

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



**PERENCANAAN DAN PEMBUATAN ALAT ISI ULANG VOUCHER  
ELEKTRIK DENGAN HANDPHONE YANG DIINTERFACEKAN  
DENGAN MIKROKONTROLLER AT89C4051**

**SKRIPSI**

**Disusun Oleh:  
INDRA SANJAYA PRIYANTO  
NIM. 01.17.110**

**MARET 2006**

---

LEMBAR PERSETUJUAN



**PERENCANAAN DAN PEMBUATAN ALAT ISI ULANG VOUCHER  
ELEKTRIK DENGAN HANDPHONE YANG DIINTERFACEKAN  
DENGAN MIKROKONTROLLER AT89C4051**

Disusun dan Diajukan untuk Melengkapi dan Memenuhi Syarat Mencapai Gelar  
Sarjana Teknik

Disusun Oleh :

**NAMA : INDRA SANJAYA PRIYANTO**

**NIM : 01.17.110**



Mengetahui,

Ketua Jurusan T. Elektro S-1

**(Dr. F. Yudi Limpraptono, MT)**

**NIP.P. 1039500274**

Diperiksa dan Disetujui,

Dosen Pembimbing

**(Joseph Dedy Irawan, ST, MT)**

**NIP. 132315178**



**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**


Nama : Indra Sanjaya Priyanto  
Nim : 01.17.110  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perencanaan Dan Pembuatan Alat Isi Ulang Voucher  
Elektrik Dengan Handphone Yang Diinterfacekan  
Dengan Mikrokontroller AT89C4051

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


Hari : Sabtu  
Tanggal : 18 Maret 2006  
Dengan Nilai : 74,5



Panitia Ujian Skripsi

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

Sekretaris

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

Anggota Penguji

Penguji I

  
(Ir. Eko Nurcahyo)

Penguji II

  
(Ir. M Luqman)

---



### LEMBAR BIMBINGAN SKRIPSI

Nama : Indra Sanjaya Priyanto  
NIM : 01.17.110  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perencanaan Dan Pembuatan Alat Isi Ulang  
Voucher Elektrik Dengan Handphone Yang  
Diinterfacekan Dengan Mikrokontroller  
AT89C4051  
Tanggal Pengajuan Skripsi : 30 Mei 2005  
Selesai Penulisan Skripsi : 16 Maret 2006  
Dosen Pembimbing : Joseph Dedy Irawan, ST, MT  
Telah Dievaluasi Dengan Nilai : 83 (A)

Mengetahui,

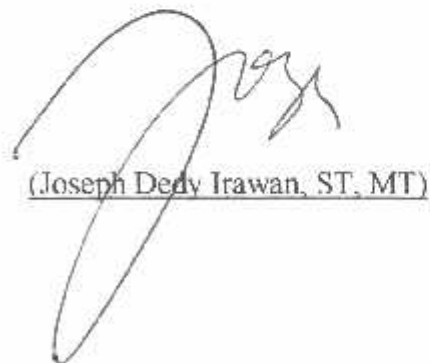
Ketua Jurusan T. Elektro S-1



(Ir. F. Yudi Limpraptono, MT)

Diperiksa Dan Disetujui,

Dosen Pembimbing



(Joseph Dedy Irawan, ST, MT)

---

## LEMBAR PERSEMBAHAN

Jalan menuju " **BAHAGIA DAN SUKSES** " tidak selalu lurus, ada tikungan bernama kegagalan, ada bundaran bernama " **kebingungan** ", tanjakan bernama " **teman** ", lampu merah bernama musuh, lampu kuning bernama " **Keluarga** ". Engkau akan mengalami ban kempes dan pecah, Itulah hidup. Tapi jika engkau membawa ban serep bernama " **Tekad** ", mesin bernama " **Mesin Ketekunan** ", asuransi bernama " **Iman** ", pengemudi bernama " **Yesus** ", sampailah di daerah yang disebut " **SUKSES DAN BAHAGIA** ".

Pertama kali saya mengucapkan terima kasih pada papa mamaku yang telah membiayai aku dan selalu mendoakan aku sehingga aku sekarang bisa lulus Dan memperoleh gelar **ST**.

Aku juga berterimakasih pada nenekku, adikku Randy, **AAN** yang telah menyemangati, mendoakan aku pada waktu aku akan menempuh ujian skripsi, sehingga berkat doa restu mereka aku dapat menyelesaikan skripsiku dan dapat lulus dalam ujian skripsi. Aku Sangat-sangat berterimakasih pada mereka.

Buat temen-temen kostku aku ucapkan selamat tinggal semuanya ya...Victor, Mr. Bond, **BN**, Bondet, Toha, dll yang tak bisa aku sebutkan, aku ucapin selamat tinggal ya....Semoga kalian baik-baik aja di sana ya...

Buat Victor cepetan lulus ya tak tunggu....sering-sering Nonton26 ya....

Buat semuanya yang tak bisa aku sebutkan satu-satu yang telah membantuku menyemangatiku maupun membantuku dalam penyusunan skripsi aku ucapin trimakasih yang sebesar-besarnya yaa...Moga tuhan membalas kebaikanmu....

Buat temen-temenku kampus yang belum nyusun skripsi cepetan nyusun yaa...Jangan lama-lama tinggal di **ITN** yaa....

Ok...Aku kira sampai disini dulu aku mengakhiri persembahanku ini....

I Love You All.....

INDRA SANJAYA P

## KATA PENGANTAR

Dengan mengucapkan puji dan syukur kehadiran Tuhan Yang Maha Esa yang telah memberikan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Perencanaan Dan Pembuatan Alat Isi Ulang Voucher Elektrik Dengan Handphone Yang Diinterfacekan Dengan Mikrokontroller AT89C4051”.

Skripsi ini dibuat sebagai salah satu persyaratan untuk memperoleh gelar Sarjana Teknik di Institut Teknologi Nasional Malang.

Dengan terselesaikannya skripsi ini, juga tidak terlepas dari kesabaran Dosen pembimbing untuk memberikan waktu asistensi dan konsultasi kepada penulis serta banyak teman-teman saya yang telah banyak membantu sehingga skripsi ini dapat diselesaikan.

Oleh karena itu penulis mengucapkan terimakasih kepada :

- Joseph Dedy Irawan, ST, MT. Selaku dosen pembimbing skripsi saya di Institut Teknologi Nasional ( ITN ) Malang.
- Dr. Ir. Abraham Lomi, MSEE. Selaku Rektor Institut Teknologi Nasional ( ITN ) Malang.
- Para Instruktur Lab.Elektronika yang membantu pelaksanaan skripsi saya.
- Teman-teman yang telah memberi dukungan dan doanya.

Penulis menyadari sepenuhnya bahwa skripsi ini masih terdapat kekurangan-kekurangan, baik dalam penyampaian materi maupun dalam penyusunannya. Oleh karena itu, penulis mengharapkan saran dan kritik yang membangun demi kesempurnaan dimasa yang akan datang. Semoga skripsi ini bermanfaat bagi semua pihak.

Malang, Maret 2006

Penulis

# DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	v
DAFTAR TABEL.....	vi
ABSTRAKSI.....	vii
BAB I    PENDAHULUAN.....	1
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	2
1.3    Tujuan Perencanaan.....	2
1.4    Batasan Masalah.....	2
1.5    Metodologi Penelitian.....	3
1.6    Sistematika Penulisan.....	4
BAB II    LANDASAN TEORI.....	5
2.1    Pendahuluan.....	5
2.2    Mikrokontroler AT89C4051.....	5
2.2.1    Pendahuluan.....	5
2.2.2    Perangkat Keras Mikrokontroler AT89C4051.....	6
2.2.2.1    Arsitektur AT89C4051.....	7
2.2.2.2    Konfigurasi Pin AT89C4051.....	8
2.2.3    Organisasi Memori.....	9
2.2.3.1    Program Memori Internal.....	9
2.2.3.2    Data Memori (RAM) Internal.....	10
2.2.3.3    SFR (Register Fungsi Khusus).....	11
2.2.3.4    PSW (Program Status Word).....	12
2.2.3.5    PCON (Power Control).....	12
2.2.3.6    Sistem Interrupt.....	13
2.2.3.7    Timer/Counter.....	14
2.2.3.8    Metode Pengalamatan.....	16
2.2.3.8.1    Pengalamatan Bit.....	16
2.2.3.8.2    Pengalamatan Tak Langsung.....	17
2.2.3.8.3    Pengalamatan Berindeks.....	17
2.2.3.8.4    Konstanta Immediate.....	17
2.3    Sistem Kerja SMS.....	17
2.3.1    ATCommand Untuk Komunikasi Dengan SMS Centre.....	18
2.3.2    Hyper Terminal.....	18
2.3.3    PDU Sebagai Bahasa SMS.....	19
2.3.3.1    PDU Untuk Kirim SMS Ke SMS Centre.....	19
2.3.3.2    PDU Untuk Terima SMS Dari SMS Centre.....	25
2.3.3.3    Membedah Kedelapan Header.....	26
2.4    LCD.....	27
2.5    Data Cable (Port Serial).....	28
2.6    Interface Handphone Dengan Mikrokontroler.....	29

2.7	Port Serial .....	30
2.8	RS MAX232 .....	32
2.9	Keypad .....	34
2.10	IC 74C922 (Keypad Encoder).....	35
BAB III	PERENCANAAN DAN PEMBUATAN ALAT.....	37
3.1	Blok Diagram .....	37
3.2	Prinsip Kerja Alat.....	38
3.3	Perancangan Perangkat Keras (Hardware).....	38
	3.3.1 Rangkaian Mikrokontroller AT89C4051 .....	38
	3.3.2 Rangkaian Reset Dan Oscilator.....	39
	3.3.3 Rangkaian Komunikasi Serial RS232 .....	41
	3.3.4 Rangkaian Penampil LCD.....	42
	3.3.5 Rangkaian Keypad Dan Encoder .....	43
3.4	Perancangan Perangkat Lunak .....	43
BAB IV	PENGUJIAN DAN ANALISIS.....	45
4.1	Tujuan.....	45
4.2	Pengujian Sistem Mikrokontroller .....	45
4.3	Pengujian Komunikasi Serial .....	47
4.4	Pengujian Rangkaian Keypad .....	50
4.5	Pengujian Sistem Secara Keseluruhan .....	52
BAB V	PENUTUP.....	55
5.1	Kesimpulan.....	56
5.2	Saran.....	56
	Daftar Pustaka .....	57



## DAFTAR GAMBAR

Gambar 2-1	Blok Diagram Mikrokontroller .....	6
Gambar 2-2	Konfigurasi Pin AT89C4051 .....	8
Gambar 2-3	Skema Mendefinisikan PSW .....	12
Gambar 2-4	Skema Mendefinisikan PCON .....	12
Gambar 2-5	Kegunaan Interrupt Enable Register .....	14
Gambar 2-6	Konfigurasi Dan Guna TMOD .....	15
Gambar 2-7	Timer/Counter Mode Control Register .....	15
Gambar 2-8	A Request SMSC .....	19
Gambar 2-9	LCD 2x16 Karakter .....	27
Gambar 2-10	Konektor HP Siemens C45 .....	30
Gambar 2-11	Port Serial .....	31
Gambar 2-12	IC MAX232 .....	32
Gambar 2-13	Level Tegangan RS232 .....	33
Gambar 2-14	Keypad 4x4 .....	34
Gambar 2-15	Aliran Arus Saat Tombol Tidak Ditekan .....	35
Gambar 2-16	IC 74C922 .....	36
Gambar 3-1	Diagram Blok Sistem .....	37
Gambar 3-2	Mikrokontroller AT89C4051 .....	39
Gambar 3-3	Rangkaian Reset .....	40
Gambar 3-4	Rangkaian Oscilator .....	41
Gambar 3-5	Rangkaian Komunikasi Serial RS232 .....	41
Gambar 3-6	Rangkaian Penampil LCD .....	42
Gambar 3-7	Rangkaian Keypad Dan Encoder .....	43
Gambar 3-8	Flowchart Software .....	44
Gambar 4-1	Diagram Blok Pengujian Mikrokontroller .....	46
Gambar 4-2	Rangkaian Pengujian Transfer Data .....	48
Gambar 4-3	Program Uji Komunikasi Data di Mikrokontroler .....	49
Gambar 4-4	Program Uji Komunikasi Data di Komputer .....	49
Gambar 4-5	Diagram blok pengujian pengkode papan tombol (keypad) .....	51
Gambar 4-6	Tampilan Kode Sms Untuk Isi Pulsa Mentari 10.000 .....	53
Gambar 4-7	Laporan Sukses Pengisian Pulsa .....	53
Gambar 4-8	Kode Sms Untuk Cek Saldo .....	54
Gambar 4-9	Laporan Dari Cek Saldo .....	54

## DAFTAR TABEL

Tabel 2-1. Pengaturan RS0-RS1 Untuk Select Register Bank.....	10
Tabel 2-2. Special Function Register .....	11
Tabel 2.3. Alamat Sumber Interupsi .....	13
Tabel 2.4. Alamat Sumber Interupsi .....	16
Tabel 2.5. Operator SMS Centre.....	21
Tabel 2.6. Operator Selluler .....	21
Tabel 2.7. Validasi SMS.....	22
Tabel 2-8. Skema 7 Bit.....	25
Tabel 2-9. Nama-nama Pin Pada LCD M1632 .....	28
Tabel 2-10. Konektor Siemens C45 .....	30
Tabel 2-11. Konektor Port Serial .....	31
Tabel 4.1 Hasil Pengujian Sistem Mikrokontroler.....	47
Tabel 4.2 Hasil Pengujian Transfer Data Serial.....	50
Tabel 4-3 Hasil pengujian pengkode papan tombol (keypad).....	51

**Perencanaan Dan Pembuatan Alat Isi Ulang Voucher Elektrik  
Dengan Handphone Yang Diinterfacekan Dengan  
Mikrokontroller AT89C4051**

(Indra Sanjaya Priyanto, Teknik Elektronika S-1, Dosen : Joseph Dedy  
Irawan, ST, MT)

**Abstraksi**

Sctiap orang menginginkan kecepatan dalam segala hal, termasuk dalam hal untuk melakukan isi ulang voucher elektrik. Biasanya apabila sebuah counter handphone akan melakukan isi ulang voucher elektrik bila ada pembeli, dia harus mengetikkan kode-kode untuk melakukan isi ulang tersebut lewat sms termasuk juga nomor HP pembeli voucher dan nominal voucher yang dibeli. Hal ini kadang-kadang dapat menimbulkan kesalahan pada penulisan nomor handphone yang akan diisi dan nominal voucher yang akan dibeli. Untuk menghindari hal tersebut kami merancang alat isi ulang voucher elektrik dengan handphone yang diinterfacekan pada mikrokontroller AT89C4051. Alat ini sangat berguna untuk proses isi ulang voucher elektrik dan tidak menimbulkan kesalahan di pihak penjual karena pembeli langsung memasukkan sendiri nomor handphone yang akan diisi.

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi sekarang ini semakin berkembang dalam berbagai bidang seiring dengan perkembangan zaman yang semakin canggih. Salah satunya adalah di bidang elektronika. Peralatan elektronika menjadi piranti yang sangat banyak dipergunakan diberbagai bidang khususnya pada piranti dengan kerumitan yang tidak dapat dilakukan oleh manusia dengan cepat dan kesukaran dalam prosesnya, maka dengan teknologi elektronika akan mampu memberikan kemudahan dan kenyamanan dalam melakukan suatu pekerjaan.

Salah satu perkembangan teknologi elektronika adalah terobosan dan penemuan baru dalam bidang Mikrokontroller dengan teknologi *single chip*. Berdasarkan pada masalah ini akan dirancang suatu sistem yang dapat melakukan pengisian ulang voucher elektrik yang diintegrasikan langsung dari handphone ke mikrokontroller sehingga penjual voucher elektrik tidak usah menulis kode-kode lewat sms untuk mengisi pulsa. Hal ini memungkinkan para pembeli dapat mengisi ulang voucher elektrik sendiri tanpa bantuan penjual.

Setiap orang menginginkan kecepatan dalam segala hal, termasuk dalam hal untuk melakukan isi ulang voucher elektrik. Biasanya apabila sebuah counter handphone akan melakukan isi ulang voucher elektrik bila ada pembeli, dia harus mengetikkan kode-kode untuk melakukan isi ulang tersebut lewat sms termasuk juga nomor HP pembeli voucher dan nominal voucher yang dibeli. Hal ini kadang-kadang dapat menimbulkan kesalahan pada penulisan nomor handphone yang akan diisi dan nominal voucher yang akan dibeli. Untuk menghindari hal tersebut kami merancang alat isi ulang voucher elektrik dengan handphone yang diinterfacekan

pada mikrokontroler AT89C4051. Alat ini sangat berguna untuk proses isi ulang voucher elektrik dan tidak menimbulkan kesalahan di pihak penjual karena pembeli langsung memasukkan sendiri nomor handphone yang akan diisi.

### **1.2. Rumusan masalah**

Dengan memperhatikan latar belakang dari usulan skripsi ini didapatkan rumusan masalah sebagai berikut :

- Bagaimana mempelajari untuk merencanakan dan membuat alat isi ulang voucher elektrik dengan handphone yang diinterfacekan dengan mikrokontroler AT89C4051.
- Bagaimana merencanakan dan membuat rangkaian interface dari handphone ke mikrokontroler.
- Bagaimana membuat perangkat lunak atau software yang berfungsi untuk mengendalikan alat yang direncanakan.

### **1.3. Tujuan Perencanaan**

Dari rumusan masalah di atas maka tujuan ini adalah untuk merencanakan dan membuat alat isi ulang voucher elektrik dengan handphone yang diinterfacekan ke mikrokontroler sebagai pengendali..

### **1.4. Batasan Masalah**

Agar pemmasalahan dalam tugas akhir ini tidak berkembang karena keterbatasan pengetahuan, waktu, dan biaya maka pembahasan dibatasi pada :

---

- a. Minimum system yang dirancang menggunakan Chip AT89C4051 sebagai unit kontrol utama.
- b. Handphone yang kami gunakan adalah handphone siemens C45.

### 1.5. Metodologi penelitian

Metode yang digunakan dalam pembuatan laporan ini adalah :

- **Study Literature**, Dengan mempelajari teori serta aplikasi system kontrol menggunakan mikrokontroller AT89C4051.

Study Literature disini akan mempelajari hal – hal sebagai berikut:

- Arsitektur mikrokontroller AT89C4051.
- Karakteristik umum
- Interface dengan Input / Output ( I / O ) mikrokontroller

- **Perencanaan dan pembuatan alat**

Dalam pembuatan alat ini menggunakan konsep sebagai berikut :

- Perencanaan system secara keseluruhan ( pembuatan blok diagram System )
  - Mendiskripsikan fungsi dari masing – masing blok diagram
  - Membuat perangkat keras ( hardware ) dan perangkat lunaknya ( software )
  - Implementasi software yang telah dirancang kedalam chip AT89C4051.
- **Field Research ( Riset Lapangan )**, Membandingkan data hasil percobaan dengan kondisi dan keadaan yang sesungguhnya dilapangan, yang merupakan data primer sebagai pembuktian kebenaran dari data sekunder hasil study literature.
-

## 1.6. Sistematika Penulisan

Pada penulisan tugas akhir ini ditulis sedemikian rupa sehingga diperoleh hubungan yang jelas antara bagian yang satu dengan bagian yang lainnya. Sistematiknya adalah sebagai berikut :

### BAB I. PENDAHULUAN

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

### BAB II. LANDASAN TEORI

Meliputi dasar teori yang mendukung perancangan alat .

### BAB III. PERENCANAAN DAN PEMBUATAN ALAT

Meliputi penjelasan mengenai perancangan dan pembuatan alat .

### BAB IV. PENGUJIAN ALAT

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

### BAB V. PENUTUP

Berisi kesimpulan dan saran.

---

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Pendahuluan**

Dalam perencanaan dan perealisasiian alat ini dibutuhkan pemahaman mengenai pengetahuan yang berhubungan dengan aplikasi alat yang akan direalisasikan, aplikasi yang dibahas dalam bab ini antara lain : mengenai Mikrokontroller, interface antara handphone dengan mikrokontroller. Ponsel yang digunakan dalam perancangan ini berupa ponsel tipe Siemens C45.

#### **2.2. Mikrokontroller AT89C4051**

##### **2.2.1. Pendahuluan**

Perbedaan mendasar antara mikrokontroller dengan mikroprosesor adalah mikrokontroller selain memiliki CPU juga dilengkapi dengan memori dan input output yang merupakan kelengkapan dari sistem minimum mikrokomputer sehingga sebuah mikrokontroller dapat dikatakan sebagai mikrokomputer dalam keping tunggal (Single Chip Mikrokontroller) yang dapat berdiri sendiri.

Mikrokontroller AT89C4051 adalah mikrokontroller ATMEL yang kompatibel penuh dengan mikrokontroller keluarga MCS-51, membutuhkan daya yang rendah, memiliki performa yang tinggi, dan merupakan mikrokomputer 8 bit yang dilengkapi 4 Kbyte EPROM (Erasable and programmable Read Only Memory) dan 128 RAM byte internal.

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

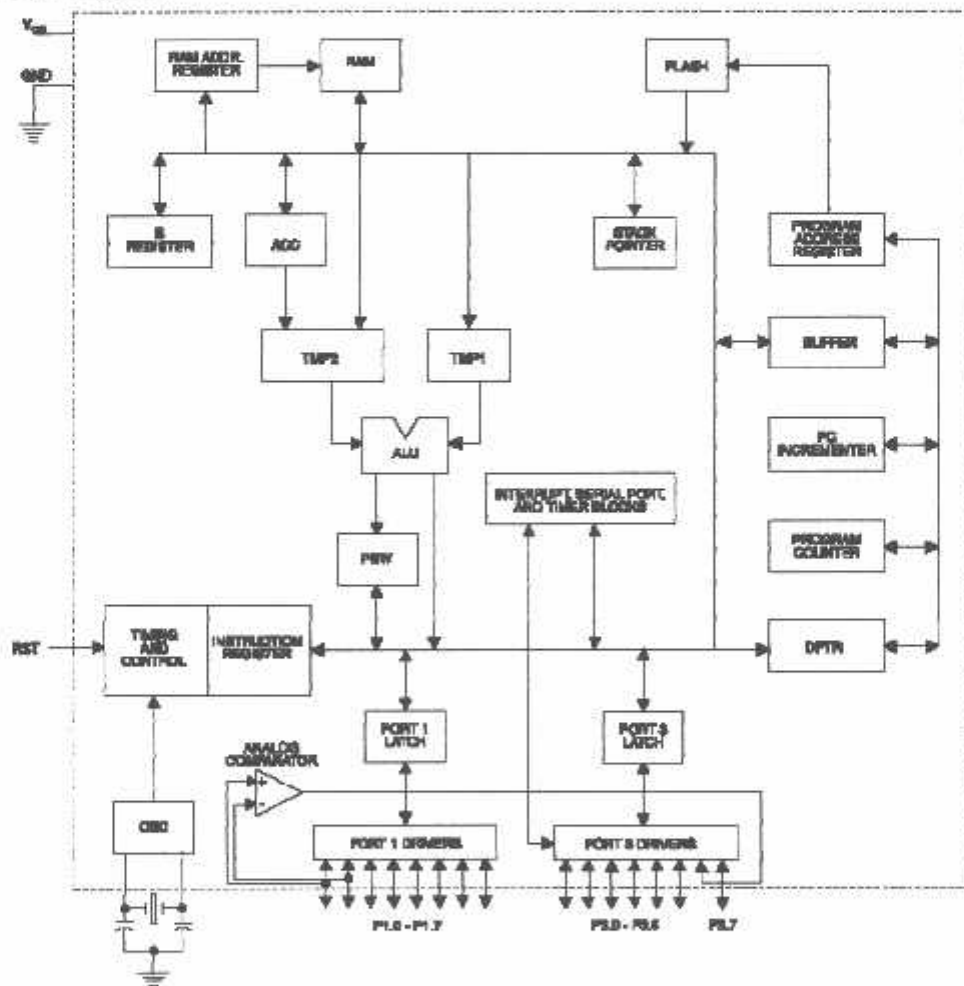


### 2.2.2. Perangkat Keras Mikrokontroller AT89C4051

Mikrokontroller AT89C4051 secara umum memiliki :

- CPU 8 bit
- Memori
- Port input output yang dapat diprogram
- Timer dan counter
- Sumber Interrupt
- Port serial yang dapat diprogram
- Oscilator dan Clock

Block Diagram



Gambar 2-1 Blok Diagram Mikrokontroller <sup>1)</sup>

<sup>1)</sup>Data Sheet AT89C4051, [www.ATMEL.com](http://www.ATMEL.com), 2003, Halaman 2

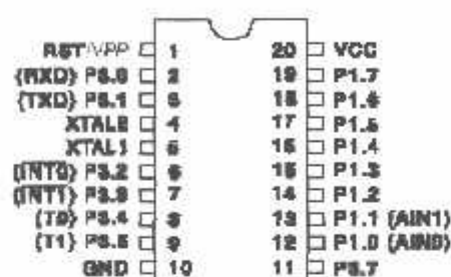
### 2.2.2.1. Arsitektur AT89C4051

Arsitektur AT89C4051 adalah sebagai berikut :

1. CPU (Central Processing Unit) 8 bit dengan register A (accumulator) dan B
  2. 16 bit program counter (PC) dan data pointer (DPTR)
  3. 8 bit program status word (PSW)
  4. 8 Bit Stack Pointer (SP)
  5. 4 Kbyte internal EPROM
  6. 128 Byte internal RAM
    - 4 bank register, masing-masing berisi 8 register.
    - 16 byte yang dapat dialami pada bit level.
    - 80 byte general purpose memory data
  7. 16 pin input output yang tersusun atas P1 dan P3, masing-masing 8 bit.
  8. 2 buah timer dan counter
  9. Receiver / transmitter data serial full duplex: SBUF
  10. Control register, yaitu TCON, TMOD, SCON, PCON, IP dan IE
  11. 5 buah sumber interrupt (2 buah sumber interrupt external dan 3 buah sumber interrupt internal)
  12. Oscilator dan clock Internal
-

### 2.2.2.2. Konfigurasi pena-pena Mikrokontroller AT89C4051

Mikrokontroller AT89C4051 terdiri atas 20 pin, dengan konfigurasi sebagai berikut :



Gambar 2.2 Konfigurasi Pena-pena AT89C4051 <sup>2)</sup>

Fungsi dari tiap-tiap pin adalah sebagai berikut :

1. VCC (Supply Tegangan)
2. GND (Ground)
3. Port 1 adalah port I/O bidireksional 8 bit. Pin port P1.2 sampai P1.7 memberikan tarikan internal. P1.0 dan P1.1 membutuhkan tarikan external. P1.0 dan P1.1 berperan sebagai input positif (AIN0) dan input negative (AIN1) dari komparator analog presisi on chip. Buffer output port 1 bisa diturunkan sampai 20mA dan bisa mengarahkan display led secara langsung. Jika 1 untuk pin port , maka bisa digunakan sebagai input. Jika P1.2 sampai P1.7 digunakan sebagai input diturunkan secara external, maka ia akan dijadikan arus sumber ( $I_{IH}$ ) karena tarikan internal. Port 1 juga menerima data kode selama pemrograman dan pembuktian flash.
4. Port 3 memasang P3.0 sampai P3.5. P3.7 dari Port 3 adalah tujuh pin I/O bidireksional dengan menarik internal. P3.6 memiliki ikatan kawat keras sebagai suatu input untuk output komparator on chip dan tidak bisa

<sup>2)</sup>Ibid, Halaman 1

dijangkau sebagai pin I/O tujuan umum. Buffer output Port 3 bisa turun sampai 20mA. Jika ditulis untuk pin port 3, maka pin bisa ditarik ke atas dengan penarikan dan bisa digunakan sebagai input. Sebagai input, pin port 3 yang ditarik ke bawah secara external akan menjadikan arus sumber ( $I_{II}$ ) karena ditarik ke atas. Port 3 berperan sebagai fungsi fitur khusus dari AT89C4051, port 3 juga menerima sinyal pengendalian dari pemrograman dan penjelasan flash.

5. RST adalah input reset
6. X-TAL1 dan X-TAL2

Kaki ini dihubungkan dengan kristal bila menggunakan oscillator internal. X-TAL1 merupakan input inverting oscillator amplifier sedangkan X-TAL2 merupakan output inverting oscillator amplifier.

### **2.2.3. Organisasi Memori**

Di dalam AT89C4051 ruang alamat telah dibedakan untuk program memori dan memori data.

#### **2.2.3.1. Program Memori Internal**

AT89C4051 memiliki program memori internal sebesar 4 Kbyte dengan ruang alamat 0000H-0FFFH. Jika alamat-alamat program lebih tinggi dari pada 0FFFH, yang melebihi kapasitas ROM internal menyebabkan AT89C4051 secara otomatis mengambil code byte dari program memori external. Code byte juga dapat diambil hanya dari memori external dengan alamat 0000H-0FFFH dengan cara menghubungkan pin ke ground.

---

### 2.2.3.2. Data Memori (RAM) Internal

Ruang alamat bawah memori data (RAM) internal dengan kapasitas 128 byte yaitu 00H-7FH yang terbagi atas 3 daerah yaitu :

- Empat bank register

Setiap bank terdiri atas 8 register (R0-R7) sehingga jumlah register untuk keempat bank register (bank0 - bank3) menjadi 32 buah register yang menempati ruang alamat 00H-1F1H. Mengaktifkan salah satu bank register dapat dilakukan dengan mengatur RS0-RS1 pada PSW (Program Status Word).

- Bit Addressable

Terdiri atas 16 byte yang berada pada alamat 20H-2FH. Masing-masing 128 bit lokasi ini dapat diamati secara langsung.

- Strach Pad area

Terdiri atas 80 byte yang secara langsung dan digunakan untuk keperluan umum (general purpose) misalnya digunakan untuk lokasi stack.

**Tabel 2-1. Pengaturan RS0-RS1 Untuk Select Register Bank <sup>3)</sup>**

RS1	RS0	Select Register Bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank 3

<sup>3)</sup>Ibid, Halaman 9

### 2.2.3.3. SFR (Register Fungsi Khusus)

Suatu peta area memori on chip yang disebut dengan fungsi khusus ditunjukkan tabel berikut :

**Tabel 2-2. Special Function Register <sup>4)</sup>**

010H								010H
014H	B 00000000							014H
018H								018H
01CH	ACC 00000000							01CH
020H								020H
024H	PSW 00000000							024H
028H								028H
034H								034H
038H	IP 00000000							038H
03CH	PC 11111111							03CH
040H	IE 00000000							040H
044H								044H
048H	SCON 00000000	SBUR XXXXXX						048H
04CH	PI 11111111							04CH
050H	LCR4 00000000	IM40 00000000	IE4 00000000	B4 00000000	IE5 00000000	IE6 00000000		050H
054H		SI 00001111	DIPI 00000000	DI44 00000000			FSN 00000000	054H

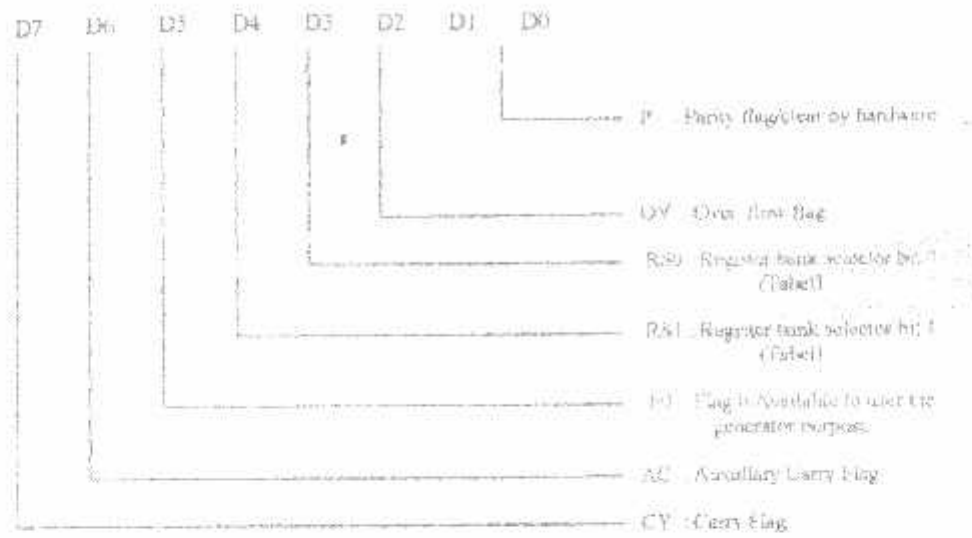
Tidak semua pengalamatan digunakan dan alamat yang digunakan tidak semua diterapkan pada chip. Akses baca pada alamat ini umumnya akan mengembalikan detik acak dan akses tulis tidak memiliki pengaruh yang tidak ditentukan.

Software user hendaknya tidak menulis satu untuk lokasi yang tidak terdaftar, karena mungkin digunakan untuk produk masa mendatang untuk mendukung feature baru. Dalam hal ini reset atau tidak aktif dari bit baru ini akan selalu menjadi nol.

<sup>4)</sup>Ibid, Halaman 5

#### 2.2.3.4. Program Status Word

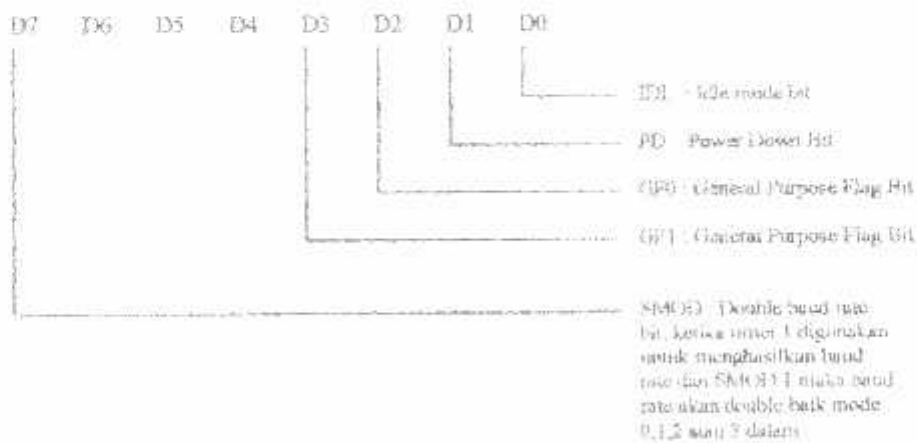
Register ini terletak pada alamat D0H. Cara mendefinisikannya adalah sebagai berikut :



Gambar 2.3. Skema Mendefinisikan PSW<sup>5)</sup>

#### 2.2.3.5. PCON (Power Control)

Register ini terletak pada alamat 87H. Cara mendefinisikannya adalah sebagai berikut :



Gambar 2.4. Skema Mendefinisikan PCON<sup>6)</sup>

<sup>5)</sup>Moh Ibnu Malik Anistardi, *Bereksperimen Dengan Mikrokontroler 8031*, PT.Elex Media Komputindo, Jakarta, 1997, halaman 30

<sup>6)</sup>Ibid, Halaman 9

### 2.2.3.6. Sistem Interrupt

Mikrokontroler AT89C4051 memiliki 5 buah sumber interrupt yang dapat membangkitkan permintaan yaitu : INT0, INT1, T0, T1 dan port serial.

Saat terjadinya interrupt mikrokontroler secara otomatis akan menuju sub rutin pada alamat tersebut setelah interrupt service selesai dikerjakan, mikrokontroler akan mengerjakan program semula. Dua sumber eksternal adalah INT0 dan INT1 dimana kedua interrupt eksternal akan aktif atau transisi tergantung isi IT0 dan IT1 pada register TCON. Interupsi T0 dan T1 aktif pada saat timer yang sesuai mengalami roll over. Interupsi serial dibangkitkan dengan melakukan operasi OR pada RI dan TI. Tiap-tiap sumber interupsi dapat enable atau disable secara software.

