retrieve active configuration profit from stored profit 0.
Z1 Reset modem and retrieve active configuration profit from stored profit 1.
Result Codes:
OK n = 0, 1
ERROR Otherwise

&Cn Data Carrier Detect (DCD) Control
Data carrier detect is a signal from the modem to your computer indicating that the carrier signal is being received from a remote modem. DCD normally turns off when the modem no longer detects the carrier signal.
&C0 The state of the carrier from the remote modem is ignored. DCD circuit is always on.
&C1* DCD turns on when the remote modem's carrier signal is detected, and off when the carrier signal is not detected.
Result Codes:
OK n = 0, 1
ERROR Otherwise

&Dn DTR Control (Data Terminal Ready)
This command interprets how the modem responds to the state of the DTR signal and changes to the DTR signal.
&D0 Ignore.
&D1 If the DTR signal is not detected while in online data mode, the modem enters command mode, issues OK result code, and remains connected.
&D2* If the DTR signal is not detected while in online data mode, the modem disconnects.
&D3 Monitor DTR signal when an on-to-off transition occurs, the modem performs a soft reset as if the ATZ command was received.
Result Codes:
OK n = 0, 1, 2, 3
ERROR Otherwise

&Fn Load Factory Settings
This command loads the configuration stored and programmed at the factory. This operation replaces all of the command options and the S-register settings in the active configuration with factory values.

&Gn V.22bis Guard Tone Control
This command determines guard tone frequency and is only used in V.22 and V.22bis mode. This option is not used in North America and is for international use only.
&G0* Guard tone disabled.
&G1 Sets guard tone to 550 Hz.
&G2 Sets guard tone to 1800 Hz.
Result Codes:
OK n = 0, 1, 2
ERROR Otherwise

&Kn Local Flow Control Selection
&K0 Disable flow control.
&K1 Reserved.
&K2 Reserved.
&K3 Enable RTS/CTS flow control.
&K4 Enable XON/XOFF flow control.
&K5 Enable Transparent XON/XOFF flow control.

Result Codes:
OK n = 0,1,2,3,4,5
ERROR Otherwise

&Pn Pulse Dial Make-to-Break Ratio Selection
&P0* 39/61 make-to-break ratio (10 pps).
&P1 33/67 make-to-break ratio (10 pps).
&P2 39/61 make-to-break ratio (20 pps)
&P3 33/67 make-to-break ratio (20 pps)
Result Codes:
OK n = 0,1,2,3
ERROR Otherwise

&Rn Clear To Send Signal Select
&R0 Modem turns on the Clear To Send signal when it detects the Request To Send (RTS) signal.
&R1* Modem turns on Clear To Send signal.
Result Codes:
OK n = 0, 1
ERROR Otherwise

&Sn Data Set Ready (DSR) Option
This command selects DSR action.
&S0* DSR always ON.
&S1 DSR comes on when establishing a connection and goes off when the connection ends.
Result Codes:
OK n = 0, 1
ERROR Otherwise

&Un Protocol Selection
&U0 Select V.PCM
&U1 Select V.34Bis/V.34
&U2 Select V.32Bis/V.22Bis
&U3 Select Bell 103 300bps
&U4 Select V21 300bps
&U5 Select V23

Result Codes:
OK n = 0,1,1,2,3,4,5
ERROR Otherwise

&V View Active Configuration and Stored Profile
This command is used to display the active profiles.

