

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



**SKRIPSI**

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI PERALATAN  
LISTRIK VIA SMS DILENGKAPI FASILITAS KONFIGURASI NOMOR HP  
BERBASIS ATMEGA16**

*Disusun Oleh:*  
**PRIYO ADI JATMIKO**  
**NIM : 0412223**

**MARET 2009**

---

# LEMBAR PERSETUJUAN

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI ON/OFF  
PERALATAN LISTRIK VIA SMS DILENGKAPI FASILITAS  
KONFIGURASI NOMOR HP BERBASIS ATMega16**

## SKRIPSI

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

**Disusun Oleh :**


**PRIYO ADI JATMIKO**


**NIM : 04.12.223**

**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

  
**Ir. H. Sidik Noertjahjono, MT**  
**NIP.Y.1028700163**

  
**Sotvohadi, ST, MSc**  
**NIP.1039700309**

**Mengetahui**

**Ketua Jurusan Teknik Elektro S-1**



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

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



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

---

**BERITA ACARA UJIAN SKRIPSI  
FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Priyo Adi Jatmiko  
NIM : 04.12.223  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengendali On/Off  
Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi No  
HP Berbasis ATMega16

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

Hari : Selasa  
Tanggal : 17 Maret 2009  
Dengan Nilai : 88,95 (A) *BA*

**Ketua Majelis Penguji**

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

**Sekretaris Majelis Penguji**

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

**Penguji I**

**(Ir. Mimien Mustikawati, MT)**  
NIP. Y.1030000352

**Penguji II**

**(Ir. M. Ibrahim Ashari, MT)**  
NIP. Y. 1030100358

---

**PERANCANGAN DAN PEMBUATAN SISTEM PENGENDALI  
ON/OFF PERALATAN LISTRIK VIA SMS DILENGKAPI FASILITAS  
KONFIGURASI NOMOR HP BERBASIS ATMEGA16**

**Priyo Adi Jamiko**

**04.12.223**

**Jurusan Teknik Elektronika – Institut Teknologi Nasional Malang**

**Jln. Raya Karanglo Km 2 Malang**

**mick\_mickq@yahoo.com**

**Dosen Pembimbing : I. Ir. Sidik Noertjahjono, MT.**

**II. Sotyohadi, ST, MSc**

**Kata Kunci : Sistem pengendali, SMS, Atmega16**

Telepon seluler yang memiliki jalur komunikasi secara serial dengan fasilitas SMS yang mampu bertukar informasi berbasis teks dengan jarak jauh dan tanpa kabel, dapat memberikan solusi yang tepat terhadap masalah pengendalian dan pemantauan peralatan listrik dari jarak jauh. sistem pengendali akan dirancang dengan menggunakan mikrokontroller ATMEGA16 sebagai pemroses utama data masukan dari telepon seluler maupun data dari detektor arus

Sistem pengendali akan mengendalikan aktif dan tidaknya peralatan listrik ketika mendapatkan data SMS dengan format tertentu, dan sistem pengendali akan memberikan informasi melalui SMS status aktif atau tidaknya peralatan listrik setelah sistem pengendali mendapatkan SMS dengan format tertentu

Dengan fasilitas konfigurasi nomor seluler yang terdapat pada sistem pengendali, maka nomor pengguna dapat disesuaikan dengan mudah, dan dengan memanfaatkan memori EEPROM internal mikrokontroler untuk menyimpan hasil konfigurasi nomor seluler, maka memungkinkan sistem pengendali untuk mampu berkomunikasi dengan lebih dari satu nomor ponsel.

---



## KATA PENGANTAR

Dengan memanjatkan puji syukur kepada Tuhan Yang Maha Esa, atas berkat, rahmat dan karunia-Nya sehingga dapat menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA16” dengan lancar. Skripsi ini merupakan persyaratan kelulusan studi di jurusan Teknik Elektro S-1 konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar sarjana teknik.

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

1. Bapak Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak Ir. Sidik Noertjahjono, MT selaku Dosen Pembimbing I.
5. Bapak Sotyohadi, ST, MSc selaku Dosen Pembimbing II.
6. Orangtua serta saudara-saudara kami yang telah memberikan doa restu, motivasi, semangat dan biaya.
7. Teman-teman instruktur Laboratorium Perancangan Elektronika yang telah membantu.
8. Semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini, untuk itu

penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

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

Malang, Maret 2009

Penulis

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>BERITA ACARA UJIAN SKRIPSI</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	xii
<b>DAFTAR TABEL</b> .....	xvi
<b>DAFTAR GRAFIK</b> .....	xviii
<b>DAFTAR LAMPIRAN</b> .....	xix
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian.....	4
1.5. Sistematika Penulisan.....	4
<b>BAB II DASAR TEORI</b> .....	6
2.1. Pendahuluan.....	6
2.2. <i>Handphone</i> Siemens C55.....	6
2.3. Kabel Data Siemens C55.....	8
2.4. <i>Short Message Service &amp; Protocol Data Unit</i> .....	9
2.4.1. <i>Short Message Service (SMS)</i> .....	10
2.4.2. Sistem Kerja SMS.....	10
2.4.3. <i>Protocol Data Unit</i> .....	11

2.4.3.1. Mengirim SMS .....	11
2.4.3.2. Memanggil Data PDU Pada SMS Yang Diterima .....	17
2.5. Dioda .....	19
2.6. Transistor Bipolar .....	20
2.6.1. Arus Transistor .....	21
2.6.2. Alpha & Beta .....	22
2.6.3. <i>Common Emitter</i> .....	22
2.6.4. Arus Basis .....	24
2.6.5. Kurva <i>Collector</i> .....	24
2.7. <i>Optocoupler</i> IC 4N25 .....	26
2.8. Max 232 Sebagai Pengubah Tegangan TTL .....	27
2.9. Relay .....	31
2.10. Keypad .....	32
2.11. LCD ( <i>Liquid Crystal Display</i> ) .....	33
2.12. Mikrokontroler ATMEGA-16 .....	36
2.12.1. Arsitektur ATMEGA-16 .....	36
2.12.2. Fitur ATMEGA-16 .....	37
2.12.3. Konfigurasi Pin ATMEGA-16 .....	38
2.12.4. Peta Memori .....	39
2.12.5. Status Register ( <i>Sreg</i> ) .....	40
<b>BAB III PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>42</b>
3.1. Pendahuluan .....	42
3.2. Perancangan Perangkat Keras .....	42

3.2.1. Blok Diagram Alat .....	43
3.2.2. Prinsip Kerja Alat .....	45
3.2.3. Rangkaian Detektor Arus .....	46
3.2.4. Driver Relay .....	48
3.2.5. Rangkaian Antarmuka RS-232 .....	51
3.2.6. Rangkaian LCD .....	52
3.2.7. Rangkaian Keypad .....	53
3.2.8. Minimum Sistem ATMEGA-16 .....	54
3.3. Perencanaan Format Teks SMS .....	57
3.4. Perancangan Perangkat Lunak .....	57
<b>BAB IV PENGUJIAN DAN HASIL ANALISA .....</b>	<b>62</b>
4.1. Pengujian Rangkaian Detektor Arus .....	62
4.1.1. Alat Yang Digunakan .....	62
4.1.2. Langkah Pengujian .....	62
4.1.3. Hasil Dan Analisa Pengujian Rangkaian Detektor Arus .....	63
4.2. Pengujian Rangkaian Driver Relay .....	69
4.2.1. Alat Dan Bahan .....	69
4.2.2. Langkah Pengujian .....	70
4.2.3. Hasil Dan Analisa Pengujian Rangkaian Driver Relay	71
4.3. Pengujian Rangkaian Minimum Sistem ATMEGA 16 .....	74
4.3.1. Alat Dan Bahan .....	74
4.3.2. Langkah Pengujian .....	74
4.3.3. Hasil Dan Analisa Pengujian .....	75

4.4. Pengujian Rangkaian Komunikasi Serial RS-232 .....	76
4.4.1. Alat Dan Bahan .....	76
4.4.2. Langkah Pengujian .....	76
4.4.3. Hasil Dan Analisa Pengujian .....	76
4.5. Pengujian Keypad .....	77
4.5.1. Alat Dan Bahan .....	77
4.5.2. Langkah Pengujian .....	78
4.5.3. Hasil Pengujian .....	78
4.6. Pengujian Rangkaian LCD .....	79
4.6.1. Alat Dan Bahan .....	79
4.6.2. Langkah Pengujian .....	80
4.6.3. Hasil Pengujian .....	80
4.7. Pengujian Keseluruhan Sistem .....	81
4.7.1. Alat Dan Bahan .....	81
4.7.2. Langkah Pengujian .....	81
4.7.3. Hasil Pengujian Dan Analisa .....	82
4.7.3.1. Proses Konfigurasi Nomor Ponsel .....	82
4.7.3.2. Proses Pengendaliar Sistem .....	87
4.8. Konsumsi Daya Sistem Pengendali .....	94
4.8.1. Alat Dan Bahan .....	94
4.8.2. Langkah Pengujian .....	94
4.8.3. Hasil Pengujian .....	94
4.9. Spesifikasi Alat .....	95

<b>BAB V PENUTUP</b> .....	97
5.1. Kesimpulan .....	97
5.2. Saran .....	98
<b>DAFTAR PUSTAKA</b> .....	99

## DAFTAR GAMBAR

Gambar 2.1.	Handphone Siemens Tipe C55 .....	7
Gambar 2.2.	Pinout Handphone Siemens Tipe C55 .....	7
Gambar 2.3.	Rangkaian Serial Interface pada Kabel Data Siemens C55 .....	9
Gambar 2.4.	Depletion Region Dan Simbol Dioda .....	19
Gambar 2.5.	Struktur Sebuah Transistor .....	20
Gambar 2.6.	Lapisan Deplesi .....	20
Gambar 2.7.	Tiga Arus Konvensional Pada Transistor .....	21
Gambar 2.8.	Aliran Elektron .....	21
Gambar 2.9.	Rangkaian Common Emitter .....	22
Gambar 2.10.	Skematik Optocoupler 4N25 .....	26
Gambar 2.11.	IC Max 232 .....	28
Gambar 2.12.	<i>1byte of Asyncron Data</i> .....	29
Gambar 2.13.	Konektor DB-9 .....	30
Gambar 2.14.	Bentuk Fisik Relay .....	32
Gambar 2.15.	Relay DPDT .....	32
Gambar 2.16.	Keypad Matrix 4X4 .....	32
Gambar 2.17.	Deskripsi Pin LCD M1632 .....	34
Gambar 2.18.	Blok Diagram Fungsional ATMEGA-16 .....	36
Gambar 2.19.	Pin-Pin ATMEGA-16 .....	38
Gambar 2.20.	Konfigurasi Memori Data ATMEGA-16 .....	39
Gambar 2.21.	Peta Memori Program ATMEGA-16 .....	40
Gambar 3.1.	Diagram Blok Sistem .....	43



Gambar 3.2.	Rangkaian Detektor Arus .....	46
Gambar 3.3.	Jumlah Tegangan Dalam Rangkaian Tertutup .....	47
Gambar 3.4.	Rangkaian Driver Relay .....	49
Gambar 2.5.	Rangkaian Antarmuka RS-232 .....	51
Gambar 3.6.	Rangkaian LCD .....	53
Gambar 2.7.	Rangkaian Minimum Sistem ATMEGA-16 .....	54
Gambar 2.8.	Rangkaian Reset Pada ATMEGA-16 .....	56
Gambar 3.10.	Flowchart bagian 1 .....	58
Gambar 3.11.	Flowchart bagian 2 .....	59
Gambar 3.12.	Flowchart bagian 3 .....	60
Gambar 3.13.	Flowchart bagian 4 .....	61
Gambar 4.1.	Rangkaian Pengujian Detektor Arus .....	63
Gambar 4.2.	Blok Diagram Detektor Arus .....	63
Gambar 4.3.	Pengukuran Tegangan Jala-jala Listrik .....	63
Gambar 4.4.	Pengukuran Tegangan Simpul Beban .....	63
Gambar 4.5.	Bentuk Gelombang Pada Simpul Beban .....	65
Gambar 4.6.	Pengukuran Tegangan Pada Titik AB .....	65
Gambar 4.7.	Bentuk Gelombang Pada Titik AB .....	66
Gambar 4.8.	Pengukuran Tegangan Output Detektor Arus Saat Tidak Ada Aliran Arus Listrik Ke Beban Listrik .....	66
Gambar 4.9.	Pengukuran Tegangan Pada Output Detektor Pada Saat Ada Aliran Arus Listrik Ke Beban Listrik .....	68
Gambar 4.10.	Rangkaian Pengujian Driver Relay .....	70
Gambar 4.11.	Blok Diagram Pengujian Driver Relay .....	70

Gambar 4.12. Hasil Pengukuran Arus Ib Trasistor A733 .....	71
Gambar 4.13. Hasil Pengukuran Arus Ie Trasistor A733 .....	73
Gambar 4.14. Diagram Blok Pengujian Mikrokontroler .....	75
Gambar 4.15. Blok Diagram Pengujian Driver Serial .....	76
Gambar 4.16. Hasil Pengujian Driver Serial Pada <i>Hyper Terminal</i> .....	77
Gambar 4.17. Blok Diagram Pengujian Keypad .....	78
Gambar 4.18. Blok Diagram Pengujian Rangkaian LCD M1632 .....	80
Gambar 4.19. Hasil Pengujian Rangkaian LCD M1632 .....	80
Gambar 4.20. Tampilan LCD Daftar Menu .....	82
Gambar 4.21. Tampilan LCD Set Nomor HP Untuk User 1 .....	83
Gambar 4.22. Tampilan LCD Set Nomor HP Untuk User 2 .....	83
Gambar 4.23. Tampilan LCD Set Nomor HP Untuk User 3 .....	83
Gambar 4.24. Tampilan LCD Set Nomor HP Untuk User 4 .....	84
Gambar 4.25. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Alamat \$000 - \$00D .....	85
Gambar 4.26. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Alamat \$00F - \$01C .....	85
Gambar 4.27. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Alamat \$01E - \$02B .....	85
Gambar 4.28. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Alamat \$02D - \$03A .....	85
Gambar 4.29. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU , Alamat \$0D2 - \$0E4 .....	86

Gambar 4.30. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU , Alamat \$0E6 - \$0F8 .....	86
Gambar 4.31. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU , Alamat \$0FA - \$10C .....	87
Gambar 4.32. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU , Alamat \$10E - \$120 .....	87
Gambar 4.33. Data Yang PDU Dikirimkan HP Server ke <i>Hyperterminal</i> Saat Isi SMS "ONN 1234" .....	89
Gambar 4.34. Data Yang PDU Dikirimkan HP <i>Server</i> ke <i>Hyperterminal</i> Saat Isi SMS "OFF 1234" .....	91
Gambar 4.35. Kondisi Beban Lampu Pijar Setelah Data SMS "ONN 1234" Diterima Sistem .....	93
Gambar 4.36. Diagram Blok Pengujian Konsumsi Daya Pada <i>Panel Box</i> Sistem Pengendali .....	94
Gambar 4.37. Hasil Pengujian Konsumsi Daya Sistem Pengendali .....	95
Gambar 4.38. Alat Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA -16 .....	96

## DAFTAR TABEL

Tabel 2.1.	Fungsi Masing-Masing Pin Out .....	8
Tabel 2.2.	Nomor SMS Center Dengan Cara <i>National Code</i> .....	12
Tabel 2.3.	Nomor SMS Center Dengan Cara <i>International Code</i> .....	12
Tabel 2.4.	Skema Encoding <i>7bit</i> .....	15
Tabel 2.5.	Keterangan Setiap Header Pada PDU .....	18
Tabel 2.6.	Fungsi Pin RS-232 dalam DB-9 .....	30
Tabel 2.7.	Fungsi Pin – Pin LCD .....	35
Tabel 4.1.	Tabel Hasil Pengujian Tegangan Beban .....	65
Tabel 4.2.	Tabel Hasil Pengujian Tegangan Pada Titik A B .....	66
Tabel 4.3.	Tabel perbandingan hasil perhitungan dan hasil pengujian Tegangan Simpul A B .....	67
Tabel 4.4.	Tabel Hasil Pengujian Tegangan Keluaran Detektor Arus ...	68
Tabel 4.5.	Tabel perbandingan hasil perhitungan dan hasil pengujian tegangan keluaran detektor arus .....	69
Tabel 4.6.	Tabel Hasil Pengujian Arus Ib Transistor A733 .....	71
Tabel 4.7.	Tabel perbandingan hasil perhitungan dan hasil pengujian arus Ib pada transistor A733 .....	72
Tabel 4.8.	Tabel Hasil Pengujian Arus Ie Transistor A733 .....	72
Tabel 4.9.	Tabel perbandingan hasil perhitungan dan hasil pengujian arus Ie pada transistor A733 .....	74
Tabel 4.10.	Hasil Pengujian Sistem Mikrokontroler .....	75

Tabel 4.11. Tabel Hasil Pengujian Driver Serial RS 232 Pada <i>Hyper Terminal</i> .....	77
Tabel 4.12. Tabel Hasil Pengujian Keypad Matrix 4x4 .....	79
Tabel 4.13. Daftar Menu Pada Sistem .....	82
Tabel 4.14. Daftar Nomor Handphone User Yang Akan Digunakan Untuk Pengujian Alat .....	83
Tabel 4.15. Pemetaan Memori EEPROM Yang digunakan Untuk Penyimpanan Nomor Selular .....	84
Tabel 4.16. Hasil Konversi Nomor Ponsel Ke Format PDU .....	86
Tabel 4.17. Data PDU Lengkap Yang Diterima Mikrokontroler Saat Perintah "ONN" .....	88
Tabel 4.18. Kondisi Beban Setelah Perintah SMS "ONN" Diproses Oleh Mikrokontroler .....	89
Tabel 4.19. Data PDU Lengkap Yang Diterima Mikrokontroler Saat Perintah "OFF" .....	90
Tabel 4.20. Kondisi Beban Setelah Perintah SMS "OFF" Diproses Oleh Mikrokontroler .....	91
Tabel 4.21. PDU Lengkap Yang Dikirimkan Mikrokontroler Ke <i>HP Server</i> .....	92
Tabel 4.22. Laporan Status Beban Yang Dikirimkan <i>HP Server</i> Ke <i>HP User</i> .....	93
Tabel 4.23. Konsumsi Daya Sistem Pengendali .....	94

## DAFTAR GRAFIK

Grafik 2.1. ....	24
Grafik 2.2. ....	25

## DAFTAR LAMPIRAN

1. Gambar Skematik Rangkaian Keseluruhan
2. Listing Program
3. Datasheet
  - a. Datasheet Dioda 1N5403
  - b. Datasheet Optocoupler 4N25
  - c. Datasheet IC Max 232
  - d. Datasheet LCD M1632
  - e. Datasheet Transistor A733
  - f. Datasheet ATMEGA-16





# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Seiring dengan berkembangnya teknologi nirkabel (*wireless*), salah satunya adalah teknologi GSM (*Global System for Mobile Communications*), yang semakin murah dan dengan kapasitas jangkauan yang semakin luas, menyebabkan pemakaian telepon seluler tidak hanya berada pada salah satu golongan masyarakat tertentu saja (kaum elit), namun pemakai telepon seluler sudah menjangkau semua lapisan. SMS (*Short Message Service*) adalah salah satu fasilitas yang terdapat pada telepon seluler yang hampir setiap orang mengenalnya. Selain memiliki biaya operasional yang cukup murah, fasilitas ini juga merupakan media komunikasi dan sarana informasi antar individu yang cukup memiliki sifat waktu nyata (*real-time*), sehingga tidaklah mengherankan apabila SMS masih tetap menjadi pilihan bagi setiap orang sebagai sarana komunikasi, meskipun saat ini teknologi yang lain (seperti EMS ataupun MMS) telah berkembang.

Tingginya aktivitas manusia dalam memenuhi kebutuhan ekonomi semakin mengurangi waktu bagi manusia untuk berada di sekitar rumah, baik untuk berkumpul bersama keluarga maupun untuk menjaga keamanan peralatan listrik pada tempat tinggal tersebut. Permasalahan ini mendorong penulis untuk mengadakan penelitian dan perancangan peralatan pemantau dan pengendali peralatan listrik dari jarak jauh yang efektif dan efisien. Telepon seluler dengan fasilitas SMS yang mampu bertukar informasi berbasis teks secara jarak jauh

(*remote*) dan tanpa kabel (*wireless*) dapat memberikan solusi yang tepat terhadap masalah pengontrolan dan pemantauan peralatan listrik dari jarak jauh.

Pengendalian pada sistem ini memiliki kelebihan dari skripsi sebelumnya yang berjudul "Alat Pengontrol Penerangan dan Keamanan Rumah Jarak Jauh Menggunakan SMS Dengan Mikrokontroler AT89S52" yang pada prinsipnya memiliki sistem yang hampir sama namun pada skripsi sebelumnya untuk mengendalikan sistem hanya ada satu nomor pemanggil yang disimpan di dalam mikrokontroler, nomor pemanggil dapat diubah hanya melalui program ulang.

Sedangkan pada skripsi ini, dengan alasan untuk kemudahan dan keamanan sistem bagi pengguna, sistem dapat dikendalikan oleh beberapa nomor handphone yang sebelumnya harus dikonfigurasi pada sistem dengan cara menginputkan nomor hand phone melalui keypad pada sistem, nomor handphone yang telah diisikan pada sistem dapat dirubah melalui keypad pada sistem. Pada skripsi ini juga memiliki fitur check status yang mana tidak dimiliki oleh sistem pada skripsi sebelumnya.

## **1.2. Rumusan Masalah**

Dalam Perancangan dan Pembuatan Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA16 dapat dirumuskan beberapa masalah yang akan dibahas yaitu:

1. Bagaimana merancang komunikasi antara hand phone dengan mikrokontroller?
2. Bagaimana memproses pengiriman maupun penerimaan informasi sms menggunakan PDU (*Protocol Data Unit*)?

3. Bagaimana cara mengendalikan on/off peralatan listrik dari jala-jala listrik menggunakan mikrokontroler?
4. Bagaimana merancang dan membuat perangkat keras atau *hardware* dan perangkat lunak atau *software* agar bekerja dengan baik?

### 1.3. Tujuan

Tujuan dari penulisan skripsi ini adalah untuk memudahkan manusia dalam pengendalian dan pemantauan peralatan listrik rumah terhadap hubungan jala-jala listrik dari jarak jauh.

### 1.4. Batasan Masalah

Agar pembahasan dari skripsi ini tidak terlalu meluas maka penyusun perlu membuat batasan-batasan masalah yang meliputi:

1. Tipe *handphone* yang digunakan adalah siemens C55.
2. Nomor seluler yang digunakan adalah nomor dari operator selular GSM Indosat, Satelindo, Telkomsel, dan Excelcom.
3. Nomor selular yang digunakan adalah nomor seluler 12 digit.
4. Menggunakan lampu pijar sebanyak 4 buah sebagai simulasi beban peralatan listrik yang akan dikendalikan.
5. Menggunakan serial interface untuk komunikasi antara *handphone* dengan mikrokontroler.

## 1.5 Metodologi

Metodologi yang dipakai dalam pembuatan skripsi ini adalah:

### 1. Studi Literatur

Mencari referensi-referensi yang berhubungan dengan perencanaan dan pembuatan alat yang akan dibuat.

### 2. Penelitian Lapangan

Melakukan penelitian secara langsung mengenai objek-objek yang berhubungan langsung dengan perencanaan alat yang akan dibuat.

### 3. Pengolahan Data

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

## 1.6. Sistematika Pembahasan

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

### **BAB I** Pendahuluan

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

### **BAB II** Landasan Teori

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

### **BAB III Perencanaan Dan Pembuatan Alat**

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

### **BAB IV Pengujian Alat**

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

### **BAB V Penutup**

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

## BAB II

### DASAR TEORI

#### 2.1. Pendahuluan

Pada bab ini akan dibahas mengenai teori penunjang dari peralatan yang direncanakan. Pokok pembahasan pada bab ini adalah :

1. *Handphone Siemens C55*
2. *Kabel Data Siemens C55*
3. *Short Messages Service dan Protocol Data Unit*
4. *Dioda*
5. *Transistor Bipolar*
6. *Optocoupler*
7. *Serial Interface Max 232*
8. *Relay*
9. *Keypad*
10. *LCD M1632*
11. *Mikrokontroler ATMEGA 16*

#### 2.2. *Handphone Siemens C55*

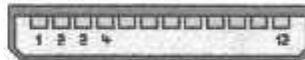
Pada saat ini, sistem komunikasi *Handphone* sudah berkembang pesat. Sistem *Handphone* merupakan bagian dari sistem komunikasi yang berfungsi sebagai media pengirim dan sekaligus sebagai media penerima, selain untuk komunikasi audio, pada *handphone* juga mampu untuk komunikasi data berupa teks yang sering disebut SMS (*Short Messages Service*).



**Gambar 2.1. Handphone Siemens Tipe C55**

**Sumber :** [www.siemens.com](http://www.siemens.com)

Untuk berkomunikasi dengan perangkat lain seperti PC (*Personal Computer*), ataupun Mikrokontroler handphone memiliki jalur data komunikasi serial yang terletak pada pinout-nya berikut merupakan konfigurasi dan fungsi masing-masing Pin Out dari handphone Siemens tipe C55.



**Gambar 2.2. Pin Out Siemens Tipe C55**

**Sumber :** [www.siemens.com](http://www.siemens.com)

**Tabel 2.1. Fungsi Masing-Masing Pin Out**

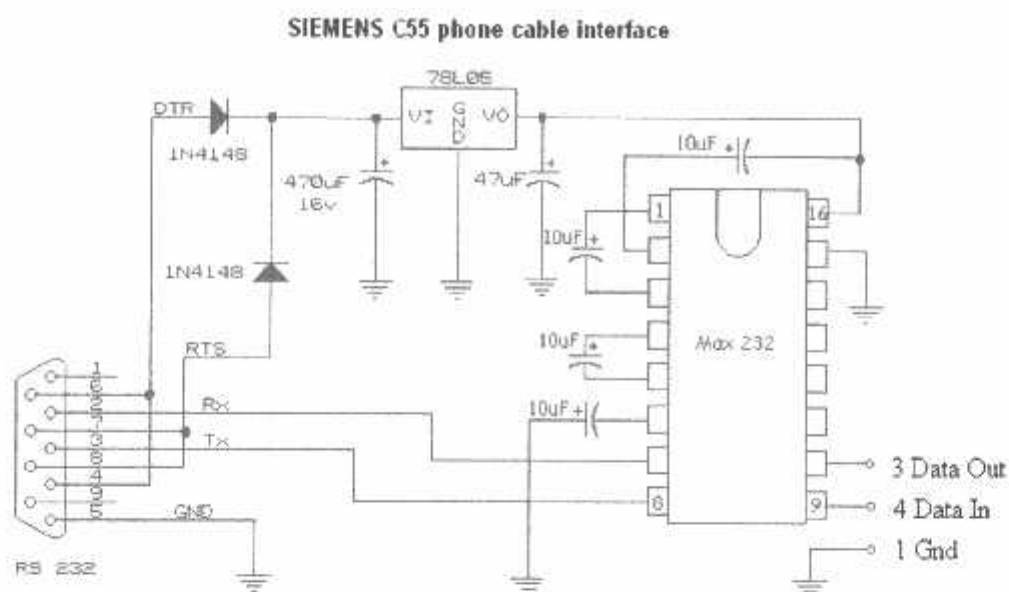
PIN	Siemens X55	Fungsi
1	V in	Power charging
2	Gnd	Ground
3	Tx	Data transfer
4	Rx	Data receiver
5	CTS	Jalur deteksi kabel aksesoris
6	RTS	Jalur data untuk aksesoris
7	DCD	Jalur clock untuk Aksesoris
8	Audio P	Audio Positif
9	-	-
10	Audio N	Audio Negatif
11	Gnd Mic	Ground Mic
12	EPP	EPP1 (Ext Mic)

Sumber : [www.siemens.com](http://www.siemens.com)

### 2.3. Kabel Data Siemens C55

Level tegangan data yang digunakan pada handphone memiliki tegangan untuk sinyal data serial sebesar 3,3V(Level LVTTTL), untuk komunikasi dengan peralatan lain, handphone siemens C55 dilengkapi dengan kabel data serial, yang mana pada kabel data tersebut berisikan rangkaian *serial interface* yang menggunakan standart komunikasi data RS-232, berikut gambar rangkaian yang terdapat pada kabel data siemens C55.





**Gambar 2.3. Rangkaian Serial Interface Pada Kabel Data Siemens C55**  
 Sumber : [www.siemens.com](http://www.siemens.com)

## 2.4. Short Messages Service dan Protocol Data Unit

### 2.4.1. Short Messages Service (SMS)

SMS adalah fasilitas yang dimiliki oleh jaringan GSM (*Global System for Mobile Communication*) yang memungkinkan pelanggan untuk mengirimkan dan menerima pesan-pesan singkat sepanjang 160 karakter. SMS ditangani oleh jaringan melalui pusat layanan atau *SMS Service Center (SMS SC)* yang berfungsi menyimpan dan meneruskan pesan dari sisi pengirim ke sisi penerima. Format SMS yang dipakai oleh produsen MS (*Mobile Station*) adalah *Protokol Data Unit (PDU)*. Format PDU akan mengubah septet kode ASCII (7 bit) menjadi bentuk Byte (8 bit) pada saat pengiriman data dan akan diubah kembali menjadi ASCII pada saat diterima oleh MS.

#### 2.4.2. Sistem Kerja SMS

Dibalik tampilan menu messages pada sebuah ponsel terdapat AT *Command* yang bertugas mengirim dan menerima data ke dan dari SMS *Center*. AT *Command* tiap-tiap SMS device bisa berbeda-beda, tetapi pada dasarnya sama. Perintah AT *Command* biasanya disediakan oleh vendor handphone.

Pada ponsel GSM terdapat fasilitas pengaksesan data melalui koneksi serial atau dengan antarmuka infra merah. Untuk mengakses data, diperlukan urutan instruksi pada antarmuka ponsel. ETSI (*European Telecommunication Standart Institute*) menstandarkan instruksi tersebut dalam spesifikasi teknik GSM pada dokumen GSM 07.07 dan GSM 07.05, dimana setiap ponsel harus mengacu pada instruksi tersebut, instruksi ponsel diawali dengan karakter AT dan diakhiri dengan enter atau 0Dh. Perintah yang diterima akan direspon dengan diterimanya data 'OK' atau 'Error'. Instruksi yang diterima oleh ponsel dan sedang diproses akan terinterupsi oleh instruksi lain yang datang sehingga setiap pengiriman instruksi harus menunggu datangnya respon dari ponsel.

Beberapa contoh AT *Command* yang penting untuk SMS adalah sebagai berikut :

AT+CMGS – Untuk mengirim SMS

AT+CMGL - Untuk memeriksa SMS

AT+CMGD – Untuk menghapus SMS

AT+CMGR – Untuk membaca SMS

### 2.4.3. Protocol Data Unit

AT Command untuk SMS biasanya diikuti oleh data I/O yang diwakili oleh unit-unit PDU. Data yang mengalir ke dan dari SMS Center harus berbentuk PDU (*Protocol Data Unit*). PDU berisi bilangan – bilangan heksadesimal yang mencerminkan I/O. PDU terdiri dari beberapa header. Header untuk mengirim SMS center berbeda dengan SMS yang diterima SMS center.

#### 2.4.3.1. Mengirim SMS

Terdapat delapan header untuk mengirim SMS, header – header tersebut adalah :

##### 1. Nomor SMS center

Header pertama ini terdiri dari atas tiga sub header, yaitu :

- a. Jumlah pasangan heksadesimal SMS center dalam bilangan heksa.
- b. National atau international kode
  - Untuk national, kode sub-headernya adalah 81
  - Untuk international, kode sub-headernya adalah 91.
- c. Nomor SMS center sendiri dalam pasangan heksa dibalik-balik. Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan F didepannya. Contoh nomor SMS Center Indosat-M3 dapat ditulis dengan dua cara
  - Cara 1 (National)  
0855000000 diubah menjadi :
    - 06 : ada 6 pasang.
    - 81 : 1 pasang.

- o 80-55-00-00-00 : 5 pasang.

Digabung menjadi : 06818055000000

- Cara 2 (International)

62855000000 diubah menjadi :

- o 07 : ada 7 pasang.
- o 91 : 1 pasang .
- o 26-58-05-00-00-F0

Digabung menjadi : 07912658050000F0

Berikut beberapa nomor SMS- Center operator seluler di Indonesia

**Tabel 2.2. Nomor SMS Center Dengan Cara *National Code***

No	Operator Seluler	SMS Center	Kode PDU
1	Telkomsel	081100000	068180110000F0
2	Satelindo	0816124	0581806121F4
3	Excelcom	0818445009	06818081440590
4	Indosat - M3	0855000000	06818055000000

Sumber : <http://home.student.utwente.nl/>

**Tabel 2.3. Nomor SMS Center Dengan Cara *International Code***

No	Operator Seluler	SMS Center	Kode PDU
1	Telkomsel	6281100000	06912618010000
2	Satelindo	62816124	059126181642
3	Excelcom	62818445009	07912618485400F9
4	Indosat - M3	62855000000	07912658050000F0

Sumber : <http://home.student.utwente.nl/>

## 2. Tipe SMS

Tipe send, tipe SMS = 1. Jadi bilangan heksanya adalah 01.

## 3. Nomor referensi SMS

Nomor referensi dibiarkan 0. Jadi bilangan heksanya 00. selanjutnya akan diberikan sebuah nomor referensi otomatis oleh ponsel atau alat SMS – gateway.

## 4. Nomor Ponsel Penerima

Sama seperti menulis PDU header untuk SMS center, header ini juga terbagi atas tiga bagian, yaitu :

- a. Jumlah bilangan decimal nomor ponsel yang dituju dalam bilangan heksa.
- b. National atau International code.
- c. Untuk National kode sub-headernya : 81.
- d. untuk international, kode sub-headernya : 91.
- e. Nomor ponsel yang dituju dalam pasangan heksa dibalik – balik.

Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut dipasangkan dengan F di depannya. Contoh nomor ponsel yang dituju : 628129573337 dapat ditulis dengan dua cara sebagai berikut :

- Cara 1 : 0812957337 diubah menjadi :
  - a. 0B : ada 11 angka
  - b. 81
  - c. 80-21-59-37-33-F7Digabung : 0B818021593733F7
- Cara 2 : 628129573337 diubah menjadi :
  - a. 0C ada 12 angka

b. 91

c. 26-81-92-75-33-73

Digabung menjadi : 0C91261892753373

#### 5. Bentuk SMS

0 : 00 >> Dikirim sebagai SMS

1 : 01 >> Dikirim sebagai telex

2 : 02 >> Dikirim sebagai fax

Dalam hal ini, pengiriman dalam bentuk SMS sehingga memakai bilangan heksa 00

#### 6. Skema Encoding Data I/O

Ada dua skema encoding data yaitu :

a. Skema 7bit >> Ditandai dengan angka 0 : 00 heksa

b. Skema 8bit >> Ditandai dengan angka lebih besar dari 0 yang diubah ke heksa.

Pada umumnya SMS gateway yang tersedia pada Service Center menggunakan Skema 7bit sehingga menggunakan kode 00 heksa

#### 7. Jangka waktu Sebelum SMS expired

Jika bagian ini tidak diisi, maka tidak ada batasan waktu berlakunya SMS, tetapi jika diisi dengan bilangan integer yang diubah ke dalam pasangan heksa tertentu, bilangan yang diberikan tersebut akan mewakili jumlah waktu validitas SMS tersebut.

#### 8. Isi SMS

Header ini terdiri atas dua sub-header, yaitu :

Contoh untuk kata "hello"

- Langkah pertama :

Bit	7	1
h	110	1000
e	110	0101
l	110	1100
l	110	1100
o	110	1111

- Langkah kedua :

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

Oleh karena total 7bit x 5 huruf = 35 bit, sedangkan yang diperlukan adalah 8 x 5 bit = 40 bit, maka diperlukan 5 bit dummy yang diisi dengan bilangan 0. Dengan demikian, kata "hello" hasil konversinya adalah **E8329BFD06**.

## 9. Menggabungkan Delapan Header

Masing – masing header maupun sub-header harus digabungkan menjadi sebuah PDU yang lengkap.