Tingkat prioritas semua sumber interupsi dapat diprogram sendiri-sendiri dengan set atau clear bit pada SFRS (Interupsi Priority)

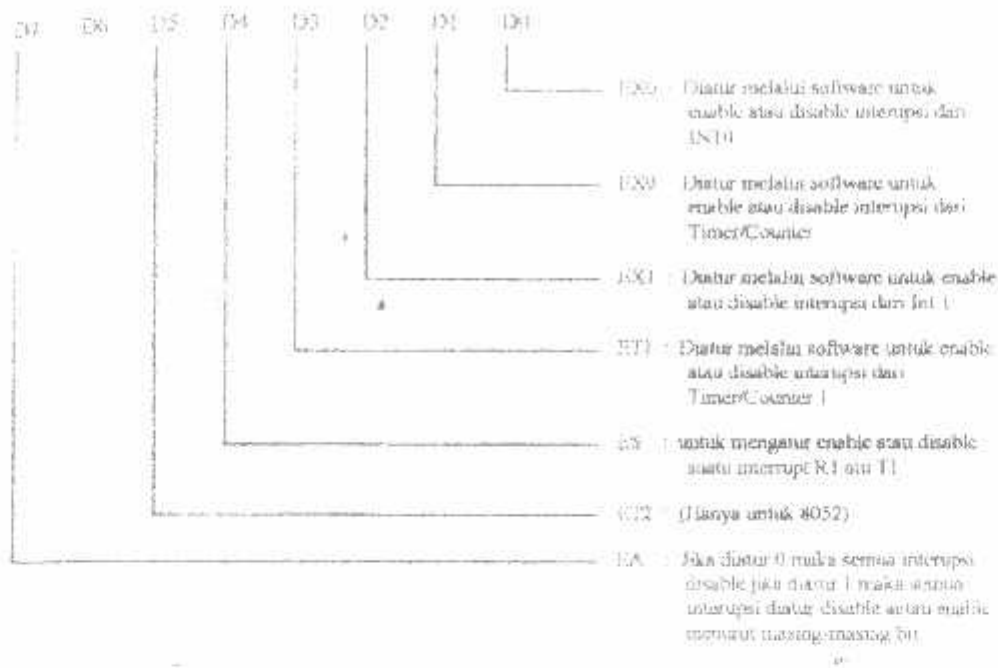
**Tabel 2.3. Alamat Sumber Interupsi <sup>7)</sup>**

Sumber Interrupt	Alamat Awal
Interupt Luar 0 (INT0)	0003h
Pewaktu/Pencacah 0 (T0)	003Bh
Interupt Luar 1 (INT1)	001Bh
Pewaktu/Pencacah 1 (T1)	001Bh
Port serial	0023h

Register yang berperan dalam mengatur aktif tidaknya interupsi adalah input enable register, berikut adalah susunan dari bit-bit beserta kegunaannya.

<sup>7)</sup>Ibid, Halaman 16





**Gambar 2.5. Kegunaan Interrupt Enable Register <sup>8)</sup>**

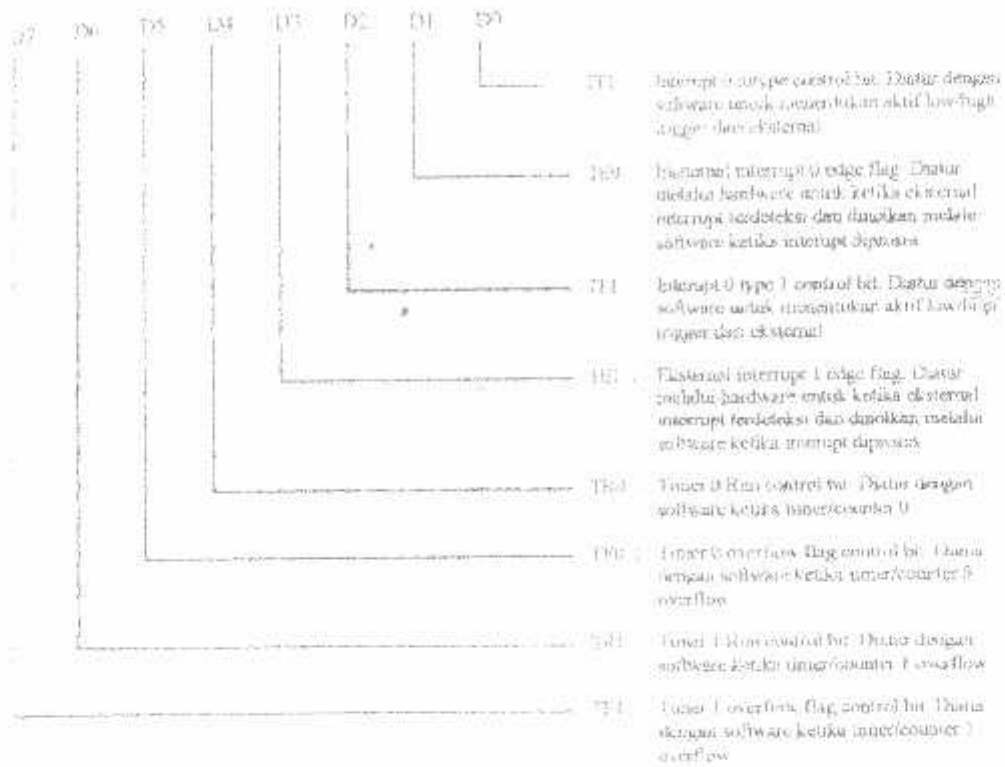
### 2.2.3.7. Timer/Counter

Pengendalian kerja dari timer/counter dilakukan dengan pengaturan register yang berhubungan dengan kinerja dari timer/counter yaitu melalui sebuah Timer/Counter Mode Control.

Untuk mengaktifkan Timer/Counter yang meliputi penentuan fungsi sebagai timer atau counter serta pemilihan mode operasi dapat diatur melalui TMOD yang beralamat pada 89 H.

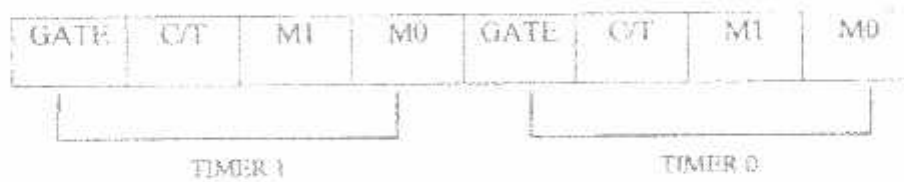
Konfigurasi yang dimaksud adalah sebagai berikut :

<sup>8)</sup>Ibid, Halaman 17



**Gambar 2.6. Konfigurasi Dan Guna TMOD**<sup>9)</sup>

TMOD : Timer/Counter Mode Control Register



**Gambar 2.7. Timer/Counter Mode Control Register**<sup>10)</sup>

<sup>9)</sup>Ibid, Halaman 24

<sup>10)</sup>Ibid, Halaman 26

**Tabel 2.4. Alamat Sumber Interupsi <sup>11)</sup>**

M1	M0	Operating Mode
0	0	Timer 13 bit
0	1	Timer/Counter 16 bit
1	0	16 bit auto reload Timer/Counter
1	1	TLO dari Timer adalah 8 bit Timer/Counter dikendalikan oleh control bit timer 0. Th0 adalah Timer 8 bit yang dikendalikan oleh timer 1 control bit

- Gate

Bila Gate = 1, Timer/Counter x enable hanya pada saat pin INT x tinggi dari trx 1. Saat gate 0, Timer/Counter enable jika bit trx 1.

- CT

Jika Bit C/T = 0, maka timer/counter x akan berfungsi sebagai timer. Jika C/T = 1 maka counter x akan berfungsi sebagai counter.

- M1 dan M2

Menentukan Mode.

### 2.2.3.8. Metode Pengalamatan

#### 2.2.3.8.1. Pengalamatan Bit (Bit Addressing)

Pengalamatan langsung tiap bit ini hanya dilakukan pada lokasi RAM internal yaitu 20H-2FH dan sebagian SFR, port 1, port 3, TCON register, SCON register, IE register, PSW register, ACC dan B register.

<sup>11)</sup>Ibid, Halaman 16

#### 2.2.3.8.2. Pengalamatan Tak Langsung

Pada pengalamatan tak langsung, instruksi menunjukkan suatu register yang isinya adalah alamat dari operand, eksternal dan internal RAM dapat dialamati secara tidak langsung. Register alamat untuk data dengan lebar 8 bit dapat berupa R0 dan R1 yang digunakan untuk memilih angka register atau stack pointer. Register alamat untuk data dengan lebar 16 bit digunakan untuk data pointer (DPTR).

#### 2.2.3.8.3. Pengalamatan Berindeks

Yang dapat diakses dengan pengalamatan berindeks hanya memori program. Mode ini dimaksudkan untuk membaca look-up table program.

#### 2.2.3.8.4. Konstanta Immediate

Pengalamatan langsung dilakukan dengan memberikan nilai ke suatu register secara langsung dengan menggunakan tanda #

Contoh : `Mov a,#100h`

### 2.3. Sistem Kerja SMS

Dibalik tampilan menu message pada sebuah ponsel sebenarnya adalah AT command 2x yang bertugas mengirim/menerima data dari/ke SMS-Centre. AT Command tiap-tiap SMS device bisa berbeda-beda, tapi pada dasarnya sama. AT command sebenarnya hampir sama dengan perintah > (prompt) pada DOS. Perintah- perintah yang dimasukkan ke port dimulai dengan kata AT, lalu diikuti oleh karakter lainnya, yang memiliki fungsi-fungsi yang unik.

---

### 2.3.1. AT Command Untuk Komunikasi Dengan SMS-Centre

Beberapa AT Command yang penting untuk SMS antara lain :

AT+CMGS → Untuk Mengirim SMS

AT+CMGL → Untuk Memeriksa SMS

AT+CMGD → Untuk Menghapus SMS

AT Command untuk SMS, biasanya diikuti oleh data I/O yang diwakili oleh unit-unit PDU.

### 2.3.2. Hyper Terminal

Salah satu software yang dapat digunakan untuk mengetes AT Command adalah Windows Hyper Terminal. Hyper Terminal biasanya telah tersedia bersama windows installer sehingga kita hanya perlu menambahkan software tersebut dari control panel > Add/Remove Windows Components > dan seterusnya. Setelah di Add carilah iconnya pada menu **Start > Program > Accessories > Communication > Hyper Terminal.**

Setelah icon tersebut diklik, akan muncul jendela setelah itu pilihlah nama connection, misalnya HT1, dan pilihlah sebuah icon dari icon-icon yang tersedia. Berikutnya akan muncul jendela tujuan connection tersebut.

Pada Combo Box : "Connect Using", pilihlah port/modem yang terdaftar. Misalnya, jika SMS Device yang ada terhubung dengan port COM1, pilihlah COM1. Selanjutnya akan muncul jendela properties untuk connection tersebut. Nilai untuk properties yang harus diisi bergantung pada jenis/alat komunikasi yang kita tuju. Jika properties ini tidak diset dengan benar, komunikasi data tidak akan terjadi. Ukuran "bits per second" disebut juga "baud rate"

Sebagai contoh : baud rate untuk SMS device berupa ponsel Siemens C45 adalah 115,2 Kbps.

### 2.3.3. PDU Sebagai Bahasa SMS

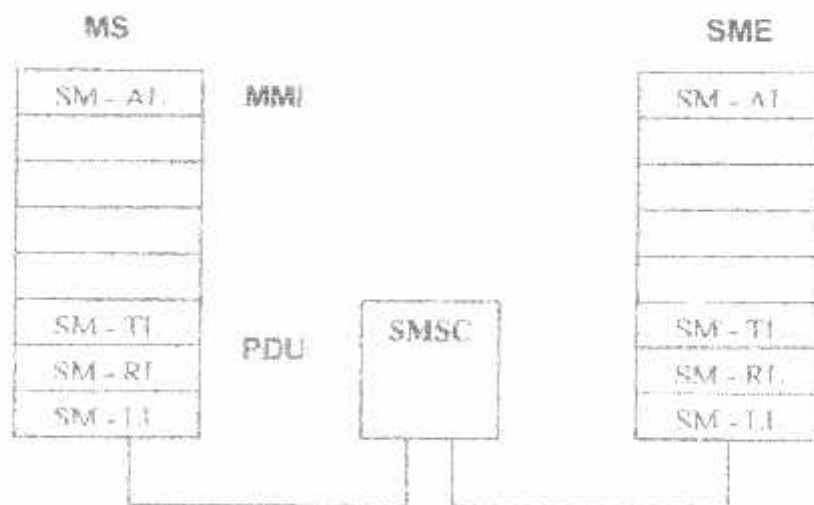
Data yang mengalir ke/dari SMS-Centre harus berbentuk PDU (Protocol Data Unit). PDU berisi bilangan-bilangan heksadesimal yang terdiri atas: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Sebagai contoh, untuk angka decimal 1000, bilangan heksadesimalnya adalah 3E8. Cara mengkonversikannya :

$$1000 : 16 = 62 \text{ Sisa } 8 \rightarrow \underline{8}$$

$$62 : 16 = 3 \text{ Sisa } 14 \rightarrow \underline{E}$$

$$3 : 16 = 0 \text{ Sisa } 3 \rightarrow \underline{3} \quad ^{12)}$$

#### 2.3.3.1. PDU Untuk Kirim SMS ke SMS Centre



Gambar 2.3 A Request SMSC<sup>13)</sup>

<sup>12)</sup>Tr. Bustam Khang, *Trik Pemrograman Aplikasi Berbasis SMS*, PT Elex Media Komputindo, Jakarta, 2003, halaman 9

<sup>13)</sup>ibid, Halaman 9

<b>MS</b>	: Mobile Station
<b>SME</b>	: Short Message Entity
<b>SMSC</b>	: Short Message Service Centre
<b>MMI</b>	: Man Machine Interface
<b>PDU</b>	: Protocol Data Units
<b>SM-AL</b>	: Short Message Application Layer
<b>SM-TL</b>	: Short Message Transport Layer
<b>SM-RL</b>	: Short Message Relay Layer
<b>SM-LL</b>	: Short Message Link Layer

**PDU** untuk mengirim SMS terdiri atas delapan header, antara lain :

➔ **Nomor SMS-Centre**

- Jumlah pasangan Heksadesimal SMS Centre dalam bilangan Heksa
- Nasional/Internasional Code

Untuk nasional, kode subheadernya yaitu 81

Untuk internasional, kode subheadernya yaitu 91

- No SMS Centrenya sendiri, dalam pasangan heksa dibolak-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya.

Contoh : Untuk SMS Centre Excelcom dapat ditulis dengan dua cara sebagai berikut :

Cara 1 :

0818445009 diubah menjadi :

- |                              |                  |     |
|------------------------------|------------------|-----|
| a. 06 ➔ ada 6 pasang         |                  |     |
| b. 81 ➔ 1 pasang             | } Total 6 Pasang | 14) |
| c. 80-81-44-05-90 ➔ 5 pasang |                  |     |

<sup>14)</sup>ibid, Halaman 10

Digabung menjadi 06818081440590

Cara 2 :

62818445009 diubah menjadi :

- a. 07 → ada 7 pasang
- b. 91 → 1 pasang
- c. 26-81-48-54-00-F9 → 6 pasang
- } Total 7 Pasang <sup>15)</sup>

Berikut ini beberapa no SMS Centre Operator selluler di Indonesia

**Cara I**

**Tabel 2.5. Operator SMS Centre <sup>16)</sup>**

No	Operator Selluler	SMS Centre NO	Kode PDU
1	Telkomsel	0811000000	06818011000000
2	Satelindo	0816125	0581806121F5
3	Excelcom	0818445009	06818081440590
4	Indo-M3	08550100000	06818055000000

**Cara II**

**Tabel 2.6. Operator Selluler <sup>17)</sup>**

No	Operator Selluler	SMS Centre NO	Kode PDU
1	Telkomsel	0811000000	07912618010000F0
2	Satelindo	0816125	059126181652
3	Excelcom	0818445009	07912618485400F9
4	Indo-M3	08550100000	07912658050000F0

<sup>15)</sup>bid, Halaman 10

<sup>16)</sup>bid, Halaman 10

<sup>17)</sup>bid, Halaman 11



→ Tipe SMS

→ Nomor Referensi SMS

→ Nomor Ponsel Penerima

→ Bentuk SMS

→ Skema Encoding Data I/O

Ada 2 skema, yaitu :

- a. Skema 7 bit → ditandai dengan angka 0 → 100
- b. Skema 8 bit → ditandai dengan angka lebih besar dari 0 → diubah ke heksa.

Kebanyakan ponsel/SMS Gateway yang ada di pasaran sekarang menggunakan skema 7 bit sehingga kita menggunakan kode **00**.

→ Jangka Waktu Sebelum SMS Expired

Jika bagian ini di-skip, itu berarti kita tidak membatasi waktu berlakunya SMS. Sedangkan jika kita isi dengan suatu bilangan akan mewakili jangka waktu validitas SMS tersebut.

Rumus untuk menghitung jangka waktu validitas SMS adalah sebagai berikut :

**Tabel 2.7. Validasi SMS**<sup>18)</sup>

Integer (INT)	Jangka Waktu Validitas SMS
0-143	(INT + 1) 5 Menit (Berarti : 5 Menit s/d 12 Jam)
144-167	12 Jam + (INT-143) x 30 Menit)
168-196	(INT-166) x 1 Hari
197-255	(INT-192) x 1 Minggu

Agar SMS yang kita buat pasti terkirim sampai ke ponsel penerimanya sebaiknya kita tidak memberikan batasan waktu validnya.

<sup>18)</sup>Ibid, Halaman 13

→ **Isi SMS**

Header ini terdiri atas dua subheader, yaitu :

- a. Panjang isi (Jumlah huruf dan isi)

Misalnya : Untuk kata "Hello" → ada 5 huruf → **05**

- b. Isi berupa pasangan bilangan heksa

Untuk ponsel/SMS gateway berskema encoding 7 bit, jika kita mengetikkan suatu huruf dari keypadnya, berarti kita membuat 7 angka 1/0 berurutan. Ada dua langkah yang harus dilakukan untuk mengkonversikan isi SMS, yaitu :

Langkah Pertama : Mengubahnya menjadi kode 7 bit.

Langkah Kedua : Mengubah kode 7 bit menjadi 8 bit yang diwakili oleh pasangan heksa.

Contoh : Untuk kata "**Hello**"

**Langkah Pertama :**

Bit	7	1	
h	110	1000	
e	110	0101	
l	110	1100	
l	110	1100	
o	110	1111	<sup>19)</sup>

<sup>19)</sup>Ibid, Halaman 14

**Langkah Kedua :**

		<b>E</b>	<b>8</b>	
h	1	110	1000	
e	00	3 11	2 0010	1
l	100	9 1	B 1011	00
l	F 1111		D 1101	100
o	0 0000	0	6 110	1111

Oleh karena total 7 bit x 5 huruf = 35 bit, sedangkan yang kita perlukan adalah 8 bit x 5 huruf = 40 bit, maka diperlukan 5 bit dummy yang diisi dengan bilangan 0. Setiap 8 bit mewakili suatu angka heksa, tentu saja karena secara logika 2 pangkat 4 sama dengan 16.

Dengan demikian kata "hello" hasil konversinya menjadi **E8329BFD06**

b4	B3	b2	b1	b7	0	0	0	0	1	1	1	1
				b6	0	0	1	1	0	0	1	1
				b5	0	1	0	1	0	1	0	1
					0	1	2	3	4	5	6	7
0	0	0	0	0	@	Δ	SP	0	-	P	"	p
0	0	0	1	1			!	1	A	Q	a	q
0	0	1	0	2	\$	Φ	"	2	B	R	b	r
0	0	1	1	3		Γ	#	3	C	S	c	s
0	1	0	0	4		Λ		4	D	T	d	t
0	1	0	1	5		Ω	%	5	E	U	e	u
0	1	1	0	6		∏	&	6	F	V	f	v
0	1	1	1	7		Ψ	'	7	G	W	g	w
1	0	0	0	8		Σ	(	8	H	X	h	x
1	0	0	1	9		⊙	)	9	I	Y	i	y
1	0	1	0	10	LF	Ξ	*	:	J	Z	j	z
1	0	1	1	11			+	:	K	Ä	k	ä
1	1	0	0	12			,	<	L	Ö	l	ö
1	1	0	1	13	CR		-	-	M		m	
1	1	1	0	14		B	.	>	N	Ü	n	
1	1	1	1	15			/	?	O		o	

Tabel 2-8. Skema 7 Bit <sup>21)</sup>

### Menggabungkan Kedelapan Header

Setelah kita mempelajari masing-masing header maupun subheader untuk mengirim SMS, kini kita akan menggabungkannya menjadi sebuah PDU yang lengkap. Contoh :

Untuk mengirimkan kata "hello" ke ponsel nomor 62812973337 lewat SMS-Centre Excelcom, tanpa membatasi jangka waktu valid, maka PDU lengkapnya adalah :

**07912618485400F901000C91261892753373000005E8329BFD06**

#### 2.3.3.2. PDU Untuk Terima SMS Dari SMS-Centre

Sama halnya dengan PDU untuk mengirim SMS, pada bagian terima SMS

21)

→ No SMS-Centre

→ Tipe SMS → untuk SMS - terima - 4 → **04**

<sup>21)</sup>Ibid, Halaman 18

- No Ponsel Pengirim
- Bentuk SMS
- Skema Encoding
- Tanggal dan waktu SMS di-stamp di SMS-Centre. Diwakili oleh 12 bilangan heksa (6 pasangan) yang berarti : yy/mm/dd hh:mm:ss.  
Contoh : 300150512380 → 03/10/05 15:32:08 → 5 Oktober 2003 15:32:08 WIB.
- Batas waktu validitas, jika tidak dibatasi dilambangkan dengan 00
- Isi SMS.

### 2.3.3.3. Membedah Kedelapan Header

Setelah mengupas satu demi satu header untuk SMS-Terima ini, maka untuk PDU berikut :

**07912658050000F00,04,0C91265816107398,00,00,300150512380,00,005E8329B  
FD06**

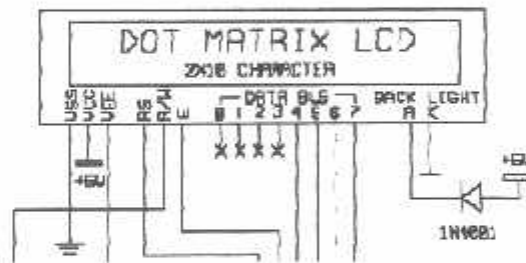
Dapat diartikan sebagai berikut :

1. SMS Tersebut dikirim lewat SMS-Centre : 62855000000
  2. SMS tersebut merupakan SMS terima
  3. SMS tersebut dikirim dari ponsel no.628561013789
  4. SMS tersebut diterima dalam bentuk SMS
  5. SMS tersebut memiliki skema encoding 7 bit
  6. SMS tersebut sampai di SMS-Centre pada tanggal : 05-10-03, pukul : 15.32:08 WIB
  7. SMS tersebut tidak memiliki batas waktu valid
  8. SMS tersebut isinya "hello"
-

## 2.4. LCD

LCD display module M1632 buatan Seiko instrument Inc terdiri dari 2 bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka 2 baris, masing-masing baris bisa menampung 16 huruf/angka.

Bagian kedua merupakan sebuah sistem yang dibentuk dengan mikrokontroler yang ditempelkan dibalik pada panel LCD, berfungsi mengatur tampilan informasi serta berfungsi mengatur komunikasi M1632 dengan mikrokontroler yang memakai tampilan LCD itu. Dengan demikian pemakaian M1632 menjadi sederhana, sistem lain yang M1632 cukup mengirimkan kode-kode ASCII dari informasi yang ditampilkan seperti layaknya memakai sebuah printer. Berikut adalah gambar LCD 2x16.



Gambar 2-9 LCD 2x16 Karakter <sup>22)</sup>

LCD M1632 mempunyai spesifikasi sebagai berikut :

1. Mempunyai 16 karakter dan 2 baris tampilan yang terdiri dari 5x7 dot matrik ditambah dengan kursor.
2. Pembangkit karakter ROM untuk 192 jenis karakter.
3. Pembangkit karakter RAM untuk 8 jenis karakter.
4. 80x8 display data RAM (max 80 karakter).
5. Isolator di dalam modul.

<sup>22)</sup>Datasheet LCD, [www.alldatasheet.com](http://www.alldatasheet.com), Halaman 1

6. Memerlukan catu daya 5 Volt.
7. Otomatis reset saat catu daya dinyalakan.

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

**Tabel 2-9. Nama-nama Pin Pada LCD M1632 <sup>23)</sup>**

<b>Nama Pin</b>	<b>Fungsi</b>
VSS	Terminal Ground
VCC	Tegangan Catu 5 Volt
VEE	Mengendalikan kecerahan LCD
RS	Sinyal pemilihan register 0 = tulis 1 = baca
R/W	Sinyal seleksi baca/tulis 0 = tulis 1 = baca
E	Sinyal operasi awal yang mengaktifkan data tulis atau baca
DB0-DB7	Merupakan saluran data berisi perintah data yang akan ditampilkan
A	Backlight supply 5 Volt (Volt)
K	Backlight supply 0 (Ground)

## 2.5. Data Cable (Port Serial)

Data cable adalah perangkat keras yang merupakan sarana penghubung antara ponsel dan mikrokontroller agar bisa berkomunikasi. Seperti kabel printer, penghubung printer dengan computer sebagai sarana komunikasi antar dua perangkat keras. Dengan kata lain, apa yang bisa dilakukan oleh data kabel tergantung dari softwarena.

Keindahan menggunakan data kabel adalah fungsinya yang tak terbatas karena setiap langkah digerakkan oleh perangkat lunak (software). Data kabel hanya sebagai penterjemah antara ponsel dan mikrokontroller. Mulai dari koneksi

<sup>23)</sup>Datasheet LCD, [www.alldatasheet.com](http://www.alldatasheet.com), Halaman 1

ke internet atau fax sampai dengan mengorganisasikan isi dari SIM Card anda ataupun fitur ponsel. Termasuk juga untuk mengganti logo dan nada dering pada ponsel yang memiliki fitur tersebut, upgrade versi, ganti bahasa, dan lain sebagainya.

Macam data kabel yaitu : FBUS, MBUS, dan gabungan FBUS/MBUS. FBUS memerlukan dua input/output untuk mengirimkan dan menerima data dimana MBUS hanya memerlukan satu input/output saja. Hanya dengan kabel MBUS anda dapat mengakses data penting di dalam ponsel untuk menservis ponsel, upgrade software, dan lain-lain. FBUS biasanya digunakan untuk memasukkan logo dan ringtone di ponsel serta kebutuhan mobilitas seperti konek ke internet dan faxing.

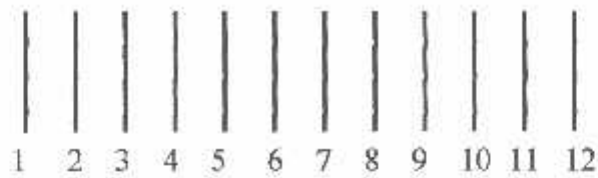
Kecepatan data ketiga untuk kirim maupun terima biasanya 9,6 K sampai dengan 14,4 K.

## **2.6. Interface Pada Ponsel/Handphone Dengan Mikrokontroller**

Handphone saat ini bukanlah alat yang mewah, hampir semua dapat memilikinya. Logo maupun ring tone-nya dapat diganti sesuai dengan keinginan user. Dunia komunikasi saat ini sudah semakin canggih, alat komunikasi sekarang tidak hanya sekedar untuk berkomunikasi saja namun telah menjadi alat bantu bagi para pengguna. Semakin banyak merek ataupun jenis handphone saat ini sering menimbulkan masalah pada interface antara mikrokontroller dan handphone tidak mempunyai standar, tidak seperti protocol komunikasi RS232 yang sudah distandarkan dalam IEEE. Oleh karena itu, agar sebuah mikrokontroller dapat berkomunikasi dengan jenis handphone maka perlu diketahui koneksi pada setiap jenis handphone. Contoh konektor Siemens C45 dapat dilihat pada gambar berikut

---





**Gambar 2-10. Konektor HP Seimens C45 <sup>24)</sup>**

Keterangan :

PIN	Nama	Fungsi	Input / Output
1	GND	Ground	-
2	Self service	Recognition/control Baterai charger	In/out
3	Load	Charging Voltage	In
4	Baterai	Baterai	Out
5	Data Out	Data sent	Out
6	Data In	Data received	In
7	Z_CLK	Recognition/Control Accessories	-
8	Z_Data	Recognition/Control Accessories	-
9	Micg	Ground For Mikrophone	In
10	Mic	Microphone input	-
11	Aud	Loudspeaker	Out
12	Audg	Ground Eksternal Speaker	-

**Tabel 2-10. Konektor Siemens C45 <sup>25)</sup>**

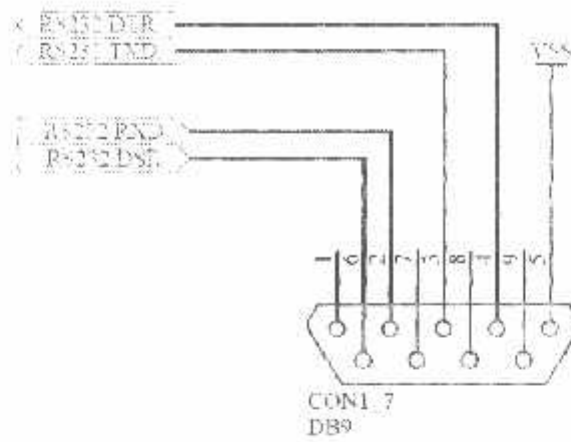
## 2.7. Port Serial

Port komunikasi yang digunakan adalah port serial yang terdiri dari berbagai jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE & DCE, semuanya terdapat 24 jenis sinyal, tapi yang umum dipakai hanyalah 9 jenis sinyal. Konektor yang dipakai untuk sinyal yang lengkap adalah DB25, sedangkan konektor DB9 hanya bisa dipakai untuk 9 sinyal yang umum dipakai.

Port serial dengan DB9 cable connection yang mempunyai 9 buah pin dengan fungsinya masing-masing ditunjukkan dalam table :

<sup>24)</sup> [www.elektroindonesia.com](http://www.elektroindonesia.com), 2004

<sup>25)</sup> [www.elektroindonesia.com](http://www.elektroindonesia.com), 2004



**Gambar 2-11. Port Serial** <sup>26)</sup>

**Keterangan :**

No	Konektor	
1	Received line signal detector (Carrier detect/DCD)	Handshake from DCE
2	Received data (RD)	Data from DCE
3	Transmit data (TD)	Data From DTE
4	Data Terminal Ready (DTR)	Handshake from DTE
5	Signal Ground	Reference point for signal
6	Data set ready (DSR)	Handshake from DCE
7	Request to send (RTS)	Handshake from DTE
8	Clear to send (CTS)	Handshake from DCE
9	Ring indicator	Handshake from DCE

**Tabel 2-11. Konektor Port Serial** <sup>27)</sup>

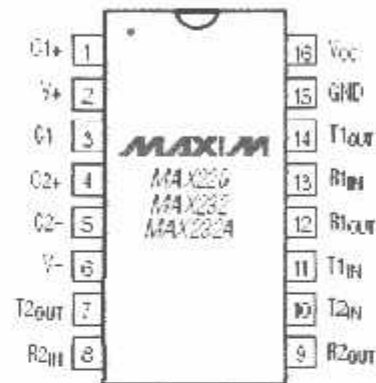
Sinyal-sinyal tersebut ada yang menuju ke DCE ada pula yang berasal dari DCE. Sinyal yang menuju ke DCE artinya DTE berfungsi sebagai output dan DCE berfungsi sebagai input, misalnya sinyal TD, pada sisi DTE kaki TD adalah output dan kaki ini dihubungkan ke kaki TD pada DCE yang berfungsi sebagai input. Kebalikan sinyal TD adalah RD, sinyal ini berasal dari DCE dan dihubungkan ke kaki RD pada DTE yang berfungsi sebagai input.

<sup>26)</sup> [www.elektroindonesia.com](http://www.elektroindonesia.com), 2005

<sup>27)</sup> Pelita 2000, Sistem Mikrokontroler

## 2.8. RS MAX232

RS MAX232 tersusun dari 2 bagian yaitu : RS232 line driver yang berfungsi merubah level tegangan TTL ke level tegangan RS232 dan RS232 line receiver yang berfungsi untuk merubah level tegangan RS232 menjadi level tegangan TTL. Gambar IC MAX 232 adalah sebagai berikut :



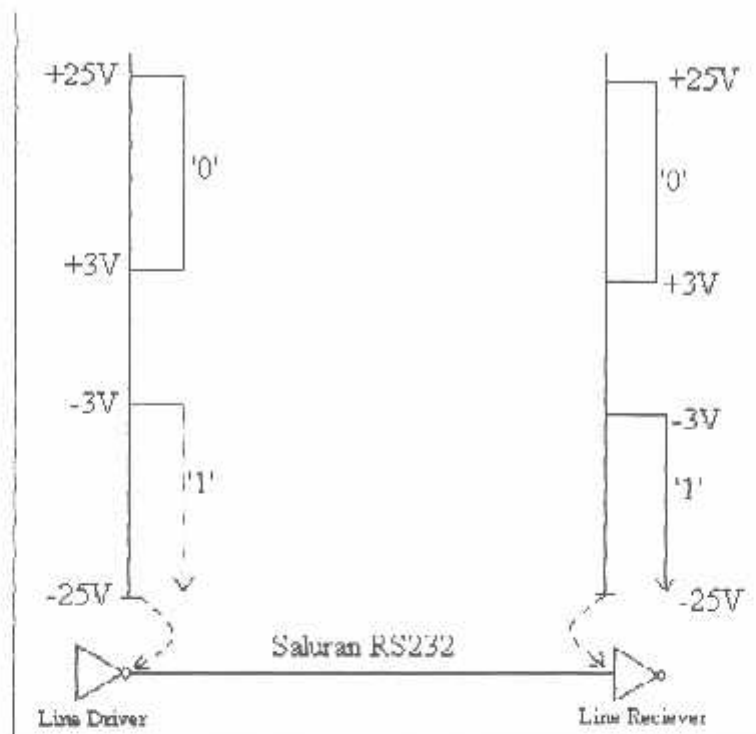
**Gambar 2-12. IC MAX232** <sup>28)</sup>

Standart RS232 ditetapkan oleh electronic industry association and telecommunication industry association pada tahun 1962. Standart ini hanya menyangkut komunikasi data antara computer (Data Terminal Equipment – DTE) dengan alat-alat pelengkap computer maupun alat-alat digital lainnya (Data Circuit Equipment – DCE).

Ada 3 hal pokok yang diatur standart RS232, antara lain adalah :

1. Bentuk sinyal dan level tegangan yang dipakai

<sup>28)</sup>Datasheet MAX232, [www.alldatasheet.com](http://www.alldatasheet.com), Halaman 17



**Gambar 2-13. Level Tegangan RS232**<sup>29)</sup>

Level tegangan signal RS232 adalah seperti terlihat pada gambar diatas. Dalam standart RS232, tegangan antara +3 sampai +25 Volt baik pada input line receiver maupun pada output line driver dianggap sebagai level tegangan '0', dan tegangan antara -3 sampai -25 V dianggap sebagai level tegangan '1'.

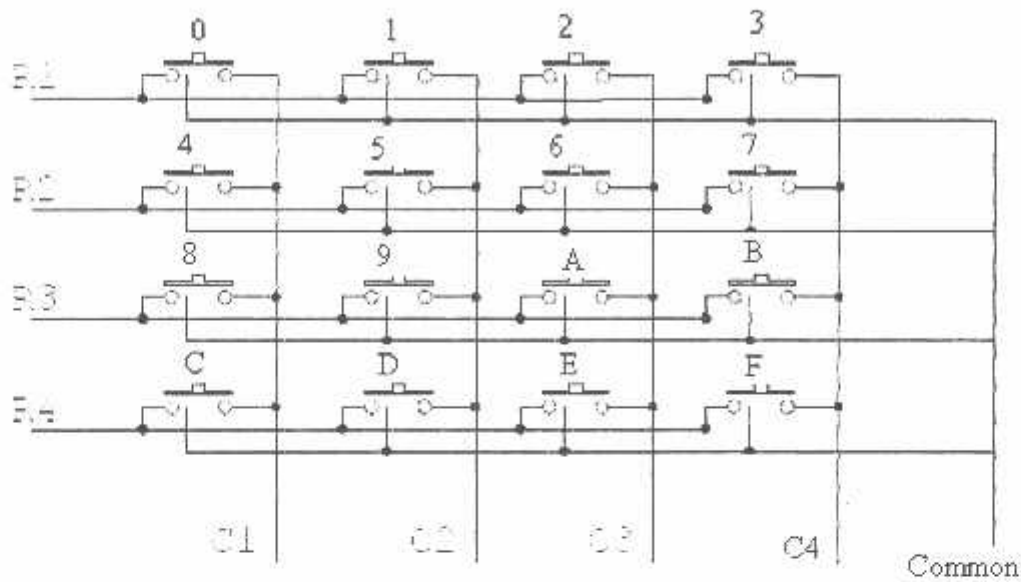
2. Penentuan jenis sinyal dan konektor yang dipakai, serta susunan sinyal pada kaki-kaki di konektor. Jenis-jenis sinyal yang dipakai mengatur pertukaran informasi antara DTE dan DCE semuanya terdapat 24 jenis sinyal, tapi yang umum dipakai adalah 9 jenis sinyal.
3. Penentuan tata cara pertukaran informasi antara computer dan alat-alat pelengkapya maupun alat-alat digital.