Result Codes: OK

&Wn Store Current Configuration
This command stores certain command options and S-register values into the modem nonvolatile memory. The ATZ command or a power up reset of the modem restores this profile.
&W0* Stores active configuration profile in configuration profit 0.
&W1 Stores active configuration profile in configuration profit 1.
Result Codes:
OK n = 0,1
ERROR Otherwise

&Yn Select Stored Profile for Power On or Reset
This command does not change the behavior of the modem but is included for compatibility with applications that issue the &Y0 command
&Y0* Select stored profile 0.
&Y1 Select stored profile 1.
Result Codes:
OK n = 0,1
ERROR Otherwise

&Zn = x Store Telephone Number
This command is used to store up to three dialing strings in the modem nonvolatile memory for later dialing. The format for the command is &Zn = stored number where n is the location 0? to which the number should be written. The dial string may contain up to 31 characters. The ATDS = n command dials using the string stored in location n.
Result Codes:
OK n = 0, 1, 2
ERROR Otherwise

***In Connect Speed Selection**
*I0 Selects connect speed 1200bps
*I1 Selects connect speed 2400bps
*I2 Selects connect speed 4800bps
*I3 selects connect speed 7200bps
*I4 Selects connect speed 9600bps
*I5 Selects connect speed 12000bps
*I6 Selects connect speed 14400bps
*I7 Selects connect speed 16800bps
*I8 Selects connect speed 19200bps
*I9 Selects connect speed 21600bps
*I10 Selects connect speed 24000bps
*I11 Selects connect speed 26400bps
*I12 Selects connect speed 28800bps
*I13 Selects connect speed 31200bps
I14 Selects connect speed 33600bps
select V.pcm
*I1 Selects connect speed 28000bps
*I2 Selects connect speed 29333bps
*I3 selects connect speed 30666bps
*I4 Selects connect speed 32000bps
*I5 Selects connect speed 33333bps
*I6 Selects connect speed 34666bps
*I7 Selects connect speed 36000bps
*I8 Selects connect speed 37333bps
*I9 Selects connect speed 38666bps
*I10 Selects connect speed 40000bps
*I11 Selects connect speed 41333bps
*I12 Selects connect speed 42666bps
*I13 Selects connect speed 44000bps
*I14 Selects connect speed 45333bps
*I15 Selects connect speed 46666bps
*I16 Selects connect speed 48000bps
*I17 Selects connect speed 49333bps
*I18 Selects connect speed 50666bps
*I19 Selects connect speed 52000ps
*I20 Selects connect speed 53333bps
*I21 Selects connect speed 54666bps
I22 Selects connect speed 56000bps

Result Codes:
OK
ERROR Otherwise

***Gn User Abort Selection**
G0 Enables key abort feature
*G1 Disables key abort feature

Result Codes:
OK n = 0,1
ERROR Otherwise

***Hn Auto Retrain Selection**
*H0 Disables auto retrain
H1 Enables auto retrain

Result Codes:
OK n = 0,1
ERROR Otherwise

***NCnn Country select**
This command provides customers chose the country for phone line regulation.

*NC40 Australia
*NC1 Austria
*NC2 Belgium
*NC3 Denmark
*NC4 Finland
*NC5 France
*NC6 Germany
*NC17 Greece
*NC10 Netherlands
*NC28 Iceland
*NC7 Ireland
*NC8 Italy
*NC29 Liechtenstein
*NC9 Luxembourg
*NC43 Japan
*NC26 Namibia
*NC11 Norway
*NC12 Portugal
*NC27 South Africa
*NC13 Spain
*NC14 Sweden
*NC15 Switzerland
*NC16 UK
*NC19 Czech Republic
*NC24 Poland
*NC25 Russia
*NC22 United States
*NC36 CTR 21 600 ohm
*NC37 CTR 21 Complex Impedance

Result Codes:
OK
ERROR Otherwise

***Tn Trellis Coding Selection**
*T0 Disables Trellis coding
T1 Enables Trellis coding

Result Codes:
OK n = 0,1
ERROR Otherwise

***On Transmission Level Selection**
O0 Selects output level -11dBm
*O1 Selects output level -12dBm
*O2 Selects output level -13dBm
*O3 Selects output level -14dBm
*O4 Selects output level -15dBm
*O5 Selects output level -16dBm
*O6 Selects output level -17dBm
*O7 Selects output level -18dBm
*O8 Selects output level -19dBm
*O9 Selects output level -20dBm
*O10 Selects output level -21dBm
*O11 Selects output level -22dBm
*O12 Selects output level -23dBm
*O13 Selects output level -24dBm
*O14 Selects output level -25dBm
*O15 Selects output level -26dBm