Contoh, akan dikirimkan kata “hello” ke ponsel nomor 6285648506944 lewat SMS center Indosat-M3, tanpa validitas waktu, maka PDU lengkap yang siap dikirimkan adalah : : 07 91 2658050000F0  
01 0D 91265846586049F4 0000 05 E8329BFD06

### 3.4.3.1. Memenggal Data PDU Pada SMS Yang Diterima

Agar data sms yang diterima dapat diproses lebih lanjut maka data pdu yang diterima harus dipilah-pilah atau dipenggal, berikut ini adalah contoh PDU yang diterima oleh HP (New SMS atau Inbox):

07 91 2658050000F0 04 0C 91 265836164900 00 00 506020 31133180  
05 E8329BFD06.

Dengan keterangan setiap header pada PDU adalah sebagai berikut:



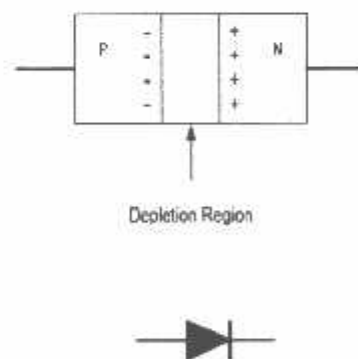
**Tabel 2.5. Keterangan Setiap Header Pada PDU**

Oktet / Digit Hexa	Keterangan
<u>07</u>	Panjang atau jumlah pasangan digit dari nomor SMSC (service number) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya)
<u>91</u>	Jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81.
<u>2658050000F0</u>	Nomor SMSC yang digunakan. Karena jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah +62855000000 (IM3)
<u>04</u>	04 Tipe SMS untuk pesan SMS yang diterima ( <i>Received</i> ), 01 kirim ( <i>Send</i> )
<u>0C</u>	Panjang digit dari nomor pengirim (0C hex = 12 desimal)
<u>91</u>	Jenis nomor pengirim (sama dengan jenis nomor SMSC)
<u>265836164900</u>	Nomor pengirim SMS, yang jika diterjemahkan adalah +628563619400
<u>00</u>	Pengenal protokol, dalam hal ini adalah 0
<u>00</u>	Skema pengkodean SMS, juga bernilai 0
<u>506020 311331 80</u>	Waktu pengiriman, yang berarti 05-06-02 (2 Juni 2005), dan jam 13:31:13. Sedangkan 80 adalah Timezone yang digunakan.
<u>04</u>	Panjang dari pesan SMS, dalam hal ini adalah 4 huruf (dalam mode 7 bit).
<u>E8329BFD06.</u>	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, kemudian dirubah ke ASCII, maka didapat pesan 'hello'

Sumber : <http://home.student.utwente.nl/>

## 2.5. Dioda

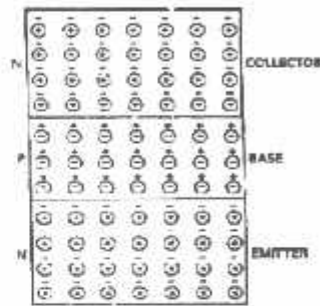
Komponen ini dibentuk dari menyambungkan bahan semikonduktor tipe-N dan tipe-P. Komponen ini memberikan resistansi yang sangat rendah terhadap aliran arus pada satu arah dan resistansi yang sangat tinggi terhadap aliran arus pada arah yang berlawanan. Sebuah diode yang ideal akan melewatkan arus tak terhingga pada satu arah dan sama sekali tidak melewatkan arus pada arah yang sebaliknya. Apabila pada sebuah diode bahan semikonduktor tipe-P dibuat lebih positif dari pada bahan tipe-N melampaui nilai ambang tegangan majunya (untuk bahan silikon 0.7V dan untuk bahan germanium 0.3V) diode akan melewatkan arus dengan bebas. Jika sebaliknya tipe-P dibuat lebih negatif dari pada bahan tipe-N, diode tidak akan mengalirkan arus sama sekali. Sebuah diode akan mengalami kerusakan apabila diberikan tegangan yang melebihi tegangan *breakdown* mundurnya. Pada sebuah diode bila tidak terdapat potensial eksternal, elektron-elektron dari bahan tipe-N akan menyeberang ke daerah tipe-P dan mengisi sebagian dari *hole-hole* yang kosong. Ini menyebabkan terbentuk suatu daerah ditengah – tengah sambungan di mana tidak terdapat pembawa muatan bebas. Zona ini dikenal dengan *Depletion region*, seperti ditunjukkan pada gambar 2-10



**Gambar 2.4. Depletion Region Dan Simbol Dioda**  
Sumber : Prinsip – prinsip Elektronik, Malvino

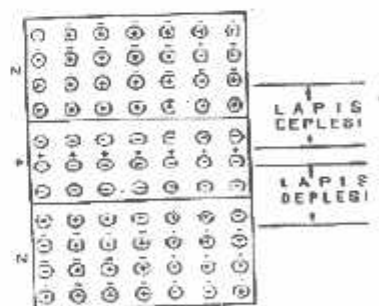
## 2.6. Transistor Bipolar

Kata bipolar pada transistor bipolar adalah singkatan dari “dua polaritas”. Transistor memiliki tiga daerah *doped*, daerah yang paling bawah disebut emiter, daerah tengah disebut basis dan daerah yang paling atas adalah kolektor. Sebuah transistor digambarkan seperti dua buah dioda yang bertolak belakang.



**Gambar 2.5. Struktur sebuah transistor**  
Sumber : Prinsip – prinsip Elektronik, Malvino

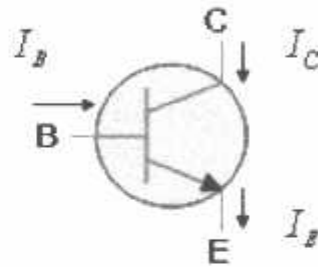
Gambar 2-5 merupakan gambaran struktur daerah transistor sebelum mengalami difusi. Ketika terdapat elektron bebas pada daerah n yang berdifusi melalui sambungan dan bergabung dengan lubang pada daerah p. Maka yang akan terjadi adalah seperti yang ditunjukkan pada gambar 2-6 yaitu timbul dua lapisan deplesi yang memiliki potensial penghalang kira-kira 0,7 V pada suhu 215 °C untuk transistor silikon (0,3 V pada suhu 25 °C untuk transistor germanium).



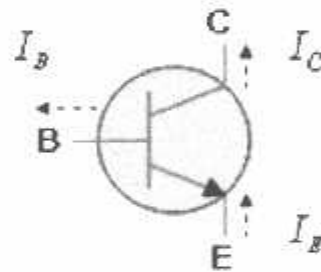
**Gambar 2.6. Lapisan Deplesi**  
Sumber : Prinsip – prinsip Elektronik, Malvino

### 2.6.1. Arus Transistor

Pada transistor terdapat tiga buah arus yang mengalir seperti yang ditunjukkan pada gambar 2.6., arus yang mengalir pada emitor ( $I_E$ ) adalah yang terbesar karena emitor merupakan sumber elektron, arus kolektor ( $I_C$ ) hampir sebesar arus emitor karena sebagian besar elektron pada emitor mengalir ke kolektor, sedangkan arus basis ( $I_B$ ) sangat kecil dibanding kedua arus lainnya



Gambar 2.7. Tiga Arus Konvensional pada Transistor  
Sumber : Prinsip – prinsip Elektronik, Malvino



Gambar 2.8. Aliran Elektron  
Sumber : Prinsip – prinsip Elektronik, Malvino

Dari hukum Kirchoff yang mengatakan jumlah semua arus yang mengalir masuk kesuatu titik sama dengan jumlah arus yang keluar dari titik tersebut. Hukum Kirchoff ini menghasilkan hubungan antara tiga arus konvensional, hubungan itu seperti ditunjukkan pada persamaan 2.1.

$$I_E = I_C + I_B \dots\dots\dots(2.1)$$

### 2.6.2. Alpha dan Beta

Alpha DC ( $a_{dc}$ ) adalah perbandingan antara arus DC yang mengalir pada kolektor terhadap arus DC yang mengalir pada emitor (seperti ditunjukkan pada persamaan 2.2).

$$a_{dc} = \frac{I_c}{I_e} \dots\dots\dots(2.2)$$

Beta DC adalah penguatan arus karena dengan arus kecil dapat dihasilkan arus kolektor yang lebih besar. Beta DC merupakan perbandingan arus kolektor DC dengan arus basis DC (seperti ditunjukkan pada persamaan 2.3).

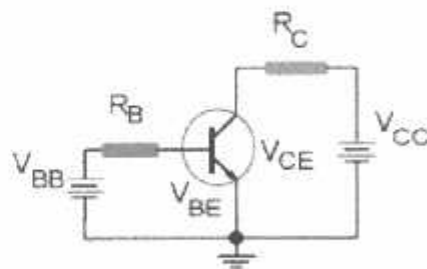
$$\beta_{dc} = \frac{I_c}{I_b} \dots\dots\dots(2.3)$$

Dari persamaan 2.2 dan 2.3 didapatkan persamaan yang ekuivalen (seperti ditunjukkan pada persamaan 2.4).

$$I_c = \beta_{dc} \cdot I_b \dots\dots\dots(2.4)$$

### 2.6.3 Common Emittor

Pada koneksi ini ground dari tiap sumber tegangan dihubungkan dengan emitor. Dari gambar 2.8 rangkaian ini memiliki dua kalang. Kalang kiri adalah kalang basis, dan kalang kanan adalah kalang kolektor.



**Gambar 2.9. Rangkaian Dasar *Common Emittor***  
Sumber : Prinsip – prinsip Elektronik, Malvino

Pada kalang basis, tegangan sumber  $V_{BB}$  membias maju dioda emitor dengan  $R_B$  sebagai resistansi pembatas arus. Dengan mengubah  $V_{BB}$  atau  $R_B$  kita dapat mengubah arus basis. Apabila kita mengubah arus basis maka arus kolektor juga akan berubah artinya arus kecil pada basis mengendalikan arus besar pada kolektor.

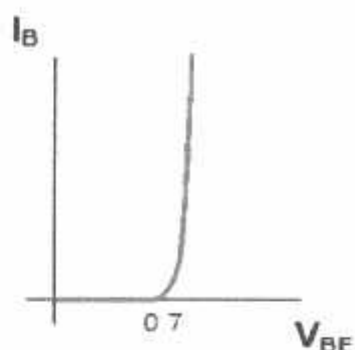
Pada kolektor, sumber tegangan  $V_{CC}$  membias balikkan dioda kolektor melalui  $R_C$ , tegangan sumber  $V_{CC}$  harus membiaskan balik dioda kolektor seperti yang ditunjukkan, atau jika tidak transistor tidak akan bekerja dengan baik. Dengan kata lain kolektor harus positif untuk mengumpulkan sebagian besar electron bebas yang diinjeksikan ke basis.

Sekilas tentang notasi, ada beberapa notasi yang sering digunakan untuk menunjukkan besar tegangan pada suatu titik maupun antar titik. Notasi dengan 1 *subscript* adalah untuk menunjukkan besar tegangan pada satu titik, misalnya  $V_C$  = tegangan kolektor,  $V_B$  = tegangan base dan  $V_E$  = tegangan emiter. Ada juga notasi dengan 2 *subscript* yang dipakai untuk menunjukkan besar tegangan antar 2 titik, yang disebut juga dengan tegangan jepit. Diantaranya adalah :

- $V_{CE}$  = tegangan jepit kolektor-emitor.
- $V_{BE}$  = tegangan jepit base-emitor.
- $V_{CB}$  = tegangan jepit kolektor-base.

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

#### 2.6.4. Arus Basis



**Grafik 2.1. Kurva  $I_B - V_{BE}$**   
Sumber : Prinsip – prinsip Elektronik, Malvino

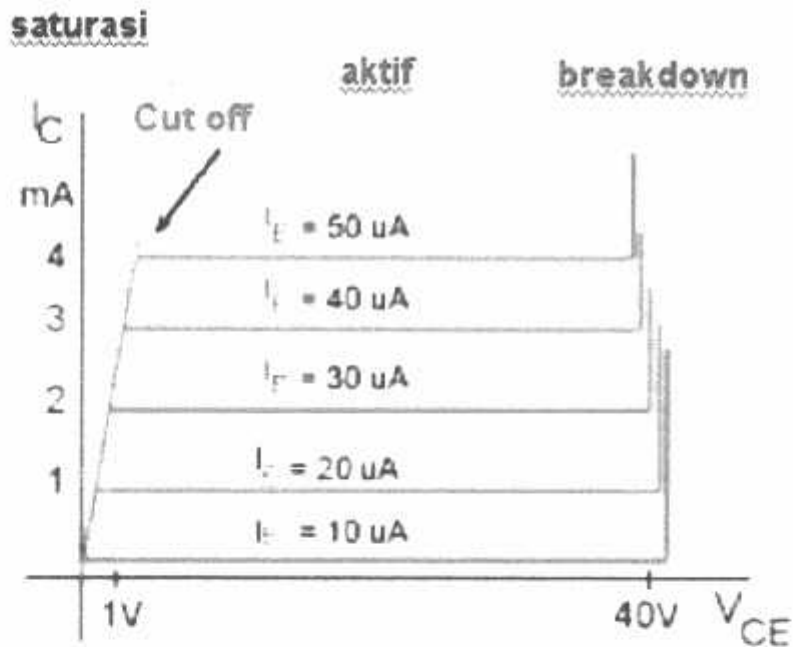
Hubungan antara  $I_B$  dan  $V_{BE}$  tentu saja akan berupa kurva dioda. Karena memang telah diketahui bahwa junction *base-emitor* tidak lain adalah sebuah dioda. Jika hukum Ohm diterapkan pada loop base diketahui adalah :

$$I_B = (V_{BB} - V_{BE}) / R_B \dots \dots \dots (2.5)$$

$V_{BE}$  adalah tegangan jepit dioda *junction base-emitor*. Arus hanya akan mengalir jika tegangan antara base-base ( $V_{BB}$ ) lebih besar dari  $V_{BE}$ . Sehingga arus  $I_B$  mulai aktif mengalir pada saat nilai  $V_{BE}$  tertentu. Besar  $V_{BE}$  umumnya tercantum di dalam *databook*. Tetapi untuk penyederhanaan umumnya diketahui  $V_{BE} = 0.7$  volt untuk transistor silikon dan  $V_{BE} = 0.3$  volt untuk transistor germanium.

#### 2.6.5. Kurva Kolektor

Sekarang sudah diketahui konsep arus base dan arus kolektor. Satu hal lain yang menarik adalah bagaimana hubungan antara arus base  $I_B$ , arus kolektor  $I_C$  dan tegangan kolektor-emiter  $V_{CE}$ . Pada grafik berikut telah diplot beberapa kurva kolektor arus  $I_C$  terhadap  $V_{CE}$  dimana arus  $I_B$  dibuat konstan.



**Grafik 2.2. Kurva Kolektor**  
 Sumber : Prinsip – prinsip Elektronik, Malvino

Dari kurva ini terlihat ada beberapa *region* yang menunjukkan daerah kerja transistor. Pertama adalah daerah *saturasi*, lalu daerah *cut-off*, kemudian daerah aktif dan seterusnya daerah *breakdown*. Pada grafik 2-2 terdapat daerah ditengah dimana besar  $V_{CE}$  antara 1 sampai 40 V. Daerah ini merupakan daerah kerja normal transistor. Pada daerah ini, dioda emitor terbias maju dan dioda kolektor terbias balikkan. Ini disebut daerah aktif.

Daerah operasi yang lain adalah daerah *Breakdown*. Transistor tidak boleh bekerja pada daerah ini karena akan rusak. Dari kurva kolektor, terlihat jika tegangan  $V_{CE}$  lebih dari 40V, arus  $I_C$  menanjak naik dengan cepat. Transistor pada daerah ini disebut berada pada daerah *breakdown*.

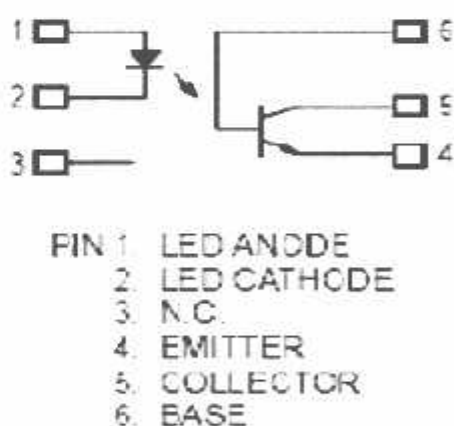


Daerah operasi yang ketiga, terdapat bagian naik awal kurva, dimana VCE antara 0 dan sepersepuluh volt. Pada daerah ini dioda kolektor tidak memiliki tegangan positif yang cukup untuk mengumpulkan semua elektron bebas yang diinjeksikan ke basis. Daerah ini disebut daerah saturasi.

Pada kurva kolektor terdapat kurva yang tidak diharapkan kurva yang terletak paling bawah. Besar arus basis pada kurva itu adalah nol akan tetapi terdapat arus kolektor yang kecil. Kurva paling bawah ini disebut daerah *cut off*.

## 2.7. Optocoupler IC 4N25

Optocoupler merupakan sebuah perangkat elektronik yang terdiri dari dua buah komponen yaitu LED (*Light Emitting Diode*) sebagai sumber cahaya *Infrared* (IR) dan photo transistor sebagai penangkap cahaya *Infrared*. Kekuatan cahaya yang dihasilkan LED tergantung dari arus yang mengalir pada LED tersebut. Sedangkan arus kolektor pada transistor juga ditentukan oleh kekuatan intensitas cahaya IR dari LED tersebut, berikut skematik dari optocoupler :



**Gambar 2.10. Skematik optocoupler 4N25**  
 Sumber : [www.jameco.com](http://www.jameco.com)

Dalam gambar diatas dapat dilihat rangkaian elektrik dalam kemasan optocoupler 4N25 .Penempatan antara LED dan phototransistor diberi jarak tertentu yang digunakan untuk melewatkan berkas cahaya dari LED ke phototransistor . Celah kosong ini berfungsi sebagai jarak agar jalur elektrik LED tidak terhubung dengan phototransistor. Komponen ini biasa digunakan sebagai pengendali terhadap beban yang bertegangan lebih tinggi dari tegangan sistem control.

Adapun cara kerja dari optocoupler yaitu bila cahaya dari LED mati maka phototransistor tidak dapat bekerja atau tersumbat (*Cutoff*) , sehingga arus kolektor tidak mengalir .Sebaliknya jika cahaya dari LED menyala maka phototransistor akan bekerja(*saturation*) ,sehingga arus kolektor akan mengalir. Disini optocoupler digunakan untuk mengisolasi mikrokontroler AT89S8253 dari adanya arus balik yang terjadi pada rangkaian driver motor dan dengan adanya tambahan diode diharapkan dapat mencegah tegangan balik yang dihasilkan kumparan relay sesaat setelah dimatikan.

## **2.8. Max 232 Sebagai Pengubah Tegangan TTL**

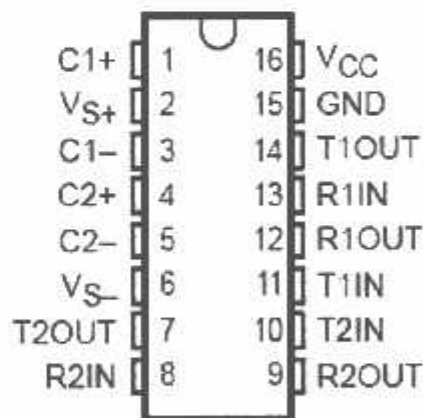
Max 232 merupakan pengubah TTL ke level tegangan RS-232. Max 232 memiliki dua *driver* mengonversikan RS-232 ke level TTL, dan dua penerima yang merubah TTL ke RS-232. Max 232 memiliki 16 pin dan dioperasikan dengan empat buah kapasitor yang memiliki nilai 1uF.

RS-232 merupakan salah satu jenis antar muka (*interface*) dalam proses transfer data antar perangkat dalam bentuk serial transfer. RS-232 merupakan singkatan dari *Recommbeded Standard number 232*. Alat ini dibuat oleh

*Electronic Industry Assosiation*, untuk *Interface* antara peralatan *terminal data* dan peralatan komunikasi data, dengan menggunakan *data binner serial* sebagai data yang ditransmisikan IC MAX 232 ini mempunyai empat buah bagian konverter yaitu dua buah *driver receiver* dan dua buah *driver transmitter*.

Saluran kabel data Siemens C55 menggunakan *standard RS-232*, dimana *logic 0 (low)* dinyatakan sebagai tegangan antara +3 volt sampai +10 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara -3 volt sampai -10 volt. Level tegangan ini tidak sesuai dengan tegangan *level* tegangan yang dipakai pada port *port seri ATmega 16* yang menggunakan *Standard TTL (Transistor Transistor Logic)*, yaitu *level* tegangan baku dalam rangkaian – rangkaian *digital*.

Dalam *standard TTL*, *logic 0 (low)*, dinyatakan sebagai tegangan antara 0 volt sampai 0.8 volt, dan *logic 1 (high)* dinyatakan sebagai tegangan antara 3.5 volt sampai 5 volt. Untuk dua MCU yang dihubungkan secara *serial* pada jarak tertentu maka dibutuhkan IC MAX 232 karena *level* tegangan TTL terlalu kecil untuk ditransfer.

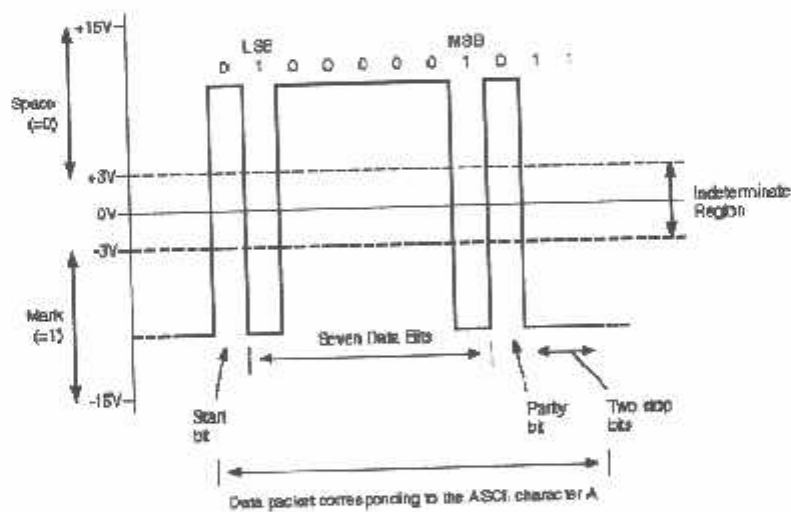


**Gambar 2.11. IC MAX 232**

**Sumber :** [www.electronictlab.com](http://www.electronictlab.com), **DataSheet MAX 232**

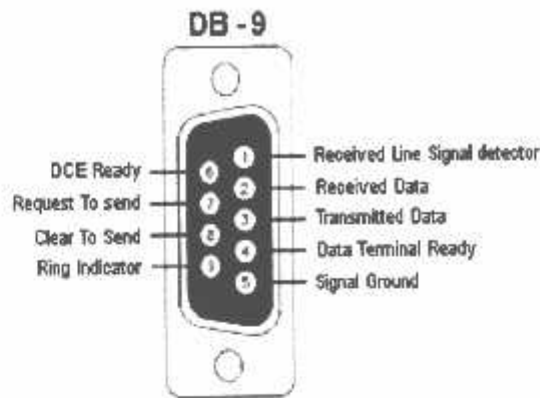
RS-232 merupakan seperangkat alat yang berfungsi sebagai *interface* dalam proses *transfer* data secara *serial*. Metode pengiriman secara *serial* RS-232 adalah *asinkron*. Pengiriman *asinkron* berarti tidak membutuhkan pewaktu sebagai *sinkronisasi*. Dalam pengiriman secara *serial asinkron*, clock dapat dikirimkan, tetapi dikondisikan oleh *timing start bit* yang merupakan isyarat dari sumber ke tujuan untuk mendekodekan adanya pengiriman karakter sudah selesai dikirim.

Karakteristik *electris* dan sistem RS-232 adalah mempunyai tegangan keluaran antara -15 volt sampai dengan +15 volt. Tegangan +3 sampai +15 volt untuk logika '0' / *spacing* dan tegangan -3 sampai -15 volt untuk logika '1' / *marking*. Hal tersebut dinyatakan dalam gambar 2-17.



**Gambar 2.12. 1 byte of Async Data**  
**Sumber :** [www.arcelect.com](http://www.arcelect.com) , rs232

Pada pin out handphone Siemens C55 menggunakan koneksi kabel data *serial* yang menggunakan *Standard* RS-232. Fasilitas ini menggunakan konektor DB-9. Gambar konektor DB-9 seperti terdapat dalam gambar 2.21.



**Gambar 2.13. Konektor DB-9**  
**Sumber : Interfacing Komputer dan mikrokontroler**

Fungsi masing – masing pin pada DB-9 seperti terdapat dalam tabel 2.4

**Tabel 2.6. Fungsi Pin RS-232 dalam DB-9**

Pin	Nama	Fungsi
1	DCD (Data Carrier Detect)	Mendeteksi Sinyal Carrier dari modem lain
2	RD (Received Data Line) / (RXD)	Pengiriman data serial dari DCE ke DTE
3	TD (Transmitted Data Line) / (TXD)	Pengiriman data serial dari DTE ke DCE
4	DTR (Data Terminal Ready)	Memberitahukan DCE bahwa DTE telah aktif dan siap untuk bekerja
5	Ground	Referensi semua tegangan antar muka
6	DSR (Data Set Ready)	Memberitahukan DTE bahwa DCE telah aktif dan siap untuk bekerja
7	RTS (Request To Send)	Memberitahukan DCE bahwa DTE akan mengirim data
8	CTS (Clear To Send)	Memberitahukan DTE bahwa DCE siap menerima data
9	RI (Ring Indikator)	Aktif jika <i>modem</i> menerima sinyal ring jalur telepon

**Sumber : Interfacing Komputer dan mikrokontroler**

Jalur data (TXD dan RXD) untuk transport data, TXD adalah jalur *output* serial handphone, data dikirim dari pin ini. Sedangkan RXD adalah penerima untuk handphone, data yang datang akan diterima oleh pin ini. Pin ke empat adalah *output* (RTS) dimana sebuah sinyal akan diberikan pada alat yang dihubungkan dengan maksud meminta kiriman data. CTS adalah sinyal masukan yang menunggu sinyal dari alat yang terhubung ketika alat tersebut menerima sinyal RTS dan bisa menerima data maka ia akan mengirimkan sinyal balik yang merupakan CTS. DTR adalah sinyal keluaran yang memberi tanda bahwa ada alat yang terhubung dan akan mengirimkan data. DSR merupakan sinyal input yang mana jika alat yang terhubung menerima sinyal DTR ia akan memberi sinyal balik kemudian diterima sebagai sinyal DSR.

## 2.9. Relay

*Relay* merupakan suatu alat untuk menghubungkan atau memutuskan kontak antara komponen satu dengan yang lain. Dalam memutus atau menghubungkan kontak digerakkan oleh *fluksi* yang timbul akibat adanya magnet listrik. Jadi fluksi inilah yang menghubungkan atau memutuskan kontak dan antara kumparan dengan bagian saklar tidak ada hubungan listrik.

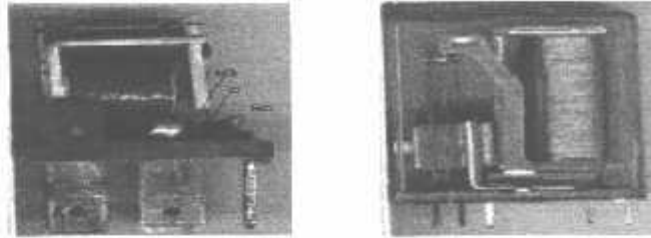
Ada beberapa jenis susunan kontak *relay* dimana semuanya terisolasi terhadap arus listrik yang ada didalam kumparan. Jenis susunan kontak sebagai berikut:

1. *Normally Open* ( normal membuka)

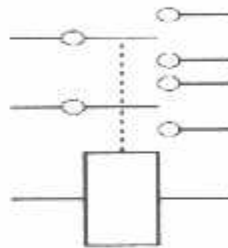
Yaitu kontak – kontak tertutup pada saat kumparan relay dialiri arus listrik.

## 2. *Normally Close* ( normal tertutup)

Yaitu kontak – kontak terbuka pada saat kumparan relay dialiri arus listrik.



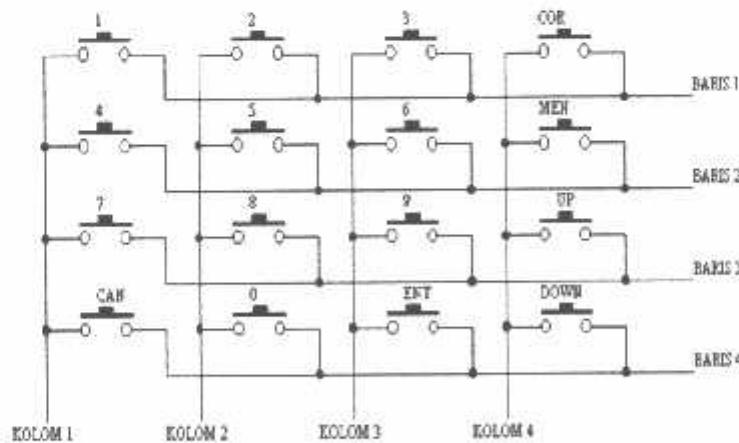
**Gambar 2.14. Bentuk Fisik Relay**  
Sumber : [www.wikipedia.org](http://www.wikipedia.org), Relay



**Gambar 2.15. Relay DPDT (Double – Pole, Double – Terminal)**  
Sumber : [www.wikipedia.org](http://www.wikipedia.org), Relay

## 2.10. Keypad

*Keypad* merupakan komponen yang digunakan sebagai sarana untuk memasukkan data ke komputer atau minimum system mikrokontroler.



**Gambar 2.16. Keypad Matrik 4x4**  
Sumber : [www.wikipedia.org](http://www.wikipedia.org), keypad

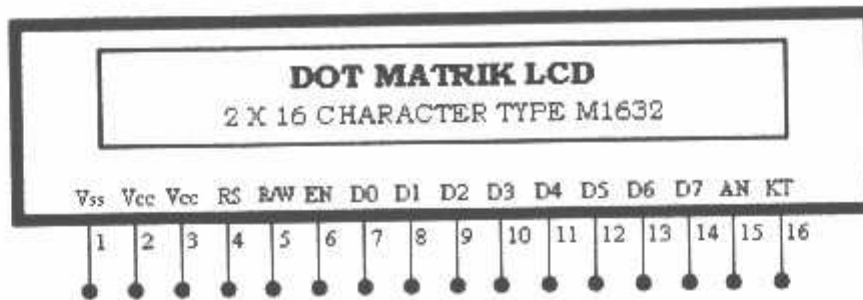
Rangkaian ini dapat dianalogikan dengan empat buah kabel terbuka yang disilangkan dengan empat buah kabel terbuka yang lain (diletakan di atasnya). Perlakuan ini akan memperoleh 16 titik persilangan. bila pada suatu titik, kabel yang bersilangan itu disentuh (salah satu ditekan sehingga menyentuh kabel yang bersilangan dibawahnya), maka diasumsikan bahwa tombol *keypad* pada posisi persilangan tersebut ditekan.

### 2.11. LCD (*Liquid Crystal Display*)

*Liquid Crystal Display* adalah modul tampilan berkonsumsi daya yang relatif rendah dan terdapat sebuah kontroller CMOS didalamnya. Kontroler tersebut sebagai pembangkit karakter dari ROM/RAM dan *display* data RAM. Semua fungsi tampilan dikontrol oleh suatu intruksi dan modul LCD dapat dengan mudah untuk diinterfacekan dengan mikroprossor/mikrokontroller. Input yang diperlukan untuk mengendalikan modul ini berupa bus data yang termultipleks dengan bus alamat dan 3 bit sinyal kontrol. Pengendali *dot matrik* LCD dilakukan secara internal pada modul LCD sendiri.

LCD merupakan suatu bentuk kristal cair yang akan beremulsi apabila dikenakan tegangan padanya. Tampilannya ini berupa *dot matrik* 5 x LCD sehingga jenis huruf yang dapat ditampilkan akan lebih banyak dan lebih baik resolusinya jika dibandingkan dengan 7 segment.





**Gambar 2.17. Deskripsi pin pada LCD Tipe M1632**  
**Sumber : LCD Manual Book**

LCD tipe M1632 memiliki ciri-ciri sebagai berikut :

- LCD ini terdiri dari 32 karakter dengan 2 baris masing-masing 16 karakter dengan *display dot matrik 5 x 7*
- Karakter generator ROM dengan 192 tipe karakter
- Karakter generator RAM dengan 8 tipe karakter
- 80 x 8 display data RAM
- Dapat diinterfacekan ke MPU 8 atau 4
- Dilengkapi fungsi tambahan : *display clear, cursor home, display ON/OFF, cursor ON/OFF, display character blink, cursor shift, dan display shift.*
- Internal Data
- Internal Otomatis, reset pada saat power ON
- +5 volt PS Tunggal

Tabel 2.7. Fungsi Pin – Pin LCD

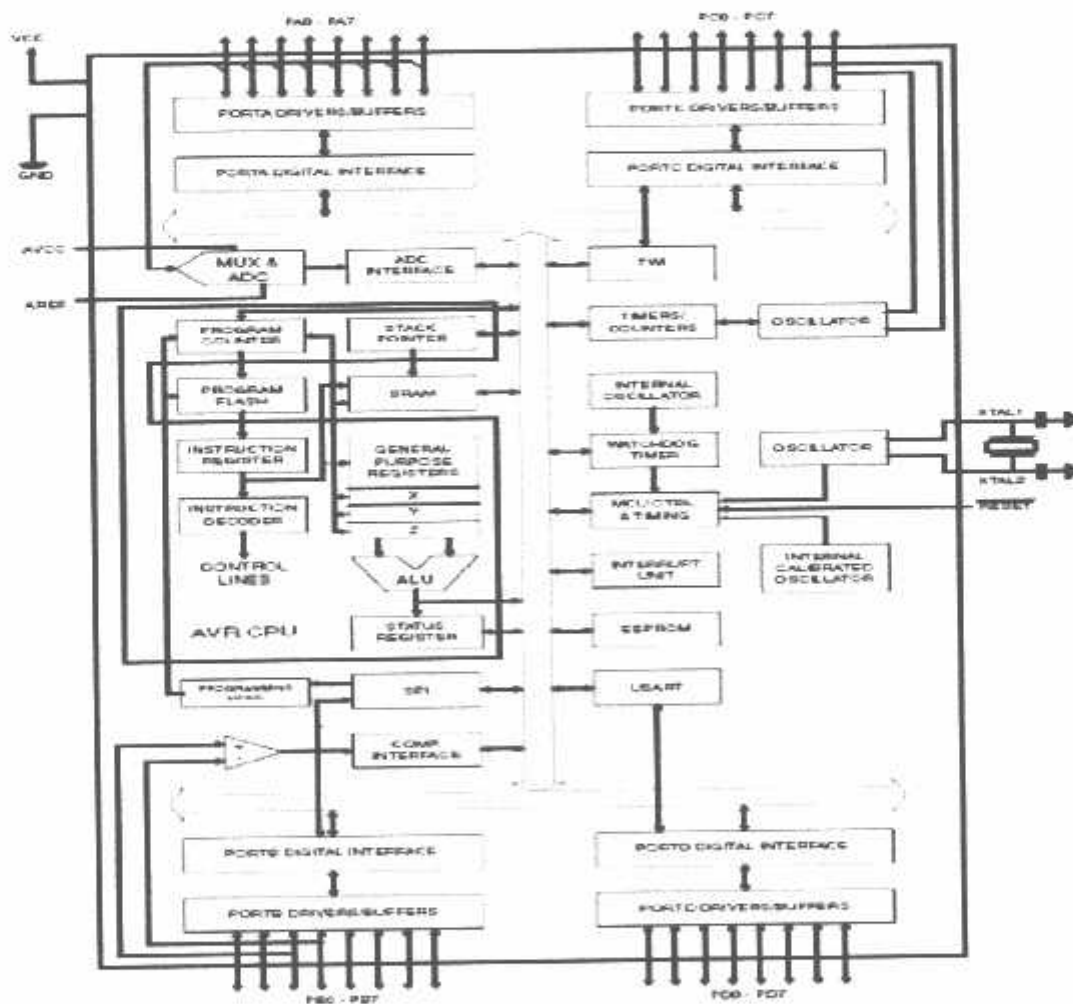
Nama Pin	Jumlah	I/O	Tujuan	Fungsi
DB0-DB3	4	I/O	MPU	Tri state bidirectional lower data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus
DB4-DB7	4	I/O	MPU	Tri state bidirectional upper four data bus: data dibaca dari modul ke MPU atau dari MPU ditulis ke modul melalui bus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

Sumber : LCD Module User Manual

## 2.12. Mikrokontroler ATMEGA 16

Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bit word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock. Secara umum, AVR dapat dikelompokkan menjadi 4 kelas, yaitu ATtiny, keluarga AT90Sxx, keluarga ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, peripheral, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, mereka bisa dikatakan hampir sama.