Sedangkan arus yang dibutuhkan rata-rata sebesar 5mA.

<sup>29)</sup>Ibid, Halaman 20

## 2.9. Keypad

Keypad 4x4 di sini adalah sebuah keypad matrix dengan susunan empat baris dan empat kolom dengan sebuah common



**Gambar 2-14. Keypad 4x4<sup>30)</sup>**

Seperti terlihat dalam gambar di atas, apabila saklar '0' ditekan, maka baris 1 dan kolom 1 akan terhubung ke common. Apabila saklar '1' ditekan, maka baris 1 dan kolom 2 akan terhubung ke common dan seterusnya.

Kondisi tidak adanya penekanan tombol diatur dengan adanya kondisi logika high dengan menghubungkan semua pin keypad (kecuali common) ke VCC melalui resistor pull up. Pada saat tombol tidak ditekan, maka arus akan mengalir dari VCC melalui resistor menuju ke port seperti tampak pada gambar berikut.

<sup>30)</sup>Delta Electronic, [www.delta-electronic.com](http://www.delta-electronic.com)



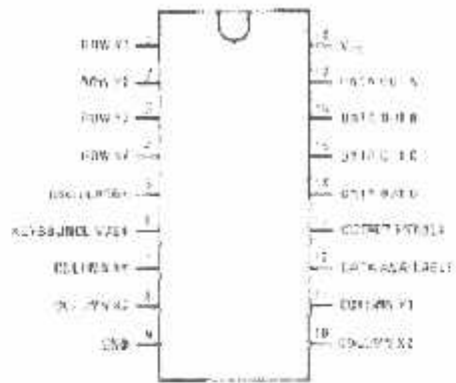
**Gambar 2-15. Aliran Arus Saat Tombol Tidak Ditekan**<sup>30)</sup>

Sedangkan saat tombol ditekan, maka baris dan kolom akan terhubung ke ground sehingga kondisi pada baris dan kolom tersebut akan menjadi low. Apabila tombol '0' ditekan, maka baris 1 dan kolom 1 akan terhubung ke ground sehingga kondisi baris dan kolom tersebut akan berubah menjadi low, demikian pula pada tombol '2' dan seterusnya. Pengambilan data dari keypad dilakukan dengan menunggu adanya penekanan tombol keypad. Kondisi tidak ada penekanan tombol adalah high untuk semua pin keypad kecuali common yang terhubung ke ground.

## 2-10. IC 74C922 (Keypad Encoder)

IC 74C922 pada alat ini dihubungkan pada keypad 4x4. Bila tidak ada tombol yang ditekan maka logic pada data bus adalah 0, apabila ada penekanan tombol misalnya angka 1 maka logic pada data bus menjadi 0001. Dalam hal ini fungsi dari IC 74C922 adalah mengubah bilangan desimal menjadi bilangan biner. Berikut adalah gambar dari IC 74C922 beserta tabel kebenarannya.

<sup>30)</sup>Delta Electronic, [www.delta-electronic.com](http://www.delta-electronic.com)



Gambar 2-16. IC 74C922 <sup>32)</sup>

Tabel 2-12. Tabel Kebenaran IC 74C922 <sup>33)</sup>

Switch Position	0	1	2	3	4	5	6	7	8	9	10	11
	Y1, X1	Y1, X2	Y1, X3	Y1, X4	Y2, X1	Y2, X2	Y2, X3	Y2, X4	Y3, X1	Y3, X2	Y3, X3	Y3, X4
E												
A	0	1	0	1	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1	0	0	1	1
C	0	0	0	0	1	1	1	1	0	0	0	0
D	0	0	0	1	0	0	0	0	1	1	1	1
E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0

(Rows 12 through 19)

Switch Position	12	13	14	15	16	17	18	19
	Y4, X1	Y4, X2	Y4, X3	Y4, X4	Y5 (Note 1), X1	Y5 (Note 1), X2	Y5 (Note 1), X3	Y5 (Note 1), X4
D								
A	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1
C	1	1	1	1	0	0	0	0
D	1	1	1	1	0	0	0	0
E (Note 1)	0	0	0	0	1	1	1	1

NOTE 1: 0 = 0V, 1 = VCC (74C922)

<sup>32)</sup>Datasheet IC 74C922, [www.fairchildsemi.com](http://www.fairchildsemi.com), Halaman 1

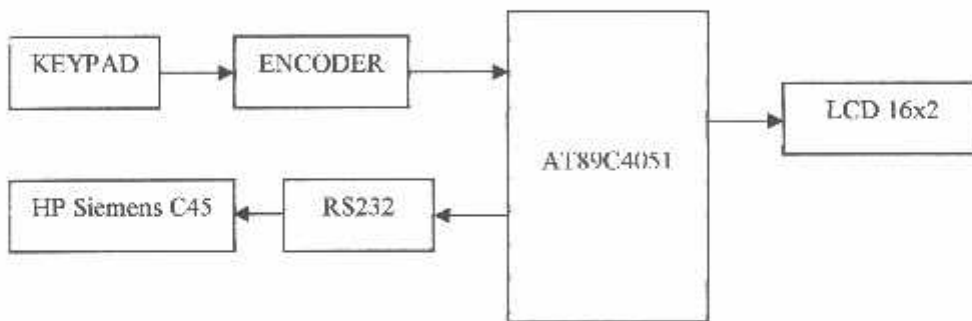
<sup>33)</sup>Datasheet IC 74C922, [www.fairchildsemi.com](http://www.fairchildsemi.com), Halaman 2

## BAB III

### PERENCANAAN DAN PEMBUTAN ALAT

#### 3.1. Diagram Blok

Di bawah ini dapat dilihat diagram blok keseluruhan dari alat yang akan dirancang.



Gambar 3-1. Diagram Blok Sistem

Fungsi dari masing-masing blok rangkaian sistem adalah sebagai berikut :

- Handphone berfungsi untuk melakukan proses isi ulang voucher.
- Keypad berfungsi untuk memasukkan nilai nominal voucher yang ingin diisi dan nomor handphone.
- Encoder berfungsi untuk penterjemah dari inputan keypad agar inputan dari keypad bisa diterima oleh mikrokontroller.
- RS232 berfungsi sebagai perantara komunikasi antara handphone dengan mikrokontroller.



- Mikrokontroler berfungsi sebagai pengendali keseluruhan system. Mikrokontroler akan memproses input yang telah diberikan dan mengendalikan output yang diinginkan.
- LCD berfungsi untuk menampilkan hasil dari eksekusi yang telah dilakukan mikrokontroler.

### **3.2. Prinsip Kerja Alat**

Pertama kali setelah power supply dihidupkan, sistem akan mendeteksi ada tidaknya sambungan / hubungan dengan pesawat HP melalui komunikasi serial RS 232. Apabila tidak ada sambungan, proses tidak akan dilanjutkan sampai pesawat HP telah diaktifkan dan terhubung dengan sistem.

Setelah terjadi sambungan antara sistem dan HP, barulah proses dapat dilakukan yang dimulai dengan memasukkan data-data yang dibutuhkan melalui tombol keypad. Urut-urutan data yang akan dimasukkan sesuai dengan instruksi yang ditampilkan di layar LCD seperti nama operator seluler, nominal voucher, nomor telepon yang akan diisi.

Setelah semua data yang diminta diisi, sistem akan meneruskan ke HP untuk dikirim ke operator. Jawaban yang diterima dari operator akan ditampilkan di layar HP.

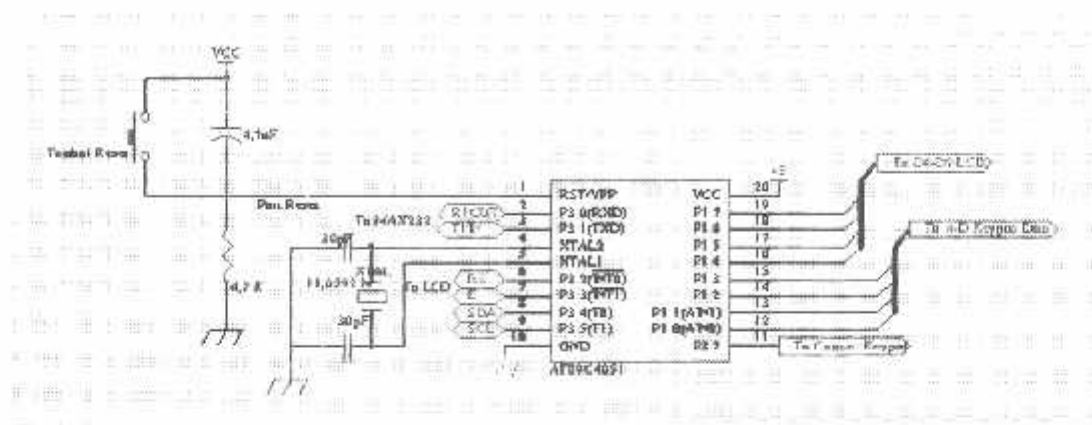
### **3.3. Perancangan Perangkat Keras ( Hardware )**

#### **3.3.1. Rangkaian Mikrokontroler**

Mikrokontroler AT89C4051 adalah mikrokontroler keluaran ATMEL, masih dalam keluarga MCS-51. IC ini mempunyai 20 pin masing-masing terbagi

---

menjadi port 1 dan port 3. Kaki-kaki yang digunakan dalam perencanaan alat pengisi voucher elektrik ini adalah sebagai berikut :

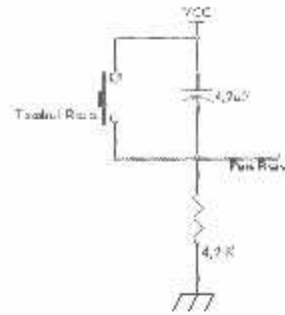


**Gambar 3-2. Mikrokontroller AT89C4051**

- P1.0 – P1.3 = Data Keypad (A-D)
- P1.4 – P1.7 = Data LCD (D4-D7)
- P3.0 dan P3.1 = RXD dan TXD IC L232
- P3.2 = Pin RS LCD
- P3.5 = Pin E LCD
- P3.7 = Pin Control Keypad
- X1 dan X2 = Kristal oscillator

### 3.3.2. Rangkaian Reset Dan Oscillator

Rangkaian reset berfungsi untuk mereset mikrokontroller ke keadaan awal baik pada saat power dihidupkan maupun reset manual tanpa harus mematikan power supply seperti gambar :



**Gambar 3-3. Rangkaian Reset**

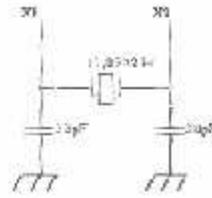
Pin AT89C4051 reset aktif high, artinya bila pin reset berlogika high, chip akan reset dan kembali ke keadaan awal. Pada saat power dijalankan, kapasitor 4,7uF akan mengisi dan membuang muatannya melalui tahanan 4,7KΩ. Proses pengisian dan pengosongan muatan sesaat ini akan mereset chip untuk kembali ke keadaan awal, dan ini disebut Rangkaian power on reset. Bila sewaktu-waktu digunakan untuk mereset chip, digunakan tombol push button (PB) yang akan memberikan logika high pada pin reset mikrokontroller AT89C4051.

Untuk Sumber Clock digunakan kristal 11,0592 MHz dan kapasitor 30 PF. Pemilihan kristal 11,0592 MHz ini didapatkan untuk mendapatkan baudrate pada komunikasi serial sesuai dengan rumus :

$$Baudrate = \frac{KxFosc}{32 \times 12(256 - TH1)}$$

$$TH1 = 256 - \frac{KxFosc}{384 \times Baudrate}$$

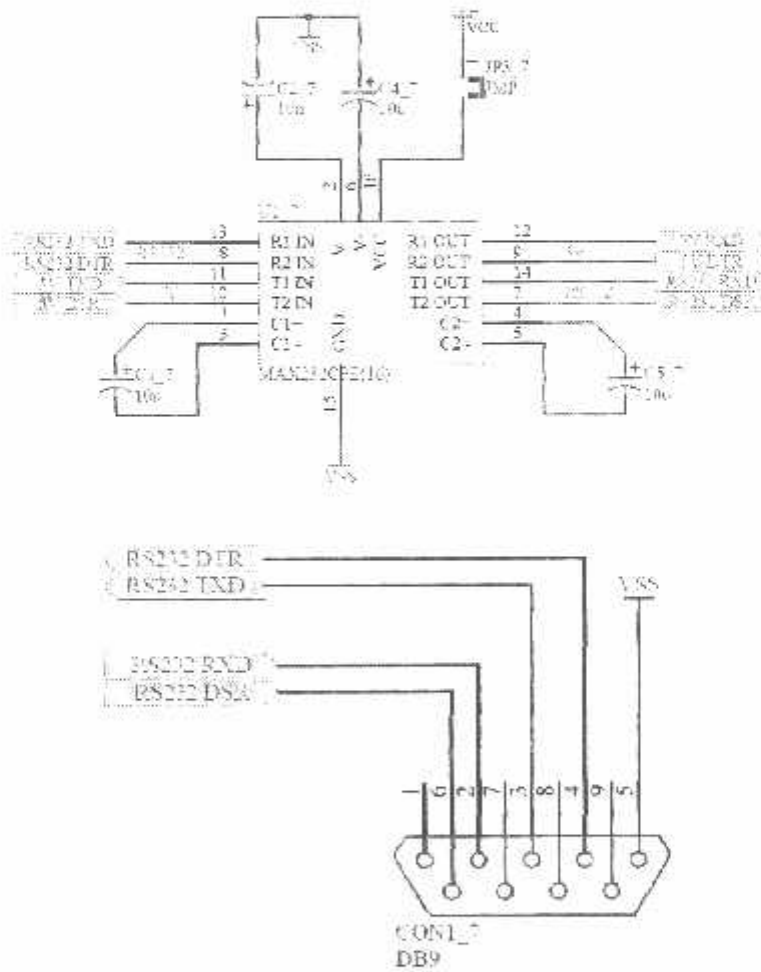
Rangkaian Oscilator seperti gambar berikut :



Gambar 3-4. Rangkaian Oscillator

### 3.3.3. Rangkaian Komunikasi Serial RS 232

Untuk melakukan komunikasi antara mikrokontroller dan HP digunakan komunikasi serial RS232 melalui kabel data Siemens C45 yang telah tersedia. IC yang digunakan adalah IC L232 dari maxim seperti gambar berikut :

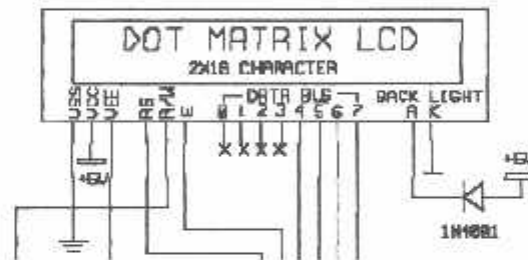


Gambar 3-5. Rangkaian Komunikasi Serial RS 232

Pin R1in (13) berhubungan dengan pin 2 DB9, pin T1out (14) dengan pin 3 DB9. Pada mikrokontroller, pin 3.0 (RXD) dari mikrokontroller berhubungan dengan pin 12 IC L232 dan pin 3.1 (TXD) dari mikrokontroller dengan T1in (11) L232. Sambungan ground terletak pada pin 5 DB9 yang berhubungan dengan ground rangkaian.

### 3.3.4. Rangkaian Penampil LCD 2x16

Sebagai tampilan dari seluruh proses yang dilakukan digunakan LCD 2x16 dot matrik tipe M162A.



**Gambar 3-6. Rangkaian Penampil LCD**

Karena keterbatasan port, maka LCD digunakan dalam mode 4 bit. Oleh karena itu sambungan pin-pin LCD adalah sebagai berikut :

Pin RS = Pin control to mikrokontroller

Pin RW = Pin Control to ground

Pin Enable = Pin Control to mikrokontroller

D0 – D3 – Pin data, not connect

D4 – D7 = Pin data to mikrokontroller

Tahanan variabel 10K berfungsi untuk mengatur kecerahan tampilan LCD sesuai keinginan, sedangkan dioda IN4002 berfungsi untuk menurunkan tegangan VCC 5 Volt menjadi 4,2 Volt sesuai dengan tegangan yang dibutuhkan pin V+ (15).

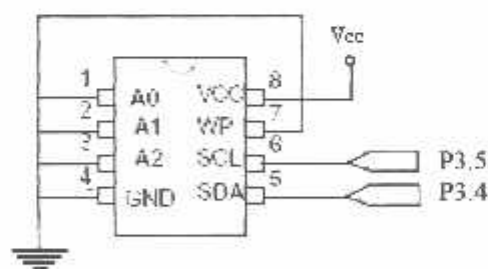
### 3.3.4. Perancangan Rangkaian Serial EEPROM AT24C64

Serial EEPROM AT24C64 digunakan sebagai penyimpan data yang bersifat Non-Volatile artinya data tidak akan hilang meskipun catu daya dihilangkan, keunggulan serial EEPROM ini adalah:

- Harga serial EEPROM lebih murah dibandingkan dengan EEPROM parallel
- Karena dioperasikan secara serial maka hanya butuh 2 Pin untuk mengaksesnya ini akan menghemat Port dimikrokontroller.
- Karena jenis memori EEPROM, data yang disimpan tetap Valid meskipun tidak ada supply tegangan yang diberikan.

Untuk mengakses serial AT24C64, yaitu Pin SDA & Pin SCL yang dipakai, A0,A1, dan A2 dikoneksi ke GND, karena memang untuk serial EEPROM AT24C64 (pin-pin ini tidak digunakan atau no connections).

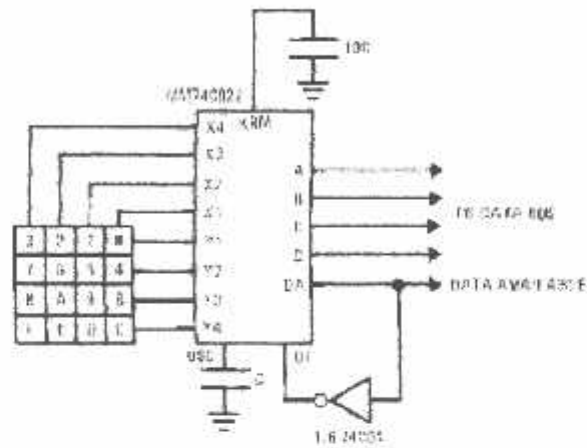
Pin WP dikoneksikan ke GND karena akan mengakses serial EEPROM secara full operation ( bisa baca/tulis), jika WP dikoneksikan ke Vcc (diberi logika high) maka kita hanya bisa baca atau read saja, tanpa bisa tulis ke EEPROM AT24C64.



**Gambar 3.5 Rangkaian Serial EEPROM AT 24C64**

### 3.3.5. Rangkaian Keypad Dan Encoder

Sebagai inputan data digunakan keypad 4x4 dan rangkaian encoder yang berfungsi menterjemahkan tombol yang ditekan ke bilangan Hexadesimal. IC encoder yang digunakan adalah IC 74C922 seperti gambar berikut :

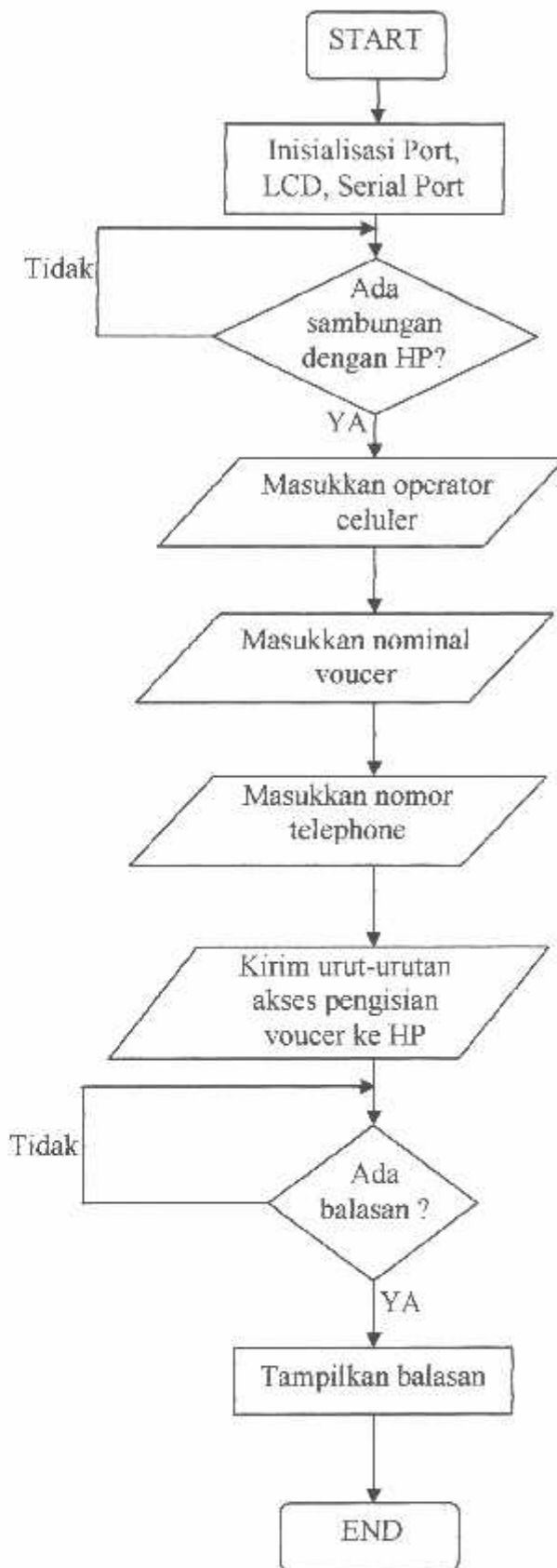


**Gambar 3-7. Rangkaian Keypad Dan Encoder**

Setiap kali tombol ditekan, encoder akan memberi informasi pada mikrokontroller melalui port control bahwa ada tombol yang ditekan dan akan menekan kode tombol tersebut pada port sampai ada tombol berikutnya yang ditekan.

### 3.4. Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak harus disesuaikan dengan perencanaan dan sistem kerja alat yang akan dibuat. Perancangan perangkat lunak dapat dilihat pada diagram alir berikut :



Gambar 3-8. Flochart Software



## BAB IV

### PENGUJIAN ALAT

#### 4.1 Tujuan

Bab ini membahas tentang pengujian dan analisis alat yang telah dibuat. Secara umum, pengujian ini bertujuan untuk mengetahui apakah alat yang telah direalisasikan dapat bekerja sesuai dengan spesifikasi perencanaan yang telah ditetapkan. Pengujian alat ini meliputi pengujian perangkat keras dan perangkat lunak. Pengujian dilakukan pada masing-masing blok terlebih dahulu, yang selanjutnya dilakukan pengujian untuk sistem secara keseluruhan. Adapun pengujian terhadap perangkat keras meliputi pengujian terhadap mikrokontroler, RS232, dan keypad.

#### 4.2 Pengujian Sistem Mikrokontroler.

- **Tujuan**

Untuk mengetahui kondisi awal dari mikrokontroler apakah sudah sesuai dengan yang direncanakan

- **Prosedur Pengujian**

1. Membuat program yang digunakan dalam pengujian mikrokontroler.

Program yang digunakan dalam pengujian mikrokontroler ini merupakan program sederhana yang meletakkan  $0F_H$  dan  $F0_H$  pada ACC secara bergantian kemudian memindahkannya pada *Port 1* AT89C4051. Program yang dibuat adalah sebagai berikut :

```
ORG 0000H  
  
JMP START
```

```

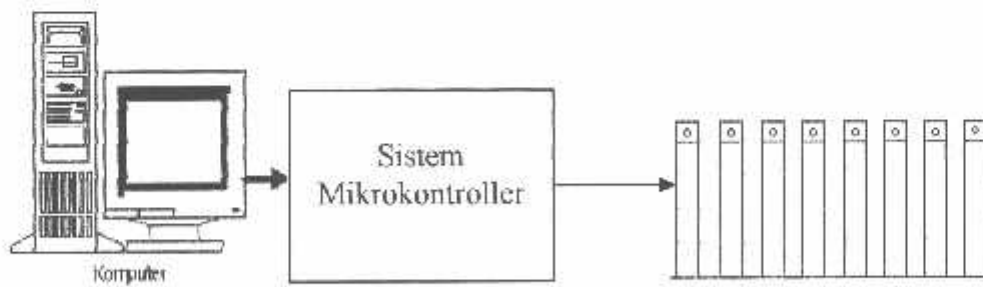
START:  MOV   A,#0FH
        MOV   P1,A
        CALL  TUNDA
        MOV   A,#F0H
        MOV   P1,A
        JMP   START

TUNDA:  MOV   R3,#0FFH

TUNDA1: MOV   R2,#0FFH
        DJNZ  R2,S
        MOV   R1,#0FH
        DJNZ  R1,S
        DJNZ  R3,TUNDA1
        RET
        END

```

2. Rangkaian dibuat seperti Gambar 4-1.
3. Memasang catu daya rangkaian sebesar 5 Volt DC
4. Download program diatas .
5. Mengamati keluaran pada LED Display .



**Gambar 4-1 Diagram blok Pengujian Mikrokontroler**

- **Hasil Pengujian**

Hasil pengujian pada sistem mikrokontroller ditunjukkan dalam Tabel 4.1 dibawah ini :

**Tabel 4.1 Hasil Pengujian Sistem Mikrokontroler**

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

- **Analisis Pengujian**

Dari hasil pengujian dalam tabel 4-1 dapat dilihat bahwa *port 1* memberikan logika 0F<sub>H</sub> dan F0<sub>H</sub> secara bergantian sesuai dengan isi program.

### 4.3 Pengujian Komunikasi Serial

- **Tujuan**

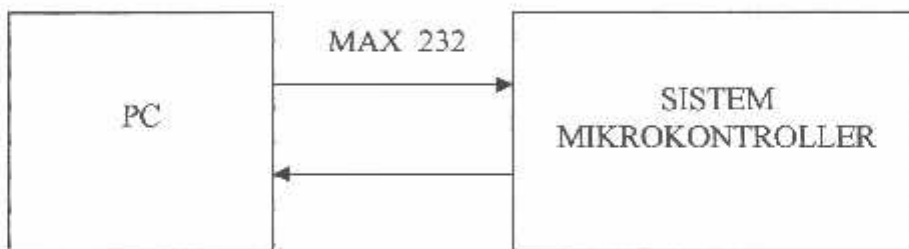
Untuk mengetahui apakah data yang dikirim dari MCU ke *handphone* dapat diterima dengan benar dengan melakukan simulasi pada komputer.

- **Peralatan yang digunakan**

- o Komputer
  - o Sistem mikrokontroller dengan antarmuka RS232.
  - o EPROM Emulator EL-TECH Model EE-02.
-

- **Prosedur pengujian**

1. Menyusun rangkaian seperti pada Gambar 4-2.
2. Membuat program transfer data pada sistem mikrokontroler seperti yang ditunjukkan dalam Gambar 4-3 dan Gambar 4-4. Dengan program tersebut komputer mengirim data '1234567890' ke alat dan oleh alat akan dikembalikan lagi ke komputer.
3. Mencatat hasil yang terlihat dalam layar komputer.



**Gambar 4-2 Rangkaian Pengujian Transfer Data**

---

```

B2400 EQU 232
PCON EQU 87H
ORG 0000H
JMP MULAI
ORG 0023H
SERIAL: JBC TI, OUT232
        PUSH ACC
        MOV A, SBUF
        MOV SBUF, A
        POP ACC
OUT232: CLR TI
        CLR RI
        RETI
INIT232: MOV TH1, #B1200
        MOV TMOD, #22H
        ANL PCON, #7FH
        MOV SCON, #50H
        SETB PS
        SETB ES
        SETB TR1
        SETB RA
        CALL DELAY

```

**Gambar 4-3 Program Uji Komunikasi Data di Mikrokontroler**

```

* Cuplikan program uji komunikasi data di komputer menggunakan program
VB 6.0

MSComm1.Output = '1234567890'

Text1.Text = MSComm1.Input

```

**Gambar 4-4 Program Uji Komunikasi Data di Komputer**

- **Hasil Pengujian**

Hasil pengujian transfer data serial ini ditunjukkan Tabel 4.2 dibawah ini :

**Tabel 4.2 Hasil Pengujian Transfer Data Serial**

<b>Data yang dikirim komputer</b>	<b>Data yang diterima komputer</b>
1234567890	1234567890

- **Analisis Hasil Pengujian**

Hasil pengujian dalam Tabel 4.2 menunjukkan bahwa proses pengiriman data serial dengan menggunakan RS 232 ke alat telah benar.

#### 4.4 Pengujian Rangkaian Keypad

- **Tujuan**

Untuk mengetahui apakah rangkaian ini dapat mengkodekan tombol-tombol input keypad yang ditekan menjadi data 4 bit yang bersesuaian

- **Peralatan yang digunakan**

- o Papan Tombol (keypad 4x4)
- o Rangkaian pengkode keypad
- o Peraga led 5 bit
- o Catu daya 5 Volt DC

- **Prosedur Pengujian**

1. Peralatan dirangkai seperti pada gambar 4-6
2. Menekan tombol pada papan tombol, mengamati dan mencatat keluaran yang ditampilkan ke peraga led 5 bit



**Gambar 4-5. Diagram blok pengujian pengkode papan tombol (keypad)**

- **Hasil pengujian dan analisis**

Hasil pengujian dan pengamatan rangkaian keypad ditunjukkan dalam tabel 4-3

**Tabel 4-3 Hasil pengujian pengkode papan tombol (keypad)**

Tombol	L <sub>4</sub>	L <sub>3</sub>	L <sub>2</sub>	L <sub>1</sub>	L <sub>0</sub>
0	1	0	0	0	0
1	1	0	0	0	1
2	1	0	0	1	0
3	1	0	0	1	1
4	1	0	1	0	0
5	1	0	1	0	1
6	1	0	1	1	0
7	1	0	1	1	1
8	1	1	0	0	0
9	1	1	0	0	1
A	1	1	0	1	0
B	1	1	0	1	1
C	1	1	1	0	0

<b>D</b>	1	1	1	0	1
<b>E</b>	1	1	1	1	0
<b>F</b>	1	1	1	1	1

Berdasarkan hasil pengujian dapat dilihat bahwa  $L_4$  yang dihubungkan ke pin *data available* berlogika high (ON) jika ada tombol ditekan, sedangkan  $L_3-L_0$  menunjukkan data keluaran yang dihasilkan. Dengan demikian maka rangkaian pengkode papan tombol dapat bekerja dengan baik sesuai perencanaan.

#### 4.5. Pengujian Sistem Secara Keseluruhan

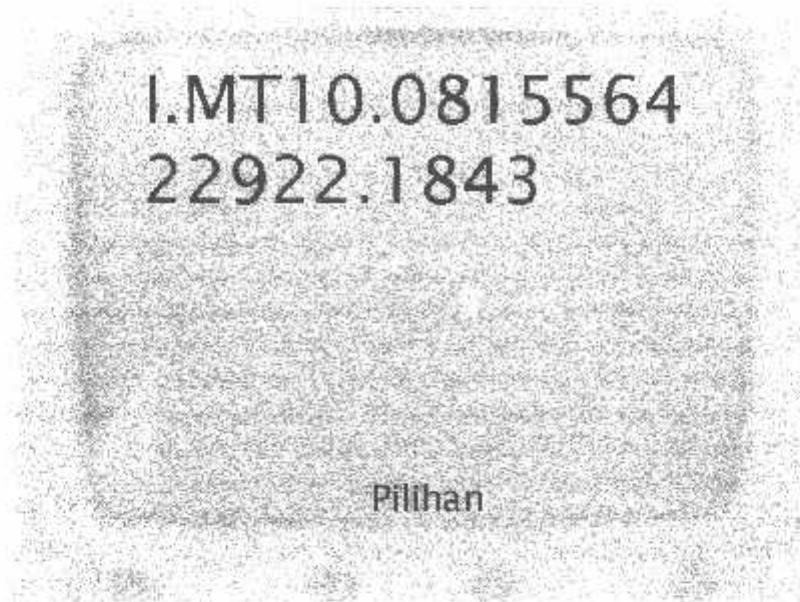
- **Petunjuk Pengoperasian alat**

Sebelum pengoperasian alat dilakukan, harus dipastikan bahwa handphone telah terhubung dengan alat lewat kabel data.

- **Pengujian Untuk Kode Pengisian Pulsa**

Kali ini saya mencoba untuk mengisi pulsa misalnya metari 10. Kode smsnya adalah : LMT10.NOHP.1843. Misalnya kita mengisi pulsa mentari Rp. 10.000 ke no 081556422922 maka tampilan pada HP secara lengkap adalah seperti gambar di bawah :





**Gambar 4-6. Tampilan Kode Sms Untuk Isi Pulsa Mentari 10.000**

Sedangkan balasan yang diterima apabila telah sukses adalah sebagai berikut :



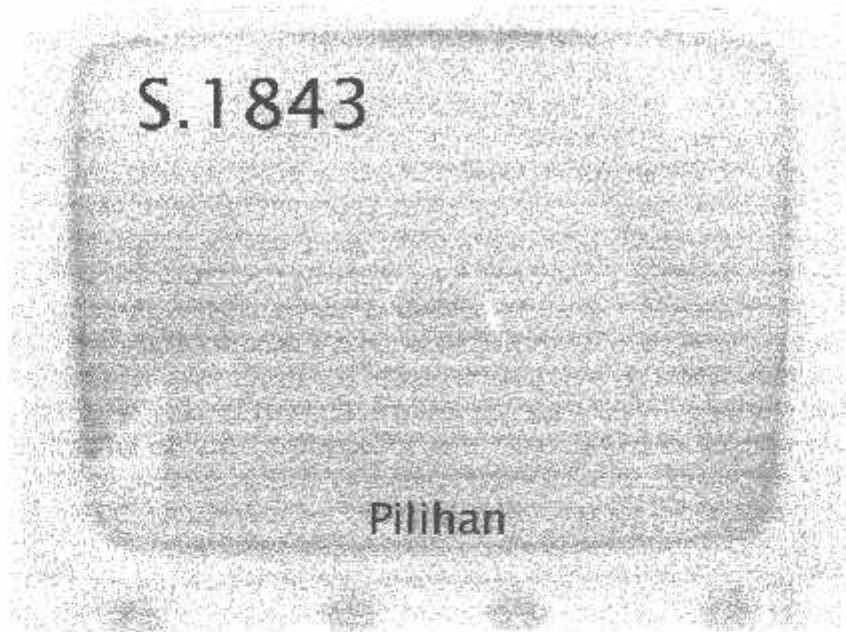
**Gambar 4-7. Laporan Sukses Pengisian Pulsa**

Laporan sukses ini berlaku untuk semua transaksi dan nominal voucher yang tersedia.