Result Codes:
OK n = 0 - 15
ERROR Otherwise

\Nn Error Control Mode Selection
This command determines the type of error control used by the modem when sending or receiving data.
\N0 Normal mode.
\N1 Direct mode.
\N2 MNP or disconnect mode. The modem attempts to connect using MNP 2-4 error control procedures. If this fails, the modem disconnects. This is also known as MNP reliable mode.
\N3 MNP, or buffer.
The modem attempts to connect using MNP 2-4 error control procedures. If this fails, the modem will connect in Normal modem
\N4 V.42 without ODP & ADP phase detection, MNP or buffer.
\N5* V.42 with ODP & ADP phase detection, MNP, or buffer (default).
The modem attempts to connect in V.42 error control mode. If this fails, the modem attempts to connect in MNP mode. If this fails, the modem connects in buffer mode and continues operation. This is also known as V.42/ MNP auto reliable mode.

\N6 V.42 without ODP & ADP phase detection or disconnect. The modem attempts to connect in V.42 error control mode. If this fails, the call will be disconnected.
\N7 V.42 with ODP & ADP phase detection or disconnect.

Result Codes:
OK n = 0, 1, 2, 3, 4, 5, 6,7
ERROR Otherwise

\Vn Protocol Result Code
\V0* Disable protocol result code appended to DCE speed.
\V1 Enable protocol result code appended to DCE speed.
Result Codes:
OK n = 0, 1
ERROR Otherwise

%Cn Data Compression Control
This command determines the operation of V.42bis and MNP class 5 data compression. Online changes do not take effect until a disconnection occurs first.
%C0 V.42bis/ MNP 5 disabled. No data compression.
%C1* V.42bis/ MNP 5 enabled. Data compression enabled (default).
Result Codes:
OK n = 0, 1
ERROR Otherwise

%Dn Blacklisting Controll
%D0 Disable blacklisting.
%D1* Enable blacklisting.
Result Codes:
OK n = 0, 1
ERROR Otherwise

Section 2. Result Code

Long Form	Short Form	n value in ATXn 0	1	2	3	4
OK	0	x	x	x	x	x
CONNECT	1	x	x	x	x	x
RING	2	x	x	x	x	x
NO CARRIER	3	x	x	x	x	x
ERROR	4	x	x	x	x	x
CONNECT 1200	5		x	x	x	x
NO DIALTONE	6			x		x
BUSY	7				x	x
NO ANSWER	8				x	x
CONNECT 300	9		x	x	x	x
CONNECT 2400	10		x	x	x	x
CONNECT 4800	11		x	x	x	x
CONNECT 9600	12		x	x	x	x
CONNECT 7200	13		x	x	x	x
CONNECT 12000	14		x	x	x	x
CONNECT 14400	20		x	x	x	x
CONNECT 19200	21		x	x	x	x
CONNECT 38400	22		x	x	x	x
CONNECT 57600	23		x	x	x	x
CONNECT 115200	24		x	x	x	x
CONNECT 1200TX/75RX	25		x	x	x	x
CONNECT 75TX/1200RX	26		x	x	x	x
CONNECT BELL 300	28		x	x	x	x
CONNECT V21	29		x	x	x	x
(Display DCE speeds by ATW0 and ATW2)						
CONNECT 16800	15		x	x	x	x
CONNECT 19200	16		x	x	x	x
CONNECT 21600	17		x	x	x	x
CONNECT 24000	21		x	x	x	x
CONNECT 26400	30		x	x	x	x
CONNECT 28800	31		x	x	x	x
CONNECT 31200	32		x	x	x	x
CONNECT 33600	33		x	x	x	x
(V.90 mode)						
CONNECT 28000	34		x	x	x	x
CONNECT 29333	35		x	x	x	x
CONNECT 30666	36		x	x	x	x
CONNECT 32000	37		x	x	x	x
CONNECT 33333	38		x	x	x	x
CONNECT 34666	39		x	x	x	x
CONNECT 36000	40		x	x	x	x
CONNECT 37333	41		x	x	x	x
CONNECT 38666	42		x	x	x	x
CONNECT 40000	43		x	x	x	x
CONNECT 41333	44		x	x	x	x
CONNECT 42666	45		x	x	x	x
CONNECT 44000	46		x	x	x	x
CONNECT 45333	47		x	x	x	x
CONNECT 46666	48		x	x	x	x
CONNECT 48000	49		x	x	x	x
CONNECT 49333	50		x	x	x	x
CONNECT 50666	51		x	x	x	x
CONNECT 52000	52		x	x	x	x
CONNECT 53333	53		x	x	x	x
CONNECT 54666	54		x	x	x	x
CONNECT 56000	55		x	x	x	x