### 2.12.1. Arsitektur ATmega16



**Gambar 2.18. Blok diagram fungsional ATmega16**  
Sumber : [www.atmel.com](http://www.atmel.com), datasheet ATmega16

Dari gambar tersebut dapat dilihat bahwa ATmega16 memiliki bagian sebagai berikut :

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah *Timer/Counter* dengan kemampuan perbandingan.
4. CPU yang terdiri atas 32 buah register.
5. *Watchdog Timer* dengan osilator internal.
6. SRAM sebesar 1 Kbyte .
7. Memori Flash sebesar 16 kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. Port antarmuka SPI.
10. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

### 2.12.2. Fitur ATmega16

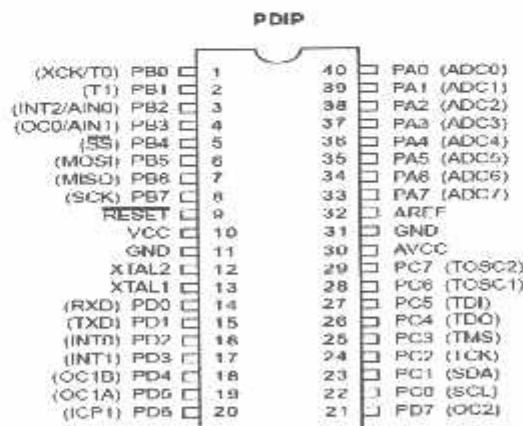
Kapabilitas detail dari ATmega16 adalah sebagai berikut :

1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori *flash* 16 KB, *SRAM* sebesar 1024 byte, dan *EEPROM* (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 byte.
3. *ADC* internal dengan fidelitas 10 bit sebanyak 8 *channel*.
4. Portal komunikasi serial (*USART*) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.

### 2.12.3. Konfigurasi Pin ATmega16

Konfigurasi pin ATmega16 bisa dilihat pada gambar 1.2. Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin ATmega16 sebagai berikut :

- A. VCC merupakan pin yang berfungsi sebagai pin masukan catu daya.
- B. GND merupakan pin *ground*.
- C. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masukan ADC.
- D. Port B (PB0..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, komparator analog, dan SPI.
- E. Port C (PC0..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan *Timer Oscilator*.
- F. Port D (PD0..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interupsi eksternal, dan komunikasi serial.
- G. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
- H. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
- I. AVCC merupakan pin masukan tegangan untuk ADC.
- J. AREF merupakan pin masukan tegangan referensi ADC.



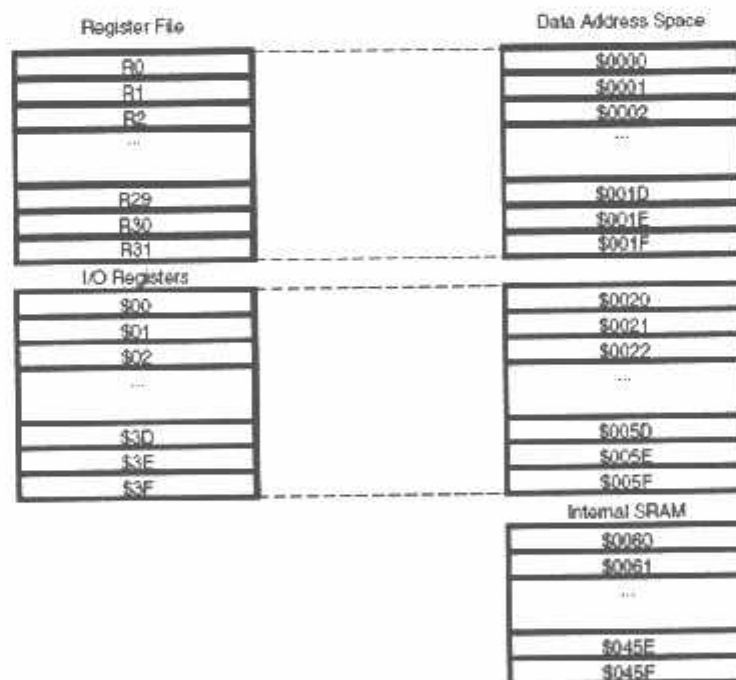
**Gambar 2.19. Pin ATmega16**

Sumber : [www.atmel.com](http://www.atmel.com), datasheet ATmega16

#### 2.12.4. Peta Memori

AVR ATmega16 memiliki ruang pengamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 buah register umum, 64 buah register I/O, dan 512 byte SRAM Internal.

Register keperluan umum menempati space data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan control terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari \$20 hingga \$5F. Register tersebut merupakan register yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler, seperti control register, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. Alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$60 sampai dengan \$25F. Konfigurasi memori data ditunjukkan pada gambar dibawah ini.



Gambar 2.19. Konfigurasi Memori Data AVR ATmega16  
Sumber : [www.atmel.com](http://www.atmel.com), *datasheet* ATmega16

Memori program yang terletak dalam *Flash PEROM* tersusun dalam word atau 2 byte karena setiap instruksi memiliki lebar 16-bit atau 32-bit. AVR ATmega16 memiliki 16KByteX16-bit *Flash PEROM* dengan alamat mulai dari \$000 sampai \$FFF. AVR tersebut memiliki 12-bit *Program Counter (PC)* sehingga mampu mengamati isi *Flash*.



**Gambar 2.20. Peta Memory Program AVR ATmega16**  
**Sumber :** [www.atmel.com](http://www.atmel.com), *datasheet* ATmega16

Selain itu, AVR ATmega16 juga memiliki memori data berupa EEPROM 8-bit sebanyak 512 *byte*. Alamat EEPROM dimulai dari \$000 sampai \$1FF.

### 2.12.5. Status Register (SREG)

Status register adalah register berisi status yang dihasilkan pada setiap operasi yang dilakukan ketika suatu instruksi dieksekusi. SREG merupakan bagian dari inti CPU mikrokontroler.

#### a. Bit 7 – I : *Global Interrupt Enable*

Bit harus diset untuk meng-enable interupsi. Setelah itu, anda dapat mengaktifkan interupsi mana yang akan anda gunakan dengan cara meng-enable bit control register yang bersangkutan secara individu. Bit

akan di-clear apabila terjadi suatu interupsi yang dipicu oleh hardware, dan bit tidak akan mengizinkan terjadinya interupsi, serta akan diset kembali oleh instruksi RETI.

b. Bit 6 – T : *Bit Copy Storage*

Instruksi BLD dan BST menggunakan bit-'T' sebagai sumber atau tujuan dalam operasi bit. Suatu bit dalam sebuah register GPR dapat disalin ke bit menggunakan instruksi BST, dan sebaliknya bit-T dapat disalin kembali ke suatu bit dalam register GPR menggunakan instruksi BLD.

c. Bit 5 – H : *Half Carry Flag*

d. Bit 4 – S : *Sign Bit*

Bit-S merupakan hasil operasi EOR antara flag-N (negatif) dan flag V (komplemen dua overflow).

e. Bit 3 – V : *Two's Complement Overflow Flag*

Bit berguna untuk mendukung operasi aritmatika.

f. Bit 2 – N : *Negative Flag*

Apabila suatu operasi menghasilkan bilangan negatif, maka *flag-N* akan diset.

g. Bit 1 – Z : *Zero Flag*

Bit akan diset bila hasil operasi yang diperoleh adalah nol.

h. Bit 0 – C : *Carry Flag*

Apabila suatu operasi menghasilkan *carry*, maka bit akan diset.



## BAB III

### PERANCANGAN DAN PEMBUATAN ALAT

#### 3.1. Pendahuluan

Pada bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan fungsi masing-masing blok rangkaian tersebut. Secara garis besar terdapat dua bagian perangkat yang ada yaitu :

- Perancangan perangkat keras (*Hardware*).
- Perancangan perangkat lunak (*Software*).

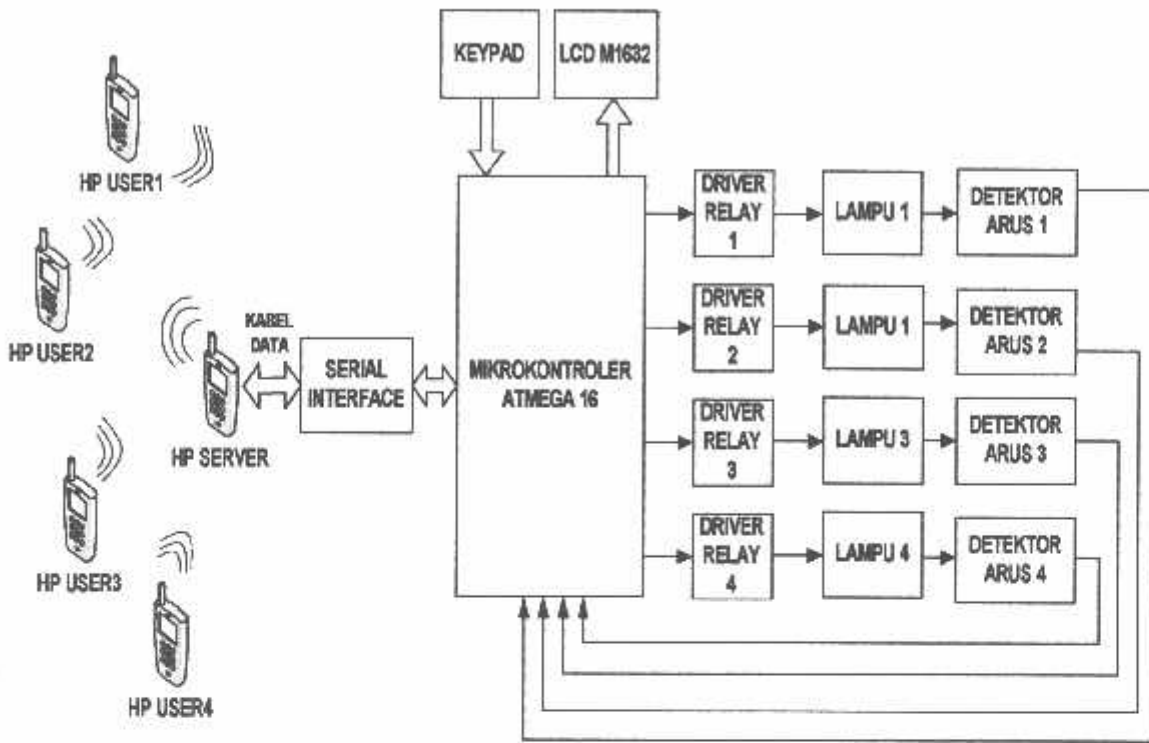
Pada perancangan perangkat keras akan meliputi seluruh *peripheral* yang digunakan pada sistem ini. Pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum. Akan tetapi kedua perangkat ini dalam kerjanya akan saling menunjang satu sama lain.

Pada perangkat keras sendiri terdiri dari beberapa rangkaian antara lain, rangkaian detektor arus, driver relay, minimum sistem mikrokontroler, serial interface. Pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum

#### 3.2. Perancangan Perangkat Keras

Perancangan perangkat keras yang direncanakan meliputi, blok diagram keseluruhan dan prinsip kerja alat, pembuatan skema seluruh rangkaian yang direncanakan, penghitungan nilai komponen yang digunakan, dan perakitan seluruh komponen.

### 3.2.1. Blok Diagram Alat



Gambar 3.1. Diagram Blok Sistem

Fungsi masing – masing diagram blok :

- **Keypad**

Berfungsi untuk menginputkan nomor hand phone yang digunakan oleh pengguna atau *user* pada mikrokontroller ATMEGA 16, keypad yang digunakan adalah keypad matrix 4x4.

- **LCD M1632**

Berfungsi untuk menampilkan langkah-langkah ataupun hasil pada saat proses memasukkan nomor handphone pada mikrokontroler.

- **Mikrokontroller**

Sebagai pemroses informasi data masukan dari handphone, keypad matrik, dan detektor arus, yang hasilnya akan digunakan untuk mengendalikan driver relay, atau mengirim informasi status sistem ke handphone.

- **Driver Relay**

Berfungsi untuk memutus aliran arus yang mengalir pada beban listrik.

- **Lampu**

Berfungsi sebagai simulasi beban listrik yang akan dikendalikan oleh sistem pengendali.

- **Detektor Arus**

Merupakan elemen untuk mendeteksi ada atau tidaknya aliran arus listrik pada beban peralatan listrik.

- **RS 232**

Berfungsi sebagai *interface* antara mikrokontroller dengan hand phone

- **Handphone**

Berfungsi sebagai fasilitas untuk menyampaikan atau menerima informasi SMS dari pengguna, handphone yang digunakan adalah Siemens C55.

### 3.2.2. Prinsip Kerja Alat

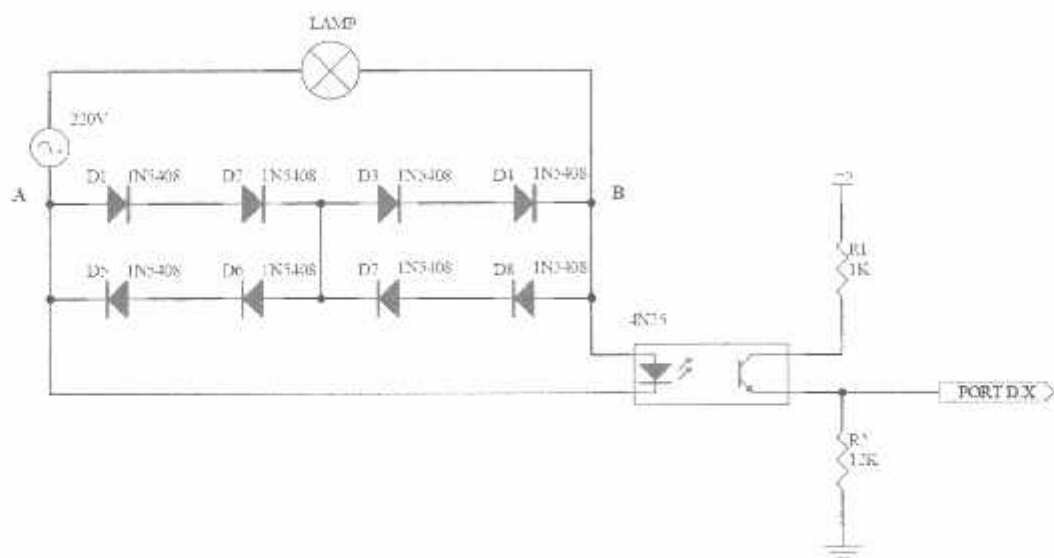
EEPROM mikrokontroler harus diisi nomor handphone pengguna (*User*) yang akan digunakan untuk mengendalikan sistem, melalui keypad matrix 4x4 nomor dapat dimasukkan ke EEPROM mikrokontroler, proses untuk memasukkan nomor handphone tersebut ditampilkan pada LCD (*Liquid Crystal Display*).

Pada prinsipnya sistem dikendalikan sepenuhnya oleh handphone pengguna, apabila sistem mendapatkan informasi dari handphone pengguna berupa SMS (*Short Message Service*) yang disimpan pada inbox handphone operator, setelah muncul *alert ringtone* SMS mikrokontroler akan memerintahkan handphone operator untuk mengeluarkan data tersebut melalui *pinoutsnya*. Melalui *pinouts* tersebut data SMS yang dikirimkan ke mikrokontroler adalah secara serial, agar level tegangan data yang dikirimkan bisa selaras antara mikrokontroler dengan handphone operator maka data dilewatkan melalui *level converter RS-232*.

Terdapat dua format data SMS yang dikirimkan, untuk format data SMS yang pertama adalah berisi informasi untuk pengendalian peralatan listrik, sedangkan format yang kedua adalah untuk *check status* sistem, mikrokontroler akan membaca data SMS yang diterima, apakah berupa pengendalian atau check status sistem, apabila berupa pengendalian maka mikrokontroler akan memproses data tersebut dan hasilnya akan digunakan untuk mengendalikan driver relay, relay akan merespon sinyal dari driver relay dengan memutus ataupun menyambung aliran arus listrik yang mengalir pada beban listrik. Sedangkan apabila data SMS yang dikirimkan ke mikrokontroler berupa check status sistem, maka mikrokontroler dengan segera membaca data inputan dari detektor arus, data

dari detektor arus ini disimpan pada EEPROM mikrokontroler, kemudian mikrokontroler memerintahkan handphone operator untuk mengirimkan data yang tersimpan pada EEPROM tersebut ke handphone pengguna tentunya berupa teks SMS.

### 3.2.3. Rangkaian Detektor Arus

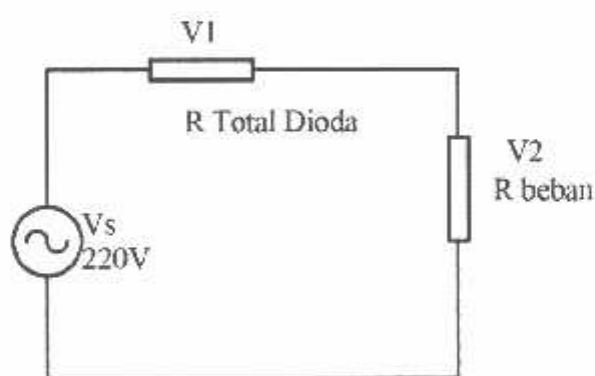


**Gambar 3.2. Rangkaian Detektor Arus**  
**Sumber : Perencanaan dan Pembuatan**

Rangkaian detektor arus terdiri dari delapan dioda yang disusun seperti pada gambar 3.2, penyusunan detektor dimaksudkan untuk mendeteksi arus yang mengalir ke beban dan menjaga agar tegangan AC tetap pada keadaan sinus. Dengan sifat dioda yang akan mengalirkan arus listrik dengan mudah atau dengan kata lain mudah konduksi apabila diberikan bias maju (*forward bias*), dan sebaliknya, dioda akan sukar konduksi apabila diberikan tegangan bias mundur (*reverse bias*). Dengan mengkonfigurasi delapan dioda seperti pada gambar 3.2 diatas nantinya akan mempertahankan tegangan yang melalui rangkaian dioda ini tetap pada kondisi AC.

Dengan tegangan jatuh yang dihasilkan pada simpul dioda sekitar 2,8 V AC dimana tegangan tersebut diambil dengan mengambil tegangan referensi tegangan baru pada rangkaian delapan dioda yang tersusun secara *forward* dan *reverse*, dan pada empat buah dioda yang disusun seri, dihubungkan ke optocoupler yang akan mentrigger led di dalam optocoupler sehingga phototransistor pada optocoupler akan mengalami saturasi sehingga Port D.X akan aktif.

Dengan asumsi bahwa jumlah tegangan dalam sebuah rangkaian tertutup sama dengan nol, maka akan didapatkan tegangan yang dibutuhkan pada suatu titik referensi.



**Gambar 3.3. Jumlah Tegangan Dalam Rangkaian Tertutup**

Disini akan dicari tegangan pada masing-masing R dalam sebuah rangkaian tertutup, berdasarkan data yang diperoleh dari *data sheet* tegangan jatuh pada dioda saat diberikan *forward bias* adalah 0.7 volt, maka :

$$\begin{aligned}
 V_d \text{ total} &= V_{d1} + V_{d2} + V_{d3} + V_{d4} \dots\dots\dots(3.1) \\
 &= 0.7 + 0.7 + 0.7 + 0.7 \\
 &= 2.8 \text{ Volt}
 \end{aligned}$$

$$\begin{aligned}
 V_R \text{ beban} &= V_s - V_d \text{ total} \dots\dots\dots(3.2) \\
 &= 220 - 2.8 \\
 &= 217,2 \text{ Volt}
 \end{aligned}$$

Sedangkan untuk perhitungan pembagi tegangan pada keluaran detektor arus

adalah :

$$VR1 = \frac{R1}{R1 + R2} \cdot (Vcc - Vbe) \dots\dots\dots(3.3)$$

$$= \frac{1}{1+12} (5 - 0.7)$$

$$= 0.33 \text{ Volt}$$

$$VR2 = \frac{R2}{R2 + R1} \cdot (Vcc - Vbe)$$

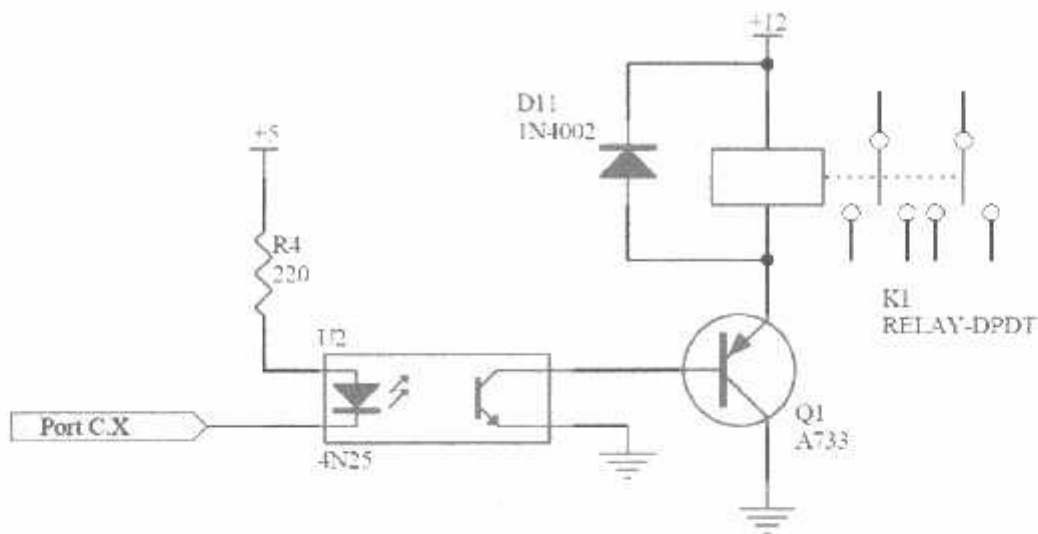
$$= \frac{12}{12+1} (5 - 0.7)$$

$$= 3.96 \text{ Volt}$$

Jadi tegangan keluaran yang menuju ke port mikrokontroler adalah 3.96 Volt

#### 3.2.4. Driver Relay

Pada perancangan *driver* ini terdiri dari sebuah *optocoupler* dan transistor A733. Pemasangan *optocoupler* berfungsi sebagai penahan *bouncing* yang muncul akibat kelebihan arus yang ditimbulkan oleh beban, sehingga dapat menyebabkan kerusakan fisik maupun nonfisik (*program*) pada mikrokontroler. Walaupun sebenarnya *bouncing* itu sendiri telah diantisipasi dengan memasang sebuah diode yang terpasang paralel dengan *relay*nya sebelum masuk ke beban. Rangkaian ini terdiri dari *optocoupler* 4N25, A733, Diode dan *relay* yang disusun seperti gambardibawah ini :



**Gambar 3.4. Rangkaian Driver Relay**

Cara kerja dari rangkaian diatas adalah untuk mengaktifkan *relay*, mikrokontroler akan mengeluarkan logika “0” pada port C.X, dengan begitu akan ada arus yang mengalir LED yang menyebabkan LED mengeluarkan cahaya dengan intensitas tertentu. Cahaya itu akan membuat transistor NPN pada *optocoupler* menjadi saturasi. Dengan rangkaian seperti diatas, kolektor dari transistor pada *optocoupler* terhubung pada basis transistor A733 dan emitornya pada *ground*, maka ketika transistor pada *optocoupler* saturasi akan membuat A733 juga mengalami saturasi. *Relay* yang digunakan memiliki tegangan 12 volt dan hambatan 345 ohm, hfe A733 adalah 270 dan  $V_{ce}$  300 mV, maka arus yang mengalir relay adalah :

$$\begin{aligned}
 I_E &= \frac{V_{EE} - V_{CE}}{R_{Relay}} \dots\dots\dots (3.4) \\
 &= \frac{12V - 0.3V}{345\Omega} \\
 &= 30.7 \text{ mA}
 \end{aligned}$$



$$I_C = I_E - I_B \dots\dots\dots (3.5)$$

$$I_B = \frac{I_C}{hfe} \dots\dots\dots (3.6)$$

Dari persamaan 3.5 dan 3.6 didapatkan :

$$I_B = \frac{I_E}{hfe + 1} \dots\dots\dots (3.7)$$

$$= \frac{0,0307}{240 + 1}$$

$$= 0.127 \text{ mA}$$

Dari datasheet dapat diketahui arus kolektor maksimum yang dapat mengalir pada transistor pada *optocoupler* 4N25 adalah 150 mA, sedangkan dengan arus basis pada TIP 32 sebesar 0.2197mA masih dibawah arus maksimum yang dapat dilewatkan pada transistor *optocoupler* 4N25. Pada *optocoupler* 4N25 memiliki  $V_{led} = 3$  volt dan  $I_{led} = 60$  mA sehingga dapat dicari *resistor* yang harus dipakai :

$$R_{LED} = \frac{V_{CC} - V_{LED}}{I_{LED}} \dots\dots\dots (3.8)$$

$$= \frac{5V - 3V}{0.06A}$$

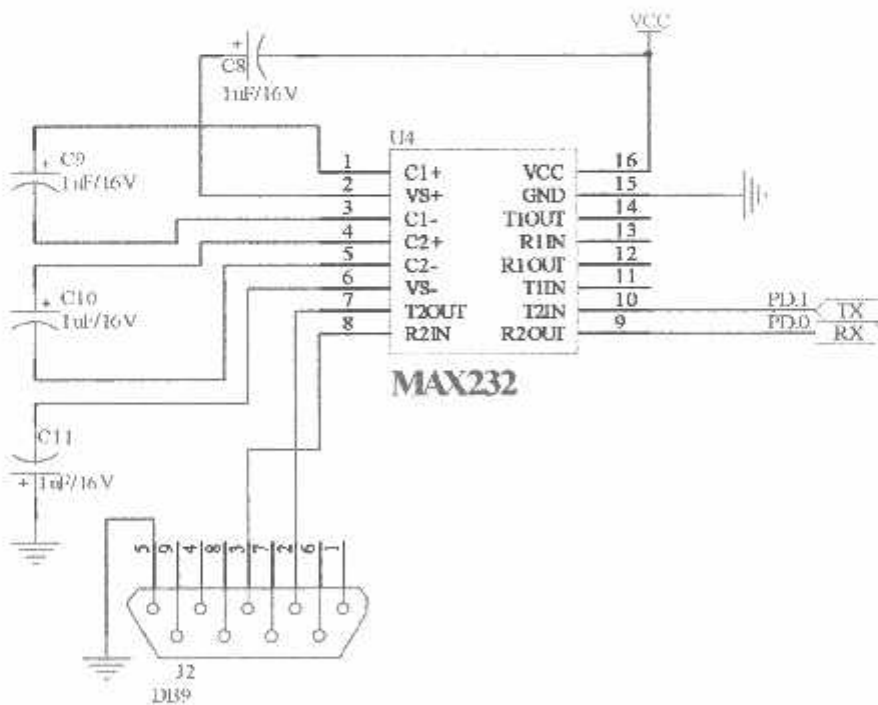
$$= 33.3$$

Karena besar hambatan dipasaran sebesar 33.3 ohm tidak ada maka dipakai hambatan sebesar 39 ohm.

### 3.2.5. Rangkaian Antar Muka RS-232

Pada perencanaan hubungan antara *MCU* dan *Handphone* menggunakan komunikasi data secara *serial*. Adapun kaki atau pin-pin yang dipakai pada kabel data serial Siemens C55 adalah pada pin no.2 sebagai *received data*, dan pin no.3 sebagai *transmitted data* dan pin no 5 untuk *Signal ground*. Level tegangan dari RS-232 harus disesuaikan ke level tegangan TTL menggunakan IC MAX 232. Kecepatan transfer data per *bit* menggunakan 19200 bps.

Data biner yang berasal dari kabel data *Handphone* mempunyai level tegangan antara +3 V sampai +15 V dan -3 V sampai -15 V, sebelum masuk pada mikrokontroler terlebih dahulu tegangannya melalui sebuah IC MAX 232, dan diubah menjadi tegangan TTL sebesar 0 Volt sampai 5 Volt yang sesuai dengan tegangan untuk mencatu kerja rangkaian mikrokontroler tersebut. Rangkaian Interface RS-232 diperlihatkan pada gambar 3-5.

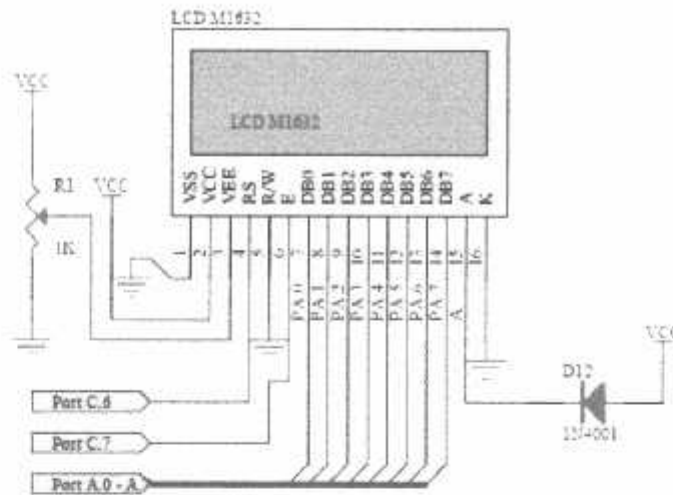


Gambar 3.5. Rangkaian Antar Muka RS-232

Perangkat Serial interface Max 232 dapat dibangun dengan menambahkan lima komponen eksternalnya. Komponen tersebut berupa capacitor  $C_1, C_2, C_3, C_4, C_5$  yang nilainya telah ditentukan pada datasheet sebesar  $1\mu\text{F}$ . Pada IC tersebut terdapat dua buah jalur pengubah level tegangan TTL menjadi level tegangan RS-232, jalur pertama adalah pin 11 (T1in) yang outputnya pada pin 14 (T1 out) serta jalur kedua adalah pin 10 (T2in) yang outputnya pada pin 7 (T2 out). Terdapat juga dua jalur pengubah level tegangan RS-232 menjadi level TTL, jalur pertama adalah pin 13 (R1 in) yang outputnya pada pin 12 (R1 out), serta jalur kedua adalah pin 8 (R2in) yang outputnya pada pin 9 (R2 out).

### 3.2.6. Perancangan Rangkaian LCD

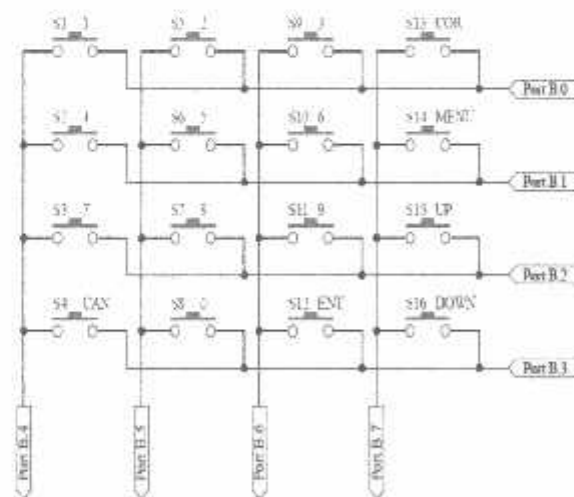
Pada perancangan ini digunakan LCD Dot Matrik 2 x 16 karakter yaitu M1632. Sinyal-sinyal yang diperlukan oleh LCD adalah RS dan Enable, sinyal RS dan Enable dipergunakan sebagai input yang outputnya dipakai untuk mengaktifkan LCD. LCD akan aktif apabila mikrokontroler memberikan instruksi tulis pada LCD. Saat kondisi RS *don't care* dan Enable 0 maka LCD tetap pada kondisi semula, pengiriman data ke LCD dilakukan saat RS berlogika 0 dan enable berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 1 dan Enable 1. Pin LCD ini untuk data terkoneksi pada *Port A* mikrokontroler ATMEGA16. Kemudian untuk RS dihubungkan pada *Port C.6*, tulis/baca (*Read/Write*) diberikan logika *low* karena disini LCD bersifat menulis data, dan yang terakhir *Enable (E)* dikendalikan dengan *Port C.7*. Gambar rangkaian LCD ditunjukkan pada gambar 3 – 10.



Gambar 3.6. Rangkaian *Liquid Crystal Display ( LCD )*

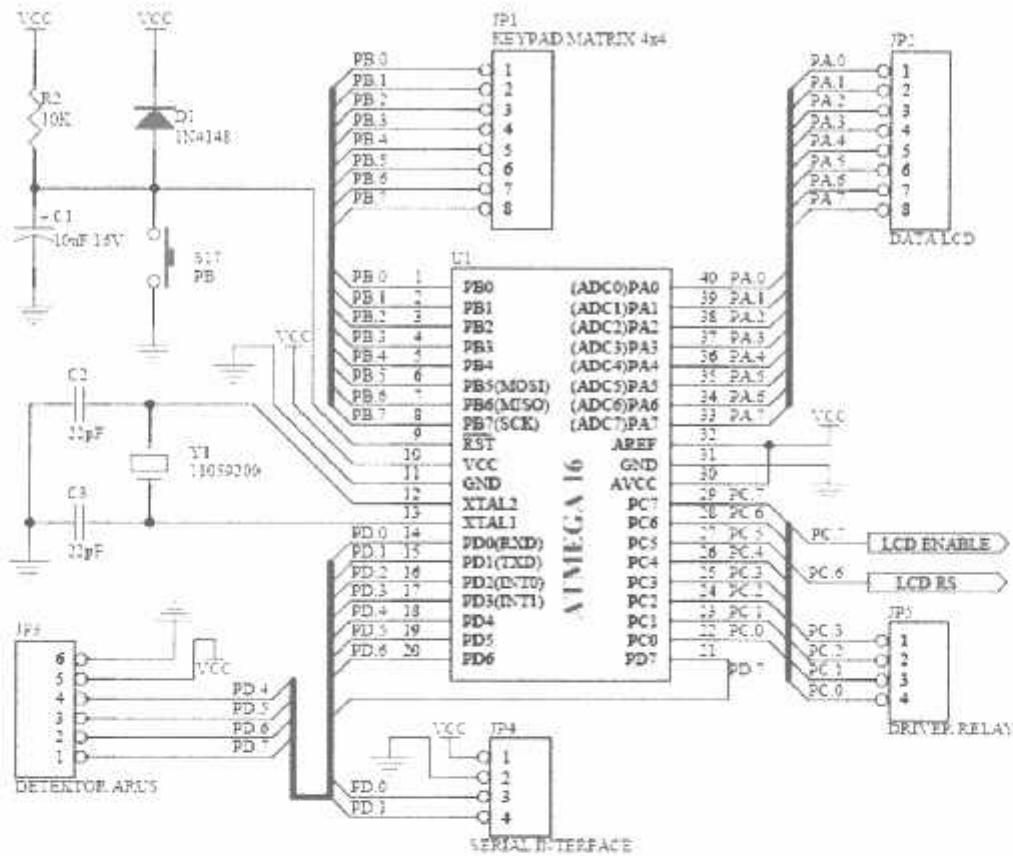
### 3.2.7. Rangkaian Keypad

Rangkaian *Keypad* dalam perencanaan ini menggunakan *Keypad matrik 4x4*. Dari tombol-tombol *Keypad* tersebut apabila ditekan akan terbentuk karakter angka (1-9), yang digunakan sebagai inputan dari mikrokontroler. Teknik pembacaan dari *Keypad* ini, yaitu model *scanning* pada jalur baris dan pembacaan jalur kolom. Bila baris dan kolom disilangkan maka akan terbentuk titik-titik potong antara keduanya seperti pada gambar 3.7. di bawah ini :



Gambar 3.7. Perancangan Keypad Matrix 4x4

### 3.2.8. Perancangan minimum sistem ATmega 16



Gambar 3.8. Minimum Sistem ATMEGA16 Dan Alokasi Pin

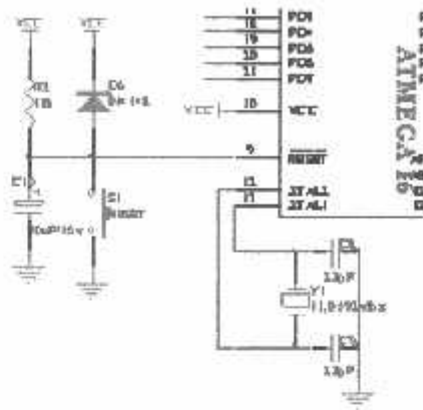
Mikrokontroler yang digunakan ialah mikrokontroler keluarga AVR ATmega 16 yang mempunyai arsitektur *RISC (Reduce Instruction Set)*. Data yang diperoleh dari handphone siemens akan diolah oleh MCU ATMEGA16 ini sebagai pemroses pengendalian maupun pemroses data jawaban atas status sistem. Untuk dapat berkomunikasi dengan handphone digunakan port serial mikrokontroler ATmega 16.