---

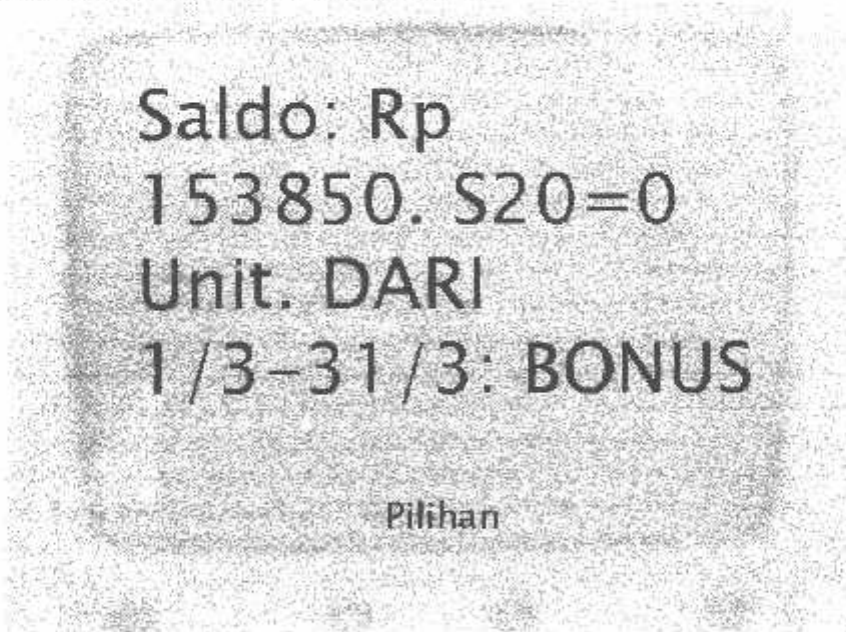
- **Pengujian Untuk Pengecekan saldo**

Kode SMS untuk pengecekan saldo adalah S.1843. Hasil dari kode sms tersebut dapat kita lihat pada gambar di bawah ini :



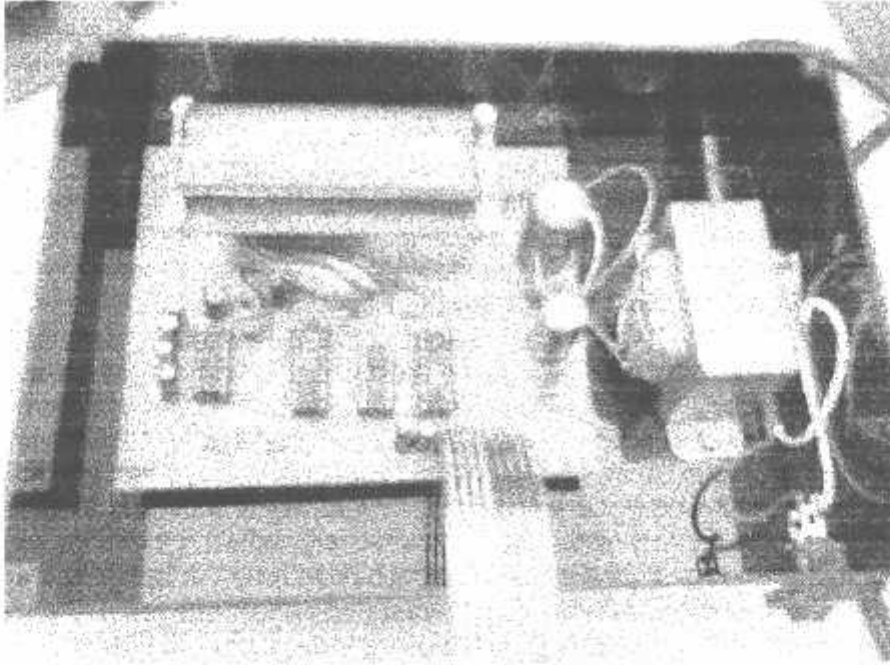
**Gambar 4-8. Kode Sms Untuk Cek Saldo**

Sedangkan balasan yang diterima adalah :



**Gambar 4-9. Laporan Dari Cek Saldo**

Foto Alat :



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan pada teori-teori yang telah dibahas pada makalah seminar hasil ini, maka dapat ditarik kesimpulan sebagai berikut :

1. Dengan adanya alat ini, maka kesalahan untuk pengisian voucer elektrik di pihak penjual dapat diminimalisir karena pembeli sendiri yang memasukkan nomor telephon.
2. Alat ini berfungsi untuk menyimpan kode-kode pengisian pulsa sehingga bila ingin mengisi voucher elektrik tidak usah mengetikkan kode-kode SMS.
3. Alat ini menggunakan komunikasi serial RS232 antara handphone dengan mikrokontroller karena RS232 adalah komunikasi serial standart Handphone yang bisa diakses oleh mikrokontroller.

#### **5.2 Saran**

1. Untuk pengembangan alat ini selanjutnya diharapkan mampu untuk menyimpan database tiap-tiap pelanggan yang telah mengisi voucer.
2. Agar fasilitas alat ini lebih sempurna, dapat ditambahkan menu untuk rckap harian pengisian voucer untuk pelanggan yang membeli voucer.

**DAFTAR PUSTAKA**

1. *Data Sheet AT89C4051* [www.Atmel.com](http://www.Atmel.com)
  2. Mohammad Ibnu Malik & Anistardi, 1997, *Bereksperimen dengan Mikrokontroller 8031*, Pt Elex Media Komputindo Jakarta
  3. Paulus Andi Nalwan, *Teknik Antarmuka Dan Pemrograman Mikrokontroller AT89C4051*, PT Elex Media Komputindo, Jakarta 2003
  4. *Training Mikrokontroller AT89C4051, Programming dan Interfacing*, Winstrich Engineering Services
  5. [www.alldatashcet.com](http://www.alldatashcet.com)
-

# LAMPIRAN



## FORMULIR BIMBINGAN SKRIPSI

Nama : Indra Sanjaya Priyanto  
Nim : 0117110  
Masa Bimbingan : 26-Jan-2006 s/d 26-Jul-2006  
Judul Skripsi : Perencanaan dan pembuatan alat isi ulang voucher elektrik dengan Handphone yang diinterfaccekan pada Mikrokontroller AT89C4051

NO	Tanggal	Uraian	Paraf Pembimbing
1.	30-01-06	Revisi Bab I kewajiban	
2.	02-02-06	Revisi BAB II Teori uang Langkah	
3.	05-02-06	Revisi Gambar Angkasan	
4.	10-02-06	ACC Bab I & II	
5.	17-02-06	Revisi BAB III	
6.	18-02-06	ACC BAB III & Makalah Lembar	
7.	02-3-06	ACC BAB IV & V	
8.	14-3-2006	Revisi KOMPRES	
9.			
10.			

Malang, 15-3-2006  
Dosen Pembimbing

Joseph Dedy Irawan, ST, MT

Form. S-4a



### Nilai Ujian Skripsi

Nama Mahasiswa : INDRA SANJAYA / Nim : 0117110  
Fakultas / Jurusan : Teknologi Industri / Teknik Elektro  
Konsentrasi : T. Energi Listrik / T. Elektronika S-1

No	Indikator Penilaian	Nilai
1.	<b>Penguasaan materi skripsi :</b> <ul style="list-style-type: none"><li>• Ketajaman perumusan masalah dan tujuan penelitian</li><li>• Ketepatan metode yang digunakan</li><li>• Ketepatan penarikan kesimpulan dengan tujuan dan hasil penelitian</li></ul>	
2.	<b>Penguasaan materi penunjang Skripsi / tinjauan pustaka :</b> <ul style="list-style-type: none"><li>• Relevansi, kemuktakhiran dan penyusunan daftar pustaka</li></ul>	
3.	<b>Kontribusi hasil penelitian</b> <ul style="list-style-type: none"><li>• Manfaat hasil penelitian bagi pengembangan IPTEKS, pembangunan dan atau pengembangan kelembagaan</li></ul>	
	Nilai rata-rata	

**Catatan :**

1. Nilai diberikan dalam bentuk angka dengan kisaran : 0 sampai dengan 100
2. Nilai kumulatif dari penguji kurang dari 56 peserta ujian dinyatakan tidak lulus

Moderator / Dosen Pembimbing

Malang, \_\_\_\_\_

Dosen Penguji





### Formulir Perbaikan Ujian Skripsi

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

NAMA

N I M

Perbaikan meliputi

:  
 :  
 :

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Malang,

( \_\_\_\_\_ )



### Nilai Ujian Skripsi

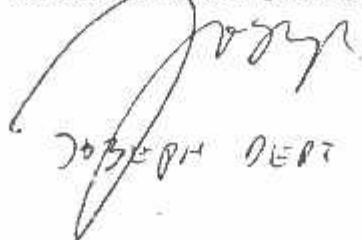
Nama Mahasiswa : INDRA SANJAYA / Nim : 01.17.110  
Fakultas / Jurusan : Teknologi Industri / Teknik Elektro  
Konsentrasi : T. Energi Listrik / T. Elektronika S-1

No	Indikator Penilaian	Nilai
1.	<b>Penguasaan materi skripsi :</b> <ul style="list-style-type: none"><li>• Ketajaman perumusan masalah dan tujuan penelitian</li><li>• Ketepatan metode yang digunakan</li><li>• Ketepatan penarikan kesimpulan dengan tujuan dan hasil penelitian</li></ul>	80
2.	<b>Penguasaan materi penunjang Skripsi / tinjauan pustaka :</b> <ul style="list-style-type: none"><li>• Relevansi, kemuktakhiran dan penyusunan daftar pustaka</li></ul>	80
3.	<b>Kontribusi hasil penelitian</b> <ul style="list-style-type: none"><li>• Manfaat hasil penelitian bagi pengembangan IPTEKS, pembangunan dan atau pengembangan kelembagaan</li></ul>	76
	Nilai rata-rata	

**Catatan :**

1. Nilai diberikan dalam bentuk angka dengan kisaran : 0 sampai dengan 100
2. Nilai kumulatif dari penguji kurang dari 56 peserta ujian dinyatakan tidak lulus

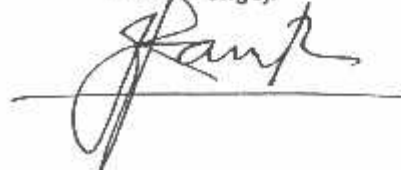
Moderator / Dosen Pembimbing

  
JOSEPH DEPT

Malang,

IR. M. L. Sudharmo

Dosen Penguji



IP - 2006.  
Maret



INSTITUT TEKNOLOGI NASIONAL  
FAKULTAS TEKNOLOGI INDUSTRI  
JURUSAN TEKNIK ELEKTRO

### Formulir Perbaikan Ujian Skripsi

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

NAMA :  
N I M :  
Perbaikan meliputi :

Lined area for writing the details of the thesis correction.

Malang,

( \_\_\_\_\_ )

## Features

- Compatible with MCS-51™ Products
- 4K Bytes of Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- 5 to 6V Operating Range
- Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 8-Bit Internal RAM
- 15 Reprogrammable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Full-Duplex Programmable Serial UART Channel
- Two LED Drive Outputs
- On-Chip Analog Comparator
- Power Idle and Power Down Modes
- Watchdog Time-Out Detection

## Description

The AT89C4051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 4K Bytes of Flash programmable and erasable read only memory (PEROM). The AT89C4051 is manufactured using Atmel's high density nonvolatile memory technology compatible with the industry standard MCS-51™ instruction set. By combining the 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C4051 is a powerful microcomputer which provides a highly flexible and cost effective solution to embedded control applications.

The AT89C4051 provides the following standard features: 4K Bytes of Flash, 128 Bytes of Internal RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt structure, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C4051 is designed with static logic for operation down to zero frequency and supports two software-selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the program memory contents but freezes the oscillator disabling all other chip functions until the next reset.

## Pin Configuration

PDIP/SOIC

RST/VPP	1	20	VCC
(RXD) P3.0	2	19	P1.7
(TXD) P3.1	3	18	P1.6
XTAL2	4	17	P1.5
XTAL1	5	16	P1.4
(INT0) P3.2	6	15	P1.3
(INT1) P3.3	7	14	P1.2
(T0) P3.4	8	13	P1.1 (AIN1)
(T1) P3.5	9	12	P1.0 (AIN0)
GND	10	11	P3.7



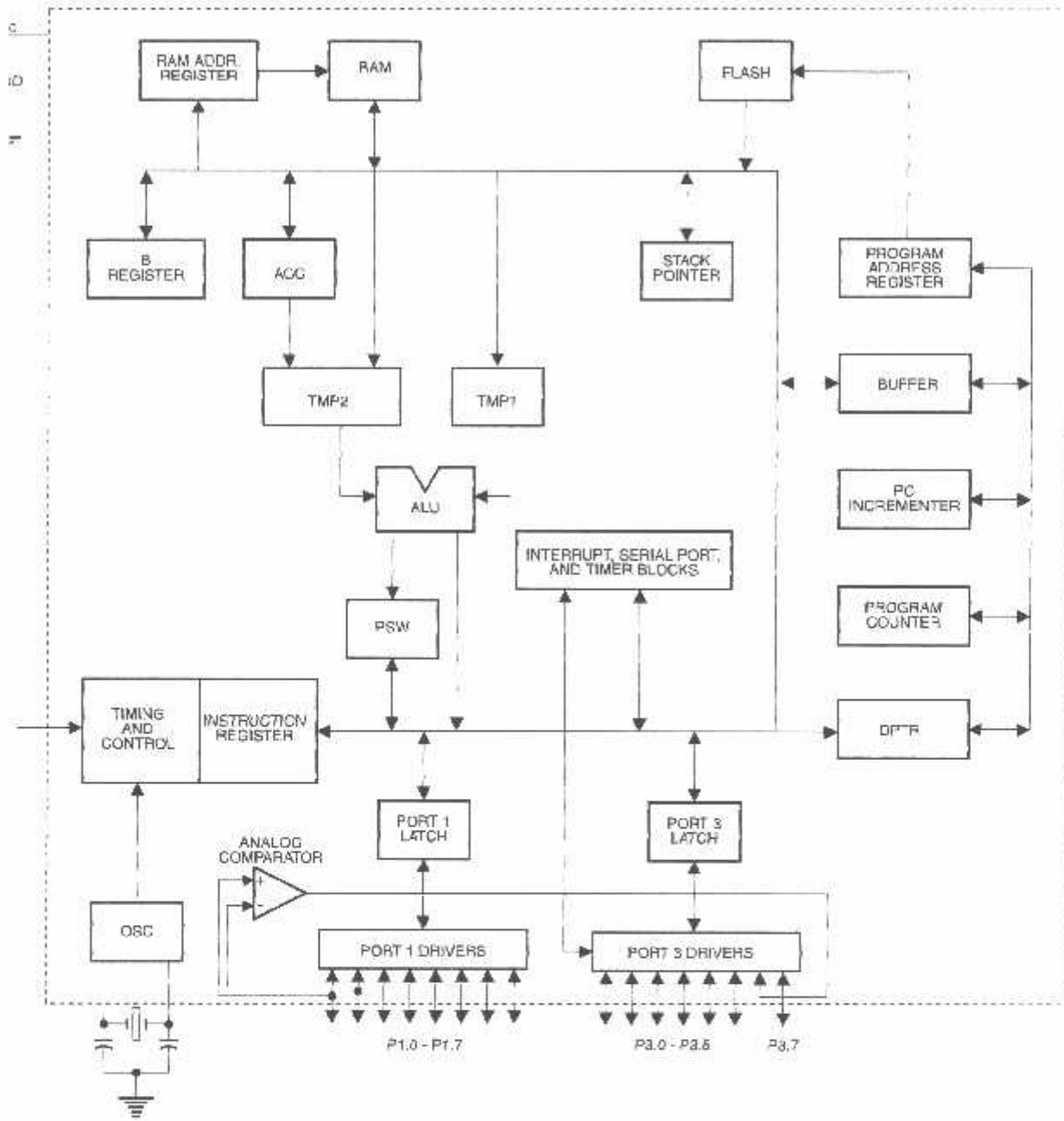
## 8-Bit Microcontroller with 4K Bytes Flash

## AT89C4051 Preliminary

Rev. 1001A-02/98



**k Diagram**



## Description

voltage.

1.

is an 8-bit bidirectional I/O port. Port pins P1.2 to provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as outputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) because of the internal pullups.

also receives code data during Flash programming and verification.

pins P3.0 to P3.5, P3.7 are seven bidirectional I/O ports with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 10 mA. When 1s are written to Port 3 pins they are high by the internal pullups and can be used as outputs. As inputs, Port 3 pins that are externally being pulled low will source current ( $I_{IL}$ ) because of the pullups.

also serves the functions of various special features of the AT89C4051 as listed below:

Pin	Alternate Functions
P1.0	RXD (serial input port)
P1.1	TXD (serial output port)
P1.2	$\overline{INT0}$ (external interrupt 0)
P1.3	INT1 (external interrupt 1)
P1.4	T0 (timer 0 external input)
P1.5	T1 (timer 1 external input)

also receives some control signals for Flash programming and verification.

input. All I/O pins are reset to 1s as soon as RST pin goes high. Holding the RST pin high for two machine cycles after the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

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

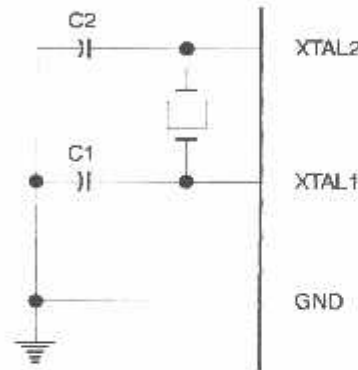
## XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

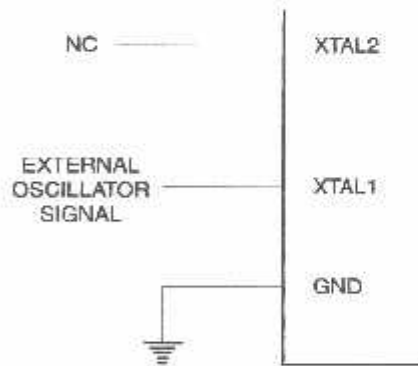
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration





## Special Function Registers

of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below:

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 0 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.

### I. AT89C4051 SFR Map and Reset Values

							0FFH
B 00000000							0F7H
							0EFH
ACC 00000000							0E7H
							0DFH
PSW 00000000							0D7H
							0CFH
							0C7H
IP XXX00000							0BFH
P3 11111111							0B7H
IE 0XX00000							0AFH
							0A7H
SCON 00000000	SBUF XXXXXXXX						9FH
P1 11111111							97H
TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		BFH
	SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	B7H

## Restrictions on Certain Instructions

The AT89C4051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 16K of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed with the MCS-51 instruction set. However, there are a few variations one must keep in mind when utilizing certain instructions to program this device.

Instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is the AT89C4051. This should be the responsibility of the software programmer. For example, LJMP 0FE0H would be a valid instruction for the AT89C4051 (with 4K of program memory), whereas LJMP 1000H would not.

### Branching instructions:

LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute only as long as the programmer keeps in mind that the destination branching address must fall within the physical program memory size (locations 00H to 0FFFH for the AT89C4051). Violating the physical space limits will cause unknown program behavior.

Instructions such as DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With conditional branching instructions the same rule applies. Again, violating the memory boundaries will cause erratic execution.

Situations involving interrupts the normal interrupt routine address locations of the 80C51 family architecture have been preserved.

### X-80C51-related instructions, Data Memory:

The AT89C4051 contains 128 bytes of internal data memory, in the AT89C4051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX instructions should be included in the program.

Even though the 80C51 assembler will still assemble instructions, they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to be aware of the physical features and limitations of the device and adjust the instructions used correspondingly.

## Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

### Lock Bit Protection Modes<sup>(1)</sup>

Program Lock Bits			Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

## Idle Mode

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

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle 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 is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

## Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V<sub>CC</sub> is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

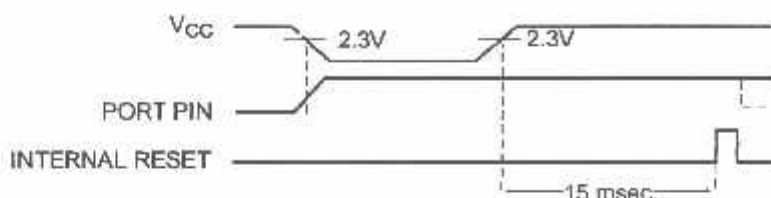




## Brown-Out Detection

$V_{CC}$  drops below the detection threshold, all port (except P1.0 and P1.1) are weakly pulled high. When

$V_{CC}$  goes back up again, an internal Reset is automatically generated after a delay of typically 15 msec. The nominal brown-out detection threshold is  $2.3V \pm 10\%$ .



## Programming The Flash

The AT89C4051 is shipped with the 4K bytes of on-chip PEROM code memory array in the erased state (i.e., conFFH) and ready to be programmed. The code memory is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the memory array needs to be erased electrically.*

**Internal Address Counter:** The AT89C4051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

**Programming Algorithm:** To program the AT89C4051, the following sequence is recommended.

**Power-up sequence:**

Apply power between  $V_{CC}$  and GND pins  
RST and XTAL1 to GND

pin RST to 'H'

pin P3.2 to 'H'

Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

**Program and Verify the Array:**

Apply data for Code byte at location 000H to P1.0 to P1.7.

Apply RST to 12V to enable programming.

Pulse pin P3.2 once to program a byte in the PEROM array. The lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.

To verify the programmed data, lower RST from 12V to 0V, set RST to 'H' level and set pins P3.3 to P3.7 to the appropriate logic levels. Output data can be read at the port P1 pins.

To program a byte at the next address location, pulse pin P3.2 once to advance the internal address counter. Apply new data to the port P1 pins.

Repeat steps 5 through 8, changing data and advancing the internal address counter for the entire 4K bytes array or until the end of the object file is reached.

**Power-off sequence:**

set XTAL1 to 'L'

set RST to 'L'

Turn  $V_{CC}$  power off

**Data Polling:** The AT89C4051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire PEROM array (4K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

**Verify the Signature Bytes:** The signature bytes are verified by the same procedure as a normal verification of memory addresses 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows:





- 00H) = 1EH indicates manufactured by Atmel
- 01H) = 41H indicates 89C4051

## Programming Interface

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

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

## 1 Programming Modes

		RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Code Data <sup>(1)(3)</sup>		12V		L	H	H	H
Code Data <sup>(1)</sup>		H	H	L	L	H	H
Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Erase		12V	 (2)	H	L	L	L
Signature Byte		H	H	L	L	L	L

1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
2. Chip Erase requires a 10-ms  $\overline{\text{PROG}}$  pulse.
3. P3.1 is pulled Low during programming to indicate RDY/ $\overline{\text{BSY}}$ .

### 3. Programming the Flash Memory

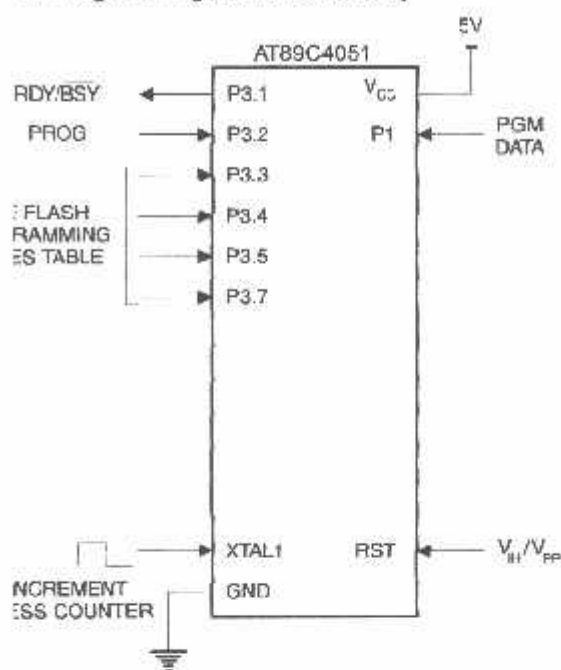
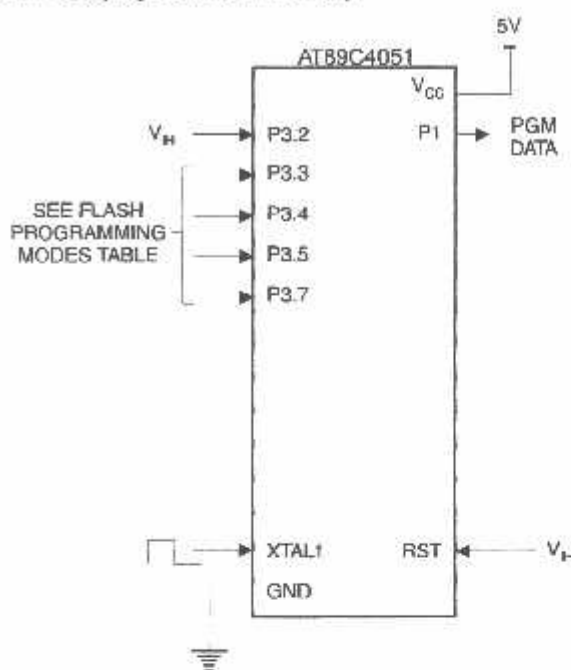


Figure 4. Verifying the Flash Memory



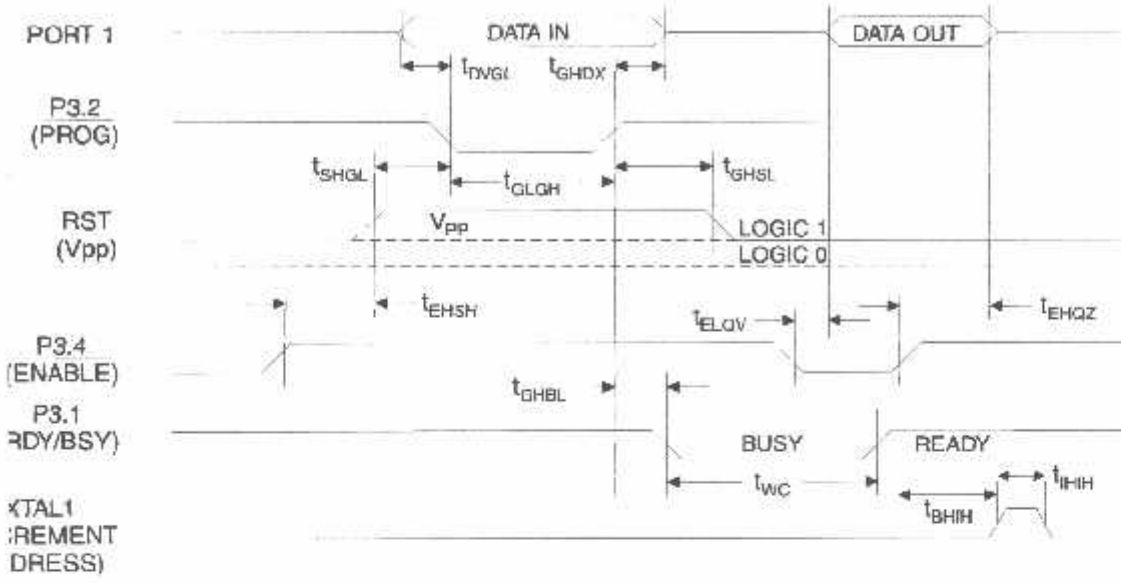
## 1 Programming and Verification Characteristics

C to 70°C,  $V_{CC} = 5.0 \pm 10\%$

Parameter	Min	Max	Units
Programming Enable Voltage	11.5	12.5	V
Programming Enable Current		250	$\mu A$
Data Setup to $\overline{PROG}$ Low	1.0		$\mu s$
Data Hold After $\overline{PROG}$	1.0		$\mu s$
P3.4 ( $\overline{ENABLE}$ ) High to $V_{PP}$	1.0		$\mu s$
$V_{PP}$ Setup to $\overline{PROG}$ Low	10		$\mu s$
$V_{PP}$ Hold After $\overline{PROG}$	10		$\mu s$
$\overline{PROG}$ Width	1	110	$\mu s$
$\overline{ENABLE}$ Low to Data Valid		1.0	$\mu s$
Data Float After $\overline{ENABLE}$	0	1.0	$\mu s$
$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
Byte Write Cycle Time		2.0	ms
RDY/BSY $\downarrow$ to Increment Clock Delay	1.0		$\mu s$
Increment Clock High	200		ns

1. Only used in 12-volt programming mode.

1 Programming and Verification Waveforms



Absolute Maximum Ratings\*

Operating Temperature.....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on Any Pin	
with respect to Ground .....	-1.0V to +7.0V
Maximum Operating Voltage.....	6.6V
Maximum Output Current.....	25.0 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.



## Characteristics

0°C to 85°C,  $V_{CC} = 3.0V$  to  $6.0V$  (unless otherwise noted)

ol	Parameter	Condition	Min	Max	Units
	Input Low Voltage		-0.5	$0.2 V_{CC} - 0.1$	V
	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
	Output Low Voltage <sup>(1)</sup> (Ports 1, 3)	$I_{OL} = 20 \text{ mA}$ , $V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}$ , $V_{CC} = 2.7V$		0.5	V
	Output High Voltage (Ports 1, 3)	$I_{OH} = -80 \mu\text{A}$ , $V_{CC} = 5V \pm 10\%$	2.4		V
		$I_{OH} = -30 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12 \mu\text{A}$	$0.9 V_{CC}$		V
	Logical 0 Input Current (Ports 1, 3)	$V_{IN} = 0.45V$		-50	$\mu\text{A}$
	Logical 1 to 0 Transition Current (Ports 1, 3)	$V_{IN} = 2V$ , $V_{CC} = 5V \pm 10\%$		-750	$\mu\text{A}$
	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		+10	$\mu\text{A}$
	Comparator Input Offset Voltage	$V_{CC} = 5V$		20	mV
	Comparator Input Common Mode Voltage		0	$V_{CC}$	V
	Reset Pulldown Resistor		50	300	K $\Omega$
	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6V/3V$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6V/3V$ P1.0 & P1.1 = 0V or $V_{CC}$		5/1	mA
	Power Down Mode <sup>(2)</sup>	$V_{CC} = 6V$ P1.0 & P1.1 = 0V or $V_{CC}$		100	$\mu\text{A}$
		$V_{CC} = 3V$ P1.0 & P1.1 = 0V or $V_{CC}$		20	$\mu\text{A}$

1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:

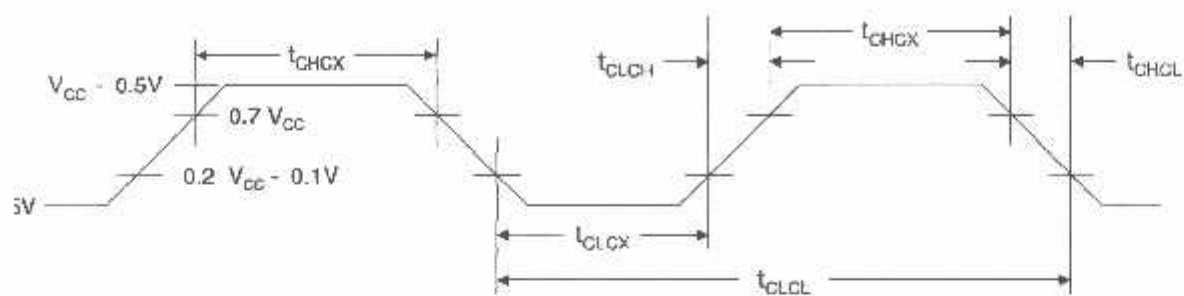
Maximum  $I_{OL}$  per port pin: 20 mA

Maximum total  $I_{OL}$  for all output pins: 80 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.

## nal Clock Drive Waveforms



## nal Clock Drive

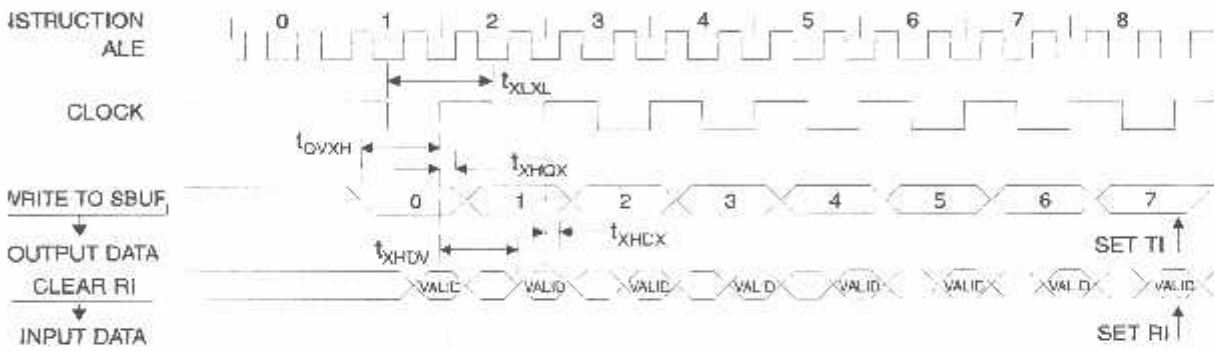
id	Parameter	$V_{CC} = 3.0V \text{ to } 6.0V$		$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	Min	Max	
	Oscillator Frequency	0	12	0	24	MHz
	Clock Period	83.3		41.6		ns
	High Time	30		15		ns
	Low Time	30		15		ns
	Rise Time		20		20	ns
	Fall Time		20		20	ns

**II Port Timing: Shift Register Mode Test Conditions**

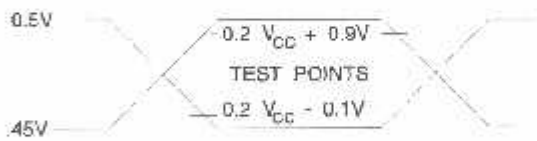
5.0V ± 20%; Load Capacitance = 80 pF)

Cl	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
	Serial Port Clock Cycle Time	1.0		12t <sub>CLCL</sub>		μs
	Output Data Setup to Clock Rising Edge	700		10t <sub>CLCL</sub> -133		ns
	Output Data Hold After Clock Rising Edge	50		2t <sub>CLCL</sub> -117		ns
	Input Data Hold After Clock Rising Edge	0		0		ns
	Clock Rising Edge to Input Data Valid		700		10t <sub>CLCL</sub> -133	ns

**Register Mode Timing Waveforms**



**Testing Input/Output Waveforms<sup>(1)</sup>**



**Float Waveforms<sup>(1)</sup>**

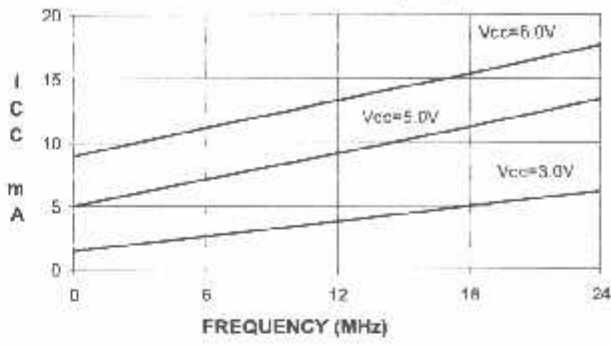


1. AC Inputs during testing are driven at V<sub>CC</sub> - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V<sub>IH</sub> min. for a logic 1 and V<sub>IL</sub> max. for a logic 0.

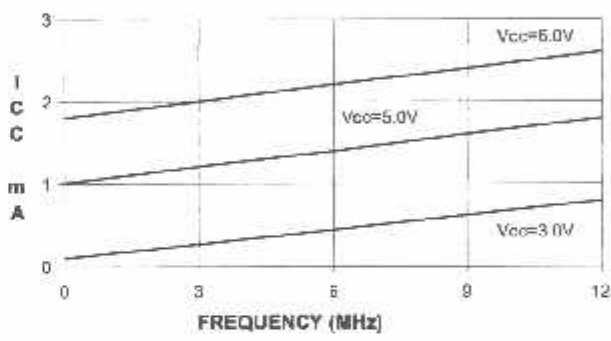
Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V<sub>OH</sub>/V<sub>OL</sub> level occurs.