Section 3. S Registers Definitions

S registers generally affect how the AT commands perform. Contents of the registers can be displayed or modified when the modem is in command mode.
To display the value of an S register: TYPE: ATSn?, where n is the register number.
PRESS: Enter To modify the value of an S register:
TYPE: ATSn = r where n is the register number, and r is the new register value.
PRESS: Enter

S0 Auto Answer Ring Number
This register determines the number of rings the modem will count before automatically answering a call. 0 (zero) is that asks modem don't do automatically answer at all. When disabled, the modem can only answer with an ATA command.
Range: 0-255
Default: 0
Units: rings

S1 Ring Counter
This register, ring counter, is read only. The value of S1 is incremented with each ring. If no rings occur over a 6 second interval, this register is cleared.
Range: 0-255
Default: 0
Units: rings

S2 AT Escape Character (User Defined)
This register determines the ASCII valued used for an escape sequence. The default is the "+" character. The escape sequence allows the modem to exit data mode and enter command mode. Values greater than 127 disable the escape sequence.
Range: 0-255
Default: 43
Units: ASCII

S3 Command Line Termination Character (User Defined)
This register determines the ASCII values as the carriage return character. This character is used to end command lines and result codes.
Range: 0-127, ASCII decimal
Default: 13 (carriage return)
Units: ASCII

S4 Response Formatting Character (User Defined)
This register determines the ASCII value used as the line feed character. The modem uses a line feed character in command mode when it responds to the computer.
Range: 0-127, ASCII decimal
Default: 10 (line feed)
Units: ASCII

S5 Command Line Editing Character (User Defined)
This register sets the character recognized as a backspace and pertains to asynchronous only.
Range: 0-127, ASCII decimal
Default: 8 (backspace)
Units: ASCII

S6 Wait Before Dialing
This register sets the length of time, in seconds, that the modem must wait (pause) after going off-hook before dialing the first digit. The modem always pauses for a minimum of two seconds, even if the value of S6 is less than two seconds. Waiting for dial tone call progress features (W dial modifier in the dial string) will override the value in register S6. This operation, however, may be affected by some ATX options according to country restrictions.
Range: 3-7
Default: 6
Units: seconds

S7 Connection Completion Timeout
This register sets the time, in seconds, that the modem must wait before hanging up because carrier is not detected. The timer is started when the modem finishes dialing (originate), or goes off-hook (answer). In originate mode, the timer is reset upon detection of an answer tone if allowed by country restriction. The timer also specifies the wait for silence time for the @ dial modifier in seconds. S7 is not associated with the W dial modifier.
Range: 1-255
Default: 60
Units: seconds

S8 Comma Dial Modifier Time
This register sets the time, in seconds, that the modem must pause when it encounters a comma (,) in the dial command string.
Range: 1-255
Default: 2
Units: seconds

S9 Carrier Detect Response Time
Register S9 sets the time the remote modem's carrier signal must be present for the local modem to recognize it. This feature ensures that your modem does not mistake a busy signal, ring, or voice for the carrier signal. The value of this register is in tenths of a second. The default value is 600 ms, although you can change it from 1 to 255. For example, entering a value 13 means that the remote modem's carrier signal must be present for 1.3 seconds for the local modem to recognize it. Setting this value higher increases the chances the modem will not mistakenly identify the carrier signal.