Alokasi penggunaan pin mikrokontroler :

- PA.0-PA.5 digunakan sebagai port output yang akan mengeluarkan data ke LCD M1632.

- PC.6 digunakan sebagai port output operasi RS (Register Select) pada LCD.
- PC.7 digunakan sebagai port output untuk operasi *enable* LCD (aktif atau tidaknya LCD).
- PB0-PB7 digunakan sebagai port input yang mendapat masukan dari keypad matrix 4x4
- PC0-PC3 digunakan sebagai port output yang akan mengendalikan driver relay
- PD5-PD7 digunakan sebagai port input, yang mendapat data masukan dari detektor arus
- PD.0 difungsikan sebagai pin RX untuk komunikasi data serial.
- PD.1 difungsikan sebagai pin TX untuk komunikasi data serial.
- Pin no.9 ialah pin reset mikrokontroler ATmega 16, reset terjadi bila pin ini diberi logika *low* dengan level tegangan 0 volt selama 1,5  $\mu$ s atau lebih.
- X<sub>1</sub> dan X<sub>2</sub> sebagai masukan dari rangkaian osilator kristal. Rangkaian osilator kristal terdiri atas osilator 11,0592 MHz, kapasitor C<sub>1</sub> dan C<sub>2</sub> yang masing-masing bernilai 22 pF yang akan membangkitkan pulsa *clock* yang digunakan sebagai penggerak bagi sejumlah operasi internal CPU.

Perancangan rangkaian reset pada mikrokontroler ATmega 16 ialah dengan memberikan logika low pada pin reset mikrokontroler ATmega 16. Rangkaian reset ini diperoleh dari *application note AVR Design Consideration* dari ATMEL. Berikut ialah gambar rancangan rangkaian reset pada ATmega 16 :



**Gambar 3.9. Rangkaian reset pada ATmega 16**

Osilator pada rangkaian minimum sistem ATmega 16 menggunakan kristal 11,0592 MHz dan kapasitor 22 pF. Nilai kapasitor ini diperoleh dari tabel datasheet tentang penggunaan kapasitor untuk rangkaian osilator / sistem clock pada ATmega 16. Penggunaan kristal 11,0592 MHz ini bertujuan agar perhitungan baudrate tidak mengalami error yang disebabkan karena selisih perhitungan. Perhitungan baudrate pada ATmega 16 dengan menggunakan kristal 11,0592 MHz :

Baudrate yang diinginkan ialah 19200 bps, maka nilai pada *UBRR(USART Baud Rate Register)* dapat ditentukan dengan perhitungan :

$$UBRR = \frac{f_{osc}}{16 \cdot Baud} - 1$$

$$UBRR = \frac{11059200}{16 \cdot 19200} - 1$$

$$UBRR = \frac{11059200}{307200} - 1$$

$$UBRR = 36 - 1 = 35 = 23H$$

Penggunaan kristal 11,0592 MHz memungkinkan hasil perhitungan *baudrate* tidak sisa dan *error* dari selisih perhitungan tidak ada.

### 3.3. Perencanaan Format Teks SMS

Pada sistem akan digunakan perintah – perintah menggunakan data dari sms, format sms yang direncanakan dalam sistem adalah sebagai berikut :

**1. ONN<spasi>(No Alat1)(No Alat2), (No Alat3)(No Alat4)**

Perintah SMS “ONN” berfungsi untuk menyalakan alat dengan nomor sesuai yang diberikan, misal “ONN 0123” akan menyalakan alat nomor 1, 2, dan 3.

**2. OFF<spasi>( No Alat1)(No Alat2)(No Alat3)(No Alat4)**

Perintah SMS “OFF” berfungsi untuk mematikan alat dengan nomor sesuai yang diberikan, misal “OFF 0123” akan menyalakan alat nomor 1, 2, dan 3.

**3. GET<spasi>(1234**

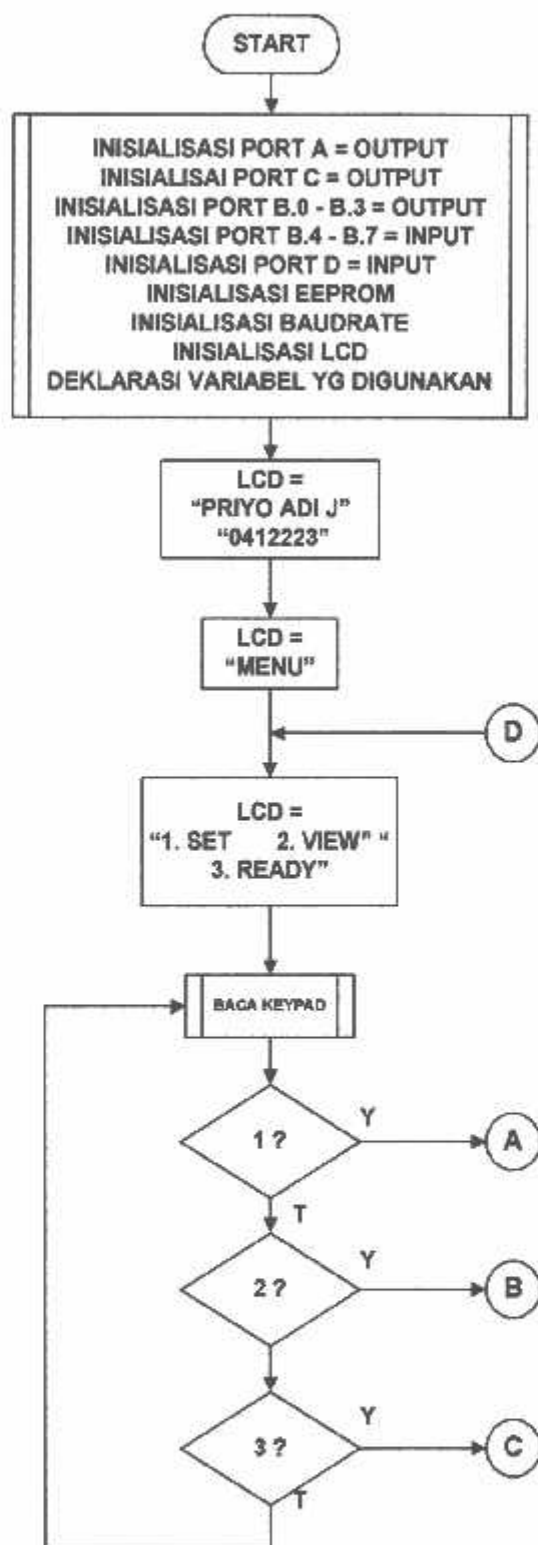
Sedangkan Perintah SMS “GET 1234” berfungsi untuk meminta laporan status. Laporan dalam bentuk empat buah angka di mana masing-masing angka menyatakan status alat. Jika angka adalah 1 maka alat menyala, dan jika angka adalah 0 maka alat mati. Contoh, jika status adalah 0001, maka alat 4 menyala, sedangkan sisanya mati.

### 3.4. Perancangan Perangkat Lunak

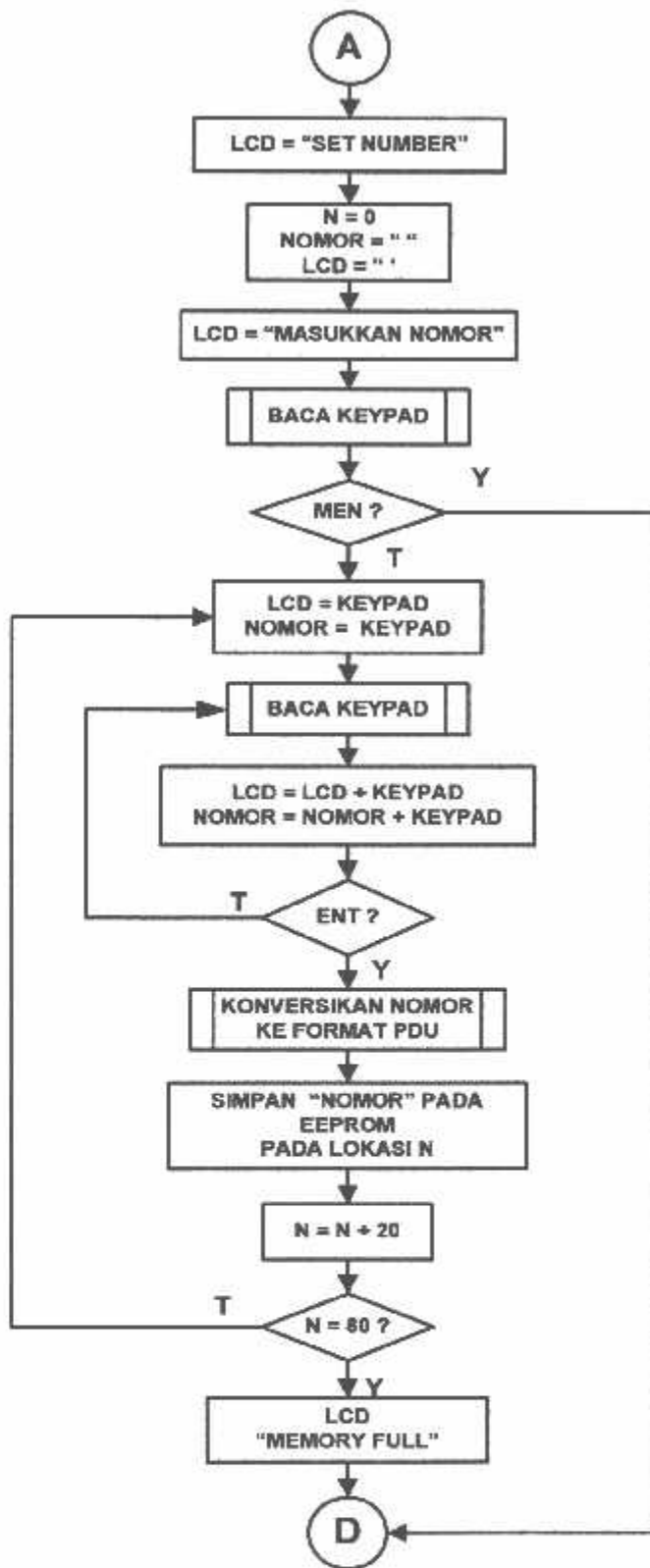
Perancangan perangkat lunak (*software*) yang digunakan dalam perencanaan dan pembuatan alat akan dipaparkan dalam *flowchart* sistem secara keseluruhan. Pembuatan *software* hanya dilakukan pada mikrokontroler menggunakan bahasa pemrograman *Basic* dengan bantuan *compiler BASCOM*



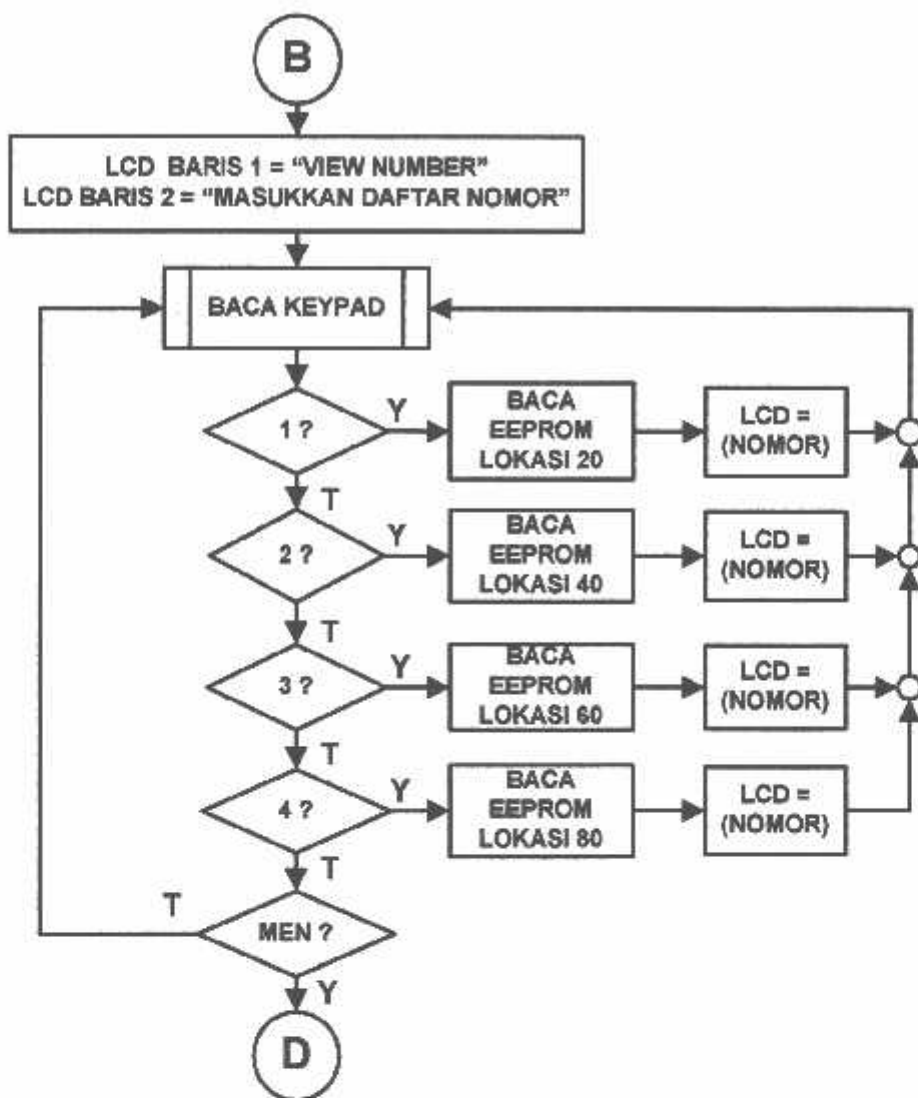
AVR. Diagram alir atau *Flowchart* program secara keseluruhan adalah sebagai berikut:



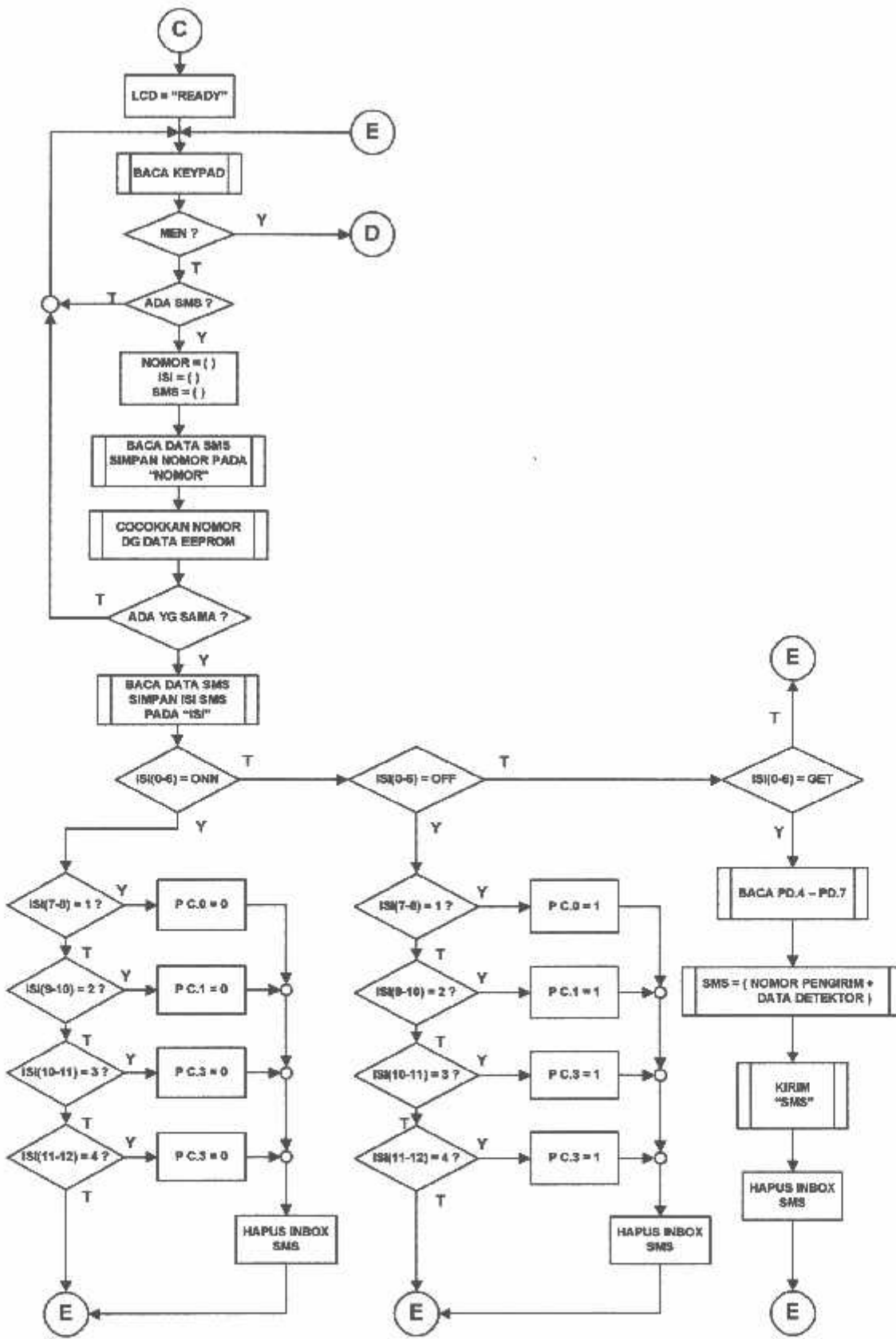
Gambar 3.10. *Flow chart* Program Bagian Ke-1



Gambar 3.11. *Flow chart* Program Bagian Ke-2



Gambar 3.12. Flow chart Program Bagian Ke-3



Gambar 3.13. Flow chart Program Bagian Ke-4



## **BAB IV**

### **PENGUJIAN DAN HASIL ANALISA**

Pengujian dilakukan untuk mengetahui sejauh mana peralatan dapat bekerja sesuai dengan perencanaan. Langkah pengujian dilakukan melalui 2 tahap, yakni pengujian pada setiap blok dan pengujian pada sistem keseluruhan. Tahap pertama dimaksudkan untuk mengetahui sejauh mana blok-blok rangkaian dapat berjalan, sedangkan tahap kedua dilakukan setelah diperoleh kepastian bahwa tiap blok rangkaian telah berjalan sesuai rencana.

#### **4.1. Pengujian Rangkaian Detektor Arus**

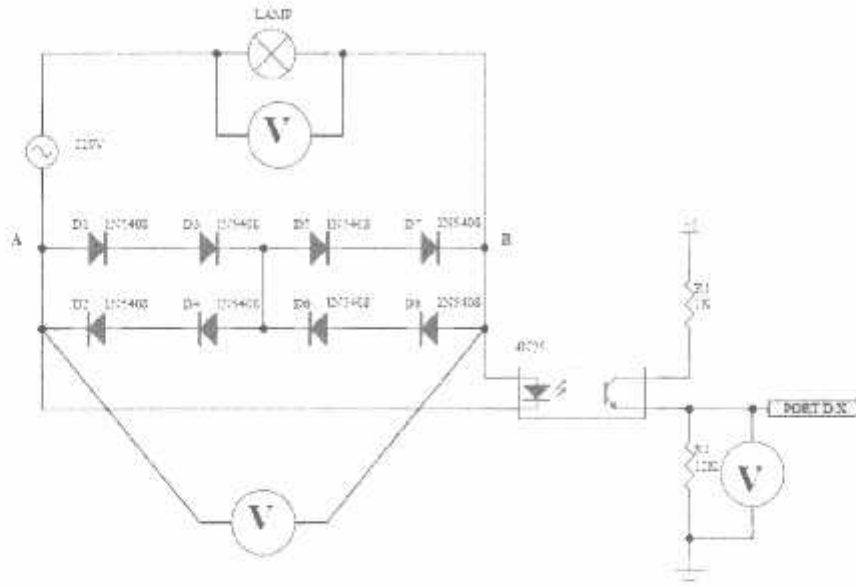
##### **4.1.1. Alat yang digunakan :**

1. Rangkaian Detektor Arus
2. Multimeter Digital SUNWA DT-930T
3. Catu daya 5 Volt
4. Tegangan Beban 200 Volt AC
5. Beban Lampu Pijar 220 Volt, 40 W
6. *Osilloscope*

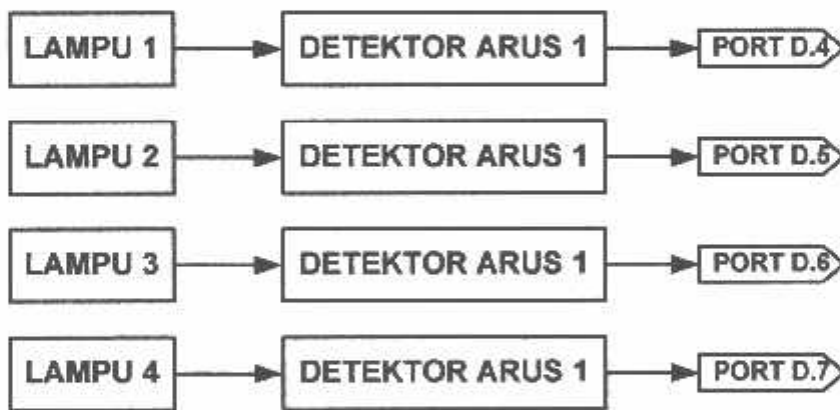
##### **4.1.2. Langkah Pengujian :**

1. Memasukkan tegangan Beban Sebesar 200V AC.
2. Memasukan catu daya sebesar 5V DC
3. Mengukur tegangan pada jala-jala listrik
4. Mengukur tegangan simpul beban pada jala-jala listrik
5. Mengukur tegangan pada titik A B.

6. Mengukur tegangan keluaran sebagai input mikrokontroler yang menuju Port.D



**Gambar 4.1. Rangkaian Pengujian Detektor Arus**



**Gambar 4.2. Blok Diagram Detektor Arus**

#### 4.1.3. Hasil dan Analisa Pengujian Rangkaian Detektor Arus

Pada hasil pengujian didapatkan data- data pengukuran sebagai berikut :

- Tegangan pada jala jala listrik sebesar : 200 AC Volt



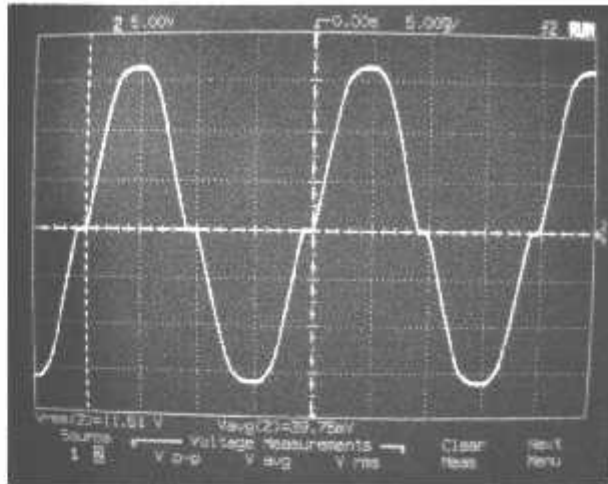
**Gambar 4.3. Pengukuran Tegangan Pada Jala-jala Listrik**

- Tegangan pada simpul beban lampu pijar 40 Watt



**Gambar 4.4. Pengukuran Tegangan Simpul Beban Lampu Pijar Pada Jala-jala Listrik**





**Gambar 4.5. Bentuk Gelombang Pada Simpul Beban**

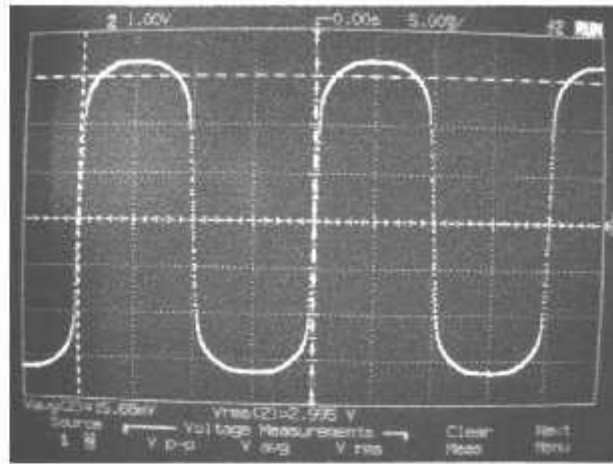
**Tabel 4.1. Tabel Hasil Pengujian Tegangan Beban**

No Beban	Tegangan Beban
1	197.3
2	198
3	197,1
4	197.5

- o Tegangan pada titik A B



**Gambar 4.6. Pengukuran Tegangan Pada Titik A B**



**Gambar 4.7. Bentuk Gelombang Pada Titik A B**

**Tabel 4.2. Tabel Hasil Pengujian Tegangan Pada Titik A B**

No Detektor	Tegangan Simpul A B (V)
1	2.8
2	2.9
3	2.9
4	2.8

Dari hasil perhitungan didapatkan tegangan pada titik AB sebesar 2.8 Volt, dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$V_1 = \text{Hasil\_perhitungan}$$

$$V_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|V_1 - V_2|}{V_1} \times 100\% \dots \dots \dots (4.1)$$

$$1. \%Error = \frac{|2.8 - 2.8|}{2.8} \times 100\% = 0\%$$

$$2. \%Error = \frac{|2.8 - 2.9|}{2.8} \times 100\% = 3.5\%$$

$$3. \%Error = \frac{|2.8 - 2.9|}{2.8} \times 100\% = 3.5\%$$

$$4. \%Error = \frac{|2.8 - 2.8|}{2.8} \times 100\% = 3.5\%$$

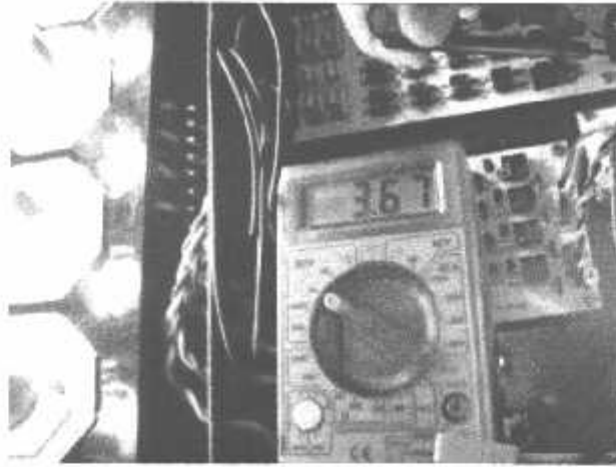
**Tabel 4.3. Tabel perbandingan hasil perhitungan dan hasil pengujian Tegangan Simpul A B**

No Detektor	Tegangan simpul A B		Error (%)
	Hasil Perhitungan (Volt)	Hasil Pengukuran (Volt)	
1	2.8	2.8	0
2	2.8	2.9	3.5
3	2.8	2.9	3.5
4	2.8	2.8	0
Rata - rata		2.85	1.75

- Tegangan keluaran detektor arus rata-rata sebesar 3.66Volt pada saat *optocoupler* saturasi atau terdapat aliran arus pada beban, dan didapatkan tegangan rata-rata sebesar 0.01 Volt pada saat *optocoupler* pada kondisi *cut off*, atau tidak ada aliran arus pada beban.



**Gambar 4.8. Pengukuran Tegangan Pada Output Detektor Arus Pada Saat Tidak Ada Aliran Arus Listrik Ke Beban**



**Gambar 4.9. Pengukuran Tegangan Pada Output Detektor Pada Saat Ada Aliran Arus Listrik Ke Beban Listrik**

**Tabel 4.4. Tabel Hasil Pengujian Tegangan Keluaran Detektor Arus**

No Detektor	<i>Saturasi (V)</i>	<i>Cut Off (V)</i>
1	3.67	0
2	3.67	0
3	3.67	0
4	3.67	0

Dari hasil perhitungan didapatkan tegangan keluaran *optocoupler* sebesar 3.96 Volt, dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$V_1 = \text{Hasil\_perhitungan}$$

$$V_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|V_1 - V_2|}{V_1} \times 100\%$$

$$1.. \%Error = \frac{|3.96 - 3.67|}{3.96} \times 100\% = 7.32\%$$

$$2.. \%Error = \frac{|3.96 - 3.67|}{3.96} \times 100\% = 7.32\%$$

$$3.. \%Error = \frac{|3.96 - 3.67|}{3.96} \times 100\% = 7.32\%$$

$$4.. \%Error = \frac{|3.96 - 3.67|}{3.96} \times 100\% = 7.32\%$$

**Tabel 4.5. Tabel perbandingan hasil perhitungan dan hasil pengujian tegangan keluaran detektor arus**

No Detektor	Tegangan keluaran		Error (%)
	Hasil Perhitungan (Volt)	Hasil Pengukuran (Volt)	
1	3.96	3.67	7.32
2	3.96	3.67	7.32
3	3.96	3.67	7.32
4	3.96	3.67	7.32
<b>Rata - rata</b>		<b>3.667</b>	<b>7.32</b>

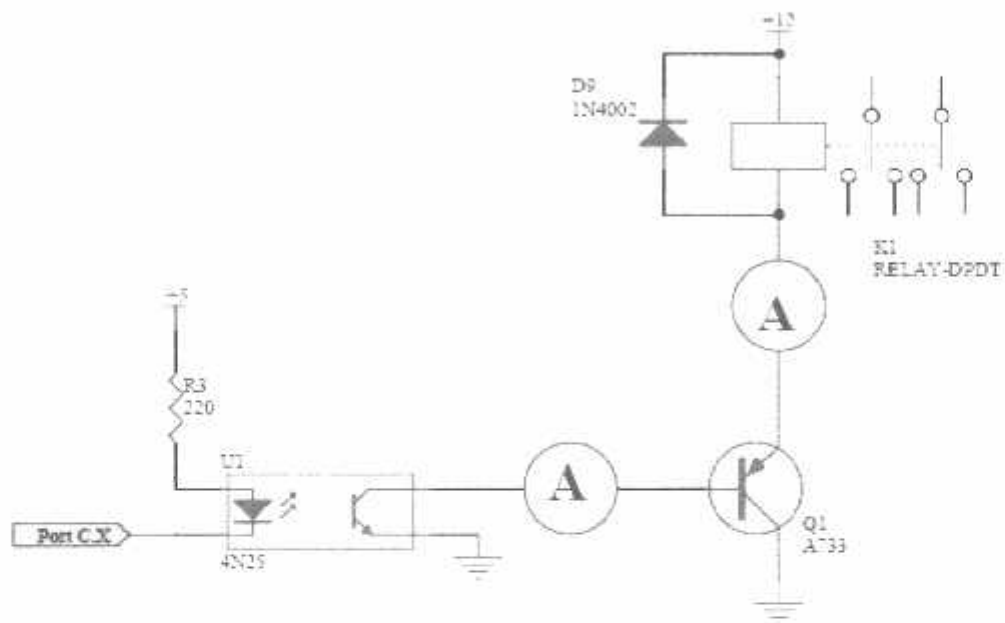
## 4.2. Pengujian Rangkaian Driver Relay

### 4.2.1. Alat dan Bahan

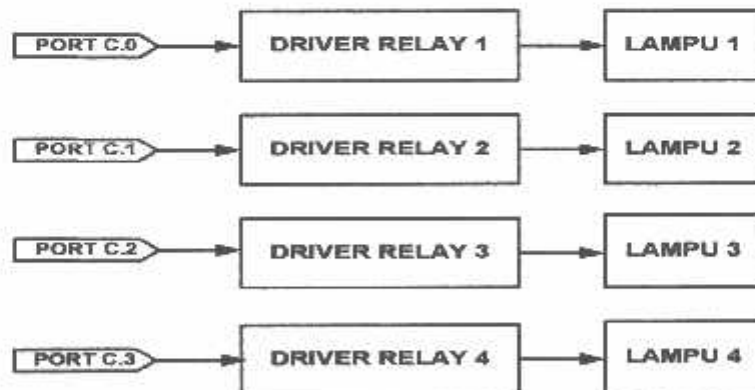
1. Rangkaian Driver Relay
2. Multimeter Digital SUNWA DT-930T
3. Rangkaian Mikrokontroler ATmega16
4. Catu daya 12 Volt
5. Catu daya 5 Volt

### 4.2.2. Langkah Pengujian

1. Menghubungkan alat dan bahan seperti gambar 4.7.
2. Berikan masukan aktif *low* pada *input* rangkaian *driver* melalui port mikrokontroler.
3. Amati hasil yang ditunjukkan amperemeter



Gambar 4.10. Rangkaian Pengujian Driver Relay



Gambar 4.11. Blok Diagram Pengujian Driver Relay

### 4.2.3. Hasil dan Analisa Pengujian Rangkaian Driver Relay

- Hasil dan Analisa Pengujian Arus Ib Trasistor A733

Tabel 4.6. Tabel Hasil Pengujian Arus Ib Trasistor A733

No Driver	Ib (mA)
1	0.13
2	0.13
3	0.13
4	0.13



Gambar 4.12. Hasil Pengukuran Arus Ib Trasistor A733

Dari hasil perhitungan didapatkan arus Ib pada transistor A733 sebesar 0.129 mA, Maka dari data tersebut dapat dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$1. \%Error = \frac{|0.127 - 0.13|}{0.0127} \times 100\% = 2\%$$

$$2. \%Error = \frac{|0.127 - 0.13|}{0.0127} \times 100\% = 2\%$$

$$3. \%Error = \frac{|0.127 - 0.13|}{0.127} \times 100\% = 2\%$$

$$4. \%Error = \frac{|0.127 - 0.13|}{0.127} \times 100\% = 2\%$$

**Tabel 4.7. Tabel perbandingan hasil perhitungan dan hasil pengujian arus Ib pada transistor A733**

No Driver	Arus Ib		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	0.127	0.13	2
2	0.127	0.13	2
3	0.127	0.13	2
4	0.127	0.13	2
<b>Rata - rata</b>		<b>0.13</b>	<b>2</b>

- **Hasil dan Analisa Pengujian Arus Ie Trasistor A733**

**Tabel 4.8. Tabel Hasil Pengujian Arus Ie Trasistor A733**

No Driver	Ib (mA)
1	30
2	30
3	30
4	30





**Gambar 4.13. Hasil Pengukuran Arus Ie Trasistor A733**

Dari hasil perhitungan didapatkan arus I<sub>b</sub> pada transistor A733 sebesar 39 mA, Maka akan dicari persentase *error* hasil pengujian terhadap perhitungan adalah:

$$I_1 = \text{Hasil\_perhitungan}$$

$$I_2 = \text{Hasil\_pengujian}$$

$$\%Error = \frac{|I_1 - I_2|}{I_1} \times 100\%$$

$$1.. \%Error = \frac{|30.7 - 30|}{33} \times 100\% = 2\%$$

$$2.. \%Error = \frac{|30.7 - 30|}{33} \times 100\% = 2\%$$

$$3.. \%Error = \frac{|30.7 - 30|}{33} \times 100\% = 2\%$$

$$4.. \%Error = \frac{|30.7 - 30|}{33} \times 100\% = 2\%$$

**Tabel 4.9. Tabel perbandingan hasil perhitungan dan hasil pengujian arus Ie pada transistor A733**

No Driver	Arus Ie		Error (%)
	Hasil Perhitungan (mA)	Hasil Pengukuran (mA)	
1	30.7	30	2
2	30.7	30	2
3	30.7	30	2
4	30.7	30	2
<b>Rata - rata</b>		<b>30</b>	<b>2</b>

### 4.3. Pengujian Rangkaian Minimum Sistem Mikrokontroler ATmega 16

#### 4.3.1. Alat dan Bahan

1. *Personal Computer* (PC) yang dilengkapi LPT1
2. *Downloader* mikrokontroler ATmega 16
3. *ISP programmer* dengan *connector* LPT1
4. Minimum sistem Mikrokontroler
5. Catu daya 5 volt

#### 4.3.2. Langkah Pengujian

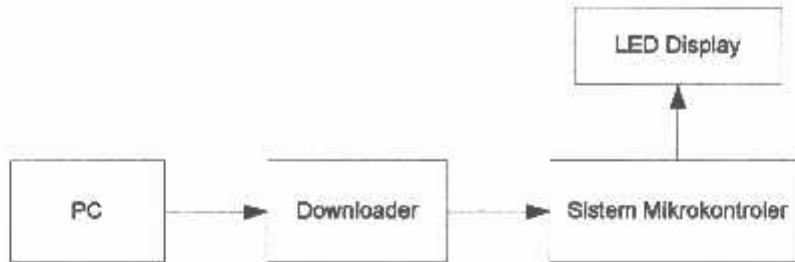
Tahapan Pengujian mikrokontroler ATmega 16 sebagai berikut:

1. Menghubungkan alat dan bahan seperti gambar 4-1.
2. Memberikan catu daya 5 volt.
3. Membuat program yang digunakan untuk menguji mikrokontroler.