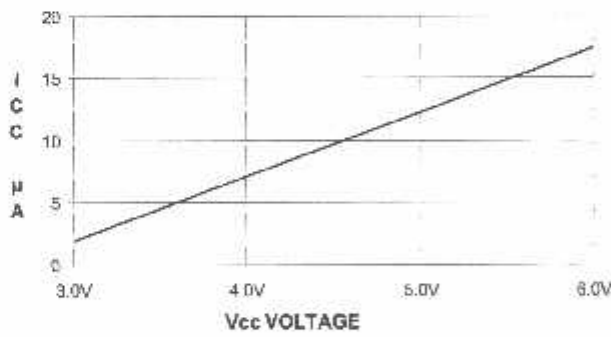
**AT89C4051**  
TYPICAL  $I_{CC}$  - ACTIVE (85°C)



**AT89C4051**  
TYPICAL  $I_{CC}$  - IDLE (85°C)



**AT89C4051**  
TYPICAL  $I_{CC}$  vs. VOLTAGE - POWER DOWN (85°C)



1. XTAL1 tied to GND for  $I_{CC}$  (power down)
2. P1.0 and P1.1 =  $V_{CC}$  or GND
3. Lock bits programmed





## Operating Information

Operating Mode	Power Supply	Ordering Code	Package	Operation Range
Normal	3.0V to 6.0V	AT89C4051-12PC	20P3	Commercial (0°C to 70°C)
		AT89C4051-12SC	20S	
		AT89C4051-12PI	20P3	Industrial (-40°C to 85°C)
		AT89C4051-12SI	20S	
		AT89C4051-12PA	20P3	Automotive (-40°C to 105°C)
		AT89C4051-12SA	20S	
High-Speed	4.0V to 6.0V	AT89C4051-24PC	20P3	Commercial (0°C to 70°C)
		AT89C4051-24SC	20S	
		AT89C4051-24PI	20P3	Industrial (-40°C to 85°C)
		AT89C4051-24SI	20S	

### Package Type

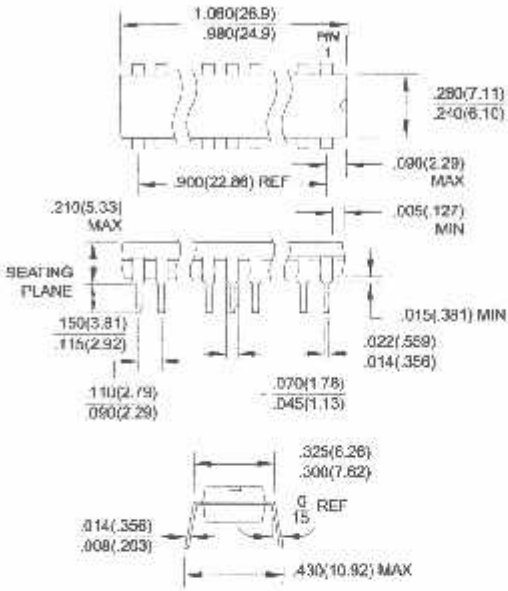
20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)

20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

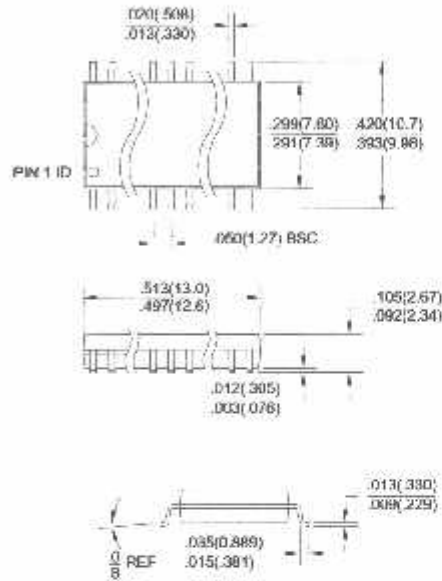
## AT89C4051

aging Information

**23**, 20-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)  
 Dimensions in Inches and (Millimeters)  
 IEC STANDARD MS-001 AD



**20S**, 20-Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)  
 Dimensions in Inches and (Millimeters)



_Tmp1	Equ	0000h
_Tmp2	Equ	0001h
_Tmp3	Equ	0002h
_Tmp4	Equ	0003h
_Tmp5	Equ	0004h
_Tmp6	Equ	0005h
_Tmp7	Equ	0006h
_Tmp8	Equ	0007h
_Tmp9	Equ	0008h
_SMS_InfoTmp	Equ	0100h
_SMS_Info	Equ	DB_SMS_InfoTmp+100
_SMS_Penerima	Equ	DB_SMS_Info+170
_SMS_Center	Equ	DB_SMS_Penerima+30
_PDU_Info	Equ	DB_SMS_Center+30
_PDU_Penerima	Equ	DB_PDU_Info+400
_PDU_Center	Equ	DB_PDU_Penerima+70
_PDU_Total	Equ	DB_PDU_Center+70
_PDU_TotalCar	Equ	DB_PDU_Total+400
DU_DPHHs	Data	20h
DU_DPLLa	Data	21h
DU_DPHHd	Data	22h
DU_DPLLd	Data	23h
p1	Data	24h
p2	Data	25h
p3	Data	26h
p4	Data	27h
-----		
SMS		
Konfirmasi	Data	28h
t1	Data	29h
agSerial	Data	2Ah
EH1	Data	2Bh
LL1	Data	2Ch
EH2	Data	2Dh
LL2	Data	2Eh
ag1	Data	2Fh
ag2	Data	30h
EH	Data	31h
LL	Data	32h
taSEE	Data	33h
erator	Data	34h
sinal	Data	35h
risMenu	Data	36h
tItem	Data	37h
norEP	Equ	40h
ceNominal	Equ	70h
tPDU	Equ	15

---

```
1_Datal Equ 0020h
1 Equ 10
```

Alamat awal lompat ke Start

```
Org 0000
Jmp StartUtama

Org 0020h
Jmp SerialInterupt
```

```
StartUtama:
Clr EA
Mov SP, #80h

Call Delay_Fix_100ms
Call Delay_Fix_100ms
Call Delay_Fix_100ms
Call LCD_Initialization
```

```
Start:
Call LCD_Clrscr
Call Send_ATE1
Cjne A, #00, SendATE_OK
Call LCD_Line1
Mov DPTR, #txt_error1
Call LCD_String
Call LCD_Line2
Mov DPTR, #txt_error2
Call LCD_String
Jmp Loop
```

```
SendATE_OK:

Call LCD_Line1
Mov DPTR, #txt_menu1
Call LCD_String
Call LCD_Line2
Mov DPTR, #txt_menu2
Call LCD_String
```

-----

cekKeypad:

```
Mov RC, #0
Call Keypad
Cjne A, #'0', CekKeypad

Call LCD_Clrscr
Call LCD_Line1
Mov DPTR, #txt_menu13
Call LCD_String
Call LCD_Line2
Mov DPTR, #txt_menu14
Call LCD_String
```

EndMenu1:

---

```

Mov     RC,#0
Call   Keypad
Cjne   A,#'1',PilihMenu2
Jmp    PilihSaldo
PilihMenu2:
Cjne   A,#'2',PilihMenu3
Jmp    PilihRekap
PilihMenu3:
Cjne   A,#'3',PilihMenu0
Jmp    PilihBeliPulsa
PilihMenu0:
Cjne   A,#'0',PilihMenu1
Jmp    Start

```

```

-----
PilihSaldo:
Call   LCD_Clrscr
Call   LCD_Line1
Mov    DPTR,#txt_menu0
Call   LCD_String
Call   LCD_Line2
Mov    DPTR,#txt_menu1
Call   LCD_String
KinCekSaldo:
Mov    R0,#0
Call   Keypad
Cjne   A,#'1',YakinCekSaldoNo
Jmp    CekSaldoProses
KinCekSaldoNo:
Cjne   A,#'2',YakinCekSaldo
Jmp    Start
kSaldoProses:
Mov    DPTR,#DB_SMS_InfoTmp
Mov    A,#'3'
Call   EEPROM_Save

    Inc    DPTR
    Mov    A,#'.'
    Call   EEPROM_Save

    Inc    DPTR
    Mov    A,#'1'
    Call   EEPROM_Save

    Inc    DPTR
    Mov    A,#'8'
    Call   EEPROM_Save

    Inc    DPTR
    Mov    A,#'4'
    Call   EEPROM_Save

    Inc    DPTR
    Mov    A,#'3'
    Call   EEPROM_Save

    Inc    DPTR

```

---

```
Mov     A,#0
Call    EEPROM_Save

Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR,#txt_menu12
Call    LCD_String

Call    Send_SMS
Jmp     Start
```

-----

libRekap:

```
Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR,#txt_menu15
Call    LCD_String
Call    LCD_Line2
Mov     DPTR,#txt_menu16
Call    LCD_String
```

libRekapTgi:

```
Call    LCD_Blink_On
Mov     R0,#20h
Call    LCD_Cursor_Position
```

;isi tahun

```
Mov     R0,#0FFh
Call    Keypad
Mov     40h,A
```

```
Mov     R0,#0FFh
Call    Keypad
Mov     41h,A
```

```
Mov     R0,#0FFh
Call    Keypad
Mov     42h,A
```

```
Mov     R0,#0FFh
Call    Keypad
Mov     43h,A
```

```
Mov     A,#'-'
Call    LCD_Data
```

;isi bulan

```
Mov     R0,#0FFh
Call    Keypad
Mov     44h,A
```

```
Mov     R0,#0FFh
Call    Keypad
Mov     45h,A
```

```
Mov     A,#'-'
Call    LCD_Data
```

```

Call    EEPROM_Save

Inc     DPTR
Mov     A, 45h
Call    EEPROM_Save

Inc     DPTR
Mov     A, 46h
Call    EEPROM_Save

Inc     DPTR
Mov     A, 47h
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'.'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'1'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'8'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'4'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'3'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #0
Call    EEPROM_Save

Call    LCD_Clear
Call    LCD_Line1
Mov     DPTR, #txt_menu2
Call    LCD_String

Call    Send_SMS
Jmp     Start

```

---

-----

11hBeliPulsa:

```

Call    LCD_Clear
Call    LCD_Line1
Mov     DPTR, #txt_menu3
Call    LCD_String
Call    LCD_Line2
Mov     DPTR, #txt_menu4
Call    LCD_String

Mov     R0, #20

```

---

```

SplashMenuA1:
    Jb      P1.4,PilihOperator
    Call   Delay_Fix_100ms
    Djnz   R0,SplashMenuA1

    Call   LCD_Cursor
    Call   LCD_Line1
    Mov    DPTR,#txt_menu5
    Call   LCD_String
    Call   LCD_Line2
    Mov    DPTR,#txt_menu6
    Call   LCD_String

    Mov    R0,#20
SplashMenuA2:
    Jc      P1.4,PilihOperator
    Call   Delay_Fix_100ms
    Djnz   R0,SplashMenuA2
    Jmp    PilihBeliPulsa
PilihOperator:
    Mov    R0,#0
    Call   Keyped
    Mov    Operator,A
-----
kOperator1:
    Cjne   A,#'1',CekOperator2
    Mov    BarisMenu,#3
    Mov    CntItem,#'9'
    Mov    DPTR,#txt_Mentari1
    Jmp    ProsesNominalPulsa
kOperator2:
    Cjne   A,#'2',CekOperator3
    Mov    BarisMenu,#2
    Mov    CntItem,#'6'
    Mov    DPTR,#txt_XLBebas1
    Jmp    ProsesNominalPulsa
kOperator3:
    Cjne   A,#'3',CekOperator4
    Mov    BarisMenu,#1
    Mov    CntItem,#'3'
    Mov    DPTR,#txt_Simpai11
    Jmp    ProsesNominalPulsa
kOperator4:
    Cjne   A,#'4',CekOperator5
    Mov    BarisMenu,#1
    Mov    CntItem,#'3'
    Mov    DPTR,#txt_XLJempol1
    Jmp    ProsesNominalPulsa
kOperator5:
    Cjne   A,#'5',CekOperator6
    Mov    BarisMenu,#1
    Mov    CntItem,#'2'
    Mov    DPTR,#txt_XLExtra1
    Jmp    ProsesNominalPulsa
kOperator6:
    Cjne   A,#'6',CekOperator7
    Mov    BarisMenu,#2

```



```

    Call    LCD_Line1
    Mov     DPTR,#txt_menu10
    Call    LCD_String
    Call    LCD_Line2
    Mov     DPTR,#txt_monull
    Call    LCD_String
kinBelii:
    Mov     R0,#0
    Call    Keypad
    Cjne    A,#'1',YakinBeliiNo
    Jmp     ProsesBeliPulsaYes
kinBeliiNo:
    Cjne    A,#'2',YakinBelii
    Jmp     Start

```

```

=====
prosesBeliPulsaYes:

```

```

    ;find operator
    Mov     A,Operator
kOperator1a:
    Cjne    A,#'1',CekOperator2a
    Mov     DPTR,#db_Mantari
    Jmp     ProsesNominalPulsaOk
kOperator2a:
    Cjne    A,#'2',CekOperator3a
    Mov     DPTR,#db_XLBebas
    Jmp     ProsesNominalPulsaOk
kOperator3a:
    Cjne    A,#'3',CekOperator4a
    Mov     DPTR,#db_Simpatti
    Jmp     ProsesNominalPulsaOk
kOperator4a:
    Cjne    A,#'4',CekOperator5a
    Mov     DPTR,#db_XLJempol
    Jmp     ProsesNominalPulsaOk
kOperator5a:
    Cjne    A,#'5',CekOperator6a
    Mov     DPTR,#db_XLExtra
    Jmp     ProsesNominalPulsaOk
kOperator6a:
    Cjne    A,#'6',CekOperator7a
    Mov     DPTR,#db_Flexi
    Jmp     ProsesNominalPulsaOk
kOperator7a:
    Cjne    A,#'7',CekOperator8a
    Mov     DPTR,#db_IM3
    Jmp     ProsesNominalPulsaOk
kOperator8a:
    Cjne    A,#'8',CekOperator9a
    Mov     DPTR,#db_AS
    Jmp     ProsesNominalPulsaOk
kOperator9a:
    Cjne    A,#'9',CekOperator10a
    Mov     DPTR,#db_Pren
    Jmp     ProsesNominalPulsaOk
sOperator10a:
    Jmp     Start

```

```

-----
prosesNominalPulsaOk:
    Clr    A
    Move   A,@A+DPTR
    Cjne   A,##',NaikkanNominalLagi
    Jmp    CekNominal
ikkanNominalLagi:
    Inc    DPTR
    Jmp    ProsesNominalPulsaOk

kNominal:
    Inc    DPTR
    Clr    A
    Move   A,@A+DPTR
    Cjne   A,Nominal,NaikkanNominalLagi

    Mov    R0,#KodeNominal
    Inc    DPTR

;-----
iKodeNominal:
    Clr    A
    Move   A,@A+DPTR
    Cjnc   A,#0,IsiKodeNominalHP
    Mov    @R0,A
    Jmp    ProsesGabungInfo
iKodeNominalHP:
    Inc    DPTR
    Mov    @R0,A
    Inc    R0
    Jmp    IsiKodeNominal

;-----
osesGabungInfo:

    Call   LCD_Clrscr
;Call   LCD_Line

    Mov    DPTR,#DB_SMS_InfoTmp

    Inc    DPTR
    Mov    R0,#KodeNominal
owKodeNominal0:
    Cjne   A,#0,ShowKodeNominal
    Jmp    ShowNomorHP
owKodeNominal:
;Call   LCD_Data
    Call   EEPROM_Save
    Inc    DPTR
    Inc    R0
    Jmp    ShowKodeNominal0
-----
owNomorHP:
;Call   LCD_Line2

```

---

```

Mov     A, #'.'
Call    EEPROM_Save

Inc     DPTR
Mov     R0, #NomorHP
owNomorHP0:
Mov     A, @R0
Cjne   A, #0, ShowNomorHP1
Jmp     ShowPassword

owNomorHP1:
;Call   LCD_Data
Call    EEPROM_Save
Inc     DPTR
Inc     R0
Jmp     ShowNomorHP0

```

---

```

owPassword:

```

```

Mov     A, #'.'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'1'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'8'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'4'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #'3'
Call    EEPROM_Save

Inc     DPTR
Mov     A, #0
Call    EEPROM_Save

```

---

```

owINFOALL:

```

```

Call    LCD_Clrscr
Call    LCD_Line1
Mov     DPTR, #txt_menu12
Call    LCD_String

Call    Send_SMS
Jmp     Start

```

```

op:

```

```

Sjmp   $

```

```

ow NominalPulse:

```

```

Mov     A, BarisMenu

```

```

        Jmp      Show_NominalPulsa3

i1xNominalVoucher:
kAmbilPulsa1:
    Mov     R0,40
    Call    Keypad
    Mov     Nominal,A
    Cjne   A,#'0',CekAmbilPulsa2a
    Jmp     CekAmbilPulsa1
kAmbilPulsa2a:
    Cjne   A,CntItem,CekAmbilPulsa3a
    Ret
kAmbilPulsa3a:
    Inc    CekAmbilPulsa1
    Ret

    ret

nd_SMS:
    Call    Serial_Initialization
    Clr     EA

-----
nd_ATE1Me:
    Call    LCD_Line2
    Mov     DPTR,#txt_ok
    Call    LCD_String
    ;Call    Delay_Fix_1s
    Mov     0,#200
i1xx4:
    Mov     1,#200
i2xx4:
    Mov     2,#10
c1xx4:
    Djnz   2,Declxx4
    Djnz   1,Isi2xx4
    Djnz   0,Isi1xx4
    ;Jmp    Send_ATE1
-----
nd_ATCMGS:
    Call    LCD_Clear
    Call    LCD_Line1
    Mov     DPTR,#txt_send_ATCMGS
    Call    LCD_String

    Call    Serial_Initialization

    Mov     DtKonfirmasi,#C
    Mov     DPTR,#txt_ATCMGS1
    Call    Serial_Transmit_String
    Mov     DPTR,#DB_PDU_TotalCar
    Call    Serial_Transmit_String_Ram
    ;Mov     DPTR,#txt_send_l3
    ;Call    Serial_Transmit_String
    Call    Serial_Initialization
    Setb   EA
nd_ATCMGS_Cek:

```

---

```

Mov     A,DtKonfirmasi
Cjne   A,#'>',Send_ATCMGS_Cek

-----
nd_PDU:
Call   LCD_Clrscr
Call   LCD_Line1
Mov    DPTR,#txt_send_PDU
Call   LCD_String
;Call  Delay_Fix_Is

Mov    R0,#0

;Mov   DPTR,#txt_PDU1
;Call  Serial_Transmit_String

Mov    DPTR,#txt_send_26
Call   Serial_Transmit_String
Setb   EA

nd_PDUcek:
Mov    A,DtKonfirmasi
Cjne   A,#'0',Send_PDUcek

-----
kses:
Call   LCD_Line2
Mov    DPTR,#txt_delivered
Call   LCD_String
Call   Delay_Fix_Is
ret

nd_ATEL:
Call   LCD_Clrscr
Call   LCD_Line1
Mov    DPTR,#txt_send_ATEL
Call   LCD_String

Call   Serial_Initialization
Clr    EA
Mov    DtKonfirmasi,#0
Mov    DPTR,#txt_ATEL
Call   Serial_Transmit_String
Setb   EA

endATE_Cek:
Call   Serial_ReceiveTimeOut
Cjne   A,#'0',hSendATE_Cek1
Mov    A,#0FFh
Ret

endATE_Cek1:
Cjne   A,#0FFh,hSendATE_Cek    ; time out
Mov    A,#00
Ret
ret

ypad:
Push   B
Jnb    P1.4,$
Nop
Jb     P1.4,$

```

---

```

Mov     A,P1
Anl    A,#0Fh
Orl    A,#30h
Mov    B,A
Mov    A,R0
Cjne  A,#CC,KeypadShowChar
Mov    A,B
Pop    B
Ret

ypadShowChar:
Mov    A,B
Call  LCD_Data
Mov    A,B
Pop    B
ret

FROM_Save:
Push  AR7
Push  AR6
Push  AR5
Push  AR4
Push  AR3
Push  AR2
Push  AR1
Push  AR0
Push  B

Mov    DataSEE,A
Call  TullisDPTRSEE16b
Mov    A,DataSEE

Pop    B
Pop    AR0
Pop    AR1
Pop    AR2
Pop    AR3
Pop    AR4
Pop    AR5
Pop    AR6
Pop    AR7
ret

FROM_Read:
Push  AR7
Push  AR6
Push  AR5
Push  AR4
Push  AR3
Push  AR2
Push  AR1
Push  AR0
Push  B

Mov    R7,#0A0h
Mov    B,DPH
Mov    R6,DPL
Call  Baca_SEE16b

```

---

```

    Pop    B
    Pop    AR0
    Pop    AR1
    Pop    AR2
    Pop    AR3
    Pop    AR4
    Pop    AR5
    Pop    AR6
    Pop    AR7

    ret
lis_SEE16b:
    Call   Siapkan16bAlamatSEE
    Jc     Wrong_Write
    Call   Buat_StopBit
    Ret
ong_Write:
    Call   Buat_StopBit
    Clr    C
    Jmp    Tulis_SEE16b
    ret
lis_SEE8b:
    Call   Siapkan8bAlamatSEE
    Jc     Wrong_Write8b
    Mov    A,DataSEE
    Call   KirimDataSEE
    Jc     Wrong_Write8b
    Call   Buat_StopBit
    Ret
ong_Write8b:
    Call   Buat_StopBit
    Clr    C
    Jmp    Tulis_SEE8b
    ret
ca_SEE16b:
    Call   Siapkan16bAlamatSEE
    Jc     Wrong_Read
    Call   Buat_StartBit      ;Kirim Device Address dengan
    Call   ModeBacaSEE
    Jc     Wrong_Read
    Call   BacaDataSEE
    Ret
ong_Read:
    Call   Buat_StopBit
    Clr    C
    Ajmp  Baca_SEE16b
    ret
ca_SEE8b:
    Clr    C
    Call   Siapkan8bAlamatSEE
    Jc     Wrong_Read8b
    Call   BacaDataSEE
    Ret
ong_Read8b:
    Call   Buat_StopBit
    Clr    C
    Jmp    Baca_SEE8b

```

---

```

    ret
ambil_Ack:
    Clr    C
    Setb   SDA
    Setb   SCL
    Mov    C,SDA
    Clr    SCL
    ret

rimDeviceAddress:
    Call   Buat_StartBit
    Push   ACC
    Mov    A,R7          ;Device Address
    Call   KirimDataSEE
    Pop    ACC
    ret

rim1WordAddress:
    Push   ACC
    Mov    A,B          ;First Word Address
    Call   KirimDataSEE
    Pop    ACC
    ret

rim2WordAddress:
    Push   ACC
    Mov    A,R6          ;Second Address
    Call   KirimDataSEE
    Pop    ACC
    ret

caDataSFR:
    Clr    A

opBacaSEE16b:
    Push   B
    Rl     A
    Setb   SDA
    Setb   SCL
    Clr    C
    Mov    C,SDA
    Mov    ACC.0,C
    Clr    SCL
    Pop    B
    Djnz   B,LoopBacaSEE16b
    Call   Ambil_Ack
    Call   Ambil_Ack
    Call   Buat_StopBit
    Pop    B
    ret

rimDataSEE:
    Push   B
    Mov    B,#8

send8_bitloop:
    Rlc    A
    Mov    SDA,C
    Call   Pulse_SEE
    Djnz   B,Send8_bitloop
    Pop    B
    Clr    C
    Call   Ambil_Ack
    ret

```

---



```

lse_SEE:
    Push    B
    Setb    SCL
    Clr     SCL
    Pop     B
    ret

at_StartBit:
    Setb    SDA
    Setb    SCL
    Clr     SDA
    Clr     SCL
    ret

at_StopBit:
    Clr     SDA
    Setb    SCL
    Setb    SDA
    ret

geSEE8bWrite:
    Clr     FO
    Push    05h
    Call    Siapkan8bAlamatSEE
    Jmp     PageSEEWrite

geSEE16bWrite:
    Setb    FO
    Push    05h

op2PageSEF16bWrite:
    Call    Siapkan16bAlamatSEE

geSEEWrite:
    Mov     A, DataSEE
    Call    KirimDataSEE
    Jc     WrongPWrite
    Mov     R5, #031

opPageSEE16bWrite:
    Mov     A, DataSEE
    Call    KirimDataSEE
    Jc     Loop2PageSEE16bWrite
    Djnz   R5, LoopPageSEM16bWrite
    Pop     05h
    Call    Buat_StopBit
    Ret

ongPWrite:
    Pop     05h
    Call    Bnat_StopBit
    Jb     FO, PageSEE16bWrite
    Jmp     PageSEE8bWrite
    ret

apkan16bAlamatSEE:
    Call    KirimDeviceAddress
    Jc     SalahTulisAlamat
    Call    Kirim1WordAddress
    Jc     SalahTulisAlamat
    Call    Kirim2WordAddress

LahTulisAlamat:
    Ret
    ret

apkan8bAlamatSEE:
    Call    KirimDeviceAddress

```

```

    Mov     DPTR,#txt_SMS_Center
    Mov     Dphh,DPH
    Mov     Dpll,DPL
    Mov     DPTR,#DB_SMS_Center
    Mov     DPHH1,DPH
    Mov     DPLL1,DPL
    Call    MemROM_RAM
    ret
mROM_RAM:
mROM_RAM0:
    Mov     DPH,Dphh
    Mov     DPL,Dpll
    Clr     A
    Movc    A,%A+DPTR
    Cjne    A,#0,memRom_Ram1
    Mov     DPH,DPHH1
    Mov     DPL,DPLL1
    Call    RAM_Write
    Ret
mRom_Ram1:
    Inc     DPTR
    Mov     Dphh,DPH
    Mov     Dpll,DPL
    Mov     DPH,DPHH1
    Mov     DPL,DPLL1
    Call    RAM_Write
    Inc     DPTR
    Mov     DPHH1,DPH
    Mov     DPLL1,DPL
    Jmp     MemROM_RAM0
    ret
mRAM_RAM:
mRAM_RAM0:
    Mov     DPH,Dphh
    Mov     DPL,Dpll
    Call    RAM_Read
    Cjne    A,#0,memRam_Ram1
    Mov     DPH,DPHH1
    Mov     DPL,DPLL1
    Call    RAM_Write
    Ret
mRam_Ram1:
    Inc     DPTR
    Mov     Dphh,DPH
    Mov     Dpll,DPL
    Mov     DPH,DPHH1
    Mov     DPL,DPLL1
    Call    RAM_Write
    Inc     DPTR
    Mov     DPHH1,DPH
    Mov     DPLL1,DPL
    Jmp     MemRAM_RAM0
    ret
eate_PDU_All:
PDU_Center

```

---

```

Mov    DPTR,#DB_SMS_Center
Mov    CPDU_DPHHs,DPH
Mov    CPDU_DPLLs,DPL
Mov    DPTR,#DB_PDU_Center
Mov    CPDU_DPHHd,DPH
Mov    CPDU_DPLLd,DPL
Call   Create_Nomor_Center

```

#### PDU Penerima

```

Mov    DPTR,#DB_SMS_Penerima
Mov    CPDU_DPHHs,DPH
Mov    CPDU_DPLLs,DPL
Mov    DPTR,#DB_PDU_Penerima
Mov    CPDU_DPHHd,DPH
Mov    CPDU_DPLLd,DPL
Call   Create_Nomor_Penerima

```

#### PDU Info

```

Mov    DPTR,#DB_SMS_Info
Mov    CPDU_DPHHs,DPH
Mov    CPDU_DPLLs,DPL
Mov    DPTR,#DB_PDU_Info
Mov    CPDU_DPHHd,DPH
Mov    CPDU_DPLLd,DPL
Call   Create_PDU

```

```
ret
```

#### create\_PDU\_Total:

##### PDU Total

##### Center

```

Mov    DPTR,#DB_PDU_Center
Mov    DPHH1,DPH
Mov    DPLL1,DPL
Call   MemRAM_RAM

```

```

Inc    DPTR
Mov    DPHH1,DPH
Mov    DPLL1,DPL

```

```

Mov    Dphh,DPH
Mov    Dpll,DPL
Call   MemRAM_RAM

```

```

Mov    DPL,DPLL1
Mov    A,#00
Call   RAM_Write_Hexa
Inc    DPTR
Mov    A,#00
Call   RAM_Write_Hexa
Inc    DPTR
Mov    DPHH1,DPH
Mov    DPLL1,DPL
Mov    Dphh,DPH
Mov    Dpll,DPL
Call   MemRAM_RAM

```

---

```

Mov     DPTR,#DB_PDU_Center
Call   LenStrDiv2
Mov     Tmpl,A

Mov     DPTR,#DB_PDU_Total
Call   LenStrDiv2
Clr     C
Subb   A,Tmpl
Mov     DPTR,#DB_PDU_TotalCar
Mov     A,R0
Swap   A
Anl    A,#0Fh
Cjne   A,#00h,Create_PDU_Total0
Anl    A,#0Fh
Cjne   A,#00h,Create_PDU_Total1
Jmp    Create_PDU_Total2
Call   RAM_Write
Inc    DPTR
Jmp    Create_PDU_Total01
eate_PDU_Total1:
Orl    A,#30h
Call   RAM_Write
Inc    DPTR
Jmp    Create_PDU_Total2
--
Mov     A,R1
Swap   A
Anl    A,#0Fh
Cjne   A,#00h,Create_PDU_Total0x
eate_PDU_Total01x:
Mov     A,R1
Anl    A,#0Fh
Cjne   A,#00h,Create_PDU_Total1x
Jmp    Create_PDU_Total2x
eate_PDU_Total0x:
Orl    A,#30h
Call   RAM_Write
Inc    DPTR
Jmp    Create_PDU_Total01x
eate_PDU_Total1x:
Orl    A,#30h
Call   RAM_Write
Inc    DPTR
Jmp    Create_PDU_Total2x
--
eate_PDU_Total2x:
Mov     A,#13
Call   RAM_Write
Inc    DPTR

Mov     A,#0
Call   RAM_Write
Inc    DPTR

Mov     DFHH1,DPH
Mov     DPLL1,DPH

```

---

```

Mov     DPTR,#DB_PDU_Total
Mov     Dphh,DPH
Mov     Dpll,DPL
Call    MemRAM_RAM
ret

rsing_RAM:
R0 tempat end parsing
rsing00:
Call    RAM_Read
Cjne   A,0,Parsing01
Inc     DPTR
Ret

rsing01:
Inc     DPTR
Jmp     Parsing00
ret

eate_Nomor_Penerima:
Push   AR0
Push   AR1
Push   AR2
Push   AR3
Push   AR4
Push   AR5
Push   AR6
Push   AR7

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Call    LenStr

Mov     DPH,CPDU_DPHHd
Mov     DPL,CPDU_DPLLd
Call    RAM_Write_Hexa
Inc     DPTR
Mov     A,#91h
Call    RAM_Write_Hexa

Inc     DPTR
Mov     CPDU_DPHHd,DPH
Mov     CPDU_DPLLd,DPL

eate_Nomor_Penerima0:
ambil nomor # 1

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Call    RAM_Read
Inc     DPTR
Mov     CPDU_DPHHs,DPH
Mov     CPDU_DPLLs,DPL
Mov     R0,A
Cjne   A,#0,Create_Nomor_Penerima1
Mov     DPH,CPDU_DPHHd
Mov     DPL,CPDU_DPLLd
Mov     A,#0
Call    RAM_Write

```

---

```

        Jmp      Create_Nomor_PenerimaExit

eate_Nomor_Penerima1:
ambil_nomor # 2
    Mov      DPH,CPDU_DPHHs
    Mov      DPL,CPDU_DPLLs
    Call     RAM_Read
    Inc      DPTR
    Mov      CPDU_DPHHs,DPH
    Mov      CPDU_DPLLs,DPL
    Mov      A,A
    Cjne    A,#0,Create_Nomor_Penerima2
    Mov      DPH,CPDU_DPHHd
    Mov      DPL,CPDU_DPLLd

    Mov      A,#'F'
    Call     RAM_Write

    Inc      DPTR
    Mov      A,R0
    Call     RAM_Write

    Inc      DPTR
    Mov      A,#0
    Call     RAM_Write
    Jmp      Create_Nomor_PenerimaExit

eate_Nomor_Penerima2:
    Mov      DPH,CPDU_DPHHd
    Mov      DPL,CPDU_DPLLd

    Mov      A,R1
    Call     RAM_Write

    Inc      DPTR
    Mov      A,R0
    Call     RAM_Write

    Inc      DPTR
    Mov      CPDU_DPHHd,DPH
    Mov      CPDU_DPLLd,DPL
    Jmp      Create_Nomor_Penerima0

eate_Nomor_PenerimaExit:
    Pop      AR7
    Pop      AR6
    Pop      AR5
    Pop      AR4
    Pop      AR3
    Pop      AR2
    Pop      AR1
    Pop      AR0
    ret

eate_Nomor_Center:
    Push     AR0
    Push     AR1
    Push     AR2
    Push     AR3

```

---

```

Push    AR4
Push    AR5
Push    AR6
Push    AR7

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Mov     DPL,CPDU_DPLLd
Call    RAM_Write_Hexa
Inc     DPTR
Mov     A,#91h
Call    RAM_Write_Hexa

Inc     DPTR
Mov     CPDU_DPHHd,DPH
Mov     CPDU_DPLLd,DPL

```

\_Center0:

```

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Call    RAM_Read
Inc     DPTR
Mov     CPDU_DPHHs,DPH
Mov     CPDU_DPLLs,DPL
Mov     R0,A
Mov     DPL,CPDU_DPLLd
Mov     A,#0
Call    RAM_Write
Jmp     CN_CenterExit

```

\_Center1:

ambil nomor # 2

```

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Call    RAM_Read
Inc     DPTR
Mov     CPDU_DPHHs,DPH
Mov     CPDU_DPLLs,DPL
Mov     R1,A
Cjne   A,#0,CN_Center2
Mov     DPH,CPDU_DPHHd
Mov     DPL,CPDU_DPLLd

```

```

Mov     A,#'F'
Call    RAM_Write

```

```

Inc     DPTR
Mov     A,R0
Call    RAM_Write

```

```

Inc     DPTR
Mov     A,#0
Call    RAM_Write
Jmp     CN_CenterExit

```

\_Center2:

```

Mov     DPH,CPDU_DPHHd

Mov     A,R1
Call    RAM_Write

Inc     DPTR
Mov     A,R0
Call    RAM_Write

Inc     DPTR
Mov     CPDU_DPHHd,DPH
Mov     CPDU_DPLLd,DPL
Jmp     CN_Center0

_CenterExit:
Mov     DPTR,#DB_PDU_Center
Inc     DPTR
Inc     DPTR
Call    LenStr
Mov     DPTR,#DB_PDU_Center
Call    RAM_Write_Hexa

Pop     AR7
Pop     AR6
Pop     AR5
Pop     AR4
Pop     AR3
Pop     AR2
Pop     AR1
Pop     AR0
ret

eate_PDU:
Push   AR0
Push   AR1
Push   AR2
Push   AR3
Push   AR4
Push   AR5
Push   AR6
Push   AR7

Mov     DPH,CPDU_DPHHs
Mov     DPL,CPDU_DPLLs
Call    LenStr

Mov     DPH,CPDU_DPHHd
Mov     DPL,CPDU_DPLLd
Call    RAM_Write_Hexa
Inc     DPTR
Mov     CPDU_DPHHd,DPH
Mov     CPDU_DPLLd,DPL

eate_PDUD:
Mov     R0,#0           ; counter ambil data

eate_PDU1:
ambil_karakter_source
Mov     DPH,CPDU_DPHHs

```

---



```

    Call    RAM_Read
cek jika sudah habis
    Cjne   A,#0,Create_PDU2
    Mov    DPH,CPDU_DPHHd
    Mov    DPL,CPDU_DPLLd
    Call   RAM_Write
    Jmp    Create_PDU_Exit
eate_PDU2:
    Anl   A,#07Fh
    Mov   R1,A           ; simpan sementara
    Inc  DPTR
    Mov  CPDU_DPHHs,DPH
    Mov  CPDU_DPLLs,DPL
    Call RAM_Read       ; ambil next karakter
    Anl  A,#07Fh
    Mov  R2,A
cek jika sudah habis
    Cjne  A,#0,Create_PDU3
    Jmp   Create_PDU4
eate_PDU3:
ambil lsb dan geser kanan sebanyak counter
    Mov  A,R0
    Mov  R7,A
    Mov  R3,#00        ; tampung geser
eate_PDU3GeserKanan:
    Mov  A,R2
    Rrc  A
    Mov  R2,A
    Mov  A,R3
    Rrc  A
    Mov  R3,A
    Sjne R7,Create_PDU3GeserKanan
    Mov  A,R1
    Call MaskingBit
    Orl  A,R3
    Mov  DPH,CPDU_DPHHd
    Mov  DPL,CPDU_DPLLd
    Call RAM_Write_Hexa
    Mov  CPDU_DPLLd,DPL
    Mov  A,R0
    Cjne A,#7,Create_PDU1
    Mov  DPH,CPDU_DPHHs
    Mov  DPL,CPDU_DPLLs
    Inc  DPTR
    Mov  CPDU_DPHHs,DPH
    Mov  CPDU_DPLLs,DPL
    Jmp  Create_PDU0
eate_PDU4:
ambil lsb dan geser kanan sebanyak counter
    Mov  A,R0
    Mov  R7,A
    Mov  R3,#00        ; tampung geser
eate_PDU3GeserKanan2:
    Mov  A,R2
    Rrc  A
    Mov  R2,A
    Mov  A,R3