S10 Automatic Disconnect Delay after Carrier Loss
This register sets the length of time, in tenths of a second, that the modem waits before hanging up after a loss of carrier. This allows for a temporary carrier loss without causing the local modem to disconnect. The actual interval the modem waits before disconnecting is the value in register S10.
Range: 1-255
Default: 14
Units: 0.1 seconds

S11 DTMF Dialing Speed
This register determines the dialing speed which is prefixed for each country.
Range: 50-255
Default: 95
Units: 0.001 seconds

S12 Escape Guard Time
This register sets the value (in 20 ms increments) for the required pause after the escape sequence (default 1 s).
Range: 0-255
Default: 50
Units: 0.02 seconds

S13 Pulse Dialing Control
This register determines pulse dialing is disabled or enabled.
ATS13=0 Disable Pulse Dialing
ATS13=1 Enable Pulse Dialing

Range: 0-1

S14 Bit-mapped Register
Bit-mapped register S14 lets you control modem echo, responses, dialing method, and the original or answer mode.

Bit	Function
0=0	Do not echo command (E0)
1	Echo command (E1) – default
1	Reserved
2	Mapping to ATQn
3=0	Numeric responses (V0)
1	Verbose (Verbal) Responses (V1) – default
4,5=	Mapping to AT&P
6=0	Use touch-tone dialing method (T) – default
1	Use pulse dialing method (P)
7=0	Answer incoming calls (A)
1	Originate calls (D) – default

S21 Bit-mapped Register
Bit-mapped register S21 lets you control certain data communication control signals.

Bit	Function
0,1=	x Undefined
2=	Mapping to AT&R
3,4=0	Modem ignores the DTR signal (&D0)
1	Modem enters command mode after ON-to-OFF DTR transition (&D1)
2	Modem hangs up after ON-to-OFF DTR transition (&D2) – default
3	Modem resets after ON-to-OFF DTR transition (&D3)
5=0	CD signal always on (&C0)
1	CD signal on when a remote carrier signal is present (&C1) – default
6 =0	AT&S=0
1	AT&S=1

S22 Bit mapped Register
Bit-mapped register S22 lets you control internal speaker and the modem responses.

Bit	Function
0,1= 0	Speaker volume off (L0)
1	Low speaker volume (L1)
2	Medium speaker volume (L2) – default
3	Loud speaker volume (L3)
2,3= 0	Speaker off (M0)
1	Speaker on until carrier detected (M1) – default
2	Speaker always on (M2)
3	Speaker on until carrier detected but off during dialing (M3)
4,5,6=0	Hayes Smart modem compatibility (X0)
1	Include CONNECT XXX responses (X1)
2	Same as 4 plus dial tone detection (X2)
3	Same as 4 plus BUSY response and blind dialing (X3)
4	Same as 6 plus dial tone detection (X4) – default
7 =0	Undefined

S23 Bit-mapped Register

Bit	Function
0,1,2,3= 0	0-300bps DTE data rate
1	1200bps DTE data rate
2	2400bps DTE data rate
3	4800bps DTE data rate
4	9600bps DTE data rate
5	14400bps DTE data rate
6	19200bps DTE data rate
7	34800bps DTE data rate
8	57600bps DTE data rate
9	115200bps DTE data rate
4,5= 0	Use even parity
1	Use none parity
2	Use odd parity
3	Use mark parity
6,7= 0	Mapping to AT&G