Program yang digunakan dalam pengujian mikrokontroler ini

merupakan program yang sederhana yang meletakkan  $F0_H$  dan  $0F_H$  secara bergantian pada *PortC* ATmega 16.

4. Mengamati keluaran pada LED Display.



**Gambar 4.14. Diagram Blok Pengujian Mikrokontroler**

### 4.3.3. Hasil dan Analisa Pengujian

**Tabel 4.10. Hasil Pengujian Sistem Mikrokontroler**

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

Keterangan :

- Kondisi bit *low* ( 0 ) = LED menyala
- Kondisi bit *height* ( 1 ) = LED mati

Dari hasil pengujian pada tabel 4-1 dapat dilihat bahwa *port C* memberikan logika  $0F_H$  dan  $F0_H$  secara bergantian sesuai dengan isi program.

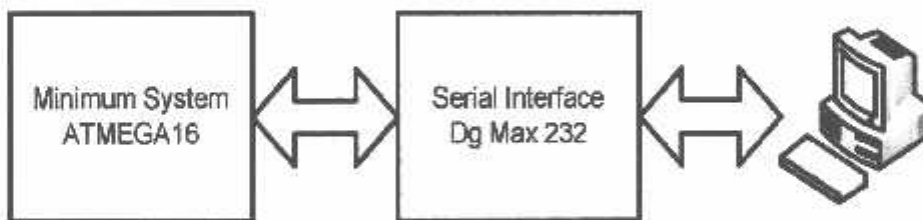
#### 4.4. Pengujian Rangkaian Komunikasi Serial RS232

##### 4.4.1. Alat dan Bahan

1. Rangkaian Driver Serial RS 232
2. Rangkaian Minimum Sistem Mikrokontroler ATmega16
3. PC dengan fasilitas *hyper terminal*
4. kabel serial RS232
5. Catu daya 5 Volt

##### 4.4.2. Langkah Pengujian

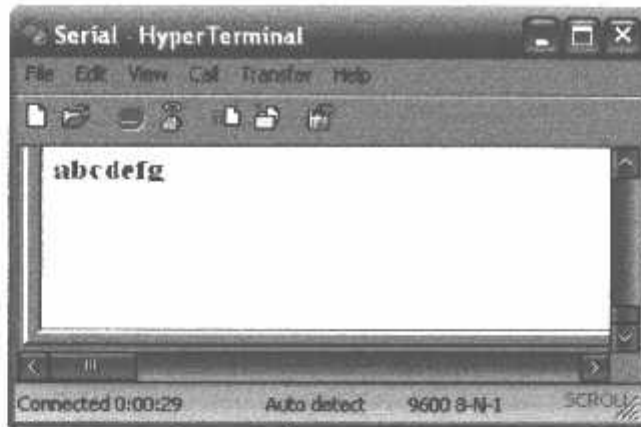
1. Mempersiapkan program mikrokontroler ATmega16 yang berfungsi untuk mengirimkan data serial yang diperintah dari *keyboard* PC
2. Menghubungkan alat dan bahan seperti gambar 4.8.
3. Mengisikan program dan melakukan penekanan pada keyboard
4. mengamati hasil pengujian



Gambar 4.15. Blok Diagram Pengujian Driver Serial

##### 4.4.3. Hasil Pengujian

Hasil pengamatan komunikasi serial RS232 pada *hyper terminal* dengan menggunakan *baudrate* sebesar 19200 dapat dilihat pada gambar 4.9. Sedangkan hasil pengujiannya dapat dilihat pada tabel 4.3.



**Gambar 4.16. Hasil Pengujian Driver Serial Pada *Hyper Terminal***

**Tabel 4.11. Tabel Hasil Pengujian Driver Serial RS 232 Pada *Hyper Terminal***

<b>Input Dari Keyboard</b>	<b>Output HT</b>
a	a
b	b
c	c
d	d
e	e
f	f
g	g

Dari hasil pengamatan di ketahui bahwa data karakter yang dikirim adalah sama dengan huruf yang kita inputkan. Dengan demikian konverter RS232 telah dapat mengirim data atau menerima data dengan baik pada kecepatan data 19200 bps, dan pada percobaan diatas tidak terdapat karakter yang rusak atau *error*.

## **4.5 Pengujian Keypad**

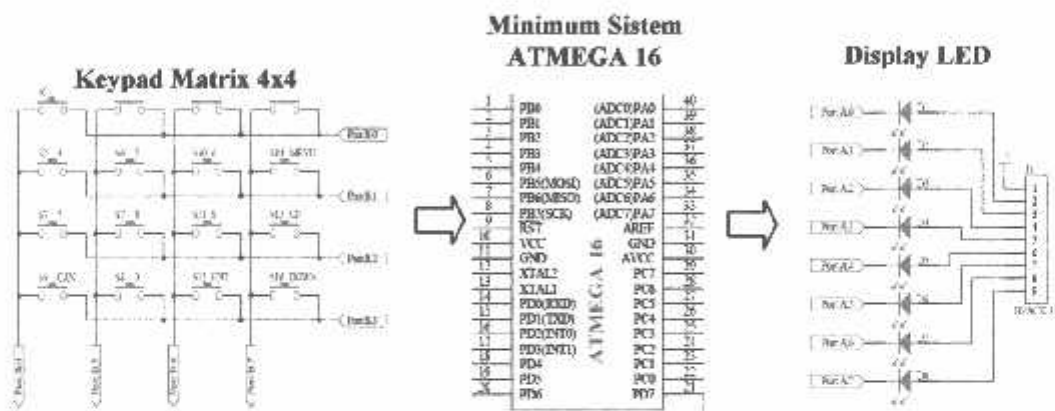
### **4.5.1. Alat dan Bahan**

1. Keypad Matrix 4x4
2. Rangkaian Minimum sistem ATMEGA16
3. Rangkaian LED

4. Catu daya 5 Volt

#### 4.5.2. Langkah Pengujian

1. Mempersiapkan program
2. Menghubungkan keypad dengan port A pada mikrokontroler dan menghubungkan rangkaian led seperti pada gambar 4.8.
3. Mengisikan program dan melakukan penekanan pada keypad
4. mengamati hasil pengujian



Gambar 4.17. Blok Diagram Pengujian Keypad

#### 4.5.3. Hasil Pengujian

Dari hasil pengujian didapat bahwa rangkaian keypad dapat memberikan data ke mikrokontroler ATMEGA16 pada saat ada penekanan tombol, terbukti pada hasil tampilan led, tampilan led mengeluarkan data seperti pada tabel 4.10, dengan demikian maka dapat dikatakan bahwa rangkaian keypad dapat bekerja dengan baik, hasil pengujian dapat dilihat pada tabel 4.4.

**Tabel 4.12. Tabel Hasil Pengujian Keypad Matrix 4x4**

Tombol	Data	Display LED							
		7	6	5	4	3	2	1	0
1	0x01	On	On	On	On	On	On	On	Off
2	0x04	On	On	On	On	On	Off	On	On
3	0x07	On	On	On	On	On	Off	Off	Off
COR	0x0e	On	On	On	On	Off	Off	Off	On
4	0x02	On	On	On	On	On	On	Off	On
5	0x05	On	On	On	On	On	Off	On	Off
6	0x08	On	On	On	On	Off	On	On	On
MEN	0x00	On	On	On	On	On	On	On	On
7	0x03	On	On	On	On	On	On	Off	Off
8	0x06	On	On	On	On	On	Off	Off	On
9	0x09	On	On	On	On	Off	On	On	Off
←	0x0f	On	On	On	On	Off	Off	Off	Off
CAN	0x0a	On	On	On	On	Off	On	Off	On
0	0x0b	On	On	On	On	Off	On	Off	Off
ENT	0x0c	On	On	On	On	Off	Off	On	On
↓	0x0d	On	On	On	On	Off	Off	On	Off

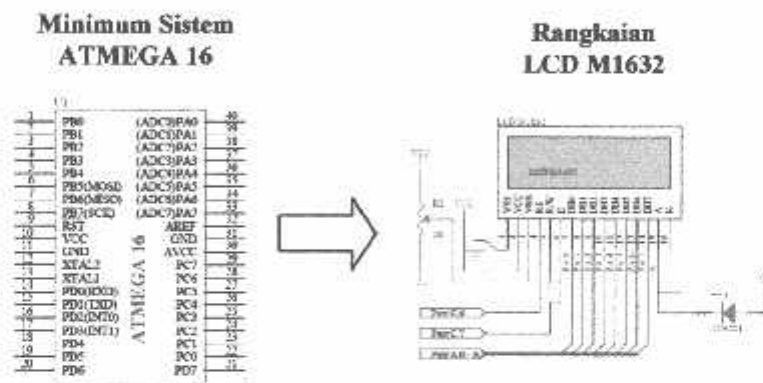
## 4.6. Pengujian Rangkaian LCD

### 4.6.1. Alat dan Bahan

1. Rangkaian LCD
2. Rangkaian Mikrokontroler ATMEGA16
3. Catu daya 5 Volt

#### 4.6.2. Langkah Pengujian

1. Mempersiapkan *software* mikrokontroler yang berfungsi untuk inialisasi LCD dan menampilkan suatu kata-kata pada baris pertama bertuliskan “Priyo Adi Jatmiko”, baris kedua bertuliskan “0412223”.
2. Mempersiapkan rangkaian pengujian seperti Rangkaian pengujian LCD pada gambar 4.9.
3. Mengamati hasil pengujian.



Gambar 4.18. Blok Diagram Pengujian Rangkaian LCD M1632

#### 4.6.3. Hasil Pengujian

Setelah melakukan pengujian LCD diatas didapatkan tulisan “Priyo Adi J” pada baris pertama dan “0412223” pada baris kedua . Dengan demikian setelah mengetahui hasil pengujian rangkaian LCD dapat disimpulkan bahwa rangkaian LCD tersebut telah berfungsi sesuai yang direncanakan.



Gambar 4.19. Hasil Pengujian Rangkaian LCD M1632



## **4.7. Pengujian Keseluruhan Sistem**

### **4.7.1. Alat dan Bahan**

1. Rangkaian minimum sistem ATMEGA16
2. Rangkaian *Serial Interface* Max 232
3. Kabel Data Handphone Siemens C55
4. Handphone Siemens C55
5. Rangkaian LCD
6. Rangkaian detektor Arus
7. Rangkaian driver Arus
8. Keypad
9. Beban listrik lampu pijar 4 buah @220 Volt / 40 Watt
10. Catu daya 12 Volt Dan 5 Volt
11. PC dengan fasilitas *hyper terminal*

### **4.7.2. Langkah Pengujian**

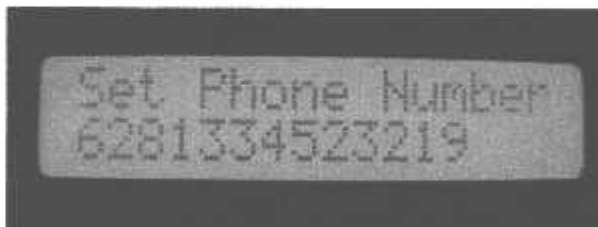
1. Menghubungkan kabel data Handphone Siemens C55 ke rangkaian serial interface Max-232.
2. Menghubungkan rangkaian serial interface ke pin Rx dan Tx pada minimum sistem.
3. Menghubungkan keypad matrix 4x4 ke port B.
4. Menghubungkan data LCD ke port A, RS ke port C.6, E ke port C.7.
5. Menghubungkan rangkaian detektor arus ke port D.4 – D7.
6. Menghubungkan rangkaian driver relay ke port C.4 – C.7.
7. Mengamati hasil pengujian.

**Tabel 4.14. Daftar Nomor Handphone User Yang Akan Digunakan Untuk Pengujian Alat**

No. User	Nomor Ponsel	Operator Seluler	SMS Center
User 1	6285648506944	Indosat	62855000000
User 2	6281334523219	Telkomsel	6281100000
User 3	6285855930231	Satelindo	62816124
User 4	6287859768061	Exelcom	62818445009



**Gambar 4.21. Tampilan LCD Set Nomor HP Untuk User 1**



**Gambar 4.22. Tampilan LCD Set Nomor HP Untuk User 2**



**Gambar 4.23. Tampilan LCD Set Nomor HP Untuk User 3**



**Gambar 4.24. Tampilan LCD Set Nomor HP Untuk User 4**

Nomor – nomor tersebut disimpan dalam dua format, yang pertama disimpan sesuai nomor yang sesungguhnya, dan yang kedua disimpan dalam bentuk PDU dan disimpan pada EEPROM mikrokontroler, alamat penyimpanan nomor - nomor selular tersebut dipetakan pada tabel berikut:

**Tabel 4.15. Pemetaan Memori EEPROM Yang digunakan Untuk Penyimpanan Nomor Selular**

No.	User	Alamat Nomor		Alamat Nomor Dalam Bentuk PDU	
		Dec	Hex	Dec	Hex
1	User 1	0 - 13	\$000 - \$00D	210 - 228	\$0D2 - \$0E4
2	User 2	15 - 28	\$00F - \$01C	230 - 248	\$0E6 - \$0F8
3	User 3	30 - 43	\$01E - \$02B	250 - 268	\$0FA - \$10C
4	User 4	45 - 58	\$02D - \$03A	270 - 288	\$10E - \$120
5	User 5	60 - 73	\$03C - \$049	290 - 308	\$122 - \$134
6	User 6	75 - 88	\$04B - \$058	310 - 328	\$136 - \$148
7	User 7	90 - 103	\$05A - \$067	330 - 348	\$14A - \$15C
8	User 8	105 - 118	\$069 - \$076	350 - 368	\$15E - \$170
9	User 9	120 - 133	\$078 - \$085	370 - 388	\$172 - \$184
10	User 10	135 - 148	\$087 - \$094	390 - 408	\$186 - \$198
11	User 11	150 - 163	\$096 - \$0A3	410 - 428	\$19A - \$1AC
12	User 12	165 - 178	\$0A5 - \$0B2	430 - 448	\$1AE - \$1C0
13	User 13	180 - 193	\$0B4 - \$0C1	450 - 468	\$1C2 - \$1D4
14	User 14	195 - 208	\$0C3 - \$0D0	470 - 488	\$1D6 - \$1E8

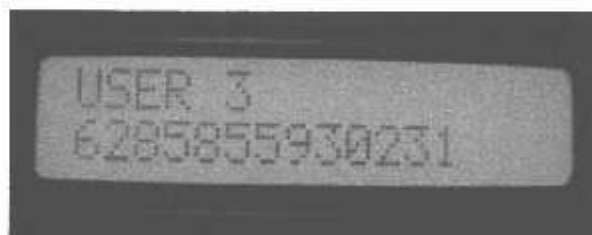
Hasil nomor *Handphone* yang telah tersimpan pada EEPROM dapat dilihat pada menu "*VIEW*" dengan menekan tombol 2 (Dua) pada keypad, berikut adalah tampilan LCD nomor yang telah tersimpan.



**Gambar 4.25. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM  
Alamat \$000 - \$00D**



**Gambar 4.26. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM  
Alamat \$00F - \$01C**



**Gambar 4.27. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM  
Alamat \$01E - \$02B**



**Gambar 4.28. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM  
Alamat \$02D - \$03A**

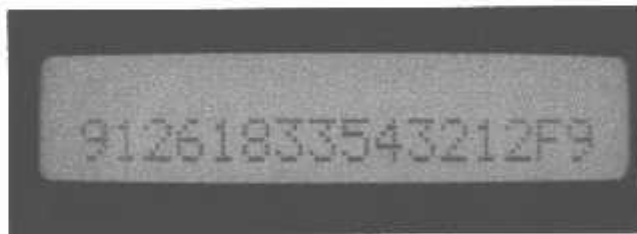
Setelah diregristasikan ke sistem nomor akan dikonfersikan ke format Protocol Data Unit (PDU), hasil konfigurasi nomor ponsel ke bentuk PDU ditunjukkan pada table berikut:

**Tabel 4.16. Hasil Konversi Nomor Ponsel Ke Format PDU**

No. User	Nomor Ponsel	Hasil Konversi
User 1	6285648506944	0D91265846586049F4
User 2	6281334523219	0D91261833543212F9
User 3	6285855930231	0D91265858950332F1
User 4	6287859768061	0D91267858798660F1



**Gambar 4.29. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU, Alamat \$0D2 - \$0E4**



**Gambar 4.30. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU, Alamat \$0E6 - \$0F8**



**Gambar 4.31. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU, Alamat \$0FA - \$10C**



**Gambar 4.32. Tampilan LCD Nomor HP Yang Tersimpan Pada EEPROM Dalam Format PDU, Alamat \$10E - \$120**

#### **4.7.3.2. Proses Pengendalian Sistem.**

Pilihan "Ready" atau pilihan tiga pada menu awal harus di tekan agar sistem bisa dikendalikan melalui SMS.

##### **1. Hasil Pengujian Perintah ONN.**

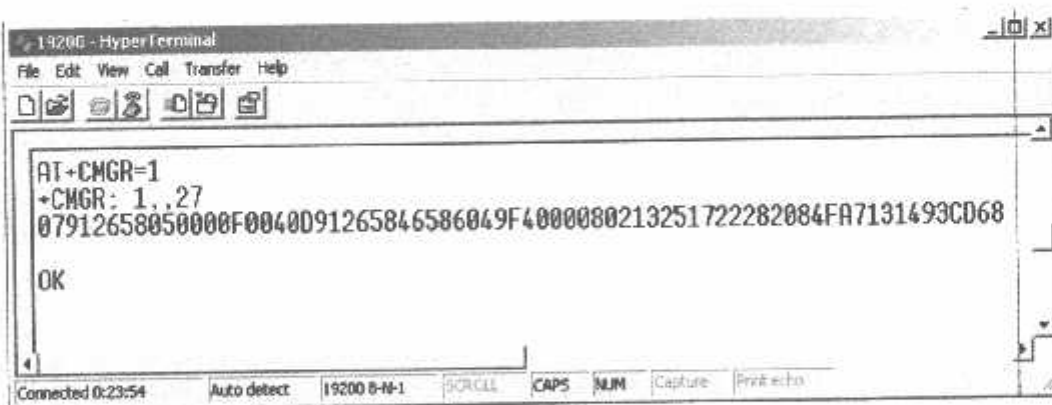
Berikut adalah data - data hasil pengujian perintah SMS "ONN" menggunakan nomor ponsel *User 1, User 2, User 3, User 4.*

**Tabel 4.17. Data PDU Lengkap Yang Diterima Mikrokontroler Saat Perintah "ONN"**

No User	Isi SMS	PDU Yang Diterima Mikrokontroler
User 1	ONN 0001	07912658050000F0040D91265846586049F4000080213251722282084FA7130483C162
User 1	ONN 0002	07912658050000F0040D91265846586049F4000080213251302282084FA7130483C164
User 1	ONN 0003	07912658050000F0040D91265846586049F4000080213251350082084FA7130483C166
User 1	ONN 0004	07912658050000F0040D91265846586049F4000080213251400082084FA7130483C168
User 2	ONN 0012	06912618010000040D91261833543212F9000080213261030082084FA7130483C564
User 2	ONN 0013	06912618010000040D91261833543212F9000080213261231082084FA7130483C566
User 2	ONN 0014	06912618010000040D91261833543212F9000080213271256182084FA7130483C568
User 2	ONN 0023	06912618010000040D91261833543212F9000080213271156482084FA7130483C966
User 3	ONN 0024	059126181642040D91265858950332F1000080213271063182084FA7130483C968
User 3	ONN 0034	059126181642040D91265858950332F1000080213281101182084FA7130483CD68
User 3	ONN 0123	059126181642040D91265858950332F1000080213281511282084FA713048BC966
User 3	ONN 0124	059126181642040D91265858950332F1000080213281612582084FA713048BC968
User 4	ONN 0134	07912618485400F9040D91267858798660F1000080213281810282084FA713048BCD68
User 4	ONN 0234	07912618485400F9040D91267858798660F1000080213291524282084FA7130493CD68
User 4	ONN 1234	07912618485400F9040D91267858798660F1000080213291231582084FA7131493CD68

**Tabel 4.18. Kondisi Beban Setelah Perintah SMS “ONN” Diproses Oleh Mikrokontroler**

No User	Isi SMS	Kondisi Beban			
		Beban 1	Beban 2	Beban 3	Beban 4
User 1	ONN 0001	1	0	0	0
User 1	ONN 0002	0	1	0	0
User 1	ONN 0003	0	0	1	0
User 1	ONN 0004	0	0	0	1
User 2	ONN 0012	1	1	0	0
User 2	ONN 0013	1	0	1	0
User 2	ONN 0014	1	0	0	1
User 2	ONN 0023	0	1	1	0
User 3	ONN 0024	0	1	0	1
User 3	ONN 0034	0	0	1	1
User 3	ONN 0123	1	1	1	0
User 3	ONN 0124	1	1	0	1
User 4	ONN 0134	1	0	1	1
User 4	ONN 0234	0	1	1	1
User 4	ONN 1234	1	1	1	1



**Gambar 4.33. Data Yang PDU Dikirimkan HP Server ke Hyperterminal Saat Isi SMS “ONN 1234”**



## 2. Hasil Pengujian Perintah OFF.

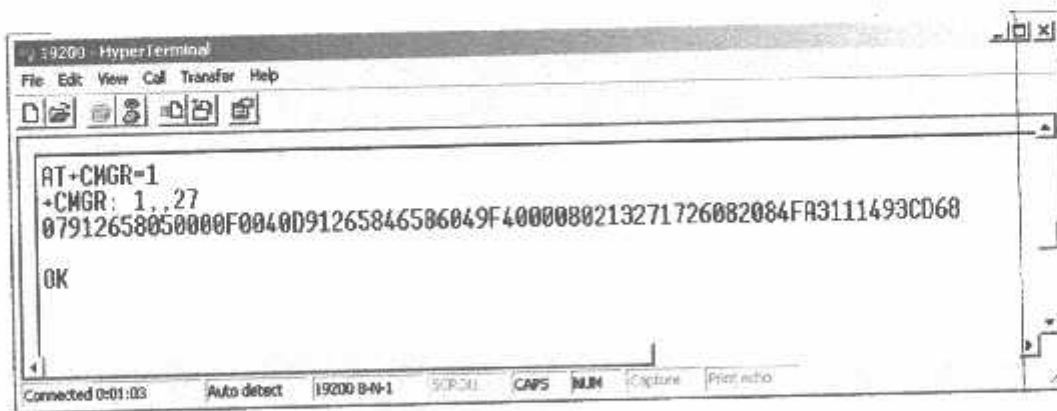
Berikut adalah data - data hasil pengujian perintah SMS "OFF" menggunakan nomor ponsel *User 1, User 2, User 3, User 4.*

**Tabel 4.19. Data PDU Lengkap Yang Diterima Mikrokontroler Saat Perintah "OFF"**

No User	ISI SMS	PDU Yang Diterima Mikrokontroler
User 1	OFF 0001	07912658050000F0040D91265846586049F4000080214260011582084FA3110483C162
User 1	OFF 0002	07912658050000F0040D91265846586049F4000080214260021482084FA3110483C164
User 1	OFF 0003	07912658050000F0040D91265846586049F4000080214260034182084FA3110483C166
User 1	OFF 0004	07912658050000F0040D91265846586049F4000080214260210182084FA3110483C168
User 2	OFF 0012	06912618010000040D91261833543212F9000080214270102582084FA3110483C564
User 2	OFF 0013	06912618010000040D91261833543212F9000080214270521082084FA3110483C566
User 2	OFF 0014	06912618010000040D91261833543212F9000080214270530182084FA3110483C568
User 2	OFF 0023	06912618010000040D91261833543212F9000080214270732582084FA3110483C966
User 3	OFF 0024	059126181642040D91265858950332F1000080214270762182084FA3110483C968
User 3	OFF 0034	059126181642040D91265858950332F1000080214280102582084FA3110483CD68
User 3	OFF 0123	059126181642040D91265858950332F1000080214280210182084FA311048BC966
User 3	OFF 0124	059126181642040D91265858950332F1000080214280301582084FA311048HC968
User 4	OFF 0134	07912618485400F9040D91267858798660F1000080214280455182084FA311048BCD68
User 4	OFF 0234	07912618485400F9040D91267858798660F1000080214280654182084FA3110493CD68
User 4	OFF 1234	07912618485400F9040D91267858798660F1000080214280951282084FA3111493CD68

**Tabel 4.20. Kondisi Beban Setelah Perintah SMS "OFF" Diproses Oleh Mikrokontroler**

No User	ISI SMS	KONDISI BEBAN			
		Beban 1	Beban 2	Beban 3	Beban 4
User 1	OFF 0001	0	1	1	1
User 1	OFF 0002	1	0	1	1
User 1	OFF 0003	1	1	0	1
User 1	OFF 0004	1	1	1	0
User 2	OFF 0012	0	0	1	1
User 2	OFF 0013	0	1	0	1
User 2	OFF 0014	0	1	1	0
User 2	OFF 0023	1	0	0	1
User 3	OFF 0024	1	0	1	0
User 3	OFF 0034	1	1	0	0
User 3	OFF 0123	0	0	0	1
User 3	OFF 0124	0	0	1	0
User 4	OFF 0134	0	1	0	0
User 4	OFF 0234	1	0	0	0
User 4	OFF 1234	0	0	0	0



**Gambar 4.34. Data Yang PDU Dikirimkan HP Server ke Hyperterminal Saat Isi SMS "OFF 1234"**

### 3. Hasil Pengujian Perintah GET.

Berikut adalah data-data hasil pengujian perintah SMS "GET 1234" menggunakan nomor ponsel *User 1*, *User 2*, *User 3*, *User 4*.

**Tabel 4.21. PDU Lengkap Yang Dikirimkan Mikrokontroler Ke HP Server**

No User	Kondisi Beban				PDU Yang Dikirimkan Mikrokontroler
	Beban 1	Beban 2	Beban 3	Beban 4	
User 1	0	0	0	0	07912658050000F001000D91265846586049F400000430180C06
User 1	1	0	0	0	07912658050000F001000D91265846586049F400000431180C06
User 1	0	1	0	0	07912658050000F001000D91265846586049F4000004B0180C06
User 1	0	0	1	0	07912658050000F001000D91265846586049F400000430580C06
User 2	0	0	0	1	07912658050000F001000D91261833543212F900000430182C06
User 2	1	1	0	0	07912658050000F001000D91261833543212F9000004B1180C06
User 2	1	0	1	0	07912658050000F001000D91261833543212F900000431580C06
User 2	1	0	0	1	07912658050000F001000D91261833543212F900000431182C06
User 3	0	1	1	0	07912658050000F001000D91265858950332F1000004B0580C06
User 3	0	1	0	1	07912658050000F001000D91265858950332F1000004B0182C06
User 3	0	0	1	1	07912658050000F001000D91265858950332F100000430582C06
User 3	1	1	1	0	07912658050000F001000D91265858950332F1000004B1580C06
User 4	1	1	0	1	07912658050000F001000D91267858798660F1000004B1182C06
User 4	1	0	1	1	07912658050000F001000D91267858798660F100000431582C06
User 4	0	1	1	1	07912658050000F001000D91267858798660F1000004B0582C06
User 4	1	1	1	1	07912658050000F001000D91267858798660F1000004B1582C06

**Tabel 4.22. Laporan Status Beban Yang Dikirimkan HP Server Ke HP User**

No User	Kondisi Beban				Perintah SMS	SMS Yang Diterima HP User
	Beban 1	Beban 2	Beban 3	Beban 4		
User 1	0	0	0	0	GET 1234	0000
User 1	1	0	0	0	GET 1234	1000
User 1	0	1	0	0	GET 1234	0100
User 1	0	0	1	0	GET 1234	0010
User 1	0	0	0	1	GET 1234	0001
User 2	1	1	0	0	GET 1234	1100
User 2	1	0	1	0	GET 1234	1010
User 2	1	0	0	1	GET 1234	1001
User 2	0	1	1	0	GET 1234	0110
User 3	0	1	0	1	GET 1234	0101
User 3	0	0	1	1	GET 1234	0011
User 3	1	1	1	0	GET 1234	1110
User 4	1	1	0	1	GET 1234	1101
User 4	1	0	1	1	GET 1234	1011
User 4	0	1	1	1	GET 1234	0111
User 4	1	1	1	1	GET 1234	1111



**Gambar 4.35. Kondisi Beban Lampu Pijar Setelah Data SMS "ONN 1234" Diterima Sistem**

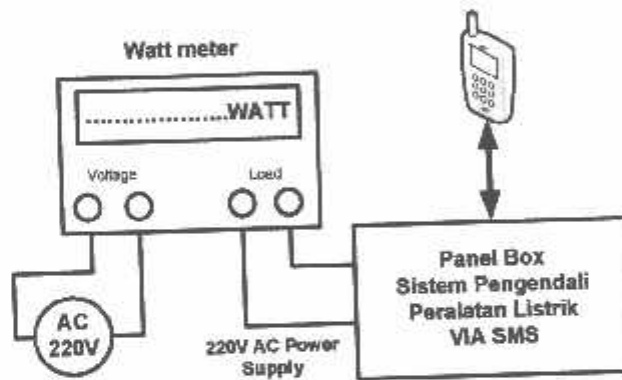
## 4.8. Konsumsi Daya Sistem Pengendali

### 4.8.1. Alat Dan Bahan

1. *Wattmeter*
2. Seluruh rangkaian sistem pengendali

### 4.8.2. Langkah Pengujian

1. Menghubungkan beban listrik dengan *panel box*
2. Memberikan catu daya pada panel box.
3. Mengukur daya dengan menggunakan *wattmeter*.
4. Mempersiapkan rangkaian pengujian seperti Rangkaian pengujian LCD pada gambar 4.9.



Gambar 4.36. Diagram Blok Pengujian Konsumsi Daya Pada *Panel Box* Sistem Pengendali

### 4.8.3. Hasil Pengujian

Dari hasil pengujian konsumsi daya pada panel box didapat hasil pada tabel dibawah ini

Tabel 4.23. Konsumsi Daya Sistem Pengendali

Tegangan	Konsumsi Daya Sistem
220 volt	5 Watt



**Gambar 4.38. Alat Sistem Pengendali On/Off Peralatan Listrik Via SMS  
Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA -16**



## BAB V

### PENUTUP

Berdasarkan perancangan, pembuatan, pengujian dan analisis alat Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA -16.

#### 5.1. Kesimpulan

1. Dari hasil pengujian rangkaian detektor Arus, tegangan AC pada detektor arus terjadi *error* terhadap perhitungan sebesar 1.75 %, ketika ada arus yang mengalir pada beban listrik maka detektor arus akan memberikan tegangan masukan *high* pada mikrokontroler sebesar 3.667 Volt dan tegangan masukan *low* sebesar 0 Volt saat tidak ada Aliran Arus
2. Detektor arus hanya mampu mengalirkan arus sebesar 3 ampere.
3. Dari hasil pengujian rangkaian *Driver Relay*, terjadi *error* terhadap perhitungan sebesar 2 %.
4. Nomor ponsel *user* harus diregistrasikan ke sistem, jika tidak maka sms yang dikirimkan oleh *user* tidak akan diproses dan data sms akan dihapus.
5. Jika format sms yang dikirimkan oleh *user* salah, maka data sms tidak akan diproses dan data sms akan dihapus.
6. Satu nomor ponsel membutuhkan memori sebesar 35 byte, maka nomor yang mampu ditampung oleh EEPROM internal 512 Byte adalah sebanyak 14 buah.



7. Konsumsi daya sistem pengendali sebesar 5 Watt, sedangkan besarnya daya maksimum pada beban adalah 440 Watt.

## 5.2 Saran

Pada alat hasil perancangan ini masih mempunyai kekurangan-kekurangan, untuk itu ada beberapa hal yang dapat dilakukan untuk melakukan pengembangan

1. Agar dapat digunakan pada beban yang lebih besar, disarankan untuk mengganti relay dan dioda pada detektor arus yang mampu mengalirkan arus listrik lebih besar.
2. Agar sistem tetap bisa berjalan dengan baik tanpa memperhatikan isi pulsa didalam simcard. nomor ponsel yang terdapat pada sistem atau nomor ponsel *server* hendaknya jenis pascabayar.

## DAFTAR PUSTAKA

- [1]. Siemens AG, 2001, "*AT Command Set for Siemens Mobile Phones and Modems*," Munich, [www.siemens.com](http://www.siemens.com)
  - [2]. Vasillis, Serasidis, 2004, "*SMS Remote Controller With Ericson GSM : T10s, T18, T28*", [www.serasidis.gr](http://www.serasidis.gr)
  - [3]. Wavcom, 2000, "*An Introduction to The SMS in PDU Mode – GSM Recommendation Phase 2*", [www.wavcom.com](http://www.wavcom.com)
  - [4]. Wardana Lingga, "*Belajar Sendiri Mikrokontroler AVR Seri ATMegu8535 Simulasi, Hardware, dan Aplikasi*", Andi, Yogyakarta, 2006.
  - [5]. Malvino, Hanapi, "*Prinsip Prinsip Elektronik*", Jakarta, 1986
  - [6]. Teddy Marcus Zakaria Josef Widiadhi, "*Aplikasi SMS untuk Berbagai Keperluan*", Informatika, Bandung, 2006
  - [7]. Widodo, Thomas. 2002. "*Elektronika Dasar*", Salemba Teknika. Jakarta
  - [8]. [www.meselectronic.com](http://www.meselectronic.com)
  - [9]. [www.atmel.com](http://www.atmel.com)
-

# LAMPIRAN

1. **Gambar Skematik Rangkaian Keseluruhan**
  2. **Listing Program**
  3. **Datasheet**
    - a. **Datasheet Dioda 1N5403**
    - b. **Datasheet Optocoupler 4N25**
    - c. **Datasheet IC Max 232**
    - d. **Datasheet LCD M1632**
    - e. **Datasheet Transistor A733**
    - f. **Datasheet ATMEGA-16**
    - g. **AT – Command Set For Siemens**
-



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

### FORMULIR PERBAIKAN SKRIPSI

Nama : Priyo Adi Jatmiko  
NIM : 04.12.223  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Elektronika  
Judul : Perancangan Dan Pembuatan Sistem  
Pengendali On/Off Peralatan Listrik Via SMS  
Dilengkapi No HP Berbasis ATMega 16  
Hari / Tanggal Ujian Skripsi : Selasa / 17 Maret 2009

No	Tanggal	Uraian	Paraf
I	17/03/2009	Variabel rangkaian detektor dan pembagi tegangan.	

Disetujui,  
Penguji II

(Ir. M. Ibrahim Ashari, MT)  
NIP. Y. 1030100358

Mengetahui,

Dosen Pembimbing I

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

Dosen Pembimbing II

(Sotvohadi, ST, MSc)  
NIP. Y. 1039700309



### FORMULIR BIMBINGAN SKRIPSI

Nama : Priyo Adi Jatmiko  
Nim : 04.12.223  
Masa Bimbingan : 15 Desember 2008 s/d 15 Juni 2008  
Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA16

No	Tanggal	Uraian	Paraf Pembimbing
1	01 Juli 2008	Konsultasi judul skripsi.	
2	19 Juli 2008	Konsultasi Bab IV, realisasikan detektor arus.	
3	10 Agustus 2008	Pengujian driver relay, amati lebih cermat agar eror lebih kecil.	
4	21 Agustus 2008	Hati – hati interface antara hp dengan mikrokontroler.	
5	29 Agustus 2008	Dari hasil pengujian Bab IV simpulkan, dan tulis di Bab V.	
6	08 September 2008	Konsultasi Bab V, eror masih besar, cari sebab-sebabnya, sispkan makalah seminar hasil.	
7	10 september 2008	Pada makalah, hati-hati dengan batasan makalah, dan tujuan penelitian.	
8	22 januari 2009	Bab III, perencanaan & hitungan, detector arus, kapasitas beban.	
9	29 Januari 2009	Bab II, Hati-hati dengan teori perihal dioda & serial interfacenya.	
10	02 Februari 2009	Bab I, lengkapi dengan sistematika penulisan, siapkan ujian komprehensif.	