```

---

```

Rrc      A
Mov      R3,A
Djnz    R7,Create_PDU3GeserKanan?
Mov      A,R1
Call    MaskingBit
Orl     A,R3
Mov     DPH,CPDU_DP HHd
Mov     DPL,CPDU_DPL Ld
Call    RAM_Write_Hexa
Inc     DPTR
Mov     A,#0
Call    RAM_Write
Jmp     Create_PDU_Exit
Create_PDU_Exit:
Pop     AR7
Pop     AR6
Pop     AR5
Pop     AR4
Pop     AR3
Pop     AR2
Pop     AR1
Pop     AR0
ret
M_Write_Hexa:
Mov     B,A
Swap    A
Anl     A,#0Fh
Orl     A,#30h
Call    Convert_Hexa
Call    RAM_Write
Inc     DPTR
Mov     A,B
Anl     A,#0Fh
Orl     A,#30h
Call    Convert_Hexa
Call    RAM_Write
ret
Convert_Hexa:
Cjne   A,#3Ah,tes_huruf_1
Mov    A,'#A'
Ret
s_huruf_1:
Cjne   A,#3Bh,tes_huruf_2
Mov    A,'#B'
Ret
s_huruf_2:
Cjne   A,#3Ch,tes_huruf_3
Mov    A,'#C'
Ret
s_huruf_3:
Cjne   A,#3Dh,tes_huruf_4
Mov    A,'#D'
Ret
s_huruf_4:
Cjne   A,#3Eh,tes_huruf_5
Mov    A,'#E'
Ret

```

---

```

s_huruf_5:
    Cjne    A,#3Fh,tes_huruf_6
    Mov     A,#'F'
    Ret
s_huruf_6:
    Ret
    ret
skingBit:
    Mov     R7,C
skingBitKanan:
    Dfz     R7,MaskingBit1
    Jmp     MaskingBitNolKanan
skingBit1:
    Clr     C
    Rrc     A
    Jmp     MaskingBitKanan
skingBitNolKanan:
    ret

nStrDiv2:
    Push    AR0
    Push    AR1
    Push    AR2
    Push    AR3
    Push    AR4

    Mov     R0,#0
nStr120:
    Mov     R1,#0
nStr12:
    Call    RAM_Read
    Cjne    A,#0,LenStr22
    Jmp     LenStrEnd2
nStr22:
    Inc     DPTR
    Inc     R1
    Mov     A,R1
    Mov     B,#2
    Div     AB
    Mov     A,B
    Jmp     LenStr120
nStrEnd2:
    Mov     A,R0
    Pop     AR3
    Pop     AR2
    Pop     AR1
    Pop     AR0
    ret
nStr:
    Push    AR0
    Push    AR1
    Push    AR2
    Push    AR3
    Push    AR4

    Mov     R0,#0

```

---

```

    Mov     R1,#0
nStr1:
    Call   RAM_Read
    Cjne   A,#0,LenStr2
    Jmp    LenStrEnd
nStr2:
    Inc    DPTR
    Mov    R3,DPH
    Mov    R4,DPL
    Mov    DPH,R0
    Mov    DPL,R1
    Inc    DPTR
    Mov    R0,DPH
    Mov    R1,DPL
    Mov    DPH,R3
    Mov    DPL,R4
    Jmp    LenStr1
nStrEnd:
    Mov    A,R1
    Mov    B,R0
    Pop    AR4
    Pop    AR3
    Pop    AR2
    Pop    AR1
    Pop    AR0
    ret
M_Read:
    ;Movx  A,@DPTR
    Call   EEPROM_Read
    ret
M_Write:
    ;Movx  @DPTR,A
    Call   EEPROM_Save
    ret
M_Read:
    Clr    A
    Movc   A,@A+DPTR
    ret
*****
subrutin clear lcd
*****
D_Clear:
    Push   ACC
    Mov    A,#01h
    Call   LCD_Command
    Pop    ACC
    ret
*****
subrutin menempatkan kursor
*****
D_Cursor_Position:
    Push   ACC
    Mov    A,R0
    Anl   A,#0F0h
    Cjne   A,#10h,lcd_cursor_position1
    Mov    A,R0
    Anl   A,#0Fh

```

---

```

    Orl    A,#80h
    Call  LCD_Command
    Jmp   lcd_cursor_position_end
3_cursor_position1:
    Cjne  A,#20h,lcd_cursor_position_end
    Mov   A,R0
    Anl   A,#0Fh
    Orl   A,#0C0h
    Call  LCD_Command
3_cursor_position_end:
    Pop   ACC
    ret
D_Command:
    Clr   LCDRs
    Mov   LCDData,A
    Setb LCDPna
    Clr   LCDEna
    Call  delayL
    ret
D_Data:
    Setb LCDRs
    Mov   LCDData,A
    Setb LCDEna
    Clr   LCDEna
    Call  delayL
    ret
layL:
    Push  1
    Push  2
    Push  3

    Mov   3,#80
ndalqqal:
    Mov   1,#10
    Djnz  1,$
    Mov   2,#1
    Djnz  2,$
    Mov   2,#1
    Djnz  2,$
    Djnz  3,tundalqqal

    Pop   3
    Pop   2
    Pop   1
    ret
*****
dirim data ke lcd1
*****
LD Data:
    Push  DPL
    Push  DPH
    Push  ACC
    Mov   DPTR,#lcd1
    Movx  @DPTR,A
    Call  LCD_Delay
    Lcall LCD_Delay
    Lcall LCD_Delay

```

---

```

    Lcall  LCD_Delay
    Pop   ACC
    Pop   DPH
    Pop   DPL
    ret

*****
kirin instruksi ke lcd1
*****
i0  Equ   0a000h    ;lcd control operation
i1  Equ   lcd0+1    ;lcd data operation
LCD_Command:
    Push  DPL
    Push  DPH
    Mov   DPTR,#lcd0
    Movx  @DPTR,A
    Lcall LCD_Delay
    Lcall LCD_Delay
    Pop   DPH
    Pop   DPL
    Ret
    ret

D_Delay:
    Push  01h
    Push  02h
    Push  03h
    Mov   3,#5

tundalqqqa:
    Mov   1,#30
    Djnz 1,$
    Mov   2,#40
    Djnz 2,$
    Mov   2,#1
    Djnz 2,$
    Djnz 3,tundalqqqa
    Pop   03h
    Pop   02h
    Pop   01h
    ret

D_Delete_Line1:
    Push  ACC
    Push  DPH
    Push  DPL
    Call  LCD_Line1
    Mov   DPTR,#txt_blank
    Call  LCD_String
    Call  LCD_Line1
    Pop   DPL
    Pop   DPH
    Pop   ACC
    ret

D_Delete_Line2:
    Push  ACC
    Push  DPH
    Push  DPL
    Call  LCD_Line2
    Mov   DPTR,#txt_blank
    Call  LCD_String

```

```

    Call    LCD_Line2
    Pop     DPL
    Pop     DPH
    Pop     ACC
    Ret

t_blank:  Db      '          ',0
    ret

D_Initialization:

    Mov     A,#3Fh
    Call    LCD_Command
    Mov     A,#0Ch
    Call    LCD_Command
    Mov     A,#06h
    Call    LCD_Command
    Mov     A,#1Ch
    Call    LCD_Command
    Mov     A,#01h
    Call    LCD_Command
    ret

D_Line1:
    Push    ACC
    Mov     A,#80h
    Lcall   LCD_Command
    Pop     ACC
    ret

D_Line2:
    Push    ACC
    Mov     A,#0C0h
    Lcall   LCD_Command
    Pop     ACC
    ret

D_Off:
    Push    ACC
    Mov     A,#00001000b
    Lcall   LCD_Command
    Pop     ACC
    ret

D_On:
    Push    ACC
    Mov     A,#00001100b
    Lcall   LCD_Command
    Pop     ACC
    ret

----
D_String:
;Push    ACC
;Push    DPL
;Push    DPH

:car1:
    Clr     A          ; mengambil data dari eprom
    Move    A,@A+DPTR
    Cjne    A,#0,tampil1 ; tes apakah data habis?
    Ljmp    :metul

:ampil1:
    ; kembalikan data ke lcd

    Call    LCD_Data

```

```

    Inc    DPTR      ; naikkan dptr
    Jmp    getcar1

ttul:
    ;Pop   DPH
    ;Pop   DPL
    ;Pop   ACC
    ret

*****
subrutin tampil text pada lcd
dari data di ram
*****
) _String_Ram:
    Push   ACC
    Push   DPL
    Push   DPH
    Push   00h

tcarlz:
    ;Movx  A,@DPTR
    Call   RAM_Read
    Cjne   A,#0,tampilz    ; tes apakah data habis?
    Ljmp   nettulz

mpillz:
    ; keluaran data ke lcd
    Call   LCD_Data
    Inc    DPTR      ; naikkan dptr
    Ljmp   getcarlz

ttulz:
    Pop    00h
    Pop    DPH
    Pop    DPL
    Pop    ACC
    ret

) _String_R0:
    Push   ACC
    Push   DPL
    Push   DPH
    Push   00h

tcarlzd:
    ;Movx  A,@DPTR
    ;Call  RAM_Read
    Mov    A,@R0
    Cjne   A,#0,tampilzd   ; tes apakah data habis?
    Ljmp   nettulzd

mpillzd:
    ; keluaran data ke lcd
    Call   LCD_Data
    ;Inc   DPTR      ; naikkan dptr
    Inc    R0
    Ljmp   getcarlzd

ttulzd:
    Pop    00h
    Pop    DPH
    Pop    DPL
    Pop    ACC
    ret

) _Blink_Off:
    Push   ACC

```



```

    Mov    A,#00001100b    ;
    Lcall  LCD_Command
    Pop    ACC
    ret

D_Blink_On:
    Push  ACC
    Mov   A,#00001101b    ;
    Lcall LCD_Command
    Pop   ACC
    ret

D_Cursor_Off:
    Push  ACC
    Mov   A,#00001100b    ; cursor on
    Lcall LCD_Command
    Pop   ACC
    ret

D_Cursor_On:
    Push  ACC
    Mov   A,#00001110b    ; cursor on
    Lcall LCD_Command
    Pop   ACC
    ret

D_Cursor_Left:
    Push  ACC
    Mov   A,#00010000b    ; cursor left
    Lcall LCD_Command
    Pop   ACC
    ret

D_Cursor_Right:
    Push  ACC
    Mov   A,#00010100b    ; cursor left
    Lcall LCD_Command
    Pop   ACC
    ret

D_Display_Left:
    Push  ACC
    Mov   A,#00011000b    ; display left
    Lcall LCD_Command
    Pop   ACC
    ret

D_Display_Right:
    Push  ACC
    Mov   A,#00011100b    ; display right
    Lcall LCD_Command
    Pop   ACC
    ret

D_Cursor_Home:
    Push  ACC
    Mov   A,#00000010b    ; display right
    Lcall LCD_Command
    Pop   ACC
    ret

LalInterrupt:
    Push  PSW
    Push  ACC
    Jbc   SCON.1,xmit

```

---

```

ve:      Clr      SCON.0
        Mov      A,SBHF
        ;Call    _CJ_Data
        Cjne    A,#'X',CekCMGs
        Mov      DtKonfirmasi,#'D'
        Jmp     endserial
kCMGs:
        Cjne    A,#'>',EndSerial
        Mov      DtKonfirmasi,#'>'
        Jmp     endserial
ll:
        Clr      TI
iSerial:
        Pop      ACC
        Pop      PSW
        Reti
        ret
nversion_Hexa_BCD:
        Push     B
        Mov      B,#100
        Div     AB
        Mov      RO,A
        Mov      A,#10
        Xch     A,B
        Div     AB
        Swap    A
        Add     A,B
        Mov      R1,A
        Pop      B
        Ret
        ret
nversion_Des_Hexa:
        Mov      A,RO
        Swap    A
        Anl     A,#0Fh      ; ambil xx9x
        Mov      B,#0Ah
        Mul     AB
        Mov      B,A
        Mov      A,RO
        Anl     A,#0Fh      ; ambil xxx9
        Add     A,B
        ret
ayMX:
        Push    1
        Push    2
aymx1:
        Mov     2,#40
aymx2:
        Mov     1,#40
aymx3:
        Djnz   1,$
        Djnz   2,Delaymx2
        Djnz   0,Delaymx1
        Pop    2
        Pop    1
        ret

```

```

lay:
  Mov    R0,#0FFh    ; Isi Register R0 dengan FF (Hex)
lay1:
  Mov    R1,#00h     ; Isi Register R1 dengan 00 (hex)
lay2:
  Nop
  Nop
  Nop
  Djnz  R1,Delay2   ; Kurangi R1 dengan 1, bila hasil belum
                  ; sama dengan 0 maka lompat ke Delay2
  Djnz  R0,Delay1   ; Kurangi R0 dengan 1, bila hasil belum
                  ; sama dengan 0 maka lompat ke Delay1
  Ret
                  ; Kembali ke alamat setelah perintah

```

```

sial_receive:
  Jnb   RI,$
  Clr   RI
  Mov   A,SBUF
  Clr   RI
  ;Call Serial_Transmit
  ret

```

```

sial_receiveTimeOut:

```

```

  Push  0
  Mov   0,#100

```

```

sialReceiveTimeOut0:

```

```

  Mov   A,D-Konfirmasi
  Cjne  A,#'0',SerialReceiveTimeOut1
  Jmp   SerialReceiveTimeOutXX

```

```

sialReceiveTimeOut1:

```

```

  Call  TundaTimeOut
  Djnz  0,SerialReceiveTimeOut0
  Mov   A,#0FFh

```

```

sialReceiveTimeOutXX:

```

```

  Pop   0
  ret

```

```

adaTimeOut:

```

```

  Push  0
  Push  1

```

```

adaTimeOut1:

```

```

  Mov   1,#100

```

```

adaTimeOut2:

```

```

  Mov   0,#100
  Djnz  0,$
  Djnz  1,TundaTimeOut2
  Pop   1
  Pop   0
  ret

```

```

erial_receive:

```

```

  Call  Serial_Initialization
  Jnb   RI,$
  Clr   RI
  Mov   A,SBUF
  Clr   RI
  ret

```

```

sial_Transmit:

```

```

  Call  Serial_Initialization
  Mov   SBUF,A

```

---

```

Ret
Jnb TI,$
Clr TI
ret
serial_Transmit:
Mov SBUF,A
Ret
Push ACC
Mov A,FlagSerial
Cjne A,#0,Serial_TransmitOut
Jnb TI,$
Clr TI
serial_TransmitOut:
Pop ACC
ret
serial_Initialization:
Mov SCON,#50h
;Mov TH1,#250 ; BAUDRATE 9600
Mov TH1,#253 ; BAUDRATE 19200
;Mov TH1,#208 ; BAUDRATE 1200
Mov PCON,#00h ; PCON
Mov TMOD,#21h
Mov TCON,#01010000B ; RUN T1 AND T0
Mov IE,#90h ; enable only serial inter
Ret
;Mov IE,#10000010B
;Mov IP,#00001000B
ret
serial_Initialization:
Mov SCON,#50h
Mov TH1,#250 ; BAUDRATE 9600
;Mov TH1,#208 ; BAUDRATE 1200
Mov 87h,#00h ; PCON
Mov TMOD,#21h
Mov TCON,#01010000B ; RUN T1 AND T0
;Mov IE,#10000010B
;Mov IP,#00001000B
ret
serial_Initialization_GPS:
Mov SCON,#50h
Mov TH1,#244 ; BAUDRATE 4800
;Mov TH1,#250 ; BAUDRATE 9600
;Mov TH1,#208 ; BAUDRATE 1200
Mov 87h,#00h ; PCON
Mov TMOD,#21h
Mov TCON,#01010000B ; RUN T1 AND T0
;Mov IE,#10000010B
;Mov IP,#00001000B
ret
serial_Initialization:
Mov SCON,#50h
Mov TH1,#250 ; BAUDRATE 9600
;Mov TH1,#208 ; BAUDRATE 1200
Mov 87h,#00h ; PCON
Mov TMOD,#21h
Mov TCON,#01010000B ; RUN T1 AND T0
;Mov IE,#10000010B

```

```

;Mov IP,#00001000B
ret
rial_Initialization_Download:
Mov SCON,#50h
Mov TH1,#250 ; BAUDRATE 9600
;Mov TH1,#200 ; BAUDRATE 1200
Mov 87h,#00h ; PCON
Mov TMOD,#21h
Mov TCON,#01010000B ; RUN T1 AND T0
ret
rial_Transmit_CrLf:
Push ACC
Mov A,#0Ah
Call Serial_Transmit
Mov A,#0Dh
Call Serial_Transmit
Pop ACC
ret
rial_Transmit_String_Ram:
Push ACC
Push DPL
Push DPH
getcarl:
;Movx A,@DPTR
Call RAM_Read

Cjne A,#0,axtampil1 ; tes apakah data habis?
Ljmp axnnettul

tampil1: ; keluarkan data ke lcd

Call Serial_Transmit
Mov RO,#10
Call DelayMX
Inc DPTR ; naikkan dptr
Ljmp axgetcarl
nettul:
Pop DPH
Pop DPL
Pop ACC
ret
rial_Transmit_String:
Push ACC
Push DPL
Push DPH
atcarl:
Clr A ; mengambil data dari eprom
Movc A,@A+DPTR
Cjne A,#0,xtampil1 ; tes apakah data habis?
Ljmp xnnettul

umnpill1: ; keluarkan data ke lcd
Lcall Serial_Transmit
Mov RO,#10
Call DelayMX
Inc DPTR ; naikkan dptr
Ljmp xgetcarl

```

```
ttul:
    Pop    DPH
    Pop    DPL
    Pop    ACC
    ret
```

----- DELAY -----

```
lay_Var_1ms:
    Call   Delay_Fix_1ms
    Djnz   R0,Delay_Var_1ms
    ret

lay_Var_10ms:
    Call   Delay_Fix_10ms
    Djnz   R0,Delay_Var_10ms
    ret

lay_Var_100ms:
    Call   Delay_Fix_1ms
    Djnz   R0,Delay_Var_100ms
    ret

lay_Var_1s:
    Call   Delay_Fix_1s
    Djnz   R0,Delay_Var_1s
    ret

lay_Var_10s:
    Call   Delay_Fix_10s
    Djnz   R0,Delay_Var_10s
    ret

lay_Var_10ms:
    Call   Delay_Fix_10ms
    Djnz   R0,Delay_Var_10ms
    ret

lay_Fix_10us:
    Push   1
    Mov    1,#20
    Djnz   1,$
    Pop    1
    ret

lay_Fix_10s:
    Push   1
    Mov    1,#100

lay_fix_10s_1:
    Call   Delay_Fix_100ms
    Djnz   1,delay_fix_10s_1
    Pop    1
    ret

lay_Fix_1s:
    Push   1
    Mov    1,#100

lay_fix_100ms_1:
    Call   Delay_Fix_10ms
    Djnz   1,delay_fix_100ms_1
    Pop    1
    ret

lay_Fix_100ms:
    Push   1
    Mov    1,#10
```

---

```

lay_fix_100ms_1:
    Djnz    1,delay_fix_100ms_1
    Pop     1
    ret

lay_fix_10ms:
    Push    ACC
    Mov     A,TMOD
    Clr     TFO                ; me-nol-kan bit limpahan
    Setb    TR0                ; timer mulai bekerja
    Jnb     TFO,$              ; tunggu di sini sampai melimpah
    Clr     TR0                ; timer berhenti kerja
    Mov     TMOD,A
    Pop     ACC
    Ret
    ret

lay_Fix_1ms:
    Push    ACC
    Mov     A,TMOD
    Mov     TMOD,#00300001b    ; Timer 1 bekerja pada mode 1
    Clr     TFO                ; me-nol-kan bit limpahan
    Setb    TR0                ; timer mulai bekerja
    Jnb     TFO,$              ; tunggu di sini sampai melimpah
    Clr     TR0                ; timer berhenti kerja
    Mov     TMOD,A
    Pop     ACC
    Ret
    ret

TIME:
    Mov     TMOD,#11h
    Mov     TH1,#00Eh
    Mov     TL1,#00Eh
    Ret

    0123456789abcdef
_error1:  Db    'HP tdk terhubung',0
_error2:  Db    '-----',0

_menu1:   Db    'Tekan C utk menu',0
_menu2:   Db    ' --- ... ---',0

    0123456789abcdef
_menu3:   Db    '1.MTRI 2.XLB',0
_menu4:   Db    '3.SIM 4.XLJ 5.XT',0
_menu5:   Db    '6.FLEXY 7.IM3',0
_menu6:   Db    '8.AS  9.FREN',0

_menu7:   Db    'Operator:',0
_menu8:   Db    'Nominal:',0
_menu9:   Db    'Nomor telp:',0

    0123456789abcdef
_menu10:  Db    ' Anda Yakin?',0
_menu11:  Db    ' 1-Y,2-T',0

_menu12:  Db    'Data dikirim...',0
_menu13:  Db    '1.Saldo 2.Rekap',0
_menu14:  Db    '3.Beli pulsa',0

```

```

t_menu15: Db 'Isi tanggal:',0
:_menu16: Db 'tttt-bb-tt',0

0123456789abcdef
:_Mentari1: Db '1.10 2.20 3.25',0
:_Mentari2: Db '4.30 5.40 6.50',0
:_Mentari3: Db '7.75 8.100 9.150',0,0

:_XLBebas1: Db '1.10 2.25 3.50',0
:_XLBebas2: Db '4.75 5.100 6.150',0,0

:_Simpatil: Db '1.20 2.50 3.100',0,0

:_XLJempoli: Db '1.5 2.15 3.35',0,0

:_XLExtral: Db '1.50 2.100',0,0

:_Flexi1: Db '1.10 2.20 3.50',0
:_Flexi2: Db '4.100 5.150',0,0

:_IM31: Db '1.10 2.20 3.25',0
:_IM32: Db '4.30 5.40 6.75',0
:_IM33: Db '7.100',0,0

:_AS1: Db '1.25 2.50',0,0

:_Fren1: Db '1.10 2.25 3.50',0
:_Fren2: Db '4.100',0,0

-----
:_SMS_Penerima: Db '628563576430',0
:_SMS_Center: Db '62818443009',0 ; PRO-XL

```

---





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

**MAX220-MAX249**

### General Description

MAX220-MAX249 family of line drivers/receivers is ideal for all EIA/TIA-232E and V.28/V.24 communication interfaces, particularly applications where  $\pm 12V$  is available.

These parts are especially useful in battery-powered systems since their low-power shutdown mode reduces power dissipation to less than  $5\mu W$ . The MAX225, MAX233, MAX235, and MAX245/MAX246/MAX247 use terminal components and are recommended for applications where printed circuit board space is critical.

### Applications

Portable Computers  
Low-Power Modems  
Interface Translation  
Battery-Powered RS-232 Systems  
Multidrop RS-232 Networks

### Features

#### Superior to Bipolar

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

### Ordering Information

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

Ordering information continued at end of data sheet.  
\*Contact factory for dice specifications.

### Selection Table

Part	Power Supply (V)	No. of RS-232 Drivers/Rx	No. of Ext. Caps	Nominal Cap. Value ( $\mu F$ )	SHDN & Three-State	Rx Active in SHDN	Data Rate (kbps)	Features
0	-5	2/2	4	0.1	No	—	120	Ultra-low-power, industry-standard product
2	+5	2/2	4	0.1	Yes	—	200	Low-power shutdown
3 (MAX213)	-5	4/5	4	1.0 (0.1)	Yes	✓	120	MAX211 and receivers active in shutdown
5	+5	5/5	0	—	Yes	✓	120	Available in SO
0 (MAX200)	+5	5/0	4	1.0 (0.1)	Yes	—	120	5 drivers with shutdown
1 (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
2 (MAX202)	+5	2/2	4	1.0 (0.1)	No	—	120 (64)	Industry standard
2A	+5	2/2	4	0.1	No	—	200	Higher slew rate, small caps
3 (MAX203)	+5	2/2	0	—	No	—	120	No external caps
3A	+5	2/2	0	—	No	—	200	No external caps, high slew rate
4 (MAX204)	+5	4/0	4	1.0 (0.1)	No	—	120	Replaces 1488
5 (MAX205)	+5	5/5	0	—	Yes	—	120	No external caps
6 (MAX206)	+5	4/3	4	1.0 (0.1)	Yes	—	120	Shutdown, three state
7 (MAX207)	+5	5/3	4	1.0 (0.1)	No	—	120	Complements IBM PC serial port
8 (MAX208)	+5	4/4	4	1.0 (0.1)	No	—	120	Replaces 1488 and 1489
9 (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
0	+5	5/5	4	1.0	Yes	—	120	DIP or flatpack package
1 (MAX211)	+5	4/5	4	1.0 (0.1)	Yes	—	120	Complete IBM PC serial port
2	+5	2/2	4	0.1	Yes	✓	200	Separate shutdown and enable
3	+5	2/2	4	0.1	No	—	200	Open-line detection simplifies cabling
4	+5	8/10	4	1.0	No	—	120	High slew rate
5	+5	5/10	0	—	Yes	✓	120	High slew rate, int. caps, two shutdown modes
6	+5	6/10	0	—	Yes	✓	120	High slew rate, int. caps, three shutdown modes
7	+5	8/9	0	—	Yes	✓	120	High slew rate, int. caps, nine operating modes
8	+5	8/8	4	1.0	Yes	✓	120	High slew rate, selective half-chip enables
9	+5	6/10	4	1.0	Yes	✓	120	Available in quad flatpack package

**MAXIM**

Maxim Integrated Products 1

Free samples and the latest literature, visit [www.maxim-ic.com](http://www.maxim-ic.com) or phone 1-800-998-8800.  
Small orders, phone 1-800-835-8769.

# ±V-Powered, Multichannel RS-232 Drivers/Receivers

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

Supply Voltage (V <sub>CC</sub> )	-0.3V to +6V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW
Input Voltages	-0.3V to (V <sub>CC</sub> + 0.3V)	16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
Receiver Output Short-Circuited to GND	Continuous	16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
Excess Power Dissipation (T <sub>A</sub> = +70°C)	±30V	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
MAX220	±25V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
(Except MAX220) (Note 1)	+15V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
(MAX220)	±13.2V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
Input Voltages	±15V	18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
Receiver Output Short-Circuited to GND	Continuous	<b>Operating Temperature Ranges</b>	
Excess Power Dissipation (T <sub>A</sub> = +70°C)	±30V	MAX2_AC, MAX2_C	0°C to +70°C
Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	MAX2_AE, MAX2_E	-40°C to +85°C
Plastic DIP (derate 11.11mW/°C above +70°C)	889mW	MAX2_AM, MAX2_M	-55°C to +125°C
		Storage Temperature Range	65°C to +160°C
		Lead Temperature (soldering, 10sec)	+300°C

1: Input voltage measured with T<sub>OUT</sub> in high impedance state, SHDN or V<sub>CC</sub> = 0V.

2: For the MAX220, V+ and V- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses in excess of 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 maximum rating conditions for extended periods may affect device reliability.

## CRITICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

V<sub>CC</sub> = +5V ±10%, C1-C4 = 0.1μF, MAX220, C1 = 0.047μF, C2-C4 = 0.33μF, T<sub>A</sub> = 1MIN to 1MAX, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
<b>32 TRANSMITTERS</b>						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Logic Threshold Low				1.4	0.8	V
Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220; V <sub>CC</sub> = 5.0V		2.4			V
Pull-Up/Input Current	All except MAX220, normal operation			5	<0	μA
	SHDN = 0V, MAX222/242, shutdown, MAX220			±0.01	±1	μA
Output Leakage Current	V <sub>CC</sub> = 5.5V, SHDN = 0V, V <sub>OUT</sub> = ±15V, MAX222/242			±0.01	±10	μA
	V <sub>CC</sub> = SHDN = 0V, V <sub>OUT</sub> = ±15V			±0.01	±10	μA
Rate				200	116	kb/s
Transmitter Output Resistance	V <sub>CC</sub> = V+ = V- = 0V, V <sub>OUT</sub> = ±2V		300	10M		Ω
Output Short-Circuit Current	V <sub>OUT</sub> = 0V		±7	±22		mA
<b>32 RECEIVERS</b>						
Input Voltage Operating Range					+30	V
Input Threshold Low	V <sub>CC</sub> = 5V	All except MAX243 R2 <sub>N</sub>	0.8	1.3		V
		MAX243 R2 <sub>N</sub> (Note 2)	-3			V
Input Threshold High	V <sub>CC</sub> = 5V	All except MAX243 R2 <sub>N</sub>		1.8	2.4	V
		MAX243 R2 <sub>N</sub> (Note 2)		0.5	0.1	V
Input Hysteresis	All except MAX243, V <sub>CC</sub> = 5V, no hysteresis in shdn. MAX243		3.2	0.5	1	V
Input Resistance			3	5	7	kΩ
CMOS Output Voltage Low	I <sub>OUT</sub> = 3.2mA			0.2	0.4	V
CMOS Output Voltage High	I <sub>OUT</sub> = -1.0mA		3.5	V <sub>CC</sub> - 0.2		V
CMOS Output Short-Circuit Current	Sourcing V <sub>OUT</sub> = GND		2	10		mA
	Sinking V <sub>OUT</sub> = V <sub>CC</sub>		13	30		mA

**MAXIM**

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

**MAX220-MAX249**
**CRITICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243 (continued)**
*V<sub>CC</sub> = +5V ±10%, C1-C4 = 0.1µF, MAX220: C1 = 0.047µF, C2-C4 = 0.33µF, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.*

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
CMOS Output Leakage Current	SHDN = V <sub>CC</sub> or EN = V <sub>CC</sub> (SHDN = 0V for MAX222), 0V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub>			±0.05	±10	µA
Input Threshold Low	MAX242			1.4	0.8	V
Input Threshold High	MAX242		2.0	1.4		V
Operating Supply Voltage			4.5		5.5	V
Supply Current (SHDN = V <sub>CC</sub> ), see 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 <sub>A</sub> = +25°C		0.1	10	µA
		T <sub>A</sub> = 0°C to +70°C		2	50	
		T <sub>A</sub> = -40°C to +85°C		2	50	
		T <sub>A</sub> = -55°C to +125°C		35	100	
$\bar{N}$ Input Leakage Current	MAX222/242				±1	µA
$\bar{N}$ Threshold Low	MAX222/242			1.4	0.8	V
$\bar{N}$ Threshold High	MAX222/242		2.0	1.4		V
Transition Slew Rate	C <sub>L</sub> = 50pF to 2500pF, R <sub>L</sub> = 3kΩ to 7kΩ, V <sub>CC</sub> = 5V, T <sub>A</sub> = +25°C, measured from +3V to -3V or -3V to +3V	MAX222/232A/233A/242/243	6	12	30	V/µs
		MAX220	15	3	30	
Driver Propagation Delay to RS-232 (normal operation), see 1	t <sub>PHLT</sub>	MAX222/232A/233A/242/243		1.3	3.5	µs
		MAX220		4	10	
	t <sub>PLHT</sub>	MAX222/232A/233A/242/243		1.5	3.5	
		MAX220		5	10	
Driver Propagation Delay to TLL (normal operation), see 2	t <sub>PHR</sub>	MAX222/232A/233A/242/243		0.5	1	µs
		MAX220		0.6	3	
	t <sub>PLHR</sub>	MAX222/232A/233A/242/243		0.6	1	
		MAX220		0.8	3	
Driver Propagation Delay to TLL (shutdown), Figure 2	t <sub>PHLS</sub>	MAX242		3.5	10	µs
	t <sub>PLHS</sub>	MAX242		2.5	10	
Driver-Output Enable Time, Figure 3	t <sub>ER</sub>	MAX242		125	500	ns
Driver-Output Disable Time, Figure 3	t <sub>DR</sub>	MAX242		160	500	ns
Driver-Output Enable Time (N goes high), Figure 4	t <sub>ET</sub>	MAX222/242, 0.1µF caps (includes charge pump start-up)		250		µs
Driver-Output Disable Time (N goes low), Figure 4	t <sub>DF</sub>	MAX222/242, 0.1µF caps		600		ns
Driver + to - Propagation Delay Difference (normal operation)	t <sub>PHLT</sub> - t <sub>PLHT</sub>	MAX222/232A/233A/242/243		200		ns
		MAX220		2000		
Driver + to - Propagation Delay Difference (normal operation)	t <sub>PHR</sub> - t <sub>PLHR</sub>	MAX222/232A/233A/242/243		100		ns
		MAX220		225		

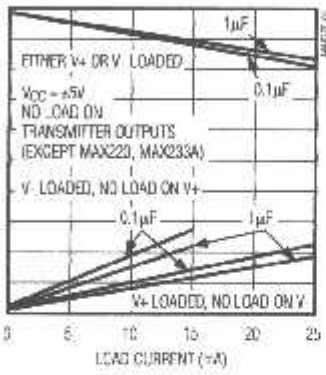
 Note: MAX243 R2<sub>OUT</sub> is guaranteed to be low when R2<sub>IN</sub> is ≥ 0V or is floating.

# ±5V-Powered, Multichannel RS-232 Drivers/Receivers

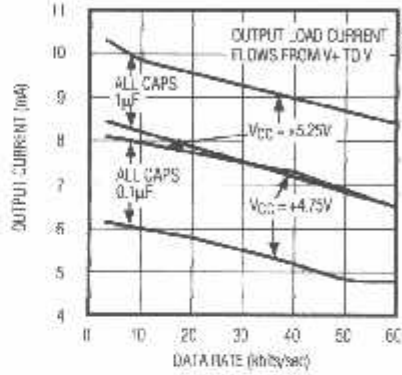
## Typical Operating Characteristics

### MAX220/MAX222/MAX232A/MAX233A/MAX242/MAX243

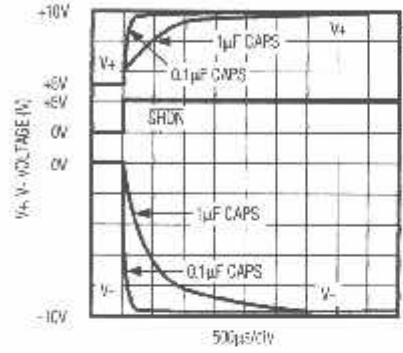
OUTPUT VOLTAGE vs. LOAD CURRENT



AVAILABLE OUTPUT CURRENT vs. DATA RATE



MAX222/MAX242 OM-TIME EXITING SHUTDOWN



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

**MAX220-MAX249**

## ABSOLUTE MAXIMUM RATINGS—MAX223/MAX230-MAX241

Input Voltages	-0.3V to +6V ( $V_{DD} - 0.3V$ ) to +14V +0.3V to -14V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)..... 800mW 24-Pin Wide SO (derate 11.76mW/°C above +70°C)..... 841mW 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 18-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 Sidebrazed (derate 20.0mW/°C above +70°C)..... 1.6W 28-Pin SSOP (derate 9.52mW/°C above +70°C)..... 762mW
Output Voltages	0.3V to ( $V_{DD} + 0.3V$ ) ±30V	Operating Temperature Ranges MAX2__C..... 0°C to +70°C MAX2__E..... -40°C to +85°C MAX2__M..... -55°C to +125°C
Input Voltages	( $V_+ + 0.3V$ ) to ( $V_- - 0.3V$ ) -0.3V to ( $V_{CC} + 0.3V$ )	Storage Temperature Range..... -65°C to +160°C Lead Temperature (soldering, 10sec)..... +300°C
Circuit Duration, $T_{OUT}$	Continuous	
Maximum Power Dissipation ( $T_A = +70^\circ\text{C}$ )		
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C).....	800mW	
18-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	
28-Pin Wide SO (derate 9.52mW/°C above +70°C).....	762mW	

Values 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 maximum rating conditions for extended periods may affect device reliability.