S25 Data Terminal Ready Delay
When Modem is on-line, it will ignore a Data Terminal Ready signal lasting less than the value of this register. In this mode, the values for this register are 0 to 255 in hundredths of a second, and the default value is 0.05 seconds. If you will be entering synchronous mode after dialing asynchronously, this register determines how long the modem waits before looking for the Data Terminal Ready signal. This lets you detach the asynchronous terminal and connect a synchronous terminal while remaining in the Data Mode. The default value is 5.
Range: 0-255
Default: 5
Units: 0.01 seconds

S26 Request To Send-to-Clear To Send Delay
This register affects synchronous operation only and applies only when you are using the &R0 command. This register determines how long the modem waits to turn on the Clear To Send signal after a Request To Send OFF-to-ON transition in 10 ms increment. The default value is 1.
Range: 0-255
Default: 1
Units: 0.01 seconds

S29

Bit	Value Function
0-4	Mapping to AT*I
5-7	Mapping to AT&U

S30 Inactivity Timer
S30 specifies the length of time (in minutes) that the modem will wait before disconnecting when no data is sent or received. This function is only applicable to buffer mode.
Range: 0-255
Default: 0 (Disable)
Units: minutes

S45 Timer to Control Sleep Mode
This command displays the number of seconds of inactivity (no characters sent from the DTE, no RING) in the off-line command state before the modem places itself into standby mode. A value of zero prevents standby mode.
Note: If a number between 1 and 4 is entered for this register, it will set the value to 5, and the inactivity before standby will be 5 s. This is done for compatibility with previous products which allowed time-outs down to 1 s.
Range: 0, 5-255
Default: 10

S95

Bit	Value Function
0,1	Mapping to ATWn
2	Mapping to AT\V
3	Mapping to ATN
4-7	Mapping to AT*O

S96 Local Phone Status
This register tells the status of the Local Phone. It is read only.
0 = local phone on-hook
1 = local phone off-hook

S97

Bit	Value Function
0-2	Mapping to AT\Nn
7	Mapping to AT%Cn

S98

Bit	Value Function
0,1	Mapping to AT*H
2	Mapping to AT*G
3	Mapping to AT*T
4-6	Mapping to AT&K

S99 Energy detection threshold
This register can use in Call Progress Mode (CPM) mode, for dial tone detection threshold. This register determines the detection threshold for Call Progress (CPM) tones. The default value is 16.

Range : 0-255

Section 4. Class 1 FAX Commands

The modem chip set supports FAX commands conforming to EIA standard 578. These commands are given here with short descriptions; complete explanations are given in the standard, available from the Electronic Industry Association.

FAX Command Set Summary

+FCLASS?	Service class indication
+FCLASS = ?	Service class capabilities
+FCLASS = n	Service class selection (n=0,1,8)
+FTM = <m>	Transmit FAX data
+FRM = <m>	Receive FAX data
+FTH = <m>	Transmit HDLC data
+FRH = <m>	Receive HDLC data
+FTM = ?	Check transmit FAX modulation
+FRM = ?	Check receive FAX modulation
+FTH = ?	Check transmit HDLC data modulation
+FRH = ?	Check receive HDLC data modulation
+FMI = ?	Check manufacturer identification
+FMM = ?	Check product identification
+FMR = ?	Check version/revision information
+FLO = ?	Select Flow Control specified
+FPR = ?	Select Serial Port Rate

+FCLASS? Service Class Indication
This command causes the modem to display the current setting. The modem can operate either as a class 0 data modem or a class 1 FAX modem.
Responses: 0 :data mode
1 if in FAX class 1
8 if in voice mode

+FCLASS = ? Service Class Capabilities
This command causes the modem to display the classes it supports.
Typical responses:
+FCLASS = ? 0, 1, 8

+FCLASS = n Service Class Selection
This command sets the modem for class n operation, where n is either a 0,1 or 8.
Parameters: 0, 1, 8
Default: 0
Command options:
+FCLASS = 0 Select data mode.
+FCLASS = 1 Select facsimile class 1.
+FCLASS = 8 Select voice mode.