Malang,

Dosen Pembimbing I

**Ir. Sidik Noertjahjono, MT**  
NIP. Y.1028700163

Form S-4b



### FORMULIR BIMBINGAN SKRIPSI

Nama : Priyo Adi Jatmiko  
Nim : 04.12.223  
Masa Bimbingan : 15 Desember 2008 s/d 15 Juni 2008  
Judul Skripsi : Perancangan Dan Pembuatan Sistem Pengendali On/Off Peralatan Listrik Via SMS Dilengkapi Fasilitas Konfigurasi Nomor HP Berbasis ATMEGA16

No	Tanggal	Uraian	Paraf Pembimbing
1	24 Juli 2008	Konsultasi Hardware & Bab I	
2	21 November 2008	Bab I, Bab II, Bab III	
3	08 Januari 2009	Bab IV, Bab V	
4	02 Februari 2009	Makalah Seminar Hasil	
5			
6			
7			
8			
9			
10			

Malang,

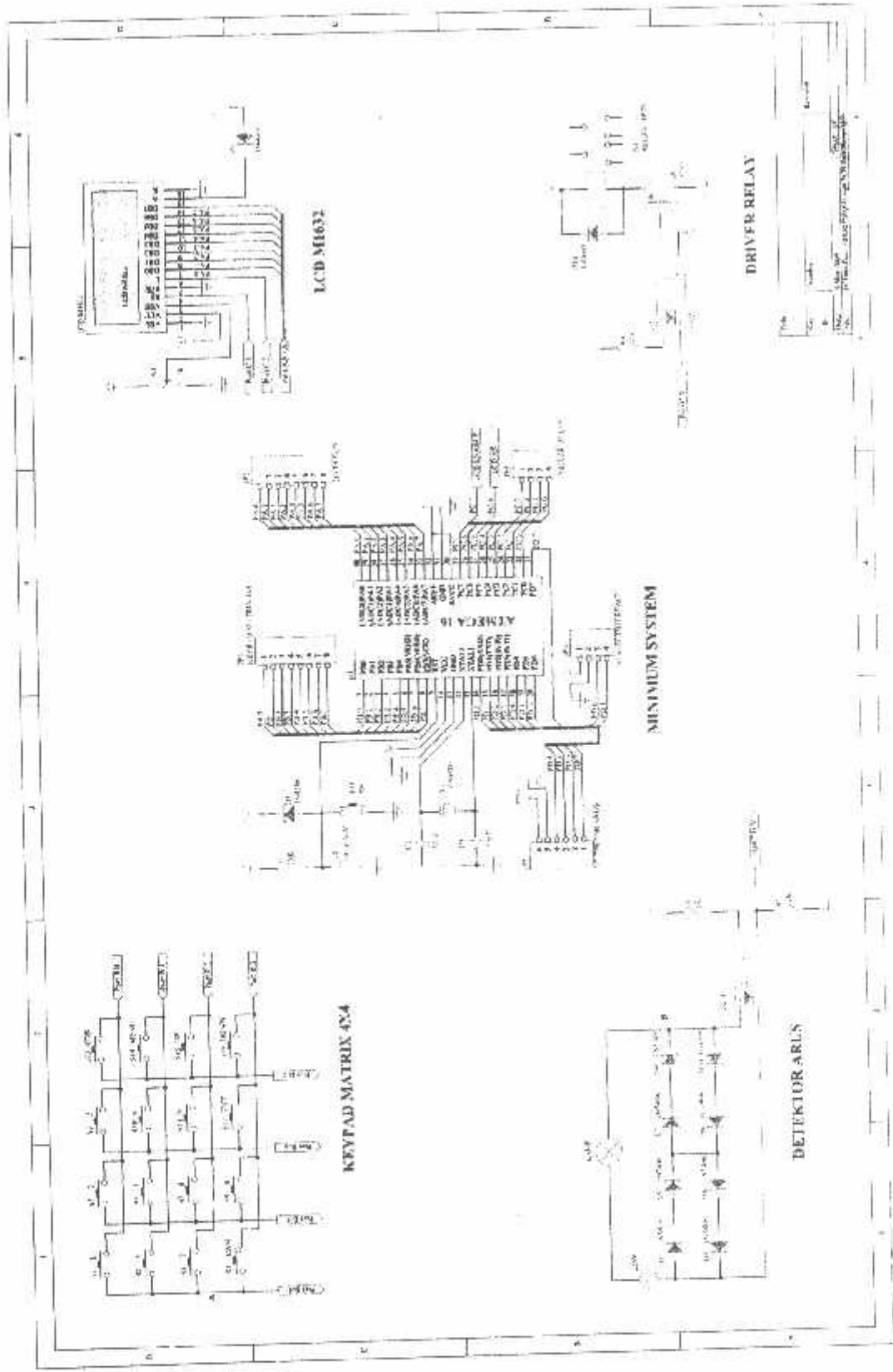
Dosen Pembimbing II

Sotvohadi, ST, MSc  
NIP.Y.1039700309

Form S-4b

# SKEMATIK RANGKAIAN

---



KEYPAD MATRIX 4x4

LCD M1632

MINIMUM SYSTEM

DRIVER RELAY

DETEKTOR ARUS

No.	Revisi	Revisi
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

Disain Oleh:   
 Nama:   
 No.   
 Kelas:   
 Jurusan:   
 Tanggal:   
 Dosen:   
 No.   
 Kelas:   
 Jurusan:   
 Tanggal:   
 Dosen:



# LISTING PROGRAM

---

```

$regfile = "m16def.dat"
$crystal = 11059200
$eeprom
$hwstack = 32
$swstack = 10
$framesize = 40
$baud = 19200

' Definisi MC
' Nilai Crystal
' Inisialisasi
' default use 3
' default use 1

Config PortA = Output

Config PinB.0 = Output
Config PinB.1 = Output
Config PinB.2 = Output
Config PinB.3 = Output
Config PinB.4 = Input
Config PinB.5 = Input
Config PinB.6 = Input
Config PinB.7 = Input

Config PortC = Output

Config PinD.4 = Input
Config PinD.5 = Input
Config PinD.6 = Input
Config PinD.7 = Input

Declare Sub Keypad

Dim W As Byte , D As Byte , Lokasi As Byte , N As Byte , E As Word , E1 As
Dim Nomer As String * 1 , Save_nomer As String * 18 , Jumlah_number As Byte
Dim Jumlah As String * 1
Dim S(14) As String * 14
Dim T(14) As String * 14 , Idx_t As Byte
Dim Simpan_number As String * 18

Dim Count As Byte , User As Byte , Alamat As Byte
Dim Aa As Byte , Bb As Byte , Cc As Byte , Dd As Byte , Ee As Byte , Ff As
Dim Gg As Byte , Hh As Byte , Ii As Byte

Dim Adams As String * 1
Dim Ss As String * 1 , Nn As String * 1 , I As String * 1 , Si As String *
Dim Lh As String * 1 , Ss1 As String * 1

Dim Nomor As String * 20 , Isi As String * 8
Dim Isi1 As String * 2 , Sisa As String * 4
Dim Lebar As String * 14 , Smac As String * 4

Dim Sta1 As String * 1 , Sta2 As String * 1 , Sta3 As String * 1 , Sta4 As

Dim Isikirim As String * 10
Dim Siapdikirim As String * 25

Config Lcdpin = Pin , Db4 = PortA.4 , Db5 = PortA.5 , Db6 = PortA.6 , Db7 =
Config Lcd = 16 * 2

'Config Lcdpin = Pin , Db4 = PortC.4 , Db5 = PortC.5 , Db6 = PortC.6 , Db7
'Config Lcd = 16 * 2

E = 0
E1 = 210
D = 0
Htg = 0

PortB = 255

```

```
Portc = &HFF

Cursor Off Noblink
Display On

Cls
Waitms 300
Lcd " Priyo Adi U"
Lowerline
Lcd " 0412223"
Wait 2

Cls
Waitms 300
Upperline
Lcd "Menu : "
Wait 1

Do
  Call Keypad
  Menu_awal:
    Upperline
    Lcd ".Set 2.View "
    Lowerline
    Lcd " 3.Ready "

    If D = 1 Then
      D = 0
      Goto Pilihan_1
    End If

    If D = 2 Then
      D = 0
      Goto Pilihan_2
    End If

    If D = 3 Then
      D = 0
      Goto Pilihan_3
    End If

Loop

Pilihan_1: 'Set

Cls
If B > 200 Then
  Cls
  Waitms 200
  Upperline
  Lcd "Memory Full"
  Wait 2
  Goto Menu_awal
End If

Cls
Upperline
Lcd "Set Phone Number"
Lokasi = 0
N = 0
Nomer = ""
Save_nomer = ""
Do
```

```
Call Keypad
If D = 1 Then
  D = 0
  Nomer = "1"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "1"
End If
If D = 2 Then
  D = 0
  Nomer = "2"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "2"
End If
If D = 3 Then
  D = 0
  Nomer = "3"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "3"
End If
If D = 5 Then
  D = 0
  Nomer = "4"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "4"
End If
If D = 6 Then
  D = 0
  Nomer = "5"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "5"
End If
If D = 7 Then
  D = 0
  Nomer = "6"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "6"
End If
If D = 9 Then
  D = 0
  Nomer = "7"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi - 1
  S(Lokasi) = Nomer
  Locate 2 , Lokasi
  Lcd "7"
End If
If D = 10 Then
  D = 0
  Nomer = "8"
  Save_nomer = Save_nomer + Nomer
  Lokasi = Lokasi + 1
```

```

    S(lokas1) = Nomer
    Locate 2 , Lokasi
    Lcd "8"
End If
If D = 11 Then
    D = 0
    Nomer = "9"
    Save_nomer = Save_nomer - Nomer
    Lokasi = Lokasi + 1
    S(lokas1) = Nomer
    Locate 2 , Lokasi
    Lcd "9"
End If
If D = 14 Then
    D = 0
    Nomer = "0"
    Save_nomer = Save_nomer + Nomer
    Lokasi = Lokasi + 1
    S(lokas1) = Nomer
    Locate 2 , Lokasi
    Lcd "0"
End If
If D = 15 Then
    D = 0
    Goto Fix_number
End If
If D = 8 Then
    D = 0

    Goto Menu_awal
End If

```

Loop

```

Fix_number:
Writeeprom Save_nomer , E
Waitms 500
E = E + 15

```

```

Cls
Upperline
Lcd " Phone Number "
Waitms 800
Lowerline
Lcd " Save_nomer"
Waitms 800
Cls
Upperline
Lcd "Press ENT"
Do
Call Keypad
    If D = 8 Then
        D = 0
        Goto Menu_awal
    End If
    If D = 15 Then
        D = 0

        Goto Proses_number
    End If
Loop

```

```

Proses_number:
Panjang = ""
Jumlah = ""
Idx_t = 0
Cls
Upperline
Lcd " Waiting..... "

```

```

Wait 1
Lowerline
Lcd Len(save_nomer)
Panjang = Str(len(save_nomer))
Jumlah_number = Val(panjang)
Jumlah = Hex(jumlah_number)
Waitms 500
Upperline
Lcd S(3) ; S(4) ; S(5) ; S(6) ; S(7) ; S(8) ; S(9) ; S(10) ; S(11) ; S(12)
Waitms 500
Lowerline
Lcd Jumlah ; "26" ; S(4) ; S(3) ; S(6) ; S(5) ; S(8) ; S(7) ; S(10) ; S(9)
Do
  Call Keypad
  If Jumlah = "0B" Then
    T(1) = Jumlah + "9126" + S(4) + S(3) + S(6) + S(5) + S(8) + S(7) + S(
    Idx_t = 1
    Goto Simpan_seting
  End If
  If Jumlah = "0C" Then
    T(2) = Jumlah + "9126" + S(4) + S(3) + S(6) + S(5) + S(8) + S(7) + S(
    Idx_t = 2
    Goto Simpan_seting
  End If
  If Jumlah = "0D" Then
    T(3) = Jumlah + "9126" + S(4) + S(3) + S(6) + S(5) + S(8) + S(7) + S(
    Idx_t = 3
    Goto Simpan_seting
  End If
  If D = 8 Then
    D = 0
    Goto Menu_awal
  End If
Loop

```

```

Simpan_seting:
Writeeprom T(idx_t) , E1
Waitms 500
E1 = E1 + 20

```

```

Upperline
Lcd "Saving Succeeded "
Lowerline
Lcd "Press MEN button"
Do
  Call Keypad
  If D = 8 Then
    D = 0
    Goto Menu_awal
  End If
Loop

```

'View

```

Pilihan_2:
Cls

Upperline
Lcd "View"
Lowerline
Lcd "Press Jp/Down"
Wait 2

User = 1
Alamat = 0
Goto Lihat

Do
  Call Keypad

```

```
If D = 16 Then
  D = 0
  Alamat = Alamat + 15
  Incr User

  If User > 14 Then
    User = 14
    Alamat = 195
    Goto Up
  End If

  Lihat:
  Readeeprom Save_nomer , Alamat
  Waitms 300
  Cls
  Upperline
  Lcd "USER " ; User
  Lowerline
  Lcd Save_nomer

End If

Up:
If D = 12 Then
  D = 0

  Alamat = Alamat - 15
  Decr User

  If User < 1 Then
    User = 1
    Alamat = 0
    Goto Down
  End If

  Readeeprom Save_nomer , Alamat
  Waitms 300
  Cls
  Upperline
  Lcd "USER " ; User
  Lowerline
  Lcd Save_nomer

End If

Down:
If D = 8 Then
  D = 0
  Cls
  Goto Menu_awal
End If

Loop
```

'Ready

```
pilihan_3:
Cls
```

```
Print "at+cmd=1"
Print Chr(13)
Waitms 500
```

```
Print "at+cmms=1"
Print Chr(13)
Waitms 500

Print "at+cmmi=1,1,0,1,1"
Print Chr(13)
Waitms 500

Aa = 0
Bb = 0
Cc = 0
Dd = 0
Ee = 0
Ff = 0
Gg = 0
Hh = 0
Ii = 0

Sta1 = ""
Sta2 = ""
Sta3 = ""
Sta4 = ""
Status = ""

Adasms = ""
Lb = ""
Ss = ""
Ss1 = ""
Nn = ""
Si = ""
Si2 = ""
I = ""
I1 = ""

Lebar = ""
Smac = ""
Nomor = ""
Isi = ""
Isi1 = ""
Sisa = ""

Count = 0

Upperline
Lcd " Ready"
Do
  Adasms = Waitkey()
  If Adasms = "1" Then
    Portc.5 = 0
    Goto Baca
  End If
Loop

End
```



## Sub Keypad

```
Portb.0 = 0
Portb.1 = 1
Portb.2 = 1
Portb.3 = 1
```

```
  If Pinb.4 = 0 Then
    Waitms 300
    D = 1
  End If
  If Pinb.5 = 0 Then
    Waitms 300
    D = 2
  End If
  If Pinb.6 = 0 Then
    Waitms 300
    D = 3
  End If
  If Pinb.7 = 0 Then
    Waitms 300
    D = 4
  End If
```

```
Portb.0 = 1
Portb.1 = 0
Portb.2 = 1
Portb.3 = 1
```

```
  If Pinb.4 = 0 Then
    Waitms 300
    D = 5
  End If
  If Pinb.5 = 0 Then
    Waitms 300
    D = 6
  End If
  If Pinb.6 = 0 Then
    Waitms 300
    D = 7
  End If
  If Pinb.7 = 0 Then
    Waitms 300
    D = 8
  End If
```

```
Portb.0 = 1
Portb.1 = 1
Portb.2 = 0
Portb.3 = 1
```

```
  If Pinb.4 = 0 Then
    Waitms 300
    D = 9
  End If
  If Pinb.5 = 0 Then
    Waitms 300
    D = 10
  End If
  If Pinb.6 = 0 Then
    Waitms 300
    D = 11
  End If
  If Pinb.7 = 0 Then
    Waitms 300
    D = 12
  End If
```

```
Portb.0 = 1
Portb.1 = 1
Portb.2 = 1
Portb.3 = 0
```

```
    If Pinb.4 = 0 Then
    Waitms 300
    D = 13
    End If
    If Pinb.5 = 0 Then
    Waitms 300
    D = 14
    End If
    If Pinb.6 = 0 Then
    Waitms 300
    D = 15
    End If
    If Pinb.7 = 0 Then
    Waitms 300
    D = 16
    End If

    End Sub
```

'CAN

'0

'ENT

'down

Baca:

'identifikasi 1

Count = 0

```
Print "at+cmgr=1"
Print Chr(13)
```

Do

Aa = Ischarwaiting()

```
If Aa > 0 Then
Ss = Waitkey()
Count = Count + 1
```

If Count = 24 Then

Do

Bb = Ischarwaiting()

```
If Bb > 0 Then
Lb = Waitkey()
Lebar = Lebar + Lb
Count = Count + 1
```

If Count = 26 Then

Do

Cc = Ischarwaiting()

```
If Cc > 0 Then
Si = waitkey()
Count = Count + 1
```

If Count = 28 Then

Do

Dd = Ischarwaiting()

```
If Dd > 0 Then
I = Waitkey()
Smec = Smec + I
Count = Count - 1
```

```

                If Count = 30 Then
                    Wait 1
                    Goto Bacalebar
                End If
            End If
        Loop

    End If
End If
Loop

Loop
End If
End If
Loop

Bacalebar:
    If Lebar = "27" Then
        Goto Bacasmc
    End If

    Goto Akhir

Bacasmc:
    If Smc = "07" Then
        Goto Bacam3
    End If

    If Smc = "06" Then
        Goto Bacasimpati
    End If

    If Smc = "05" Then
        Goto Bacamentari
    End If

    Goto Akhir

Bacam3:
    Aa = 0
    Bb = 0
    Cc = 0
    Dd = 0
    Ee = 0
    Ff = 0

    Sual = ""
    Sta2 = ""
    Sta3 = ""
    Sta4 = ""
    Status = ""

    Adasms = ""
    Ss = ""
    Nn = ""
    S1 = ""
    Si2 = ""
    I = ""
    I1 = ""

' baca lebar dat
' baca lebar sms
' baca lebar sms
' baca lebar sms
' baca lebar sms
'prosehnya indo
```

```
Nomor = ""
Isi = ""
Isi1 = ""
Sisa = ""

Count = 0
```

```
Print "at+cmgr=1"
Print Chr(13)
```

```
Do
Aa = Ischarwaiting()
```

```
If Aa > 0 Then
Ss = Waitkey()
Count = Count + 1
```

```
If Count = 47 Then
```

```
Do
Bb = Ischarwaiting()
```

```
If Bb > 0 Then
Nn = Waitkey()
Nomor = Nomor + Nn
Count = Count + 1
```

```
If Count = 65 Then
```

```
Do
Cc = Ischarwaiting()
```

```
If Cc > 0 Then
Si = Waitkey()
Count = Count + 1
```

```
If Count = 85 Then
```

```
Do
Dd = Ischarwaiting()
```

```
If Dd > 0 Then
I = Waitkey()
Isi = Isi + I
Count = Count + 1
```

```
If Count = 91 Then
```

```
Do
Ee = Ischarwaiting()
```

```
If Ee > 0 Then
I1 = Waitkey()
Isi1 = Isi1 + I1
Count = Count + 1
```

```
If Count = 99 Then
```

```
Do
Ff = Ischarwaiting()
```

```
If Ff > 0 Then
```

```
Si2 = Waitkey()
Count = Count + 1
```

```
If Count = 102 Then
```

```
Waitms 500
Goto Prosed 'PR
```

```

End If
End If
End If
End If
End If
Loop
End If
End If
End If
Loop
End If
End If
End If
End If
Loop
End If
Loop
End If
Loop
End If

```

Bacasimpat:

'prosesnya telk

```

Aa = 0
Bb = 0
Cc = 0
Dd = 0
Ee = 0
Ff = 0

```

```

Sta1 = ""
Sta2 = ""
Sta3 = ""
Sta4 = ""
Status = ""

```

```

Adasms = ""
Sa = ""
Nh = ""
Si = ""
Si2 = ""
I = ""
I1 = ""

```

```

Nomor = ""
Isi = ""
Isil = ""
Sisa = ""

```

```
Count = 0
```

```
Print "at-cmgr=1"
Print Chr(13)
```

```
Do
As = Ischarwaiting()
```

```
If As > 0 Then
Ss = Waitkey()
Count = Count + 1
```

```
If Count = 45 Then
Do
Bb = Ischarwaiting()
```

```

If Bb > 0 Then
  Nn = Waitkey()
  Nomor = Nomor + Nn
  Count = Count + 1

  If Count = 63 Then
    Do
      Cc = Ischarwaiting()

      If Cc > 0 Then
        Si = Waitkey()
        Count = Count + 1

        If Count = 83 Then
          Do
            Dd = Ischarwaiting()

            If Dd > 0 Then
              I = Waitkey()
              Isi = Isi + I
              Count = Count + 1

              If Count = 89 Then
                Do
                  Ee = Ischarwaiting()

                  If Ee > 0 Then
                    I1 = Waitkey()
                    Isi1 = Isi1 + I1
                    Count = Count + 1

                    If Count = 97 Then
                      Do
                        Ff = Ischarwaiting()

                        If Ff > 0 Then
                          Si2 = Waitkey()
                          Count = Count + 1

                          If Count = 100 Then
                            Waitms 500
                            Goto Proses 'PR
                          End If
                        End if
                      Loop
                    End If
                  End If
                Loop
              End If
            End If
          Loop
        End If
      End If
    Loop
  End If
End If
Loop
End If
Loop
End If
Bacamentari:

```

'prosesnya sate

```
Aa = 0
Bb = 0
Cc = 0
Dd = 0
Ee = 0
Ff = 0

Sta1 = ""
Sta2 = ""
Sta3 = ""
Sta4 = ""
Status = ""

Adasms = ""
Ss = ""
Nn = ""
Si = ""
Si2 = ""
I = ""
I1 = ""

Nomor = ""
Isi = ""
Isi1 = ""
Sisa = ""

Count = 0

Print "at+cmgr=1"
Print Chr(13)

Do
Aa = Ischarwaiting()

If Aa > 0 Then
Ss = Waitkey()
Count = Count + 1

If Count = 43 Then
Do
Bb = Ischarwaiting()

If Bb > 0 Then
Nn = Waitkey()
Nomor = Nomor + Nn
Count = Count + 1

If Count = 61 Then

Do
Cc = Ischarwaiting()

If Cc > 0 Then
Si = Waitkey()
Count = Count + 1

If Count = 81 Then

Do
Id = Ischarwaiting()

If Dd > 0 Then
I = Waitkey()
Isi = Isi + I
Count = Count + 1
```





```
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 290
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 310
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 330
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 350
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 370
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 390
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 410
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 430
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
Readeeprom Simpan_number , 450
Waitms 500
  If Nomor = Simpan_number Then
    Goto Proses2
  End If
```

```
Readeprom Simpan_number , 470  
Waitms 500
```

```
  If Nomor = Simpan_number Then  
    Goto Proses2  
  End If
```

```
Goto Akhir
```

```
Proses2:
```

```
'identifikasi 1
```

```
  If Isi = "C72215" Then  
    Goto Kondisi
```

```
  ElseIf Isi = "4FA713" Then
```

```
    Goto Nyalakan
```

```
  ElseIf Isi = "4FA311" Then
```

```
    Goto Matikan
```

```
  End If
```

```
Goto Akhir
```

```
Kondisi:
```

```
'KONDISI SYSTEM
```

```
  If Portd.0 = 1 Then  
    Sta1 = "1"  
    Waitms 100  
  End If
```

```
  If Portd.0 = 0 Then  
    Sta1 = "0"  
    Waitms 100  
  End If
```

```
  If Portd.1 = 1 Then  
    Sta2 = "1"  
  End If
```

```
  If Portd.1 = 0 Then  
    Sta2 = "0"  
    Waitms 100  
  End If
```

```
  If Portd.2 = 1 Then  
    Sta3 = "1"  
    Waitms 100  
  End If
```

```
  If Portd.2 = 0 Then  
    Sta3 = "0"  
    Waitms 100  
  End If
```

```
  If Portd.3 = 1 Then  
    Sta4 = "1"  
    Waitms 100  
  End If
```

```
  If Portd.3 = 0 Then  
    Sta4 = "0"  
    Waitms 100  
  End If
```

```
Status = Sta4 + Sta3 + Sta2 + Sta1  
Waitms 300
```

```
If Status = "0000" Then
```

```
  Isikirim = "04B1582C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0001" Then
```

```
  Isikirim = "04B0582C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0010" Then
```

```
  Isikirim = "0431582C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0011" Then
```

```
  Isikirim = "0430582C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0100" Then
```

```
  Isikirim = "04B1182C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0101" Then
```

```
  Isikirim = "04B0182C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0110" Then
```

```
  Isikirim = "0431182C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "0111" Then
```

```
  Isikirim = "0430182C06"  
  Waitms 200  
  Goto Siapkirim
```

```
Elseif Status = "1000" Then
```

```
Isikirim = "04B1580C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1001" Then
```

```
Isikirim = "04B0580C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1010" Then
```

```
Isikirim = "0430580C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1011" Then
```

```
Isikirim = "0430580C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1100" Then
```

```
Isikirim = "04B1180C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1101" Then
```

```
Isikirim = "04B0180C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1110" Then
```

```
Isikirim = "0431180C06"  
Waitms 200  
Goto Siapkirim
```

```
Elseif Status = "1111" Then
```

```
Isikirim = "0430180C06"  
Waitms 200  
Goto Siapkirim
```

```
End if
```

```
Goto Akhir
```

```
Siapkirim:
```

```
Siapdikirim = "07912658050000F00100" + Nomor + "0000" + Isikirim + "{026}"
```

```
Waitms 500
Print "at+cmgn=18"
Print Chr(13)
Wait 1

Print Siapdikirim
Wait 1

Goto Akhir
```

Nyalakan:

'NYALAKAN

```
If Isil = "0483C162" Then
Portc.0 = 0
Waitms 100

Elseif Isil = "0483C164" Then
Portc.1 = 0
Waitms 100

Elseif Isil = "0483C166" Then
Portc.2 = 0
Waitms 100

Elseif Isil = "0483C168" Then
Portc.3 = 0
Waitms 100

Elseif Isil = "0483C564" Then
Portc.0 = 0
Portc.1 = 0
Waitms 100

Elseif Isil = "0483C566" Then
Portc.0 = 0
Portc.2 = 0
Waitms 100

Elseif Isil = "0483C568" Then
Portc.0 = 0
Portc.3 = 0
Waitms 100

Elseif Isil = "0483C966" Then
Portc.1 = 0
Portc.2 = 0
Waitms 100

Elseif Isil = "0483C968" Then
Portc.1 = 0
Portc.3 = 0
Waitms 100

Elseif Isil = "0483CD68" Then
Portc.2 = 0
Portc.3 = 0
Waitms 100

Elseif Isil = "048BC966" Then
Portc.0 = 0
Portc.1 = 0
Portc.2 = 0
Waitms 100
```

```
Elseif Isil = "048BC968" Then
Portc.0 = 0
Portc.1 = 0
Portc.3 = 0
Waitms 100
```

```
Elseif Isil = "048BCD68" Then
Portc.0 = 0
Portc.2 = 0
Portc.3 = 0
Waitms 100
```

```
Elseif Isil = "0493CD68" Then
Portc.1 = 0
Portc.2 = 0
Portc.3 = 0
Waitms 100
```

```
Elseif Isil = "1493CD68" Then
Portc.0 = 0
Portc.1 = 0
Portc.2 = 0
Portc.3 = 0
Waitms 100
```

```
Goto Akhir
End If
```

```
Goto Pilihan_3
```

Matikan:

'MATIKAN

```
If Isil = "0483C162" Then
Portc.0 = 1
Waitms 100
```

```
Elseif Isil = "0483C164" Then
Portc.1 = 1
Waitms 100
```

```
Elseif Isil = "0483C166" Then
Portc.2 = 1
Waitms 100
```

```
Elseif Isil = "0483C168" Then
Portc.3 = 1
Waitms 100
```

```
Elseif Isil = "0483C564" Then
Portc.0 = 1
Portc.1 = 1
Waitms 100
```

```
Elseif Isil = "0483C566" Then
Portc.0 = 1
Portc.2 = 1
Waitms 100
```

```
Elseif Isil = "0483C568" Then
Portc.0 = 1
Portc.3 = 1
Waitms 100
```

```
Elseif Isil = "0483C966" Then
```

```
Portc.1 = 1
Portc.2 = 1
Waitms 100

Elseif Is11 = "0483C968" Then
Portc.1 = 1
Portc.3 = 1
Waitms 100

Elseif Is11 = "0483CD58" Then
Portc.2 = 1
Portc.3 = 1
Waitms 100

Elseif Is11 = "048BC966" Then
Portc.0 = 1
Portc.1 = 1
Portc.2 = 1
Waitms 100

Elseif Is11 = "048BC968" Then
Portc.0 = 1
Portc.1 = 1
Portc.3 = 1
Waitms 100

Elseif Is11 = "048BCL68" Then
Portc.0 = 1
Portc.2 = 1
Portc.3 = 1
Waitms 100

Elseif Is11 = "0493CD68" Then
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Waitms 100

Elseif Is11 = "1493CD68" Then
Portc.0 = 1
Portc.1 = 1
Portc.2 = 1
Portc.3 = 1
Waitms 100

Goto Akhir
End If

Goto Pilihan_3

Akhir:
Print "at+cmgd=1"
Print Chr(13)
Waitms 500

Goto Pilihan_3
```

# DATASHEET

---



# DATASHEET

## DIODA

---

# 1N5400 - 1N5408

## Features

- 3.0 ampere operation at  $T_A = 75^\circ\text{C}$  with no thermal runaway.
- High current capability.
- Low leakage.



DO-201AD  
COLOR BAND DENOTES CATHODE

## General Purpose Rectifiers

### Absolute Maximum Ratings\* $T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value										Units
		5400	5401	5402	5403	5404	5405	5406	5407	5408		
$V_{RRM}$	Maximum Repetitive Reverse Voltage	50	100	200	300	400	500	600	800	1000		V
$I_{F(AV)}$	Average Rectified Forward Current, .375" lead length @ $T_A = 75^\circ\text{C}$	3.0										A
$I_{FSM}$	Non-repetitive Peak Forward Surge Current 8.3 ms Single Half-Sine-Wave	200										A
$T_{stg}$	Storage Temperature Range	-55 to +150										$^\circ\text{C}$
$T_J$	Operating Junction Temperature	-55 to +150										$^\circ\text{C}$

\*These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

## Thermal Characteristics

Symbol	Parameter	Value	Units
$P_D$	Power Dissipation	6.25	W
$R_{JA}$	Thermal Resistance, Junction to Ambient	20	$^\circ\text{C}/\text{W}$

## Electrical Characteristics $T_A = 25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Device										Units
		5400	5401	5402	5403	5404	5405	5406	5407	5408		
$V_F$	Forward Voltage @ 3.0 A	1.2										V
$I_{rr}$	Maximum Full Load Reverse Current, Full Cycle $T_A = 105^\circ\text{C}$	0.5										mA
$I_R$	Reverse Current @ rated $V_R$ $T_A = 25^\circ\text{C}$ $T_A = 100^\circ\text{C}$	5.0 500										$\mu\text{A}$ $\mu\text{A}$
$C_T$	Total Capacitance $V_R = 4.0\text{ V}$ , $f = 1.0\text{ MHz}$	30										pF

Typical Characteristics

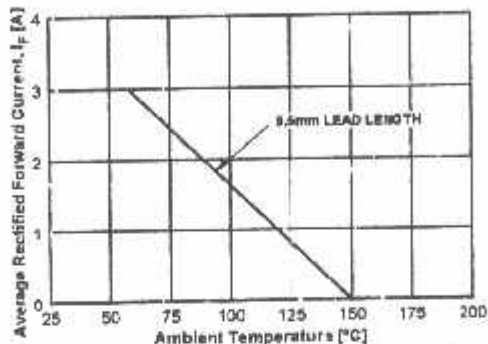


Figure 1. Forward Current Derating Curve

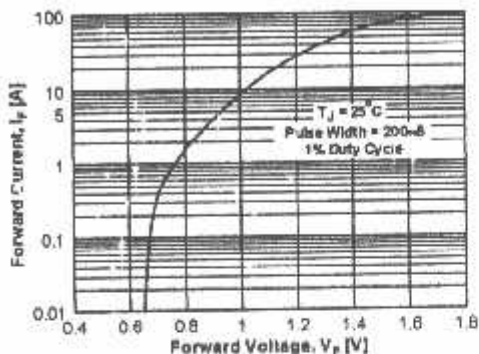


Figure 2. Forward Voltage Characteristics

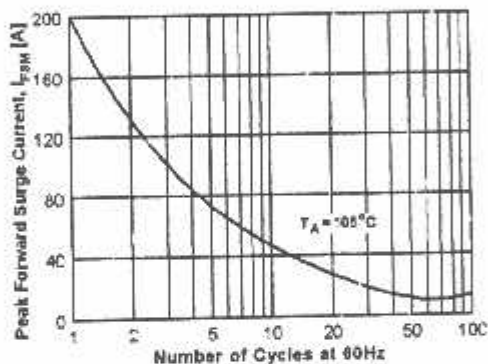


Figure 3. Non-Repetitive Surge Current

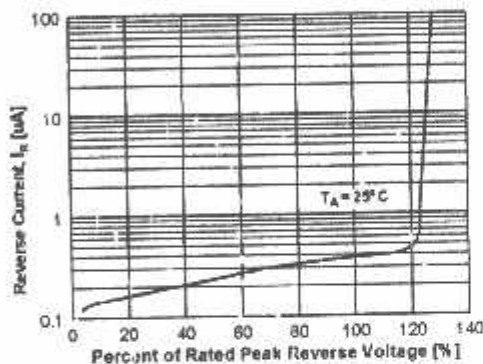


Figure 4. Reverse Current vs Reverse Voltage

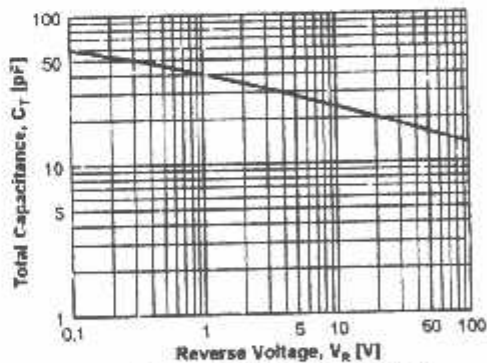


Figure 5. Total Capacitance

DATASHEET  
OPTOCOUPLER

---



## 6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N26, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

Most Economical Optoisolator Choice for Medium Speed, Switching Applications  
Meets or Exceeds All JEDEC Registered Specifications

To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a test option.

### Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

### MAXIMUM RATINGS (T<sub>A</sub> = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
<b>INPUT LED</b>			
Reverse Voltage	V <sub>R</sub>	3	Volts
Forward Current — Continuous	I <sub>F</sub>	60	mA
LED Power Dissipation @ T <sub>A</sub> = 25°C with Negligible Power in Output Detector Derate above 25°C	P <sub>D</sub>	120 1.41	mW mW/°C
<b>OUTPUT TRANSISTOR</b>			
Collector-Emitter Voltage	V <sub>CEO</sub>	30	Volts
Emitter-Collector Voltage	V <sub>ECO</sub>	7	Volts
Collector-Base Voltage	V <sub>CBO</sub>	70	Volts
Collector Current — Continuous	I <sub>C</sub>	150	mA
Detector Power Dissipation @ T <sub>A</sub> = 25°C with Negligible Power in Input LED Derate above 25°C	P <sub>D</sub>	150 1.76	mW mW/°C
<b>MECHANICAL DEVICE</b>			
Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 sec Duration)	V <sub>ISO</sub>	7500	Vac(pk)
Total Device Power Dissipation @ T <sub>A</sub> = 25°C Derate above 25°C	P <sub>D</sub>	250 2.94	mW mW/°C
Ambient Operating Temperature Range <sup>(2)</sup>	T <sub>A</sub>	-55 to +100	°C
Storage Temperature Range <sup>(2)</sup>	T <sub>stg</sub>	-55 to +150	°C
Soldering Temperature (10 sec, 1/16" from case)	T <sub>L</sub>	280	°C

Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common. Refer to Quality and Reliability Section in Opto Data Book for information on test conditions.