## CRITICAL CHARACTERISTICS—MAX223/MAX230-MAX241

MAX223/230/232/234/236/237/238/240/241,  $V_{CC} = +5V \pm 10\%$ ; MAX233/MAX235,  $V_{CC} = 5V \pm 5\%$ ,  $C_1-C_4 = 1.0\mu\text{F}$ ; MAX231/MAX239,  $5V \pm 10\%$ ;  $V_+ = 7.5V$  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
Power Supply Current	No load, $T_A = +25^\circ\text{C}$	MAX232/233	5	10	mA
		MAX223/230/234-238/240/241	7	15	
		MAX231/239	0.4	1	
Power-Supply Current		MAX231	1.8	5	mA
		MAX239	5	15	
Quiescent Supply Current	$T_A = +25^\circ\text{C}$	MAX223	15	50	μA
		MAX230/235/236/240/241	1	10	
Logic Threshold Low	$T_{IN}$ : EN, SHDN (MAX233); EN, SHDN (MAX230/235-241)			0.8	V
	$T_{IN}$	2.0			
Logic Threshold High	EN, SHDN (MAX223); EN, SHDN (MAX230/235/236/240/241)	2.4			V
Pull-Up Current	$T_{IN} = 0V$		15	200	μA
Receiver Input Voltage Operating Range		-30		30	V

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

### CRITICAL CHARACTERISTICS—MAX223/MAX230—MAX241 (continued)

MAX223/230/232/234/236/237/238/240/241,  $V_{CC} = -5V \pm 10\%$ ; MAX233/MAX235,  $V_{CC} = 5V \pm 5\%$ ,  $C_1$ – $C_4 = 1.0\mu F$ ; MAX231/MAX239,  $V_{CC} = 5V \pm 10\%$ ;  $V_+ = 7.5V$  to  $13.2V$ ;  $T_A = T_{MIN}$  to  $T_{MAX}$ ; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS	
32 Input Threshold Low	$T_A = +25^\circ C$ , $V_{CC} = 5V$	Normal operation SHDN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)	0.8	1.2		V	
		Shutdown (MAX223) SHDN = 0V, EN = 5V ( $R_{4IN}$ , $R_{5IN}$ )	0.6	1.5			
32 Input Threshold High	$T_A = +25^\circ C$ , $V_{CC} = 5V$	Normal operation SHDN = 5V (MAX223) SHDN = 0V (MAX235/236/240/241)		1.7	2.4	V	
		Shutdown (MAX223) SHDN = 0V, EN = 5V ( $R_{4IN}$ , $R_{5IN}$ )		1.5	2.1		
32 Input Hysteresis	$V_{CC} = 5V$ , no hysteresis in shutdown		0.2	0.6	1.0	V	
32 Input Resistance	$T_A = +25^\circ C$ , $V_{CC} = 5V$		3	5	7	k $\Omega$	
CMOS Output Voltage Low	$I_{OUT} = 1.8mA$ (MAX231/232/233), $I_{OUT} = 3.2mA$				0.4	V	
CMOS Output Voltage High	$I_{OUT} = -1mA$		3.5	$V_{CC} - 0.4$		V	
CMOS Output Leakage Current	$0V \leq R_{OUT} \leq V_{CC}$ ; EN = 0V (MAX223); EN = $V_{CC}$ (MAX235–241)			0.05	$\pm 10$	$\mu A$	
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 SHDN = 0V (MAX223)	PHLS		4	40	$\mu s$
			PLHS		6	40	
		MAX223/MAX230/MAX231–241, $T_A = +25^\circ C$ , $V_{CC} = 5V$ , $R_L = 3k\Omega$ to $7k\Omega$ , $C_L = 50pF$ to $2500pF$ , measured from $+3V$ to $-3V$ or $-3V$ to $+3V$		3	5.1	30	
Transition Region Slew Rate		MAX231/MAX232/MAX233, $T_A = +25^\circ C$ , $V_{CC} = 5V$ , $R_L = 3k\Omega$ to $7k\Omega$ , $C_L = 50pF$ to $2500pF$ , measured from $+3V$ to $-3V$ or $-3V$ to $+3V$		4	30	V/ $\mu s$	
Driver Output Resistance	$V_{CC} - V_+ - V_- = 0V$ , $V_{OUT} = \pm 2V$		300			$\Omega$	
Driver Output Short-Circuit Current				$\pm 10$		mA	

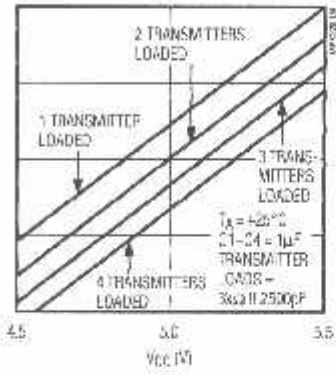
# +5V-Powered, Multichannel RS-232 Drivers/Receivers

## Typical Operating Characteristics

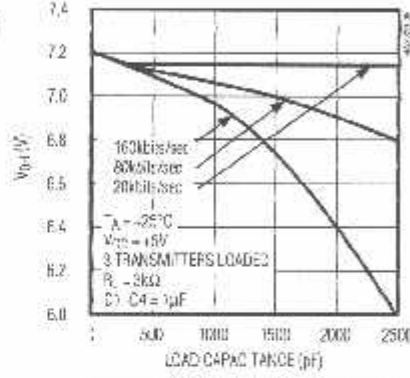
MAX220-MAX249

### MAX223/MAX230-MAX241

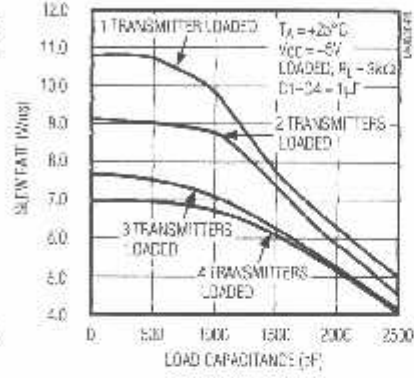
**TRANSMITTER OUTPUT VOLTAGE ( $V_{OH}$ ) vs.  $V_{CC}$**



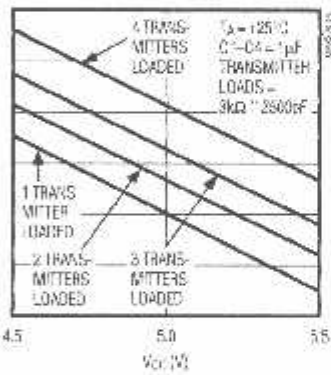
**TRANSMITTER OUTPUT VOLTAGE ( $V_{OH}$ ) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES**



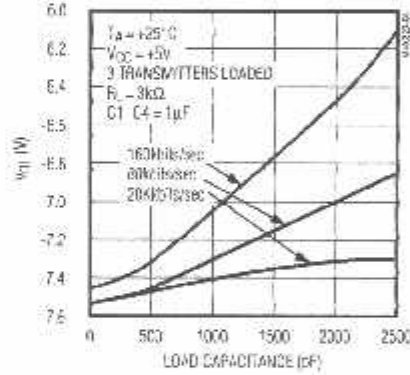
**TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE**



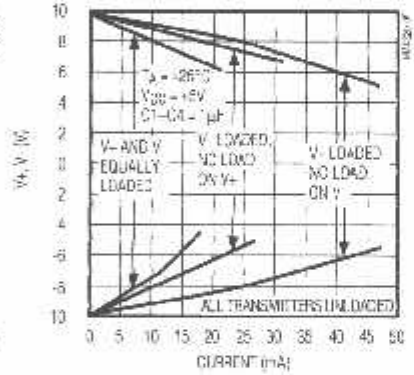
**TRANSMITTER OUTPUT VOLTAGE ( $V_{OL}$ ) vs.  $V_{CC}$**



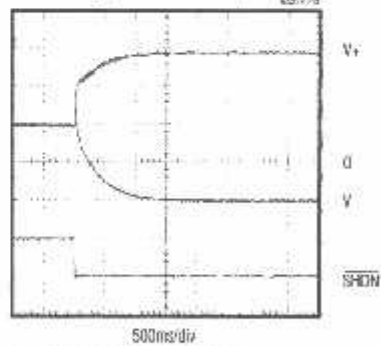
**TRANSMITTER OUTPUT VOLTAGE ( $V_{OL}$ ) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES**



**TRANSMITTER OUTPUT VOLTAGE ( $V_+$ ,  $V_-$ ) vs. LOAD CURRENT**



**$V_+$ ,  $V_-$  WHEN EXITING SHUTDOWN ( $1\mu\text{F}$  CAPACITORS)**



\*SHUTDOWN POLARITY IS REVERSE FOR NON-MAX241 PARTS

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

## ABSOLUTE MAXIMUM RATINGS—MAX225/MAX244—MAX249

Supply Voltage (V <sub>CC</sub> )	0.3V to +6V
Input Voltages	
EN <sub>A</sub> , EN <sub>B</sub> , EN <sub>F</sub> , EN <sub>T</sub> , EN <sub>RA</sub> , EN <sub>TA</sub> , EN <sub>TB</sub>	-0.3V to (V <sub>CC</sub> + 0.3V)
Outputs	
(Note 3)	±25V
	±15V
Output Voltage (one output at a time) to GND	0.3V to (V <sub>CC</sub> + 0.3V)
Output Current (one output at a time) to GND	Continuous
Output Current (one output at a time) to GND	Continuous

Continuous Power Dissipation (T <sub>A</sub> = +70°C)	
28-Pin Wide SO (derate 12.50mW/°C above +70°C)	1W
40-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	611mW
44-Pin PLCC (derate 13.33mW/°C above +70°C)	1.07W
Operating Temperature Ranges	
MAX225C, MAX24_C	0°C to +70°C
MAX225E, MAX24_E	-40°C to +85°C
Storage Temperature Range	65°C to +160°C
Lead Temperature (soldering, 10sec)	+300°C

1: Input voltage measured with transmitter output in a high-impedance state, shutdown, or V<sub>CC</sub> = 0V.

2: 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 maximum rating conditions for extended periods may affect device reliability.

## CRITICAL CHARACTERISTICS—MAX225/MAX244—MAX249

25: V<sub>CC</sub> = 5.0V ±5%, MAX244—MAX249, V<sub>CC</sub> = +5.0V ±10%, external capacitors C1—C4 = 1μF; T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>; unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
<b>32 TRANSMITTERS</b>					
Logic Threshold Low			1.4	0.8	V
Logic Threshold High		2	1.4		V
Input Pull-Up/Input Current	Tables 1a–1d		10	50	μA
	Normal operation				
	Shutdown		±0.3	–	
Rate	Tables 1a–1d, normal operation		120	64	kbits/sec
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND	±5	±7.5		V
Input Leakage Current (shutdown)	Tables 1a–1d		±0.01	±25	μA
	EN <sub>A</sub> , EN <sub>B</sub> , EN <sub>T</sub> , EN <sub>TA</sub> , EN <sub>TB</sub> = V <sub>CC</sub> , V <sub>OUT</sub> = ±15V				
	V <sub>CC</sub> = 0V, V <sub>OUT</sub> = ±15V		±0.01	±25	
Transmitter Output Resistance	V <sub>CC</sub> = V <sub>I</sub> = V <sub>O</sub> = 0V, V <sub>OUT</sub> = ±2V (Note 4)	300	10M		Ω
Output Short-Circuit Current	V <sub>OUT</sub> = 0V	±7	±30		mA
<b>32 RECEIVERS</b>					
Input Voltage Operating Range				±25	V
Input Threshold Low	V <sub>CC</sub> = 5V	0.8	1.3		V
Input Threshold High	V <sub>CC</sub> = 5V		1.8	2.4	V
Input Hysteresis	V <sub>CC</sub> = 5V	0.2	0.5	1.0	V
Input Resistance		3	5	7	kΩ
MOS Output Voltage Low	I <sub>OUT</sub> = 3.2mA		0.2	0.4	V
MOS Output Voltage High	I <sub>OUT</sub> = -1.0mA	3.5	V <sub>CC</sub> - 0.2		V
MOS Output Short-Circuit Current	Sourcing V <sub>OUT</sub> = GND	2	-10		mA
	Sinking V <sub>OUT</sub> = V <sub>CC</sub>	10	30		
MOS Output Leakage Current	Normal operation, outputs disabled, Tables 1a–1d, 0V ≤ V <sub>OUT</sub> ≤ V <sub>CC</sub> , EN <sub>RL</sub> = V <sub>CC</sub>		±0.05	±0.10	μA

**MAXIM**



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

**MAX220-MAX249**

## CRITICAL CHARACTERISTICS—MAX225/MAX244-MAX249 (continued)

225:  $V_{CC} = 5.0V \pm 5\%$ ; MAX244-MAX249,  $V_{CC} = -5.0V \pm 10\%$ ; external capacitors C1-C4 = 1 $\mu$ F;  $T_A = T_{MIN}$  to  $T_{MAX}$ ; unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
<b>OPER SUPPLY AND CONTROL LOGIC</b>						
Operating Supply Voltage		MAX225	4.75		5.25	V
		MAX244-MAX249	4.5		5.5	
Supply Current (normal operation)	No load	MAX225		10	20	mA
		MAX244-MAX249		11	30	
	3k $\Omega$ loads on all outputs	MAX225		40		
		MAX244-MAX249		57		
Shutdown Supply Current	$T_A = +25^\circ\text{C}$			8	25	$\mu$ A
	$T_A = T_{MIN}$ to $T_{MAX}$				50	
Input	Leakage current				$\pm 1$	$\mu$ A
	Threshold low			1.4	0.8	V
	Threshold high		2.4	1.4		
<b>CHARACTERISTICS</b>						
Transition Slew Rate	$C_L = 50\text{pF}$ to $2500\text{pF}$ , $R_L = 3\text{k}\Omega$ to $7\text{k}\Omega$ , $V_{CC} = 5V$ , $T_A = +25^\circ\text{C}$ measured from $+3V$ to $-3V$ or $-3V$ to $+3V$		5	13	30	V/ $\mu$ s
Driver Propagation Delay to RS-232 (normal operation), see 1	$t_{PHLT}$			1.3	3.5	$\mu$ s
	$t_{PLHT}$			1.5	3.5	
Receiver Propagation Delay to RS-232 (normal operation), see 2	$t_{PHLF}$			3.6	1.9	$\mu$ s
	$t_{PLHF}$			3.6	1.5	
Receiver Propagation Delay to RS-232 (low-power mode), see 2	$t_{PHLS}$			0.6	10	$\mu$ s
	$t_{PLHS}$			3.0	10	
Driver + to - Propagation Difference (normal operation)	$t_{PHLT} - t_{PLHT}$			350		ns
Receiver + to - Propagation Difference (normal operation)	$t_{PHLF} - t_{PLHF}$			350		ns
Receiver-Output Enable Time, Figure 3	$t_{ER}$			100	500	ns
Receiver-Output Disable Time, Figure 3	$t_{DR}$			100	500	ns
Driver Enable Time	$t_{ET}$	MAX246-MAX249 (excludes charge-pump start-up)		5		$\mu$ s
		MAX225/MAX245-MAX249 (includes charge-pump start-up)		10		ms
Driver Disable Time, Figure 4	$t_{DT}$			100		ns

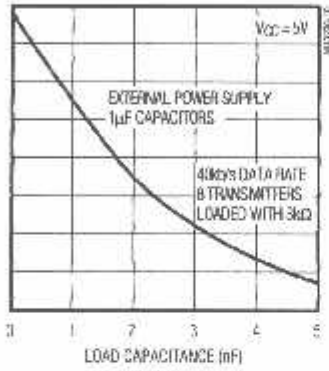
3: The 300 $\Omega$  minimum specification complies with EIA/TIA-232E, but the actual resistance when in shutdown mode or  $V_{CC} = -3V$  is 10M $\Omega$  as is implied by the leakage specification.

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

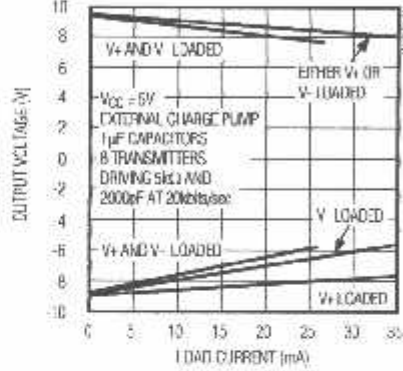
## Typical Operating Characteristics

### MAX225/MAX244-MAX249

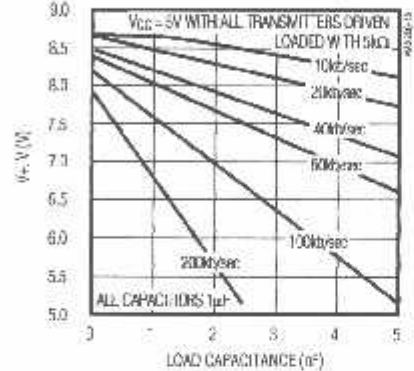
**TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE**



**OUTPUT VOLTAGE vs. LOAD CURRENT FOR V+ AND V-**

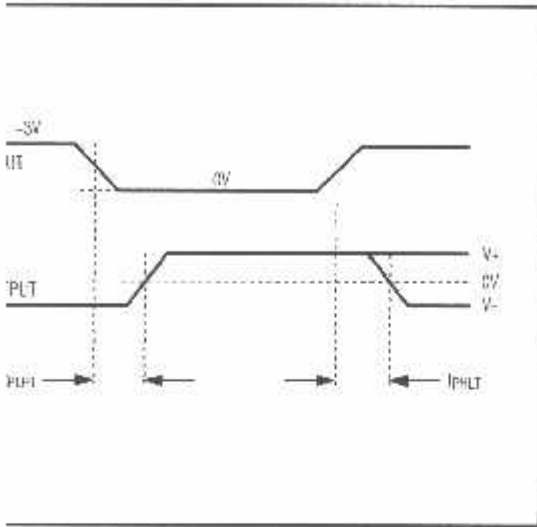


**TRANSMITTER OUTPUT VOLTAGE (V+, V-) vs. LOAD CAPACITANCE AT DIFFERENT DATA RATES**



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

MAX220-MAX249



1. Transmitter Propagation-Delay Timing

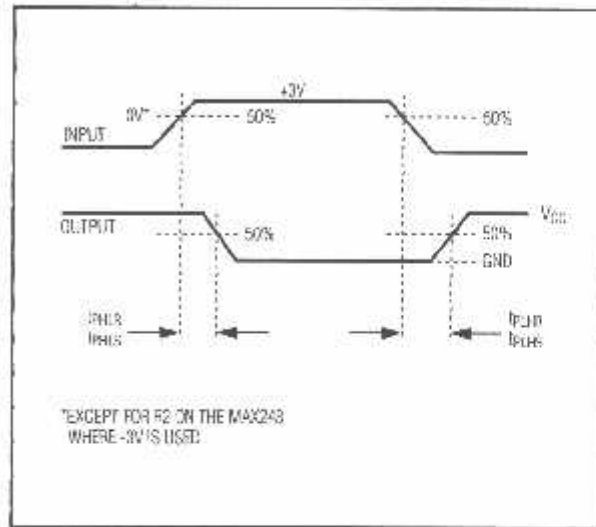
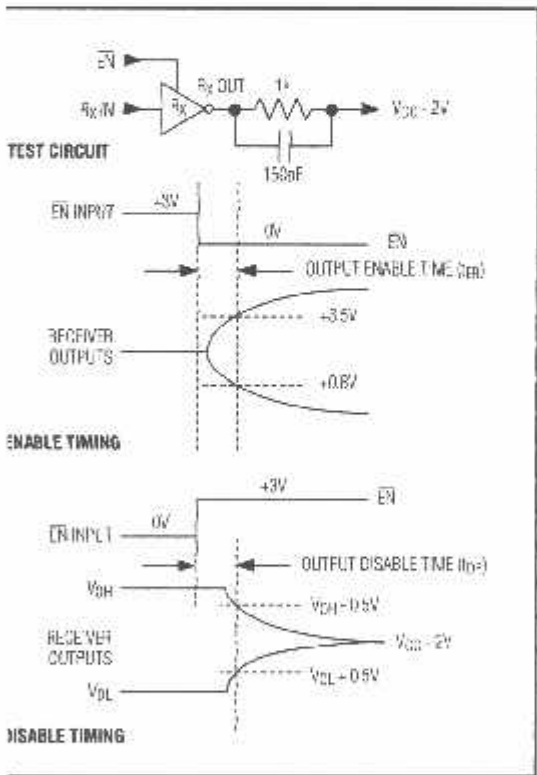


Figure 2. Receiver Propagation-Delay Timing



3. Receiver-Output Enable and Disable Timing

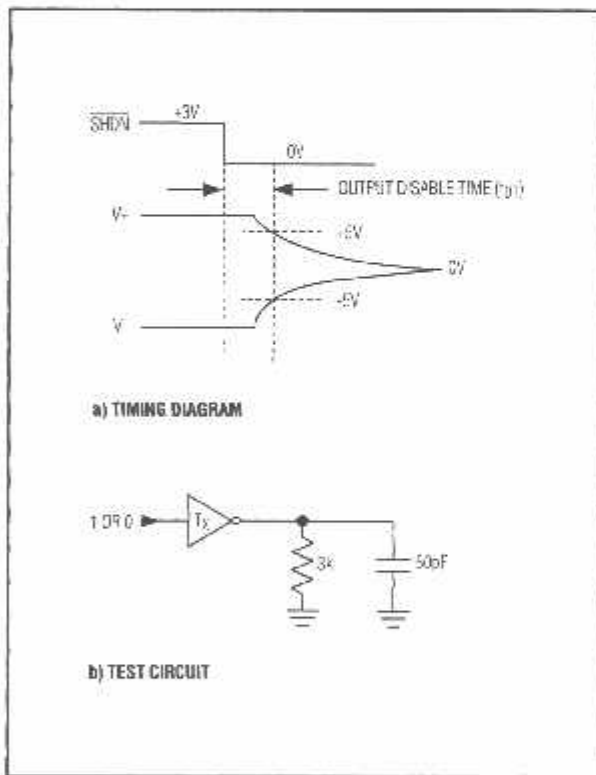


Figure 4. Transmitter-Output Disable Timing

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

Table 1a. MAX245 Control Pin Configurations

ENT	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

NT	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

IA	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

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

**MAX220-MAX249**

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

ENTB	ENRA	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	Normal Operation	All Active	All Active	All Active	All Active	
0	0	1	Normal Operation	All Active	All Active	All Active	All 3-State, except RB5 stays active on MAX247	
0	1	0	Normal Operation	All Active	All Active	All 3-State	All Active	
0	1	1	Normal Operation	All Active	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247	
1	0	0	Normal Operation	All Active	All 3-State	All Active	All Active	
1	0	1	Normal Operation	All Active	All 3-State	All Active	All 3-State, except RB5 stays active on MAX247	
1	1	0	Normal Operation	All Active	All 3-State	All 3-State	All Active	
1	1	1	Normal Operation	All Active	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247	
0	0	0	Normal Operation	All 3-State	All Active	All Active	All Active	
0	0	1	Normal Operation	All 3-State	All Active	All Active	All 3-State, except RB5 stays active on MAX247	
0	1	0	Normal Operation	All 3-State	All Active	All 3-State	All Active	
0	1	1	Normal Operation	All 3-State	All Active	All 3-State	All 3-State, except RB5 stays active on MAX247	
1	0	0	Shutdown	All 3-State	All 3-State	Low-Power Receive Mode	Low-Power Receive Mode	
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	0	Shutdown	All 3-State	All 3-State	All 3-State	Low-Power Receive Mode	
1	1	1	Shutdown	All 3-State	All 3-State	All 3-State	All 3-State, except RB5 stays active on MAX247	

# ±V-Powered, Multichannel RS-232 Drivers/Receivers

## Detailed Description

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

### Dual Charge-Pump Voltage Converter

MAX220–MAX249 have two internal charge-pumps convert +5V to ±10V (unloaded) for RS-232 driver operation. The first converter uses capacitor C1 to double +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.

The amount of power may be drawn from the +10V and -10V (V-) outputs to power external circuitry (see *Typical Operating Characteristics* section), but on the MAX225 and MAX245–MAX247, where the pins are not available. V+ and V- are not regulated, and 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 drawing 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- for 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+ instead of using the internal charge pump to generate +10V, the C1 capacitor must not be installed and the DN pin must be tied to VCC. This is because V+ is internally connected to VCC in shutdown mode.

### RS-232 Drivers

Typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver and VCC = 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. Conditions include a minimum 3kΩ load, VCC = +4.5V, and a maximum operating temperature. Unloaded driver output voltage ranges from (V+ -1.3V) to (V- +0.5V).

Input thresholds are both TTL and CMOS compatible. Outputs of unused drivers can be left unconnected (400kΩ input pull-up resistors to VCC are built in for the MAX220). The pull-up resistors force the outputs of unused drivers low because all drivers invert. External input pull-up resistors typically source 12μA, but in shutdown mode where the pull-ups are disabled. Driver outputs turn off and enter a high-impedance state—where leakage current is typically 100 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 VCC.

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 (VCC + 6V). Below -0.5V, the transmitter is diode clamped to ground with 1kΩ series impedance. The transmitter is also zener clamped to approximately VCC + 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 30Ω 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	SHDN	EN	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

MAXIM

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

MAX220-MAX249

typical 5k $\Omega$  values. The receivers implement Type 1 interpretation of the fault conditions of V.28 and V.232E.

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

### Low-Power Receive Mode

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 activates it on command and prepare it for communication at faster data rates. This operation conserves minimum power.

### Negative Threshold—MAX243

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

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

Members of the RS-232 family implement the optional cable fault protection as specified by EIA/TIA specifications. This means a receiver output goes active whenever its input is driven negative, left floating, or tied to ground. The high output tells the serial communications IC to stop sending data. To avoid this, control lines must either be driven or connected 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 2.5 $\mu$ s 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 (EN for the MAX242 and EN for the MAX223) that allows receiver output control independent of SHDN (SHDN for MAX241). With all other devices, 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 25 $\mu$ A 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 ENR input. On the MAX245, eight of the receiver outputs are controlled by the 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 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.

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

Tables 1a–1d define the control states. The MAX244 has four control pins and is not included in those tables.

MAX246 has ten receivers and eight drivers with four 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 ( $\overline{ENA} = \overline{ENB} = +5V$ ).

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  $\overline{ENTB}$ .

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 20kbts/sec.

### Applications Information

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



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

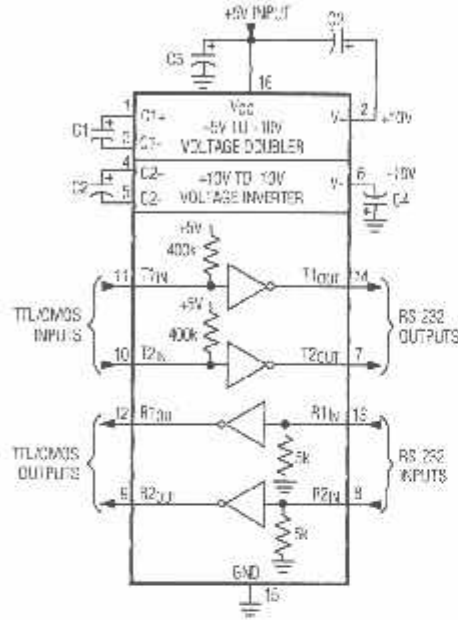
**MAX220-MAX249**

TOP VIEW



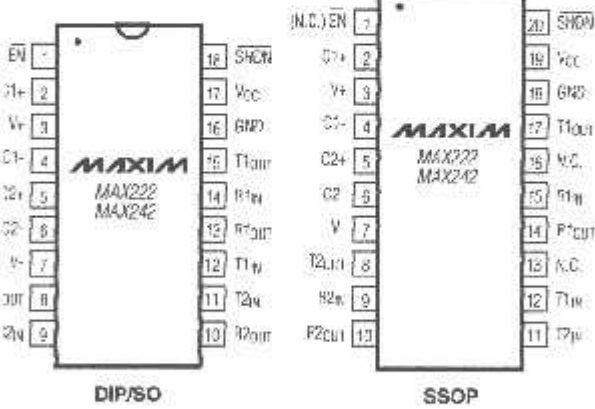
DIP/SO

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



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

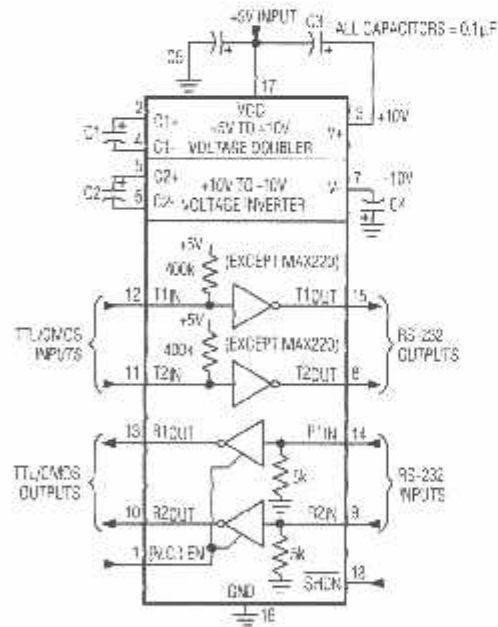
TOP VIEW



DIP/SO

SSOP

1. ARE FOR MAX222 ONLY.  
PIN NUMBERS IN TYPICAL OPERATING CIRCUIT ARE FOR DIP/SO PACKAGES ONLY.



6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

1.3.2. AT Commands According to GSM 07.05 for SMS

The GSM 07.05 commands are used for operating the SMS functions of the GSM mobile phone. The GSM module MOBILE supports the SMS PDU mode.

AT+CSMS		Selection of message service Revision according to GSM 07.05 Version 5.0.0
Test command AT+CSMS=?	Response +CSMS: (list of supported <service>S)  Parameter <service> 0 GSM 3.40 and 3.41 1 GSM 3.40 and 3.41 and compatibility of the AT command syntax for phase 2+  NOTE: Deactivating the phase 2+ compatibility is only possible if the direct output of short messages +CNMI=1,2 or +CNMI=1,3 is not activated. If necessary, the latter should be deactivated first.	
Read command AT+CSMS?	Response +CSMS: <service>,<mt>,<mo>,<bm>  Parameter <service> 0 GSM 3.40 and 3.41 <mt> 1 Mobile terminated messages Type supported <mo> 1 Mobile originated messages Type supported <bm> 1 Broadcast type messages Type supported 0 Type not supported	
Write command AT+CSMS= <service>	Parameter <service> 0 GSM 3.40 and 3.41  Response +CSMS: <mt>,<mo>,<bm> OK/ERROR/+CMS ERROR	

AT+CPMS		Selection of SMS memory Revision according to GSM 07.05 Version 4.7.0	
Test command AT+CPMS=?	Response +CPMS: (list of supported <mem1>s),( list of supported <mem2>s) ,(list of supported <mem3>s)	Parameter <mem1> Memory from which messages are read and deleted "SM" SIM-messages memory	
		<mem2> Memory to which messages are written and sent "SM" SIM-messages memory	
		<mem3> Memory in which received messages are stored, if forwarding to the ("CNMI") "SM" SIM-messages memory	PC is not set
Read command AT+CPMS?	Response +CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2> <total2> ,<mem3>,<used3> <total3>	Parameter <memx> Memory from which messages are read and deleted <usedx> Number of messages currently in <memx> <totalx> Number of storable messages in <memx>	
Write command AT+CPMS= <mem1> [,<mem2> [,<mem3>]]		Parameter <mem1> See Test command <mem2> See Test command <mem3> See Test command	
	Response +CPMS: <used1> <total1>,<used2>,<total3>,<used3>,<total3> OK/ERROR/+CMS ERROR		

AT+CMGF		SMS format	
Test command AT+CMGF=?	Response +CMGF: (list of supported <mode>s)	Parameter <mode>: 0 PDU mode	
Read command AT+CMGF?	Response +CMGF: <mode>	Parameter <mode>: 0 PDU mode	
Write command AT+CMGF=[<mode>]		Parameter <mode>: 0 PDU mode	
	Response OK/ERROR		

AT+CSCA	Address of the SMS service center
Test command AT+CSCA=?	Response OK
Read command AT+CSCA?	Response +CSCA: <sca>, <tosca>  Parameter <sca>                   Service-center address in string format <tosca>                Service-center address format
Write command AT+CSCA= <sca>[, <tosca>]	Parameter <sca>                   Service-center address in string format <tosca>                Service-center address format  Response OK/ERROR

AT+CNMI	Display new incoming SMS Revision according to GSM 07.05 Version 4.7.0
Test command AT+CNMI=?	Response +CNMI: (list of supported <mode>s),(list of supported <mt>s),(list of supported <bm>s),(list of supported <ds>s),(list of supported <bfr>s)  Parameter <mode> 0            Buffers unexpected messages (but is equivalent to rejecting; see <bfr>) 1            Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved. Otherwise forward them directly to the TE (only with S25ff) 2            Buffers unexpected messages if serial interface is occupied, otherwise they are output (only models before S25)  <mt>                0            Suppresses unexpected messages for incoming short messages 1            Unexpected messages of a received short message (SMS-DELIVER) that is stored on a chip card are output in the form +CMT: <mem>,<index> 2            Unexpected messages of a received short message (SMS-DELIVER) (except class 2 and the message "Waiting Indication Group: store message") are output in the form +CMT: [<alpha>],<length><CR><LF><pdu> (<alpha> is not supported) 3            Class 2 and the message "Waiting Indication Group: store message" are output as <mt>=1 3            Unexpected messages of a received short message (SMS-DELIVER) class 3 are output as <mt>=2. Messages with other data coding schemes are output as <mt>=1.  <b>NOTE:</b> <mt>=2 and <mt>=3 are not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1  <bm>                0            Suppresses unexpected messages for incoming cell broadcast messages 2            Outputs unexpected messages for cell broadcast messages in the form +CBM: <length><CR><LF><pdu>  <ds>                0            Suppresses unexpected messages for incoming SMS status reports 2            Outputs unexpected messages for SMS status reports in the form +CDS: <length><CR><LF><pdu>

	<p>&lt;bfr&gt; 1 Buffered unexpected messages are rejected when switching from &lt;mode&gt; 0 to &lt;mode&gt; 2</p> <p>&lt;mem&gt; See +CPMS</p> <p>&lt;index&gt; Index of the record on the chip card</p> <p>&lt;alpha&gt; alphanumeric representation of the sender address</p> <p>&lt;length&gt; Length of &lt;pdu&gt;</p> <p>&lt;pdu&gt; See +CMGL</p>
Read command <b>AT+CNMI?</b>	<p>Response +CNMI: &lt;mode&gt;,&lt;mt&gt;,&lt;bm&gt;,&lt;ds&gt;,&lt;bfr&gt;</p> <p>Parameter &lt;mode&gt; See Test command &lt;mt&gt; See Test command &lt;bm&gt; See Test command &lt;ds&gt; See Test command &lt;bfr&gt; See Test command</p>
Write command <b>AT+CNMI=[&lt;mode&gt;[,&lt;mt&gt;[,&lt;bm&gt;[,&lt;ds&gt;[,&lt;bfr&gt;]]]]</b>	<p>Parameter &lt;mode&gt; See Test command &lt;mt&gt; See Test command &lt;bm&gt; See Test command &lt;ds&gt; See Test command &lt;bfr&gt; See Test command</p> <p>Response OK/ERROR/+CMS ERROR</p>
	<p>Unexpected message +CMTI: &lt;mem&gt;,&lt;index&gt; Indication that new message has arrived</p> <p>+CMTI: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the short message</p> <p>+CDS: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the status report</p> <p>+CBM: &lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt; Direct output of the cell broadcast message</p>

<b>AT+CNMA</b>	<p>Acknowledgment of a short message directly output (without storing on the chip card)</p> <p><b>Revision according to GSM 07.05 Version 5.0.0</b></p> <p><i>(NOTE: This command is not possible unless the Phase 2+ compatibility has been activated by means of +CSMS=1)</i></p>
Test command <b>AT+CNMA=?</b>	<p>Response +CNMA: (list of supported &lt;n&gt;s)</p> <p>Parameter &lt;n&gt; 0 Mode of functioning analogous to GSM 07.05 text mode</p>
Write command <b>AT+CNMA[=&lt;n&gt;]</b>	<p>Parameter &lt;n&gt; See Test command</p> <p>Response OK/ERROR/+CMS ERROR: &lt;err&gt;</p>

<p><b>AT+CMGL</b> List SMS  <b>Revision according to GSM 07.05 Version 4.7.0</b></p>											
<p>Test command  <b>AT+CMGL=?</b></p>	<p>Response  +CMGL: (list of supported &lt;stat&gt;s)</p> <p>Parameter  &lt;stat&gt;</p> <table border="0"> <tr> <td>0</td> <td>'REC UNREAD': received unread messages (default)</td> </tr> <tr> <td>1</td> <td>'REC READ': received read messages</td> </tr> <tr> <td>2</td> <td>'STO UNSENT': stored unsent messages</td> </tr> <tr> <td>3</td> <td>'STO SENT': stored sent messages</td> </tr> <tr> <td>4</td> <td>'ALL': all messages</td> </tr> </table>	0	'REC UNREAD': received unread messages (default)	1	'REC READ': received read messages	2	'STO UNSENT': stored unsent messages	3	'STO SENT': stored sent messages	4	'ALL': all messages
0	'REC UNREAD': received unread messages (default)										
1	'REC READ': received read messages										
2	'STO UNSENT': stored unsent messages										
3	'STO SENT': stored sent messages										
4	'ALL': all messages										
<p>Write command  <b>AT+CMGL  [=&lt;stat&gt;]</b></p>	<p>Parameter  &lt;stat&gt; See Test command</p> <p>Response  If PDU mode (+CMGF=0) and command are successful:  +CMGL:&lt;index&gt;,&lt;stat&gt;[&lt;alpha&gt;],&lt;length&gt;  &lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;[&lt;CR&gt;&lt;LF&gt;  +CMGL: &lt;index&gt; &lt;stat&gt; [alpha],&lt;length&gt;  &lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;&lt;CR&gt;&lt;LF&gt;  [...]]</p>										
	<p>Parameter  &lt;pdu&gt; The PDU begins with the service-center address (according to GSM04.11), followed by the TPDU according to GSM03.40 in hexadecimal format  otherwise,  +CMS ERROR: &lt;err&gt;</p>										