+FTM = <m> Transmit FAX Data
This command causes the modem to transmit data at the modulation specified by <m>. The following table shows the values you can enter for this command and the meaning of those values.

+FTM = 24	V.27ter 2400
+FTM = 48	V.27ter 4800
+FTM = 72	V.29 7200
+FTM = 96	V.29 9600
+FTM = 73	V.17 7200
+FTM = 74	V.17 (short train) 7200
+FTM = 97	V.17 9600
+FTM = 98	V.17 (short train) 9600
+FTM = 121	V.17 12000
+FTM = 122	V.17 (short train) 12000

+FTM = 145 V.17 14400
+FTM = 146 V.17 (short train) 14400

+FRM = <m> Receive FAX Data
This command causes the modem to receive data at the modulation specified by <m>.

+FRM = 24 V.27ter 2400
+FRM = 48 V.27ter 4800
+FRM = 72 V.29 7200
+FRM = 96 V.29 9600
+FRM = 73 V.17 7200
+FRM = 74 V.17 (short train) 7200
+FRM = 97 V.17 9600
+FRM = 98 V.17 (short train) 9600
+FRM = 121 V.17 12000
+FRM = 122 V.17 (short train) 12000
+FRM = 145 V.17 14400
+FRM = 146 V.17 (short train) 14400

+FTH = <m> Transmit HDLC Data
This command causes the modem to transmit data framed in the HDLC protocol at the modulation specified by <m>.

+FTH = 3 V.21 Channel2 300

+FRH = <m> Receive HDLC Data
This command causes the modem to receive data framed in the HDLC protocol at the modulation specified by <m>.

+FRH = 3 V.21 Channel2 300

+FLO = <V> Flow Control Select

+FLO=0 Xon/Xoff and RTS/CTS flow control Turned off
+FLO=1 Xon/Xoff flow control
+FLO=2* RTS/CTS flow control

+FPR = <V> Serial Port Rate
Please refer **AT+VPR** command

Section 5. Voice Commands

+VIP Initial voice parameter
+VCID=<pmode> Caller ID
pmode=0:Disable Caller ID
pmode=1: Enable ,formatted caller report
pmode=2:Enable ,unformatted caller report
+VCID? Report current parameter
+VCID=? Queries the range that DCE support
+VDR Enable the distinctive ring feature
+VDR? Report current parameter
+VDR=? Queries the range that DCE support
+VGT Setup playback gain
+VGT? Report current parameter
+VGT=? Queries the range that DCE support
+VGR Setup record gain
+VGR? Report current parameter
+VGR=? Queries the range that DCE support
+FMI? Report manufacturer ID
+FMM? Report product information
+FMR? Report product revision
+VIT Enable inactive timer
+VIT? Report current parameter
+VIT=? Queries the range that DCE support
+VTD Set the default duration for DTMF/tone generation in 10ms increments
+VTD? Report current parameter
+VTD=? Queries the range that DCE support
+VSD=<sds,sdi>
sds=128 Normal level of sensitivity(-40dbm)
sds>= 128 More aggressive ;ex. sds=129 is –39dbm
sds<= 128 Less aggressive :ex sds=127 is –40dbm
sdi :Specify the amount of time the modem will report silence to DTE
+VSD? Report current parameter
+VSD=? Queries the range that DCE support
+VTS DTMF and tone generation
+VTS=? Queries the range that DCE support
ex1: AT+VTS=1 play DTMF 1 with a duration by the +VTD command
ex2: AT+VTS=2 play DTMF 2 with a duration by the +VTD command
ex3: AT+VTS=[1000,1300,50] play tone pair 1000Hz and 1300hz with a duration of 50ms
ex4: AT+VTS={*,6} play DTMF * with a duration of 60ms
ex5: AT+VTS=9 play DTMF 9 with a duration by the +VTD command
+VTX Enter voice playback mode
+VRX Enter voice record mode
+VSM=<cml>,<vsr>
cml=128 , select linear 8bit PCM
vsr =xx, Sample rate is always 8000
+VSM? Report current parameter
+VSM=? Queries the range that DCE support
+VPR=<rate> DTE/DCE rate will be equal 2400*rate , but auto-baud if rate=0
+VPR? Report current parameter
+VPR=? Queries the range that DCE support
+VLS=<pmode>
pmode=0: DCE on-hook, local phone connected to Telco.
pmode=1: DCE off-hook, DCE connected to Telco
pmode=2: DCE off-hook, local phone connected to DCE
pmode=3: DCE off-hook, local phone connected to Telco, DCE to local phone
pmode=4: Speaker connected to DCE,DCE on-hook(playback message)
pmode=5: Speaker connected to DEC, DCE off-hook(call screening)
pmode=6: Microphone connected to DCE, DCE on-hook (Record greeting)
pmode=7: Microphone and speaker connected , DCE off-hook (speakerphone)