4N25, 4N26, 4N27 and 4N28 devices are Motorola's recommended choices for future use and best overall value. Optoisolator is a trademark of Motorola, Inc.

**4N25\***  
**4N25A\***  
**4N26\***  
[CTR = 20% Min]  
**4N27**  
**4N28**  
[CTR = 10% Min]

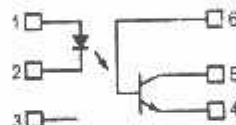
\*Motorola Preferred Devices

### STYLE 1 PLASTIC



STANDARD THRU-HOLE  
CASE 730A-04

### SCHEMATIC



- PIN 1: LED ANODE
- 2: LED CATHODE
- 3: N.C.
- 4: EMITTER
- 5: COLLECTOR
- 6: BASE



**MOTOROLA**

# 4N25 4N25A 4N26 4N27 4N28

ELECTRICAL CHARACTERISTICS ( $T_A = 25^\circ\text{C}$  unless otherwise noted)<sup>(1)</sup>

Characteristic	Symbol	Min	Typ <sup>(1)</sup>	Max	Unit
<b>PUT LED</b>					
Forward Voltage ( $I_F = 10\text{ mA}$ )	$V_F$	—	1.15 1.3 1.05	1.5	Volts
			$T_A = 25^\circ\text{C}$		
			$T_A = -55^\circ\text{C}$		
			$T_A = 100^\circ\text{C}$		
Reverse Leakage Current ( $V_R = 3\text{ V}$ )	$I_R$	—	—	100	$\mu\text{A}$
Capacitance ( $V = 0\text{ V}$ , $f = 1\text{ MHz}$ )	$C_J$	—	18	—	pF
<b>INPUT TRANSISTOR</b>					
Collector-Emitter Dark Current ( $V_{CE} = 10\text{ V}$ , $T_A = 25^\circ\text{C}$ )	$I_{CEO}$	—	1	50	nA
			1	100	
( $V_{CE} = 10\text{ V}$ , $T_A = 100^\circ\text{C}$ )	$I_{CEO}$	—	1	—	$\mu\text{A}$
Collector-Base Dark Current ( $V_{CB} = 10\text{ V}$ )	$I_{CBO}$	—	0.2	—	nA
Collector-Emitter Breakdown Voltage ( $I_C = 1\text{ mA}$ )	$V_{(BR)CEO}$	30	45	—	Volts
Collector-Base Breakdown Voltage ( $I_C = 100\text{ }\mu\text{A}$ )	$V_{(BR)CBO}$	70	100	—	Volts
Emitter-Collector Breakdown Voltage ( $I_E = 100\text{ }\mu\text{A}$ )	$V_{(BR)ECO}$	7	7.8	—	Volts
DC Current Gain ( $I_C = 2\text{ mA}$ , $V_{CE} = 5\text{ V}$ )	$h_{FE}$	—	500	—	—
Collector-Emitter Capacitance ( $f = 1\text{ MHz}$ , $V_{CE} = 0$ )	$C_{CE}$	—	7	—	pF
Collector-Base Capacitance ( $f = 1\text{ MHz}$ , $V_{CB} = 0$ )	$C_{CB}$	—	19	—	pF
Emitter-Base Capacitance ( $f = 1\text{ MHz}$ , $V_{EB} = 0$ )	$C_{EB}$	—	9	—	pF
<b>OUTPUT</b>					
Output Collector Current ( $I_F = 10\text{ mA}$ , $V_{CE} = 10\text{ V}$ )	$I_C$ (CTR) <sup>(2)</sup>		2 (20) 1 (10)	7 (70) 5 (50)	mA (%)
				—	
				—	
Collector-Emitter Saturation Voltage ( $I_C = 2\text{ mA}$ , $I_F = 50\text{ mA}$ )	$V_{CE(sat)}$	—	0.15	0.5	Volts
Turn-On Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>	$t_{on}$	—	2.8	—	$\mu\text{s}$
Turn-Off Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>	$t_{off}$	—	4.5	—	$\mu\text{s}$
Rise Time ( $I_F = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>	$t_r$	—	1.2	—	$\mu\text{s}$
Fall Time ( $I_C = 10\text{ mA}$ , $V_{CC} = 10\text{ V}$ , $R_L = 100\text{ }\Omega$ ) <sup>(3)</sup>	$t_f$	—	1.3	—	$\mu\text{s}$
Isolation Voltage ( $f = 60\text{ Hz}$ , $t = 1\text{ sec}$ ) <sup>(4)</sup>	$V_{ISO}$	7500	—	—	Vac(pk)
Isolation Resistance ( $V = 500\text{ V}$ ) <sup>(4)</sup>	$R_{ISO}$	$10^{11}$	—	—	$\Omega$
Isolation Capacitance ( $V = 0\text{ V}$ , $f = 1\text{ MHz}$ ) <sup>(4)</sup>	$C_{ISO}$	—	0.2	—	pF

<sup>(1)</sup> Always design to the specified minimum/maximum electrical limits (where applicable).

<sup>(2)</sup> Current Transfer Ratio (CTR) =  $I_C/I_F \times 100\%$ .

<sup>(3)</sup> For test circuit setup and waveforms, refer to Figure 11.

<sup>(4)</sup> For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

TYPICAL CHARACTERISTICS

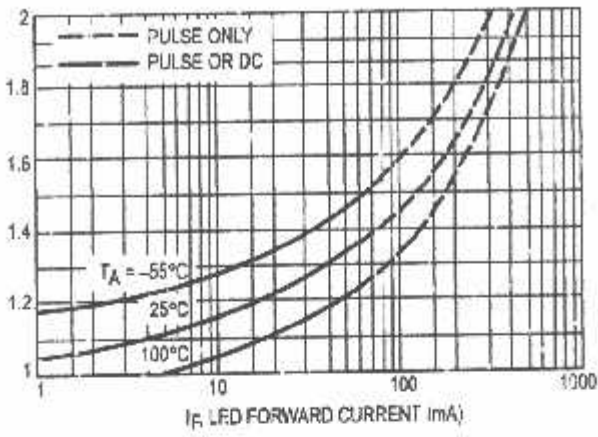


Figure 1. LED Forward Voltage versus Forward Current

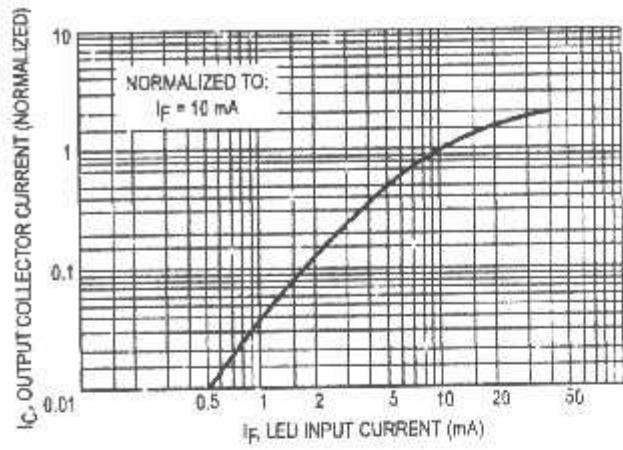


Figure 2. Output Current versus Input Current

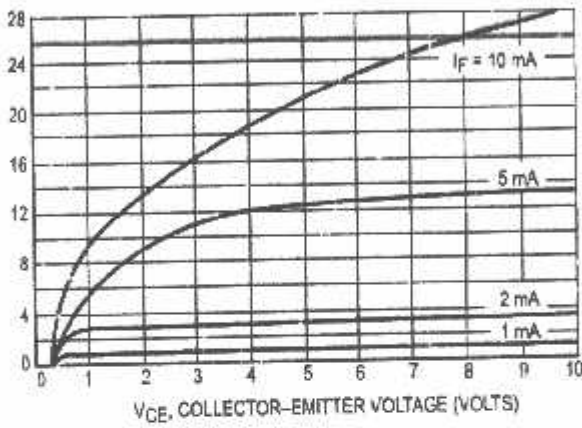


Figure 3. Collector Current versus Collector-Emitter Voltage

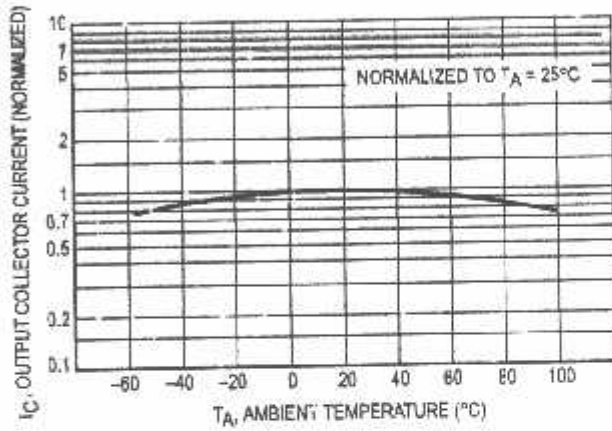


Figure 4. Output Current versus Ambient Temperature

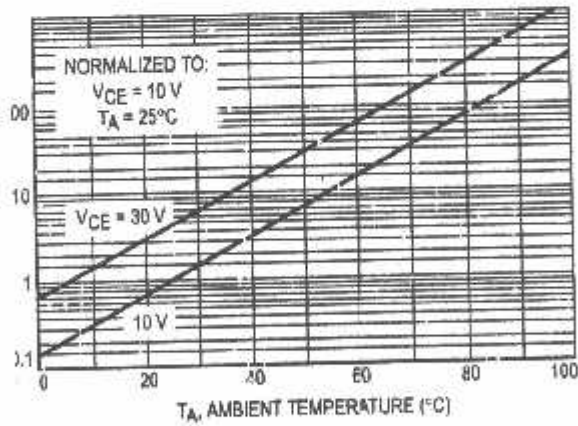


Figure 5. Dark Current versus Ambient Temperature

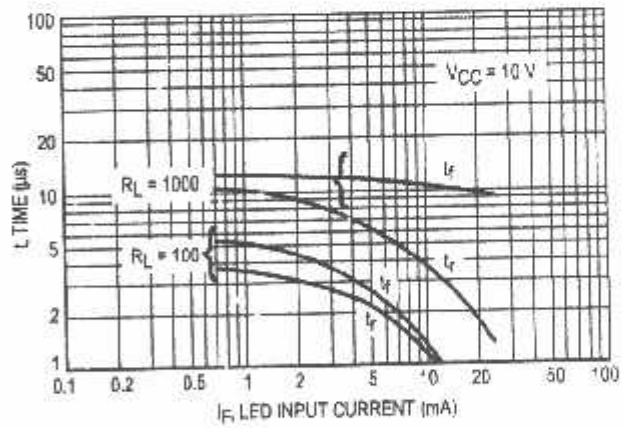
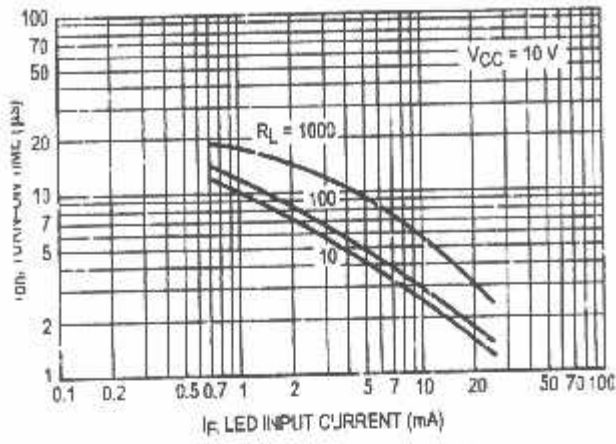
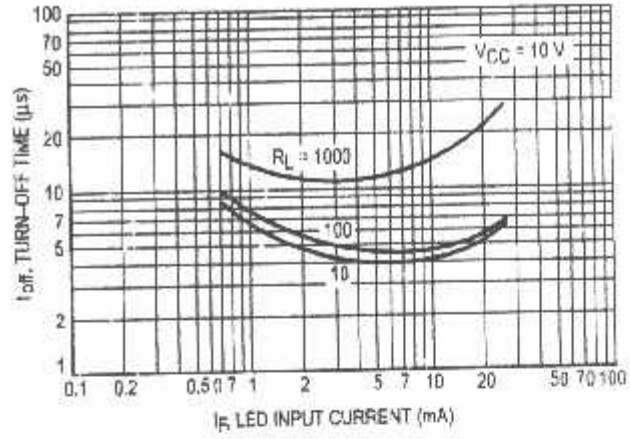


Figure 6. Rise and Fall Times (Typical Values)

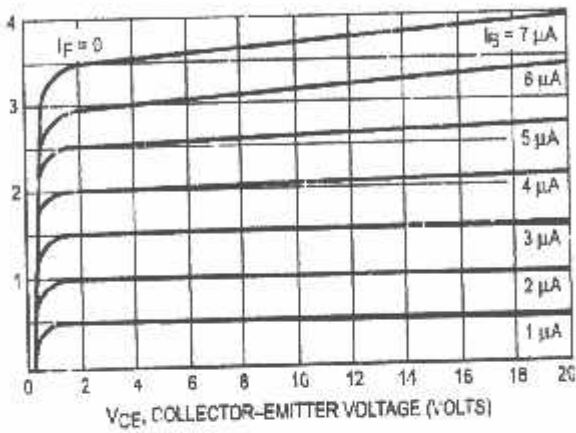
**N25 4N25A 4N26 4N27 4N28**



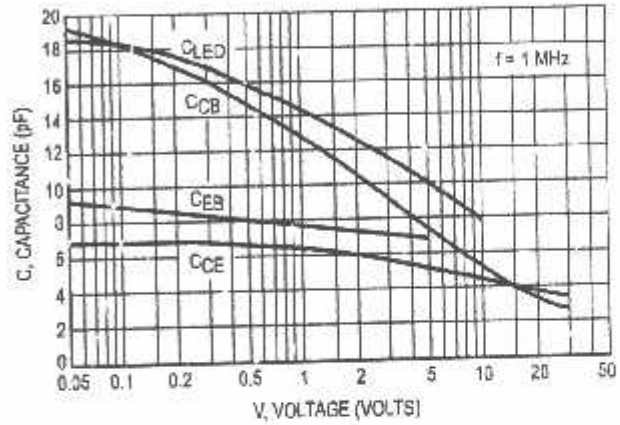
**Figure 7. Turn-On Switching Times (Typical Values)**



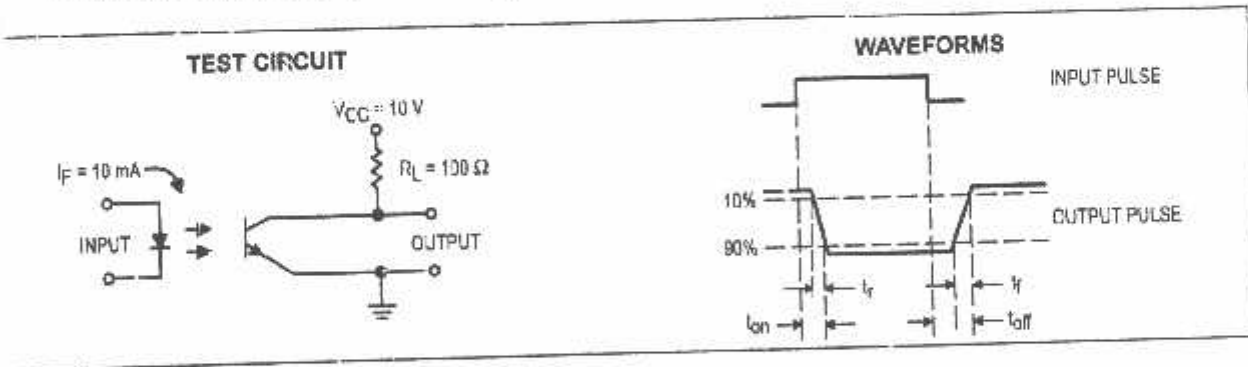
**Figure 8. Turn-Off Switching Times (Typical Values)**



**Figure 9. DC Current Gain (Detector Only)**



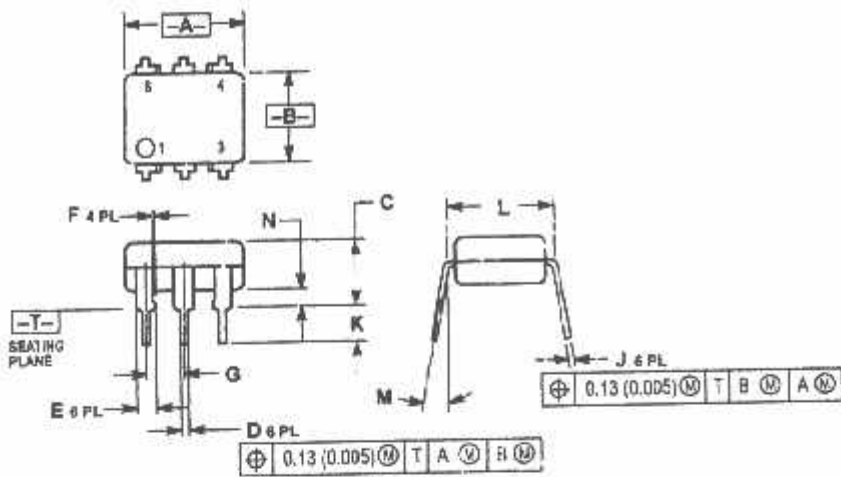
**Figure 10. Capacitances versus Voltage**



**Figure 11. Switching Time Test Circuit and Waveforms**



PACKAGE DIMENSIONS

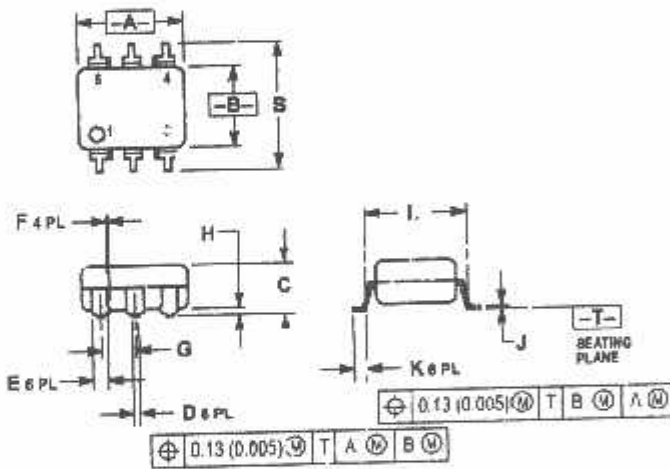


- NOTES  
 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982  
 2. CONTROLLING DIMENSION INCH  
 3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.292	0.260	7.40	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.35
G	0.100 BSC		2.54 BSC	
J	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.015	0.100	0.38	2.54

- STYLE 1:  
 PIN 1: ANODE  
 2: CATHODE  
 3: NC  
 4: EMITTER  
 5: COLLECTOR  
 6: BASE

CASE 730A-04  
 ISSUE G

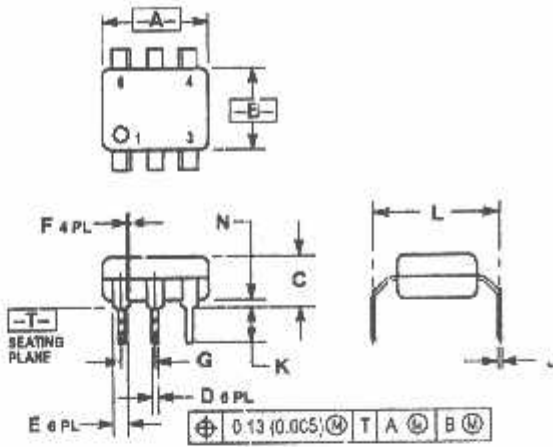


- NOTES  
 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982  
 2. CONTROLLING DIMENSION INCH

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.35
G	0.100 BSC		2.54 BSC	
H	0.020	0.025	0.51	0.63
J	0.008	0.012	0.20	0.30
K	0.008	0.035	0.16	0.88
L	0.320 BSC		8.13 BSC	
S	0.332	0.380	8.43	9.60

\*Consult factory for leadform option availability

CASE 730C-04  
 ISSUE D



- NOTES:  
 1. DIMENSIONING /ND TOLERANCING PER ANSI Y14.5A, 1982.  
 2. CONTROLLING DIMENSION: INCH.  
 3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.320	0.350	8.13	8.89
B	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.016	0.020	0.41	0.50
E	0.040	0.070	1.02	1.77
F	0.010	0.014	0.25	0.36
G	0.100 BSC		2.54 BSC	
J	0.005	0.212	0.21	5.30
K	0.100	0.150	2.54	3.81
L	0.430	0.425	10.91	10.90
N	0.015	0.040	0.38	1.02

\*Consult factory for leadform option availability

CASE 730D-05  
 ISSUE D

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of a Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such intended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

to reach us:  
 EUROPE: Motorola Literature Distribution;  
 Box 20912; Phoenix, Arizona 85086 1-800-441-2447

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki,  
 6F Setbu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

INTERNET: RMPAX0@mail.sps.mot.com - TOUCHSTONE (602) 244-0809  
 INTERNET: http://Design-NET.com

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,  
 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26625288

MOTOROLA

4N25/D



DATASHEET  
IC MAX 232

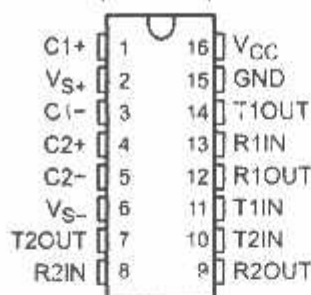
---

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- $\mu$ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 -- 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- $\mu$ F Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
MAX232I . . . D, DW, OR N PACKAGE  
(TOP VIEW)



## Description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept  $\pm 30$ -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

## ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
	Package	Quantity		
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

INDUCTION DATA information is current as of publication date. Inductors conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2004, Texas Instruments Incorporated

**TEXAS  
INSTRUMENTS**

POST OFFICE BOX 855303 • DALLAS, TEXAS 75265

**232, MAX232I**  
**1 EIA-232 DRIVERS/RECEIVERS**

17L - FEBRUARY 1989 - REVISED MARCH 2004

**Function Tables**

**EACH DRIVER**

INPUT TIN	OUTPUT TOUT
L	H
H	L

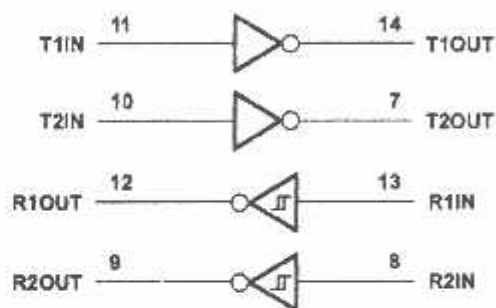
H = high level, L = low level

**EACH RECEIVER**

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

**diagram (positive logic)**



 **TEXAS  
INSTRUMENTS**

POST OFFICE BOX 655308 • DALLAS, TEXAS 75265

# MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLSD47L - FEBRUARY 1989 - REVISED MARCH 2004

## absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, $V_{CC}$ (see Note 1)	.....	-0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	.....	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	.....	-0.3 V to -15 V
Input voltage range, $V_i$ : Driver	.....	-0.3 V to $V_{CC} + 0.3$ V
Receiver	.....	$\pm 30$ V
Output voltage range, $V_o$ : T1OUT, T2OUT	.....	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	.....	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	.....	Unlimited
Package thermal impedance, $\theta_{JA}$ (see Notes 2 and 3): D package	.....	73°C/W
DW package	.....	57°C/W
N package	.....	67°C/W
NS package	.....	64°C/W
Operating virtual junction temperature, $T_J$	.....	150°C
Storage temperature range, $T_{stg}$	.....	-65°C to 150°C

Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

- NOTES: 1. All voltages are with respect to network GND.  
 2. Maximum power dissipation is a function of  $T_{J(max)}$ ,  $\theta_{JA}$ , and  $T_A$ . The maximum allowable power dissipation at any allowable ambient temperature is  $P_D = (T_{J(max)} - T_A) / \theta_{JA}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.  
 3. The package thermal impedance is calculated in accordance with JEDEC 51-7.

## recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CC}$	Supply voltage	4.5	5	5.5	V
$V_{IH}$	High-level input voltage (T1IN, T2IN)	2			V
$V_{IL}$	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			$\pm 30$	V
A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

## electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$I_{CC}$ Supply current	$V_{CC} = 5.5$ V, All outputs oper., $T_A = 25^\circ\text{C}$		8	10	mA

†All typical values are at  $V_{CC} = 5$  V and  $T_A = 25^\circ\text{C}$ .

NOTE 4 - Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$ .



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

**(232, MAX232)**  
**(L EIA-232 DRIVERS/RECEIVERS)**

17L - FEBRUARY 1989 - REVISED MARCH 2004

**DRIVER SECTION**

Typical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
High-level output voltage	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND	5	7		V
Low-level output voltage‡	T1OUT, T2OUT	$R_L = 3\text{ k}\Omega$ to GND		-7	-5	V
Output resistance	T1OUT, T2OUT	$V_{S+} = V_{S-} = 0, V_O = \pm 2\text{ V}$	300			$\Omega$
Short-circuit output current	T1OUT, T2OUT	$V_{CC} = 5.5\text{ V}, V_O = 0$		$\pm 10$		mA
Short-circuit input current	T1IN, T2IN	$V_I = 0$			200	$\mu\text{A}$

† Typical values are at  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$ .

‡ Algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage only.

§ More than one output should be shorted at a time.

¶ Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

Timing characteristics,  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$  (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
Driver slew rate		$R_L = 3\text{ k}\Omega$ to 7 $\text{k}\Omega$ , See Figure 2			30	V/ $\mu\text{s}$
Driver transition region slew rate		See Figure 3		3		V/ $\mu\text{s}$
Data rate		One TOUT switching		120		kbit/s

† Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

**RECEIVER SECTION**

Typical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
High-level output voltage	R1OUT, R2OUT	$I_{OH} = -1\text{ mA}$	3.5			V
Low-level output voltage‡	R1OUT, R2OUT	$I_{OL} = 3.2\text{ mA}$			0.4	V
Receiver positive-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$		1.7	2.4	V
Receiver negative-going input threshold voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$	0.8	1.2		V
Input hysteresis voltage	R1IN, R2IN	$V_{CC} = 5\text{ V}$	0.2	0.5	1	V
Receiver input resistance	R1IN, R2IN	$V_{CC} = 5, T_A = 25^\circ\text{C}$	3	5	7	$\text{k}\Omega$

† Typical values are at  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$ .

‡ Algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage only.

§ Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

Timing characteristics,  $V_{CC} = 5\text{ V}, T_A = 25^\circ\text{C}$  (see Note 4 and Figure 1)

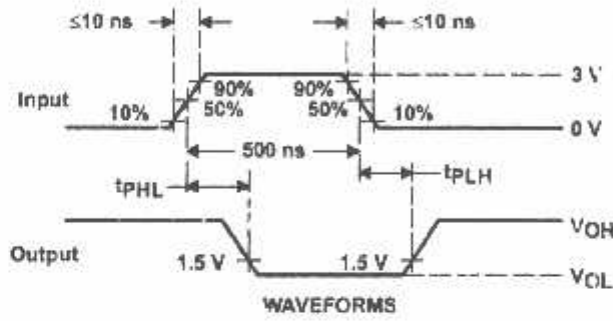
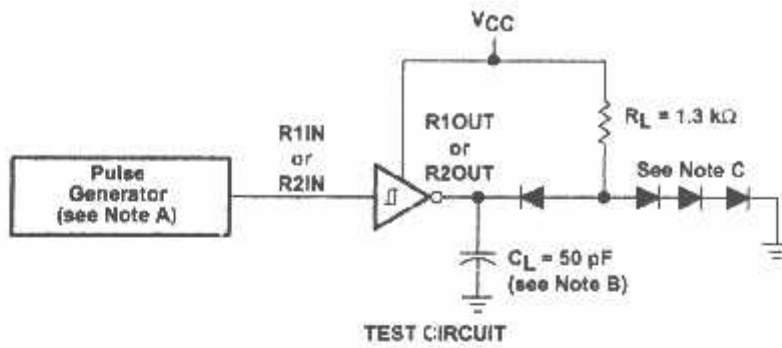
PARAMETER	TYP	UNIT
Receiver propagation delay time, low- to high-level output	500	ns
Receiver propagation delay time, high- to low-level output	500	ns

† Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .



POST OFFICE BOX 655303 • DALLAS, TEXAS 75285

PARAMETER MEASUREMENT INFORMATION

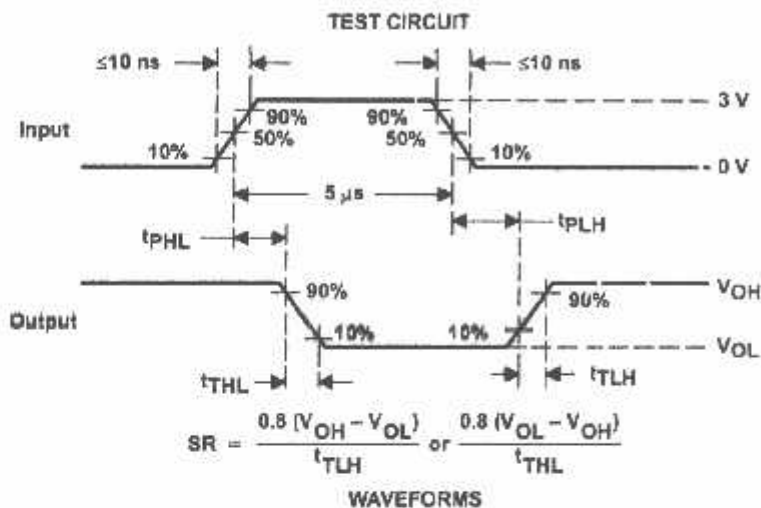
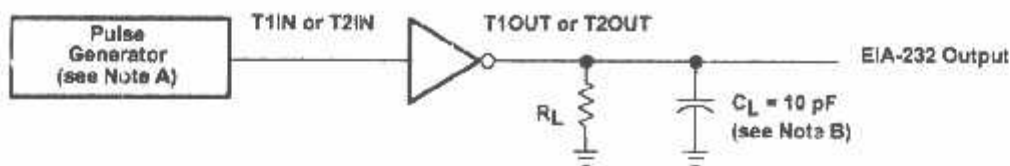


- NOTES: A. The pulse generator has the following characteristics:  $Z_0 = 50 \Omega$ , duty cycle  $\leq 50\%$ .  
 B.  $C_L$  includes probe and jig capacitance.  
 C. All diodes are 1N3064 or equivalent.

Figure 1. Receiver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements

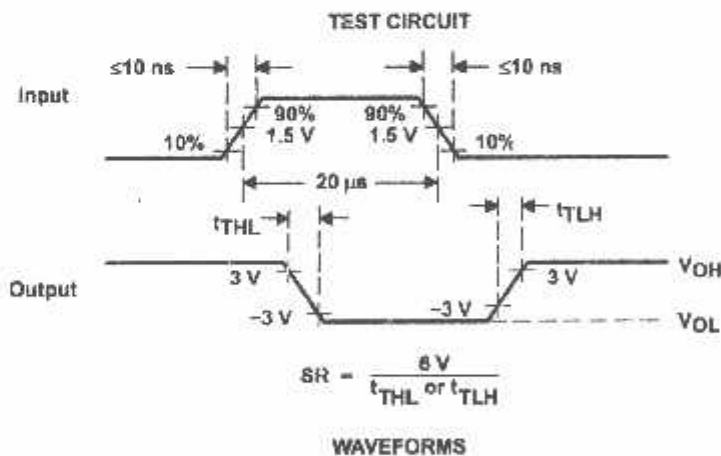
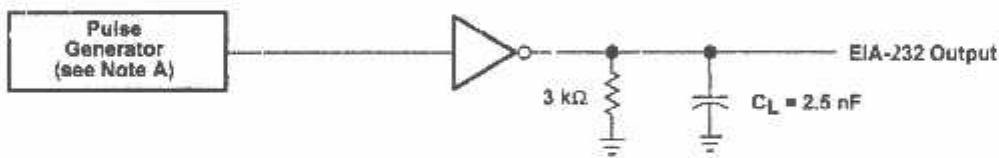


**PARAMETER MEASUREMENT INFORMATION**



- S: A. The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .
- B.  $C_L$  includes probe and jig capacitance

**Figure 2. Driver Test Circuit and Waveforms for  $t_{PHL}$  and  $t_{PLH}$  Measurements (5- $\mu$ s Input)**

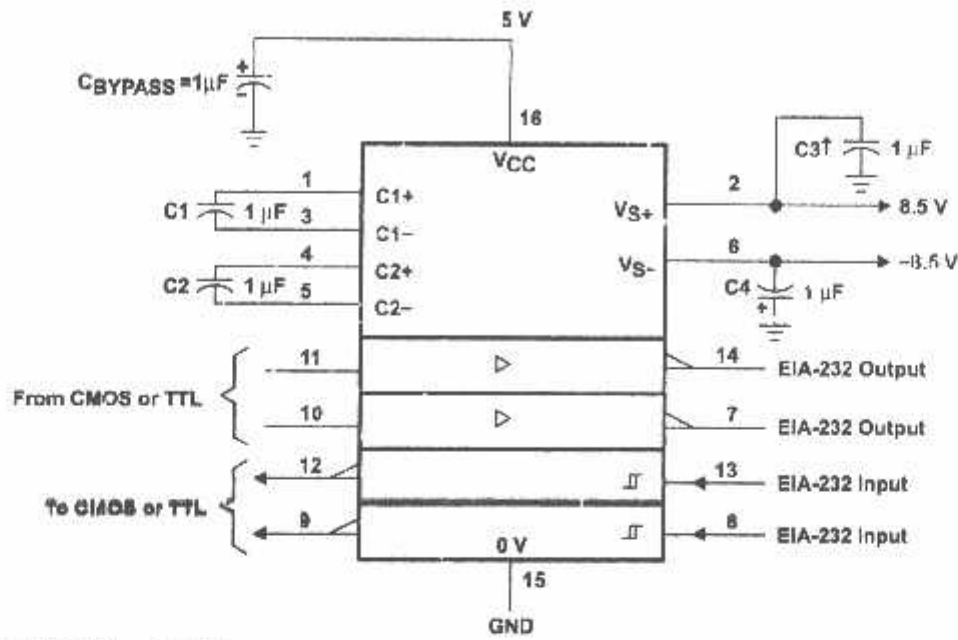


- A: The pulse generator has the following characteristics:  $Z_O = 50 \Omega$ , duty cycle  $\leq 50\%$ .

**Figure 3. Test Circuit and Waveforms for  $t_{THL}$  and  $t_{TLH}$  Measurements (20- $\mu$ s Input)**



APPLICATION INFORMATION



† C3 can be connected to  $V_{CC}$  or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1- $\mu$ F capacitors shown, the MAX202 can operate with 0.1- $\mu$ F capacitors.