<b>AT+CMGR</b>		<b>Read in an SMS</b> <b>Revision according to GSM 07.05 Version 4.7.0</b>
Test command <b>AT+CMGR=?</b>	Response OK	
Write command <b>AT+CMGR= &lt;index&gt;</b>	Parameter <index>                      Index of message in selected memory <mem1>	
	Response <b>If PDU mode (+CMGF=0) and command are successful:</b> +CMGR: <stat>,[<alpha>],<length><CR><LF><pdu>	
	Parameter <pdu>                              Siehe "AT+CMGL"	
	otherwise: +CMS ERROR: <err>	

<b>AT+CMGS</b>		<b>Send an SMS</b>
Test command <b>AT+CMGS=?</b>	Response OK	
Write command <b>If PDU mode (+CMGF=0) +CMGS=&lt;length&gt;&lt;CR&gt;PDU is given &lt;ctrl-Z/ESC&gt;</b>	Parameter <length>                      Length of PDU <pdu>                              See "AT+CMGL" <mr>                              Message reference	
	Response <b>If sending is successful:</b> +CMGS: <mr>	
	<b>If sending is not successful:</b> +CMS ERROR: <err>	

<b>AT+CMSS</b>		<b>Send an SMS from the SMS memory</b>
Test command <b>AT+CMSS=?</b>	Response OK	
Write command <b>+CMSS=&lt;index&gt;[,&lt;da&gt;[,&lt;tda&gt;]]</b>	Parameter <index>                      Index of message in selected memory <mem1> <da>                              Destination address in string format <tda>                              Format of destination address  <mr>                              Message reference	
	Response <b>If sending is successful:</b> +CMSS: <mr>	
	<b>If sending is not successful:</b> +CMS ERROR: <err>	

AT+CMGW		Write an SMS to the SMS memory	
Test command AT+CMGW=?	Response OK		
Write command <b>If PDU mode (+CMGF=0)</b> AT+CMGW=<length>[,<stat>]<CR> <b>PDU is given</b> <ctrl-Z/ESC>	Parameter <length> Length of PDU <stat> See command +CMGL <pdu> See "AT+CMGL" <index> Index of message in selected memory <mem1>		
	Response +CMGW: <index> +CMS ERROR: <err>		

AT+CMGD		Delete an SMS in the SMS memory	
Test command AT+CMGD=?	Response OK		
Write command AT+CMGD= <index>	Parameter <index> Index of message in the selected memory <mem1>		
	Response OK/ERROR/+CMS ERROR		

AT+CSCB		Select cell broadcast messages	
Test command AT+CSCB=?	Response +CSCB: (list of supported <mode> s)		
	Parameter <mode> 0 Accepts messages that are defined in <mids> and <dcss> 1 Does not accept messages that are defined in <mids> and <dcss>		
Test command AT+CSCB?	Response +CSCB: <mode>,<mids>,<dcss>		
	Parameter <mode> See Test command <mids> String type; combinations of CBM message IDs <dcss> String type; combinations of CBM data coding schemes		
Write command AT+CSCB=[<mode>[,<mids> >[,<dcss>]]]			

AT+CMGC		Send an SMS command	
Test command AT+CMGC=?	Response OK		
Write command <b>f PDU mode (+CMGF=0)</b> +CMGC=<length><CR> <b>PDU is given</b> <ctrl-Z/ESC>	Parameter <length> Length of PDU <pdu> See "AT+CMGL" <mr> Message reference		
	Response <b>If sending is successful:</b> +CMGC: <mr> <b>If sending is not successful:</b> +CMS ERROR: <err>		



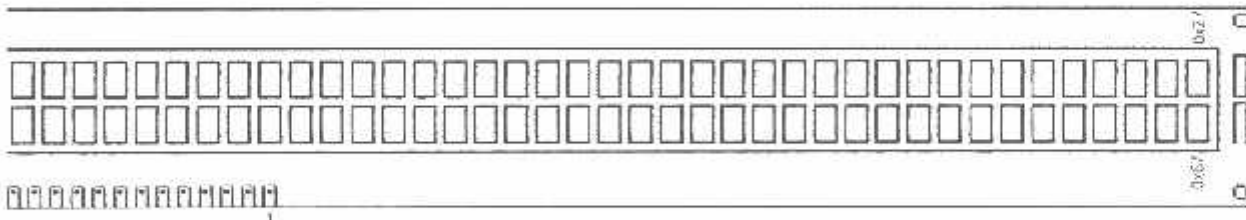
# Extended Concise LCD Data Sheet

for HD44780

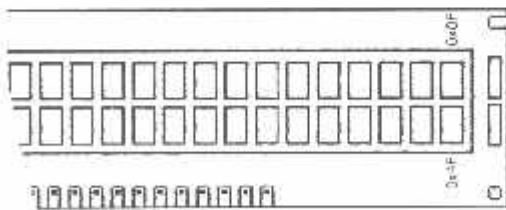
25.6.1999

Action	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clock-Cycles
	0	0	0	0	0	0	0	0	0	0	No Operation	0
Display	0	0	0	0	0	0	0	0	0	1	Clear display & set address counter to zero	165
Home	0	0	0	0	0	0	0	0	1	x	Set address counter to zero, return shifted display to original position. DDRAM contents remains unchanged.	3
Mode	0	0	0	0	0	0	0	1	I/D	S	Set cursor move direction (I/D) and specify automatic display shift (S)	3
	0	0	0	0	0	0	1	D	C	B	Turn display (D), cursor on/off (C), and cursor blinking (B)	3
Shift	0	0	0	0	0	1	S/C	R/L	x	x	Shift display or move cursor (S/C) and specify direction (R/L)	3
Function Set	0	0	0	0	1	DL	N	F	x	x	Set interface data width (DL), number of display lines (N) and character font (F)	3
RAM	0	0	0	1	CGRAM Address					Set CGRAM address. CGRAM data is sent afterwards.		3
RAM	0	0	1	DDRAM Address					Set DDRAM address. DDRAM data is sent afterwards.		3	
Flag &	0	1	BF	Address Counter					Read busy flag (BF) and address counter		0	
Write Data	1	0	Data					Write data into DDRAM or CGRAM		3		
Read Data	1	1	Data					Read data from DDRAM or CGRAM		3		
Bit	I/D	1 0	Increment Decrement						R/L	1 0	Shift to the right Shift to the left	
	S	1 0	Automatic display shift						DL	1 0	8 bit interface 4 bit interface	
	D	1 0	Display ON Display OFF						N	1 0	2 lines 1 line	
	C	1 0	Cursor ON Cursor OFF						F	1 0	5x10 dots 5x7 dots	
	B	1 0	Cursor blinking						DDRAM : Display Data RAM CGRAM : Character Generator RAM			
	S/C	1 0	Display shift Cursor move									

Display with 2 lines x 40 characters :



Display with 2 lines x 16 characters :

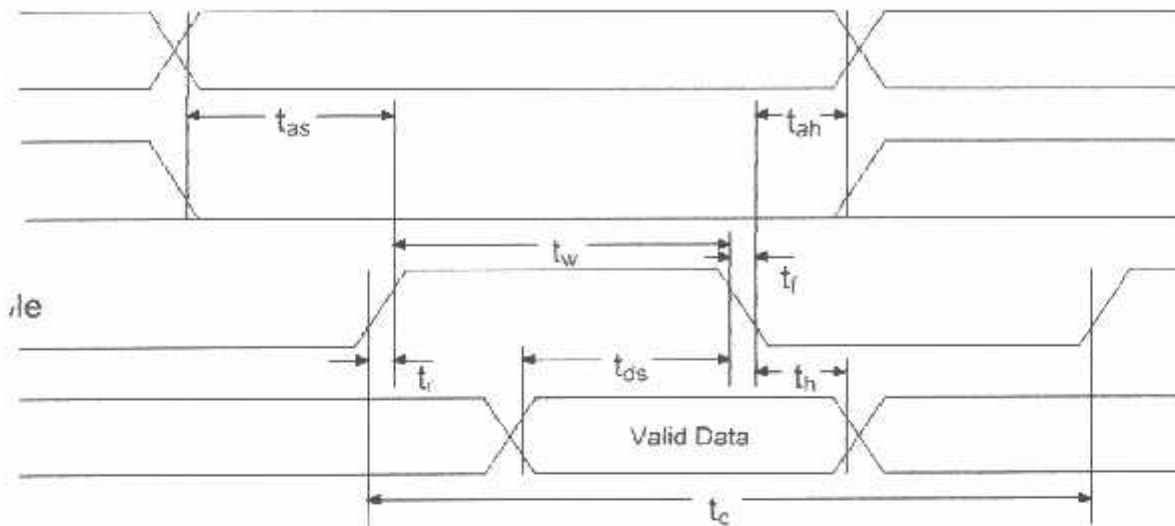


Pin No	Name	Function	Description
1	Vss	Power	GND
2	Vdd	Power	+ 5 V
3	Vee	Contrast Adj.	(-2) 0 - 5 V
4	RS	Command	Register Select
5	R/W	Command	Read / Write
6	E	Command	Enable (Strobe)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB

# Timing Characteristics

(20 to +75°C)

## Write Cycle



Cycle	$V_{DD}$	2.7 - 4.5 V <sup>(2)</sup>	4.5 - 5.5 V <sup>(2)</sup>		2.7 - 4.5 V <sup>(2)</sup>	4.5 - 5.5 V <sup>(2)</sup>	
Parameter	Symbol	Min <sup>(1)</sup>	Typ <sup>(1)</sup>	Max <sup>(1)</sup>	Unit		
Write Cycle Time	$t_c$	1000	500	-	ns		
Write Pulse Width (High)	$t_w$	450	230	-	ns		
Write Rise/Fall Time	$t_r, t_f$	-	-	25	ns	20	
Write Setup Time	$t_{as}$	60	40	-	ns		
Write Hold Time	$t_{ah}$	20	10	-	ns		
Write Setup Time	$t_{ds}$	195	80	-	ns		
Write Hold Time	$t_{dh}$	10	10	-	ns		

The above specifications are indications only (based on Hitachi HD44780). Timing will vary from manufacturer to manufacturer.

Power Supply : HD44780 S :  $V_{DD} = 4.5 - 5.5 V$   
 HD44780 U :  $V_{DD} = 2.7 - 5.5 V$

This sheet refers to specifications for the Hitachi HD44780 LCD Driver chip, which is used for most LCD displays.

Supported types are :

- 1 line x 20 characters
- 2 lines x 16 characters
- 2 lines x 20 characters
- 2 lines x 40 characters
- 4 lines x 20 characters
- 4 lines x 40 characters

©1999 by Craig Peacock, Australia  
 Peter Luethi, Switzerland

<http://www.beyondlogic.org>  
<http://www.electronic-engineering.ch>

## MM74C922 • MM74C923 16-Key Encoder • 20-Key Encoder

### General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

### Features

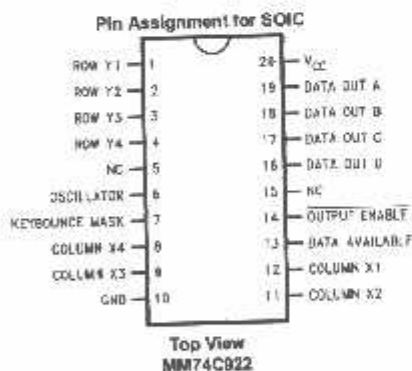
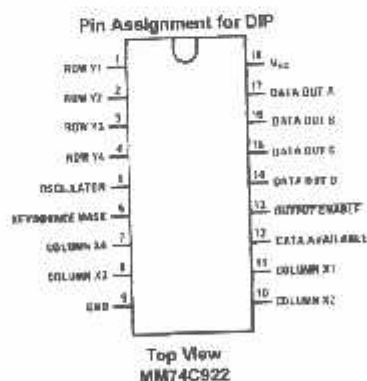
- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keyboard elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

### Ordering Code:

Order Number	Package Number	Package Description
MM74C922WM	M20R	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C922N	N18B	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

### Connection Diagrams



**Absolute Maximum Ratings**(Note 2)

Voltage at Any Pin	$V_{CC} - 0.3V$ to $V_{CC} + 0.3V$
Operating Temperature Range	-40°C to +85°C
Storage Temperature Range	-65°C to +150°C
Power Dissipation ( $P_D$ )	
Dual-In-Line	700 mW
Small Outline	500 mW
Operating $V_{CC}$ Range:	3V to 15V
$V_{CC}$	18V
Lead Temperature (Soldering, 10 seconds)	260°C

Note 2: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

**DC Electrical Characteristics**

Min/Max limits apply across temperature range unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
<b>CMOS TO CMOS</b>						
$V_{T+}$	Positive-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IN} \geq 0.7 mA$ $V_{CC} = 10V, I_{IN} \geq 1.4 mA$ $V_{CC} = 15V, I_{IN} \geq 2.1 mA$	3.0 6.0 9.0	3.8 6.8 10	4.3 8.6 12.9	V
$V_{T-}$	Negative-Going Threshold Voltage at Osc and KBM Inputs	$V_{CC} = 5V, I_{IN} \geq 0.7 mA$ $V_{CC} = 10V, I_{IN} \geq 1.4 mA$ $V_{CC} = 15V, I_{IN} \geq 2.1 mA$	0.7 1.4 2.1	1.4 3.2 5	2.0 4.0 6.0	V
$V_{NH(1)}$	Logical "1" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$	3.5 8.0 12.5	4.5 9 13.5		V
$V_{NH(0)}$	Logical "0" Input Voltage, Except Osc and KBM Inputs	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$		0.5 1 1.5	1.5 2 2.5	V
$I_{IP}$	Row Pull-Up Current at Y1, Y2, Y3, Y4 and Y5 Inputs	$V_{CC} = 5V, V_{IN} = 3.1 V_{CC}$ $V_{CC} = 10V$ $V_{CC} = 15V$		-2 -10 -22	5 20 45	$\mu A$
$V_{OUT(1)}$	Logical "1" Output Voltage	$V_{CC} = 5V, I_O = -10 \mu A$ $V_{CC} = 10V, I_O = -10 \mu A$ $V_{CC} = 15V, I_O = -10 \mu A$	4.5 9 13.5			V
$V_{OUT(0)}$	Logical "0" Output Voltage	$V_{CC} = 5V, I_O = 10 \mu A$ $V_{CC} = 10V, I_O = 10 \mu A$ $V_{CC} = 15V, I_O = 10 \mu A$			0.5 1 1.5	V
$R_{OH}$	Column "ON" Resistance at X1, X2, X3 and X4 Outputs	$V_{CC} = 5V, V_O = 0.5V$ $V_{CC} = 10V, V_O = 1V$ $V_{CC} = 15V, V_O = 1.5V$		500 300 200	1400 700 500	$\Omega$
$I_{CC}$	Supply Current Osc at 0V, (one Y low)	$V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$		0.55 1.1 1.7	1.1 1.9 2.6	mA
$I_{IN(1)}$	Logical "1" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 15V$		0.005	1.0	$\mu A$
$I_{IN(0)}$	Logical "0" Input Current at Output Enable	$V_{CC} = 15V, V_{IN} = 0V$	-1.0	-0.005		$\mu A$
<b>CMOS/LPTTL INTERFACE</b>						
$V_{NH(1)}$	Except Osc and KBM inputs	$V_{CC} = 4.75V$		$V_{CC} - 1.5$		V
$V_{NH(0)}$	Except Osc and KBM inputs	$V_{CC} = 4.75V$			0.8	V
$V_{OUT(1)}$	Logical "1" Output Voltage	$I_O = 380 \mu A$ $V_{CC} = 4.75V$ $I_O = 360 \mu A$	2.4			V

**DC Electrical Characteristics** (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$V_{OH(1)}$	Logical '1' Output Voltage	$I_O = -360 \mu A$ $V_{CC} = 4.75V$ $I_O = -360 \mu A$			0.4	V
<b>OUTPUT DRIVE (See Family Characteristics Data Sheet) (Short Circuit Current)</b>						
$I_{SOURCE}$	Output Source Current (P-Channel)	$V_{CC} = 5V, V_{OUT} = 0V,$ $T_A = 25^\circ C$	-1.75	-3.3		mA
$I_{SOURCE}$	Output Source Current (P-Channel)	$V_{OUT} = 10V, V_{OUT} = 0V,$ $T_A = 25^\circ C$	-8	-15		mA
$I_{SEK}$	Output Sink Current (N-Channel)	$V_{CC} = 5V, V_{OUT} = V_{CC},$ $T_A = 25^\circ C$	1.75	3.8		mA
$I_{SEK}$	Output Sink Current (N-Channel)	$V_{CC} = 10V, V_{OUT} = V_{CC},$ $T_A = 25^\circ C$	8	18		mA

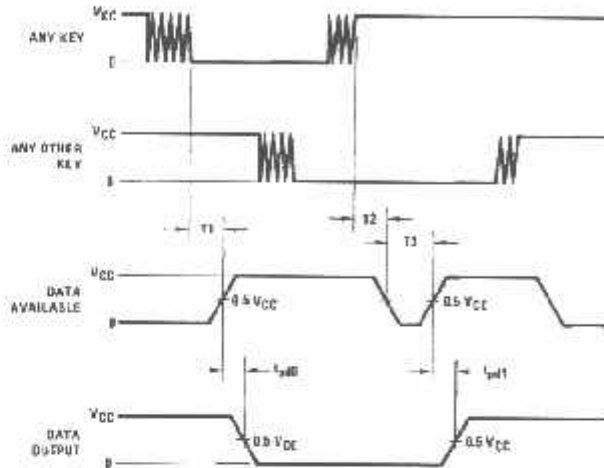
**AC Electrical Characteristics** (Note 3) $T_A = 25^\circ C, C_L = 50 \text{ pF}$ , unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
$t_{PH}, t_{PL}$	Propagation Delay Time to Logical '0' or Logical '1' from D.A.	$C_L = 50 \text{ pF}$ (Figure 1) $V_{CC} = 5V$ $V_{CC} = 10V$ $V_{CC} = 15V$		60 35 25	150 80 60	ns ns ns
$t_{PH}, t_{PL}$	Propagation Delay Time from Logical '0' or Logical '1' into High Impedance State	$R_L = 10k, C_L = 10 \text{ pF}$ (Figure 2) $V_{CC} = 5V, R_L = 10k$ $V_{CC} = 10V, C_L = 10 \text{ pF}$ $V_{CC} = 15V$		60 85 50	200 150 110	ns ns ns
$t_{PH}, t_{PL}$	Propagation Delay Time from High Impedance State to a Logical '0' or Logical '1'	$R_L = 10k, C_L = 50 \text{ pF}$ (Figure 2) $V_{CC} = 5V, R_L = 10k$ $V_{CC} = 10V, C_L = 50 \text{ pF}$ $V_{CC} = 15V$		100 55 40	250 125 90	ns ns ns
$C_{IN}$	Input Capacitance	Any Input (Note 4)		5	7.5	pF
$C_{OUT}$	3-STATE Output Capacitance	Any Output (Note 4)		10		pF

Note 3: AC Parameters are guaranteed by DC correlated testing.

Note 4: Capacitance is guaranteed by periodic testing.

Switching Time Waveforms



$T_1 - T_2 = RC$ ,  $T_3 = 0.7 RC$ , where  $R = 10k$  and  $C$  is external capacitor at KBM input.

FIGURE 1.

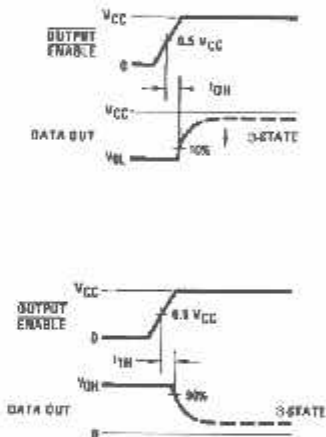
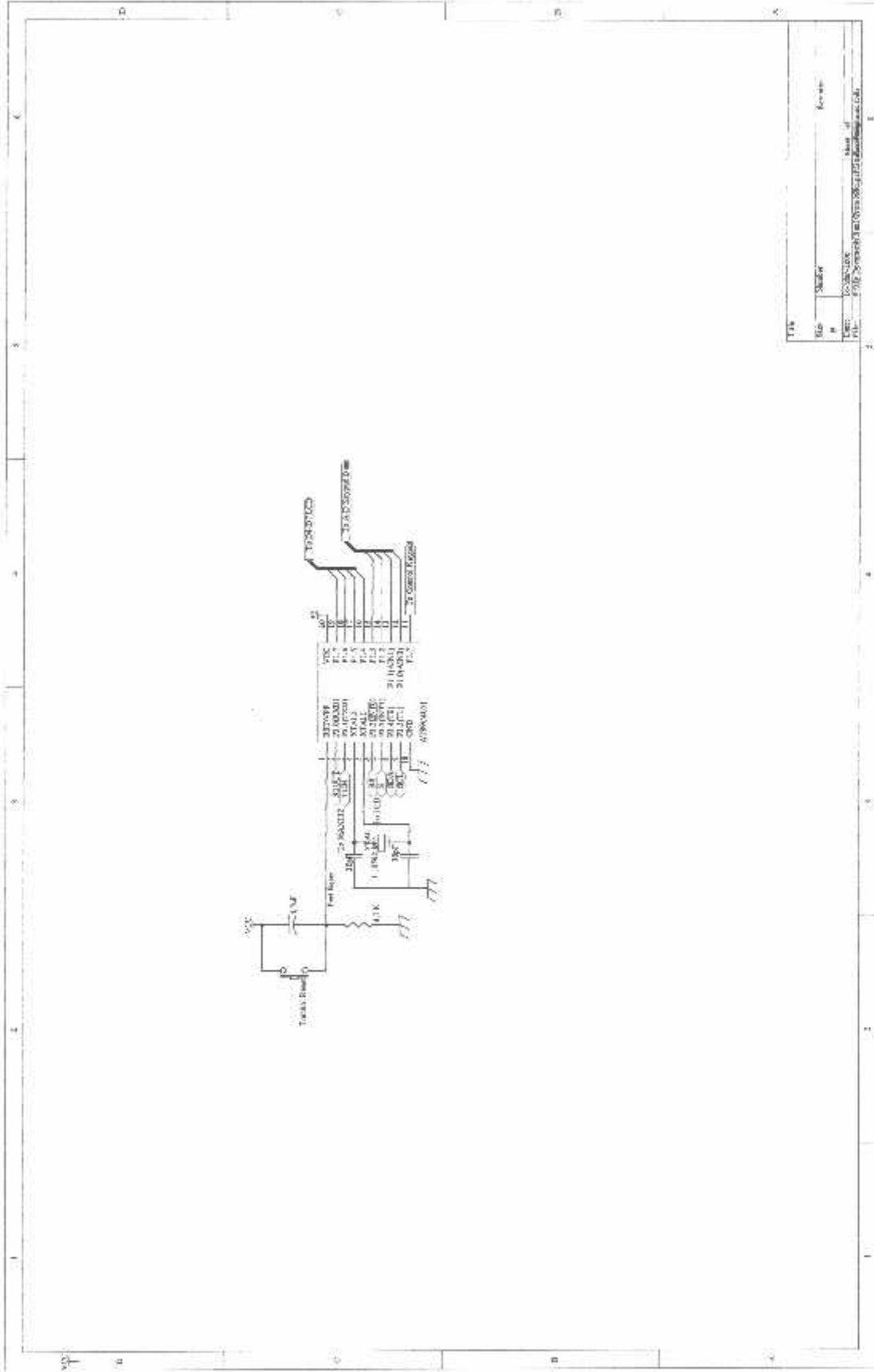
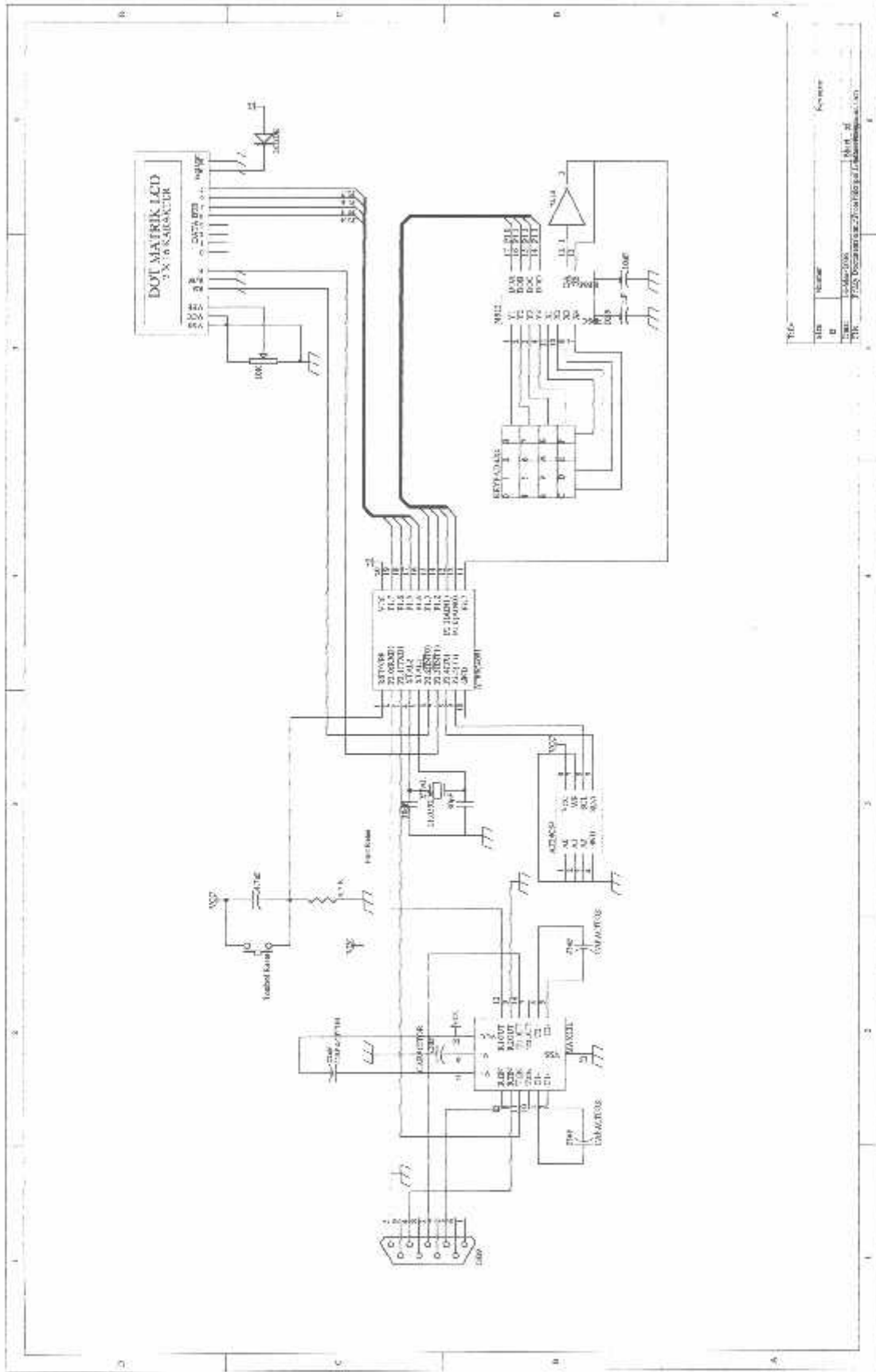


FIGURE 2.



No	Skema	Revisi
1	1000-100	
2	1000-100	
3	1000-100	
4	1000-100	
5	1000-100	
6	1000-100	
7	1000-100	
8	1000-100	
9	1000-100	
10	1000-100	
11	1000-100	
12	1000-100	
13	1000-100	
14	1000-100	
15	1000-100	
16	1000-100	
17	1000-100	
18	1000-100	
19	1000-100	
20	1000-100	
21	1000-100	
22	1000-100	
23	1000-100	
24	1000-100	
25	1000-100	
26	1000-100	
27	1000-100	
28	1000-100	
29	1000-100	
30	1000-100	
31	1000-100	
32	1000-100	
33	1000-100	
34	1000-100	
35	1000-100	
36	1000-100	
37	1000-100	
38	1000-100	
39	1000-100	
40	1000-100	
41	1000-100	
42	1000-100	
43	1000-100	
44	1000-100	
45	1000-100	
46	1000-100	
47	1000-100	
48	1000-100	
49	1000-100	
50	1000-100	



FILE: \_\_\_\_\_

DATE	REVISION	REVISION
15-Mar-2008	1	1
FILE: D:\Programs\27\04\16f877\16f877.asm		



PRICE LIST per 10 JUNI 2005



FULL ELEKTRONIK / ONLINE 24 JAM

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
MENTARI	MT10	Rp 9.900	Nominal 10.000 M10=10.750	07 hari	37 hari
MENTARI	MT20	Rp 19.400	Nominal 20.000 M20=20.000	10 hari	37 hari
MENTARI	MT25	Rp 24.500	Nominal 25.000 M25=25.000	15 hari	37 hari
MENTARI	MT30	Rp 32.500	Nominal 30.000 M30=30.500	15 hari	37 hari
MENTARI	MT40	Rp 42.000	Nominal 40.000 M40=40.000	20 hari	37 hari
MENTARI	MT50	Rp 51.500	Nominal 50.000 M50=51.500	30 hari	37 hari
MENTARI	MT75	Rp 73.500	Nominal 75.000 M75=73.000	45 hari	37 hari
MENTARI	MT100	Rp 96.500	Nominal 100.000 M100=96.000	60 hari	37 hari
MENTARI	MT150	Rp 146.500	Nominal 150.000 M150=142.000	100 hari	37 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
PRO XL / BEBAS	XR10	Rp 11.500	Reguler Nominal 10.000	07 hari	30 hari
PRO XL / BEBAS	XR25	Rp 25.500	Reguler Nominal 25.000	15 hari	30 hari
PRO XL / BEBAS	XR50	Rp 49.500	Reguler Nominal 50.000	30 hari	30 hari
PRO XL / BEBAS	XS50	Rp 49.500	SMS Nominal 50.000	30 hari	10 hari
PRO XL / BEBAS	XCR75	Rp 73.900	Reguler Nominal 75.000	60 hari	30 hari
PRO XL / BEBAS	XB75	Rp 73.900	Bicara Nominal 75.000	60 hari	30 hari
PRO XL / BEBAS	XR100	Rp 97.000	Reguler Nominal 100.000	60 hari	30 hari
PRO XL / BEBAS	XS100	Rp 97.000	SMS Nominal 100.000	90 hari	30 hari
PRO XL / BEBAS	XB100	Rp 97.000	Bicara Nominal 100.000	90 hari	30 hari
PRO XL / BEBAS	XR150	Rp 148.000	Reguler Nominal 150.000	180 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
SIMPATI	S20	Rp 23.000	Nominal 20.000	15 hari	30 hari
SIMPATI GROSIR	GS20-30	Rp 645.000	Per Unit Rp. 21.500		
SIMPATI GROSIR	GS20-50	Rp 1.050.000	Per Unit Rp. 21.000		
SIMPATI	S50	Rp 50.900	Nominal 50.000	30 hari	30 hari
SIMPATI GROSIR	GS50-30	Rp 1.500.000	Per Unit Rp. 50.000		
SIMPATI GROSIR	GS50-50	Rp 2.495.000	Per Unit Rp. 49.900		
SIMPATI	S100	Rp 97.500	Nominal 100.000	60 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
XL JEMPOL	XJ5	Rp 7.000	Nominal 5.000	05 hari	30 hari
XL JEMPOL	XJ15	Rp 17.000	Nominal 15.000	15 hari	30 hari
XL JEMPOL	XJ35	Rp 35.250	Nominal 35.000	35 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
XL EXTRA	XX50	Rp 49.500	Nominal 50.000	07 hari	30 hari
XL EXTRA	XX100	Rp 97.000	Nominal 100.000	30 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
FLEXY TRENDY	F10	Rp 11.500	Nominal 10.000	15 hari	30 hari
FLEXY TRENDY	F20	Rp 21.000	Nominal 20.000	15 hari	30 hari
FLEXY TRENDY	F50	Rp 48.000	Nominal 50.000	30 hari	30 hari
FLEXY TRENDY	F100	Rp 95.000	Nominal 100.000	60 hari	30 hari
FLEXY TRENDY	F150	Rp 148.000	Nominal 150.000	90 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
IM3	I10	Rp 10.900	Nominal 10.000	07 hari	40 hari
IM3	I20	Rp 21.000	Nominal 20.000	10 hari	40 hari
IM3	I25	Rp 28.500	Nominal 25.000	15 hari	40 hari
IM3	I30	Rp 32.100	Nominal 30.000	15 hari	40 hari
IM3	I40	Rp 42.600	Nominal 40.000	20 hari	40 hari
IM3	I50	Rp 52.000	Nominal 50.000	30 hari	40 hari
IM3	I75	Rp 73.000	Nominal 75.000	45 hari	40 hari
IM3	I100	Rp 96.750	Nominal 100.000	60 hari	40 hari

### FULL ELEKTRONIK / ONLINE 24 JAM

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
AS	A25	Rp. 29.500	Nominal 25.000		
AS	A50	Rp. 54.900	Nominal 50.000	60 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
FREN	R10	Rp. 12.500	Nominal 10.000		
FREN	R25	Rp. 28.000	Nominal 25.000	07 hari	30 hari
FREN	R50	Rp. 53.000	Nominal 50.000	15 hari	30 hari
FREN	R100	Rp. 106.000	Nominal 100.000	30 hari	30 hari
				60 hari	30 hari

### SEMI ELEKTRONIK / OFFLINE 24 JAM

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
IM3 FISIK	IP10	Rp. 11.000	Nominal 10.000		
IM3 FISIK	IP10-5	Rp. 52.500	Per Unit Rp. 10.500	15 hari	40 hari
IM3 FISIK	IP10-8	Rp. 84.000	Per Unit Rp. 10.500		
IM3 FISIK	IP20	Rp. 23.500	Nominal 20.000	10 hari	40 hari
IM3 FISIK	IP25	Rp. 28.500	Nominal 25.000	15 hari	40 hari
IM3 FISIK	IP50	Rp. 52.000	Nominal 50.000	30 hari	40 hari
IM3 FISIK	IP100	Rp. 96.000	Nominal 100.000	60 hari	40 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
SIMPATI FISIK	SP50	Rp. 52.500	Nominal 50.000	30 hari	30 hari
SIMPATI FISIK	SP100	Rp. 96.900	Nominal 100.000	60 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
MENTARI FISIK	MP10	Rp. 12.500	Nominal 10.000	07 hari	37 hari
MENTARI FISIK	MP10-5	Rp. 60.000	Per Unit Rp. 12.000		
MENTARI FISIK	MP10-8	Rp. 83.600	Per Unit Rp. 10.450		
MENTARI FISIK	MP25	Rp. 25.500	Nominal 25.000	15 hari	37 hari
MENTARI FISIK	MP25-5	Rp. 122.500	Per Unit Rp. 24.500		
MENTARI FISIK	MP25-8	Rp. 192.000	Per Unit Rp. 24.000		
MENTARI FISIK	MP50	Rp. 58.000	Nominal 50.000	30 hari	37 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
STAR ONE FISIK	O10	Rp. 12.000	Nominal 10.000	07 hari	30 hari
STAR ONE FISIK	O20	Rp. 23.000	Nominal 20.000	20 hari	30 hari
STAR ONE FISIK	O50	Rp. 49.250	Nominal 50.000	30 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
AS FISIK	AP50	Rp. 54.500	Nominal 50.000	60 hari	30 hari

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
LIPPO FISIK	LP10	Rp. 12.000	Nominal 10.000		
LIPPO FISIK	LP25	Rp. 27.000	Nominal 25.000		
LIPPO FISIK	LP50	Rp. 48.500	Nominal 50.000		
LIPPO FISIK	LP100	Rp. 96.000	Nominal 100.000		

PRODUK	KODE NOMINAL	HARGA	KETERANGAN	AKTIF	TENGGANG
FREN FISIK	RP10	Rp. 13.000	Nominal 10.000	07 hari	30 hari
FREN FISIK	RP10-5	Rp. 62.500	Per Unit Rp. 12.500		
FREN FISIK	RP10-8	Rp. 100.000	Per Unit Rp. 12.500		
FREN FISIK	RP25	Rp. 28.500	Nominal 25.000	15 hari	30 hari
FREN FISIK	RP25-5	Rp. 140.000	Per Unit Rp. 28.000		
FREN FISIK	RP25-8	Rp. 224.000	Per Unit Rp. 28.000		