Section 6. AT%TTn Commands

%tt0=0 erase %TT command

%tt2=A,B,C,D Tx v34 signal parameter
A: baudrate (parameter range:0-5)
B: data rate (parameter range:1-9 if a=0)
(parameter range:2-11 if a=1)
(parameter range:2-11 if a=2)
(parameter range:2-12 if a=3)
(parameter range:2-13 if a=4)
(parameter range:2-14 if a=5)

C: Min/Exp (parameter range:0-1)
D: Preemphasise (parameter range:0-9)

example:
at%tt2=5,14,1,8

%tt3=0 : DTMF '0'
%tt3=1 : DTMF '1'
%tt3=2 : DTMF '2'
%tt3=3 : DTMF '3'
%tt3=4 : DTMF '4'
%tt3=5 : DTMF '5'

Section 3. Voice Commands

Section 3. ATTACTIN Commands

%tt3=6 : DTMF '6'
%tt3=7 : DTMF '7'
%tt3=8 : DTMF '8'
%tt3=9 : DTMF '9'
%tt3=10 : DTMF 'A'
%tt3=11 : DTMF 'B'
%tt3=12 : DTMF 'C'
%tt3=13 : DTMF 'D'
%tt3=14 : DTMF '*'
%tt3=15 : DTMF '#'
%tt3=16 : V32 9600
%tt3=17 : V32 14400
%tt3=18 : 2100HZ ANSWER TONE
%tt3=19 : 1300HZ CALLING TONE
%tt3=20 : 1100HZ CALLING TONE
%tt3=21 :
%tt3=22 : Silence

%tt4=0 : V21 Org tx mark
%tt4=1 : V21 Org tx space
%tt4=2 : V21 Ans tx mark
%tt4=3 : V21 Ans tx space
%tt4=4 : V23 Org tx mark
%tt4=5 : V23 Org tx space
%tt4=6 : V23 Ans tx mark
%tt4=7 : V23 Ans tx space
%tt4=8 : V22 org
%tt4=9 : V22bis org
%tt4=10 : V22 ans (tx guard tone if &g1 or &g2)
%tt4=11 : V22bis ans (tx guard tone if &g1 or &g2)

%tt5=0 : V27ter 2400
%tt5=1 : V27ter 4800
%tt5=2 : V29 7200
%tt5=3 : V29 9600
%tt5=4 : V17 7200 short train
%tt5=5 : V17 7200 long train
%tt5=6 : V17 9600 short train
%tt5=7 : V17 9600 long train
%tt5=8 : V17 12000 short train
%tt5=9 : V17 12000 long train
%tt5=10 : V17 14400 short train
%tt5=11 : V17 14400 long train