Figure 4. Typical Operating Circuit

**PACKAGING INFORMATION**

Orderable Device	Status <sup>(1)</sup>	Package Type	Package Drawing	Pins	Package Qty	Eco Plan <sup>(2)</sup>	Lead/Ball Finish	MSL Peak Temp <sup>(3)</sup>
MAX232D	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232ID	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWG4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRG4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IN	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232INE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232N	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NSR	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232NSRE4	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM

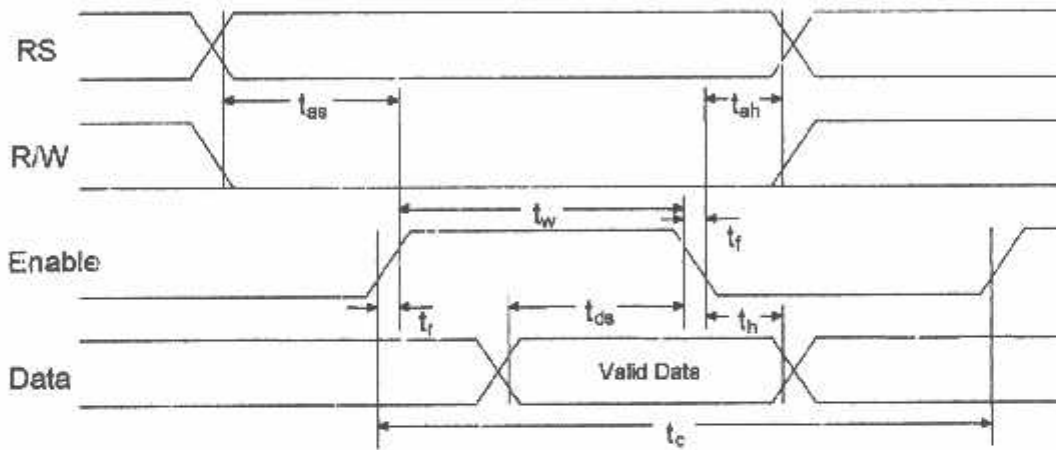
The marketing status values are defined as follows:

**DATASHEET**  
**LCD M1632**

---

Instruction	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Description	Clocks
No Operation	0	0	0	0	0	0	0	0	0	0	No Operation	0
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears display & sets address counter to zero.	165
Cursor Home	0	0	0	0	0	0	0	0	1	0	Sets address counter to zero, returns shifted display to original position. DDRAM contents remains unchanged.	3
Cursor Mode Set	0	0	0	0	0	0	0	1	L/D	S	Sets cursor move direction, and specifies automatic shift.	3
Display Control	0	0	0	0	0	0	1	D	C	B	Turns display (D), cursor on/off (C) or cursor blinking (B).	3
Cursor/display shift	0	0	0	0	0	1	S/C	R/L	0	0	Moves cursor and shift display. DDRAM contents remains unchanged.	3
Function Set	0	0	0	0	1	DL	N	M	G	0	Sets interface data width (DL), number of display lines (N,M) and voltage generator control (G).	3
CGRAM Addr	0	0	0	1	Character Generator RAM						Sets CGRAM Address	3
DDRAM Addr	0	0	1	Display Data RAM Address						Sets DDRAM Address	3	
Busy Flag & Addr	0	1	BF	Address Counter						Reads Busy Flag & Address Counter	0	
Read Data	1	0	Read Data						Reads data from CGRAM or DDRAM	3		
Write Data	1	1	Write Data						Writes data from CGRAM or DDRAM	3		

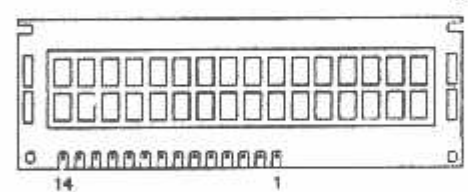
Write Cycle



Parameter	Symbol	Min <sup>(1)</sup>	Typ <sup>(1)</sup>	Max <sup>(1)</sup>	Unit
Cycle Time	$t_c$	500	-	-	ns
Pulse Width	$t_w$	230	-	-	ns
Rise/Fall Time	$t_r, t_f$	-	-	20	ns
RS Setup Time	$t_{ss}$	40	-	-	ns
RS Hold Time	$t_{sh}$	10	-	-	ns
Data Setup Time	$t_{ds}$	80	-	-	ns
Data Hold Time	$t_{dh}$	10	-	-	ns

The above specifications are a indication only. Timing will vary from manufacturer to manufacturer.

A 2 line by 16 Character LCD Module is Pictured. Data will work on most 1 line x character, 1 line x 20 character, 2 line x 16 character, 2 line x 20 character, 4 lines x character, 2 lines x 40 character etc. modules compatible with the HD44780 LCD module.



Pin No	Name	I/O	Description
1	Vss	Power	GND
2	Vdd	Power	+5v
3	V0	Analog	Contrast Control
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	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

**DATASHEET**  
**TRANSISTOR A733**

---

## PNP general purpose transistor

2PA733

## FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 50 V).

## APPLICATIONS

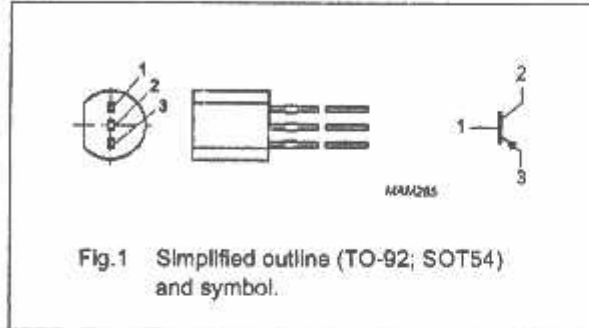
- General purpose switching and amplification.

## DESCRIPTION

PNP transistor in a TO-92 (SOT54) plastic package.  
NPN complement: 2PC945.

## PINNING

PIN	DESCRIPTION
1	base
2	collector
3	emitter



## LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
$V_{CB0}$	collector-base voltage	open emitter	-	-60	V
$V_{CE0}$	collector-emitter voltage	open base	-	-50	V
$V_{EB0}$	emitter-base voltage	open collector	-	-5	V
$I_C$	collector current (DC)		-	-100	mA
$I_{CM}$	peak collector current		-	-200	mA
$I_{BM}$	peak base current		-	-100	mA
$P_{tot}$	total power dissipation	$T_{amb} \leq 25^\circ\text{C}$ ; note 1	-	500	mW
$T_{stg}$	storage temperature		-65	+150	$^\circ\text{C}$
$T_J$	junction temperature		-	150	$^\circ\text{C}$
$T_{amb}$	operating ambient temperature		-65	+150	$^\circ\text{C}$

## Note

1. Transistor mounted on an FR4 printed-circuit board.

PNP general purpose transistor

2PA733

THERMAL CHARACTERISTICS

SYMBOL	PARAMETER	CONDITIONS	VALUE	UNIT
$R_{th(j-a)}$	thermal resistance from junction to ambient	note 1	250	K/W

Note

1. Transistor mounted on an FR4 printed-circuit board.

CHARACTERISTICS

$T_j = 25\text{ }^\circ\text{C}$  unless other specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
$I_{CBO}$	collector cut-off current	$I_E = 0; V_{CB} = -60\text{ V}$	-	-	-100	nA
$I_{EBO}$	emitter cut-off current	$I_C = 0; V_{EB} = -5\text{ V}$	-	-	-100	nA
$h_{FE}$	DC current gain 2PA733P	$I_C = -1\text{ mA}; V_{CE} = -6\text{ V}$	200	-	400	
$V_{CEsat}$	collector-emitter saturation voltage	$I_C = -100\text{ mA}; I_B = -10\text{ mA}$	-	-	-300	mV
$V_{BE}$	base-emitter voltage	$I_C = -1\text{ mA}; V_{CE} = -6\text{ V}$	-600	-	-700	mV
$C_c$	collector capacitance	$I_E = I_B = 0; V_{CB} = -10\text{ V}; f = 1\text{ MHz}$	-	4.5	6	pF
$C_e$	emitter capacitance	$I_C = I_C = 0; V_{EB} = -0.5\text{ V}; f = 1\text{ MHz}$	-	10	-	pF
$f_T$	transition frequency	$I_C = -10\text{ mA}; V_{CE} = -6\text{ V}; f = 100\text{ MHz}$	100	180	-	MHz
F	noise figure	$I_C = -200\text{ }\mu\text{A}; V_{CE} = -5\text{ V}; R_S = 2\text{ k}\Omega;$ $f = 1\text{ kHz}; B = 100\text{ Hz}$	-	-	10	dB



# DATASHEET

## ATMEGA16



## Features

High performance, Low-power AVR<sup>®</sup> 8-bit Microcontroller  
RISC Architecture  
31 Powerful Instructions – Most Single-clock Cycle Execution  
2 x 8 General Purpose Working Registers  
Fully Static Operation  
Up to 16 MIPS Throughput at 16 MHz  
On-chip 2-cycle Multiplier  
Volatile Program and Data Memories  
6K Bytes of In-System Self-Programmable Flash  
Endurance: 10,000 Write/Erase Cycles  
Optional Boot Code Section with Independent Lock Bits  
In-System Programming by On-chip Boot Program  
True Read-While-Write Operation  
128 Bytes EEPROM  
Endurance: 100,000 Write/Erase Cycles  
2K Byte Internal SRAM  
Programming Lock for Software Security  
(IEEE std. 1149.1 Compliant) Interface  
Boundary-scan Capabilities According to the JTAG Standard  
Extensive On-chip Debug Support  
Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface  
Special Features  
Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes  
One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode  
Real Time Counter with Separate Oscillator  
Four PWM Channels  
One Channel, 10-bit ADC  
8 Single-ended Channels  
7 Differential Channels in TQFP Package Only  
2 Differential Channels with Programmable Gain at 1x, 10x, or 200x  
Host-oriented Two-wire Serial Interface  
Programmable Serial USART  
Master/Slave SPI Serial Interface  
Programmable Watchdog Timer with Separate On-chip Oscillator  
On-chip Analog Comparator  
Microcontroller Features  
Power-on Reset and Programmable Brown-out Detection  
External Calibrated RC Oscillator  
Internal and External Interrupt Sources  
Low Power Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby  
and Extended Standby  
Packages  
Programmable I/O Lines  
8-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF  
Operating Voltages  
V<sub>DD</sub> = 1.8V to 5.5V for ATmega16L  
V<sub>DD</sub> = 1.8V to 5.5V for ATmega16  
Grades  
8 MHz for ATmega16L  
16 MHz for ATmega16  
Consumption @ 1 MHz, 3V, and 25°C for ATmega16L  
Active: 1.1 mA  
Sleep Mode: 0.35 mA  
Power-down Mode: < 1 µA



8-bit AVR<sup>®</sup>  
Microcontroller  
with 16K Bytes  
In-System  
Programmable  
Flash

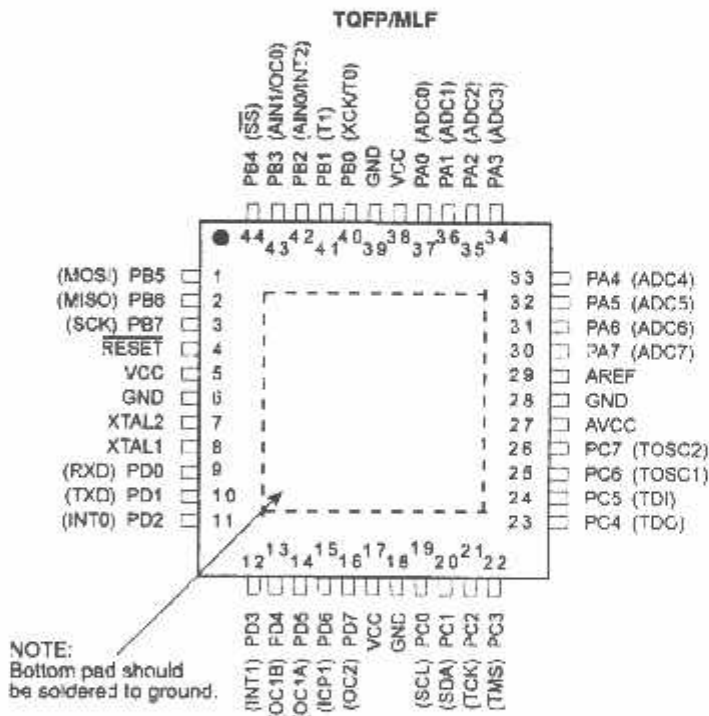
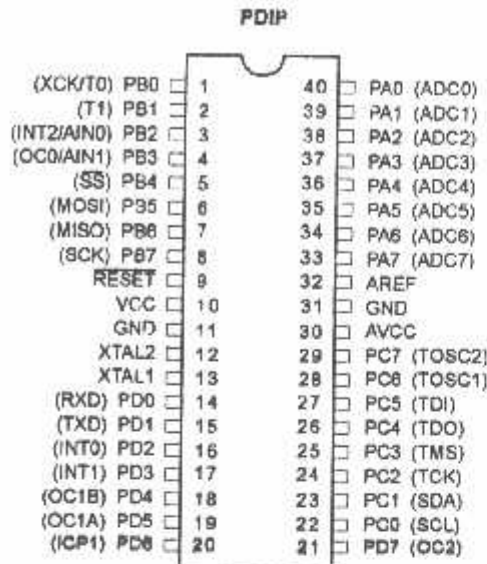
ATmega16  
ATmega16L

2466L-AVR-06/05



# Configurations

Figure 1. Pinout ATmega16



## Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

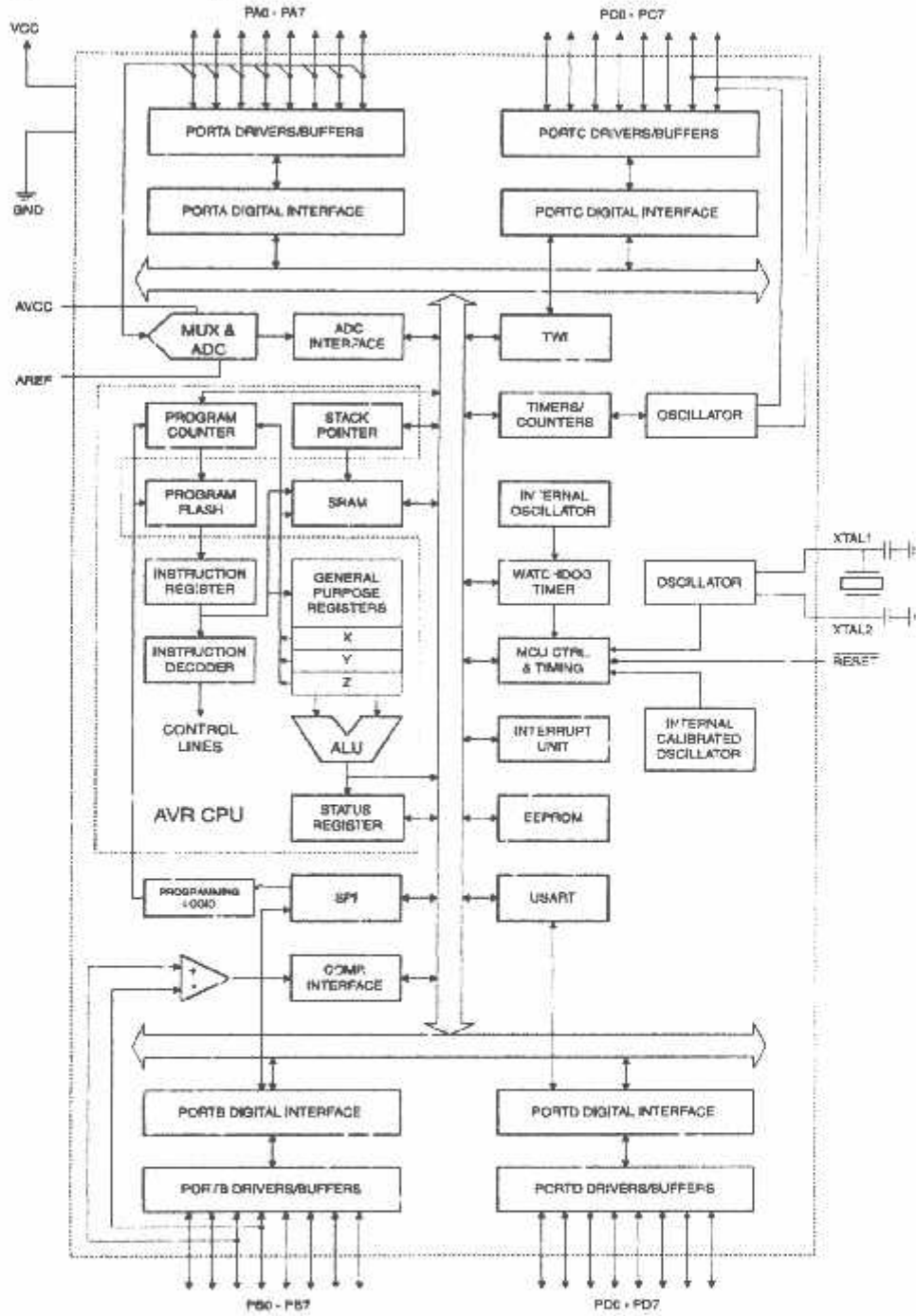
# ATmega16(L)

## view

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

## Diagram

Figure 2. Block Diagram





The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## Descriptions

Digital supply voltage.

Ground.

### PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

## ATmega16(L)

2466L-AVR-06/05

**(PB7..PB0)**

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega16 as listed on page 56.

**(PC7..PC0)**

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 59.

**(PD7..PD0)**

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega16 as listed on page 61.

**Reset Input.** A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 36. Shorter pulses are not guaranteed to generate a reset.

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

Output from the inverting Oscillator amplifier.

**AVCC** is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

**AREF** is the analog reference pin for the A/D Converter.

**IRCS**

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

**: Code  
ples**

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

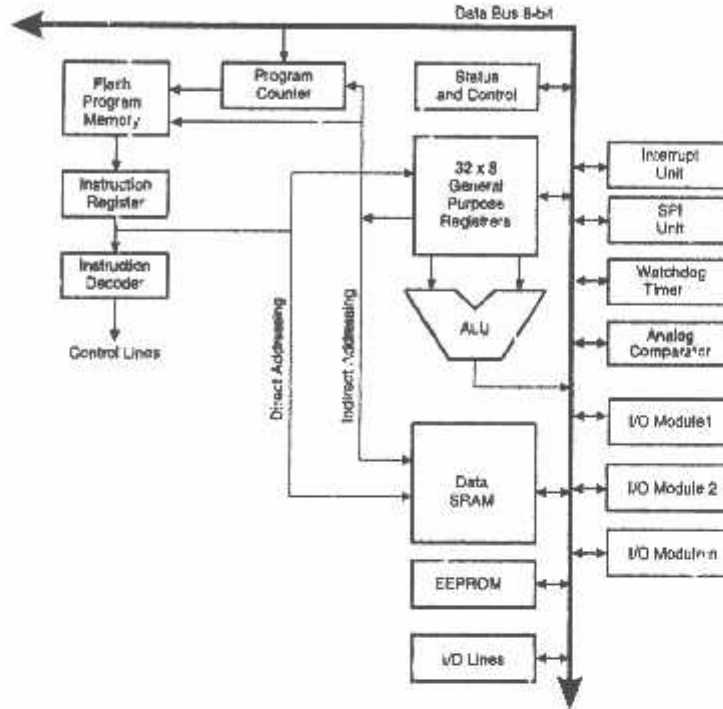
## CPU Core

### Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

### Structural Overview

Figure 3. Block Diagram of the AVR MCU Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After



an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, \$20 - \$5F.

## Arithmetic Logic

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

## Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	



- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the Instruction Set Reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy Instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit,  $S = N \oplus V$**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

## General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC Instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4 shows the structure of the 32 general purpose working registers in the CPU.

**Figure 4. AVR CPU General Purpose Working Registers**

	7	0	Addr.	
General Purpose Working Registers	R0		\$00	
	R1		\$01	
	R2		\$02	
	...			
	R13		\$0D	
	R14		\$0E	
	R15		\$0F	
	R16		\$10	
	R17		\$11	
	...			
	R26		\$1A	X-register Low Byte
	R27		\$1B	X-register High Byte
	R28		\$1C	Y-register Low Byte
	R29		\$1D	Y-register High Byte
	R30		\$1E	Z-register Low Byte
	R31		\$1F	Z-register High Byte

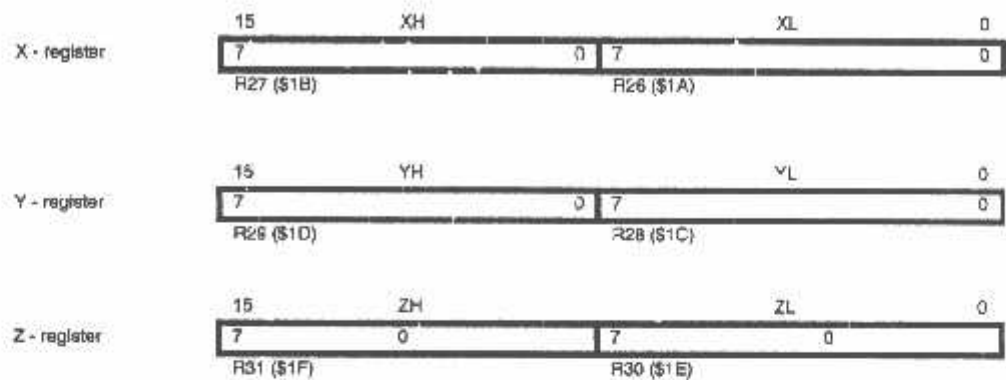
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-pointer Registers can be set to index any register in the file.

**register, Y-register and Z-register**

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as described in Figure 5.

**Figure 5.** The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the Instruction Set Reference for details).

**Pointer**

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer. If software reads the Program Counter from the Stack after a call or an interrupt, unused bits (15:13) should be masked out.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above \$60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

## Instruction Execution

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock  $clk_{CPU}$ , directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 6 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

**Figure 6. The Parallel Instruction Fetches and Instruction Executions**

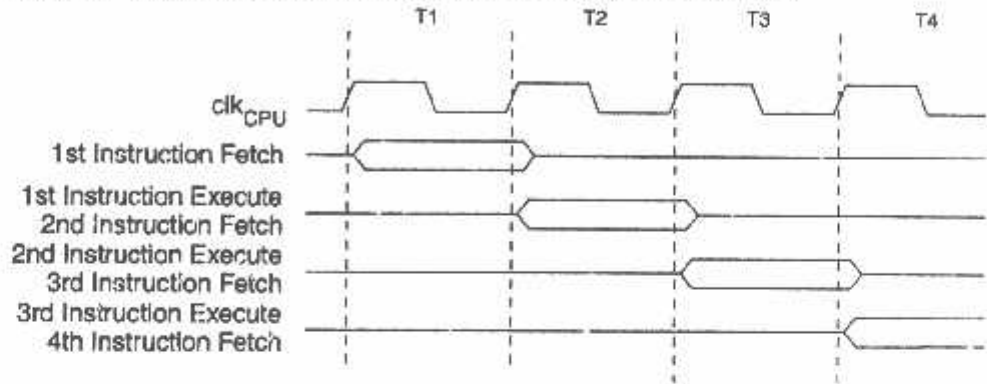
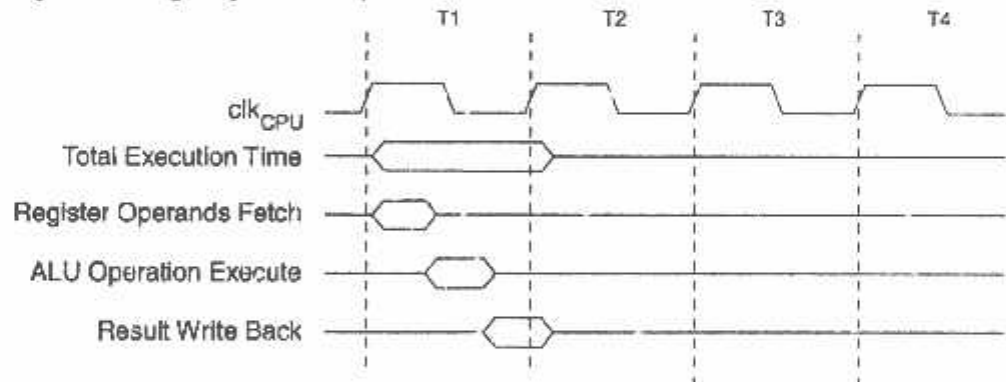


Figure 7 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

**Figure 7. Single Cycle ALU Operation**



## Interrupt

The AVR provides several different interrupt sources. These interrupts and the separate reset vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 259 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 43. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0

– the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to "Interrupts" on page 43 for more information. The Reset Vector can also be moved to the start of the boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 246.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the global interrupt enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example
<pre> in r16, SREG      ; store SREG value cli              ; disable interrupts during timed sequence sbi EECR, EEMWE  ; start EEPROM write sbi EECR, EEWE out SREG, r16    ; restore SREG value (I-bit) </pre>
C Code Example
<pre> char cSREG; cSREG = SREG; /* store SREG value */ /* disable interrupts during timed sequence */ _cli(); EECR  = (1&lt;&lt;EEMWE); /* start EEPROM write */ EECR  = (1&lt;&lt;EEWE); SREG = cSREG; /* restore SREG value (I-bit) */ </pre>

## ATmega16 Features

This section describes the different memories in the ATmega16. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega16 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

### System In-System Reprogrammable Flash Program Memory

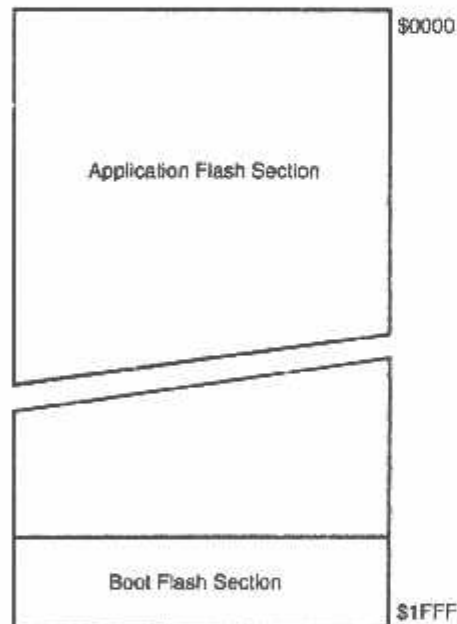
The ATmega16 contains 16K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 8K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega16 Program Counter (PC) is 13 bits wide, thus addressing the 8K program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 246. "Memory Programming" on page 259 contains a detailed description on Flash data serial downloading using the SPI pins or the JTAG interface.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory Instruction Description).

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 11.

Figure 8. Program Memory Map



**Data Memory**

Figure 9 shows how the ATmega16 SRAM Memory is organized.

The lower 1120 Data Memory locations address the Register File, the I/O Memory, and the internal data SRAM. The first 96 locations address the Register File and I/O Memory, and the next 1024 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

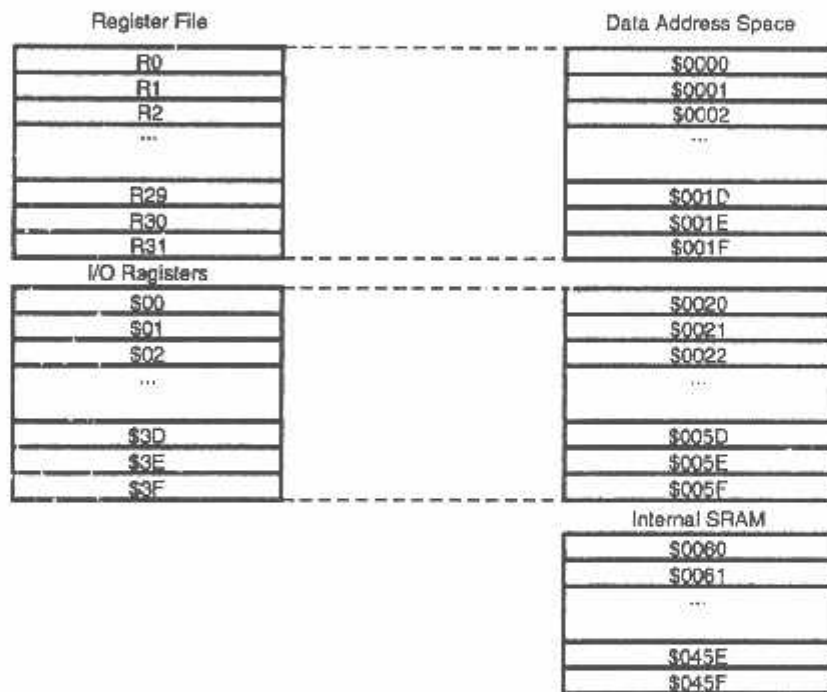
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, and the 1024 bytes of internal data SRAM in the ATmega16 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 9.

**Figure 9. Data Memory Map**

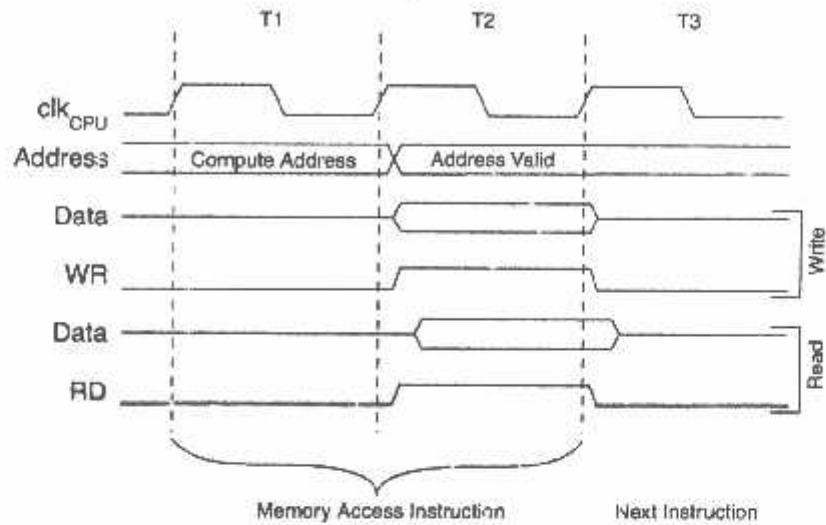




## Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two  $clk_{CPU}$  cycles as described in Figure 10.

**Figure 10.** On-chip Data SRAM Access Cycles



## EEPROM Data Memory

The ATmega16 contains 512 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For a detailed description of SPI, JTAG, and Parallel data downloading to the EEPROM, see page 273, page 278, and page 262, respectively.

### EEPROM Read/Write Access:

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 1. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies,  $V_{CC}$  is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 20 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.



## EEPROM Address

32 -- EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	EEAR8	EEARH
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	X	
	X	X	X	X	X	X	X	X	

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bits 8..0 – EEAR8..0: EEPROM Address**

The EEPROM Address Registers - EEARH and EEARL - specify the EEPROM address in the 512 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 511. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

## EEPROM Data Register –

Bit	7	6	5	4	3	2	1	0	
	MSB							LSS	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..0 – EEDR7..0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEAR Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

## EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EWE	EERE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

- **Bits 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16 and will always read as zero.

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EWE is cleared.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EWE to one causes the EEPROM to be written. When EEMWE is set, setting EWE within four clock cycles will write data to the EEPROM at the selected address. If EEMWE is zero, setting EWE will have no effect.

When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEMWE bit for an EEPROM write procedure.

• **Bit 1 – EEMWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEMWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEMWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEMWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEMWE becomes zero.
2. Wait until SPMEN in SPMCR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMWE bit while writing a zero to EEMWE in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEMWE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See "Boot Loader Support – Read-While-Write Self-Programming" on page 246 for details about boot programming.

Caution: An interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM Access, the EEAR or EEDR register will be modified, causing the interrupted EEPROM Access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEMWE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEMWE has been set, the CPU is halted for two cycles before the next instruction is executed.

• **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal – EERE – is the read strobe to the EEPROM. When the correct address is set up in the EEAR Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEMWE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEAR Register.

The calibrated Oscillator is used to time the EEPROM accesses. Table 1 lists the typical programming time for EEPROM access from the CPU.

**Table 1. EEPROM Programming Time**

Symbol	Number of Calibrated RC Oscillator Cycles <sup>(1)</sup>	Typ Programming Time
EEPROM write (from CPU)	8448	8.5 ms

Note: 1. Uses 1 MHz clock, independent of CKSEL Fuse setting.

The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (for example by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

#### Assembly Code Example

```
EEPROM_write:
    ; Wait for completion of previous write
    sbic EECR,EEWE
    rjmp EEPROM_write
    ; Set up address (r18:r17) in address register
    out EEARH, r18
    out EEARL, r17
    ; Write data (r16) to data register
    out EEDR,r16
    ; Write logical one to EEMWE
    sbi EECR,EEMWE
    ; Start eeprom write by setting EEWE
    sbi EECR,EEWE
    ret
```

#### C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    /* Wait for completion of previous write */
    while((EECR & (1<<EEWE))
        +
        /* Set up address and data registers */
        EEAR = uiAddress;
        EEDR = ucData;
        /* Write logical one to EEMWE */
        EECR |= (1<<EEMWE);
        /* Start eeprom write by setting EEWE */
        EECR |= (1<<EEWE);
    )
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

#### Assembly Code Example

```
EEPROM_read:
    ; Wait for completion of previous write
    sbic EECR,EEWE
    rjmp EEPROM_read
    ; Set up address (r18:r17) in address register
    out  EEARH, r18
    out  EARL, r17
    ; Start eeprom read by writing EERE
    sbi  EECR,EERE
    ; Read data from data register
    in   r16,EEDR
    ret
```

#### C Code Example

```
unsigned char EEPROM_read(unsigned int uiAddress)
{
    /* Wait for completion of previous write */
    while(EECR & (1<<EEWE))
        ;
    /* Set up address register */
    EEAR = uiAddress;
    /* Start eeprom read by writing EERE */
    EECR |= (1<<EERE);
    /* Return data from data register */
    return EEDR;
}
```

#### Write During Power-down Mode

When entering Power-down Sleep mode while an EEPROM write operation is active, the EEPROM write operation will continue, and will complete before the Write Access time has passed. However, when the write operation is completed, the Oscillator continues running, and as a consequence, the device does not enter Power-down entirely. It is therefore recommended to verify that the EEPROM write operation is completed before entering Power-down.

#### EEPROM Corruption

During periods of low  $V_{CC}$ , the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low  $V_{CC}$  Reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

## memory

The I/O space definition of the ATmega16 is shown in "Register Summary" on page 331.

All ATmega16 I/Os and peripherals are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions, transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the Instruction Set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O Registers as data space using LD and ST instructions, \$20 must be added to these addresses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.

The I/O and Peripherals Control Registers are explained in later sections.

## Electrical Characteristics

### Absolute Maximum Ratings\*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground	-0.5V to +13.0V
Maximum Operating Voltage	6.0V
Current per I/O Pin	40.0 mA
Current $V_{CC}$ and GND Pins	200.0mA PDIP and TQFP/MLF

\*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 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.

### Electrical Characteristics

0°C to 85°C,  $V_{CC} = 2.7V$  to 5.5V (Unless Otherwise Noted)

Parameter	Condition	Min	Typ	Max	Units
Input Low Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC}=2.7 - 5.5$	-0.5		$0.2 V_{CC}^{(1)}$	V
Input High Voltage except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC}=2.7 - 5.5$	$0.6 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Input High Voltage XTAL1 pin	$V_{CC}=2.7 - 5.5$	$0.7 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Input Low Voltage XTAL1 pin	$V_{CC}=2.7 - 5.5$	-0.5		$0.1 V_{CC}^{(1)}$	V
Input High Voltage $\overline{\text{RESET}}$ pin	$V_{CC}=2.7 - 5.5$	$0.9 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
Input Low Voltage $\overline{\text{RESET}}$ pin	$V_{CC}=2.7 - 5.5$	-0.5		$0.2 V_{CC}$	V
Output Low Voltage <sup>(3)</sup> (Ports A,B,C,D)	$I_{OL} = 20 \text{ mA}, V_{CC} = 5V$ $I_{OL} = 10 \text{ mA}, V_{CC} = 3V$			0.7 0.5	V V
Output High Voltage <sup>(4)</sup> (Ports A,B,C,D)	$I_{OH} = -20 \text{ mA}, V_{CC} = 5V$ $I_{OH} = -10 \text{ mA}, V_{CC} = 3V$	4.2 2.2			V V
Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin low (absolute value)			1	$\mu A$
Input Leakage Current I/O Pin	$V_{CC} = 5.5V$ , pin high (absolute value)			1	$\mu A$
Reset Pull-up Resistor		30		60	$k\Omega$
I/O Pin Pull-up Resistor		20		50	$k\Omega$

0°C to 85°C,  $V_{CC} = 2.7V$  to  $5.5V$  (Unless Otherwise Noted) (Continued)

Parameter	Condition	Min	Typ	Max	Units
Power Supply Current	Active 1 MHz, $V_{CC} = 3V$ (ATmega16L)		1.1		mA
	Active 4 MHz, $V_{CC} = 3V$ (ATmega16L)		3.8	5	mA
	Active 8 MHz, $V_{CC} = 5V$ (ATmega16)		12	15	mA
	Idle 1 MHz, $V_{CC} = 3V$ (ATmega16L)		0.35		mA
	Idle 4 MHz, $V_{CC} = 3V$ (ATmega16L)		1.2	2	mA
	Idle 8 MHz, $V_{CC} = 5V$ (ATmega16)		5.5	7	mA
Power-down Mode <sup>(6)</sup>	WDT enabled, $V_{CC} = 3V$		<8	15	$\mu A$
	WDT disabled, $V_{CC} = 3V$		<1	4	$\mu A$
Analog Comparator Input Offset Voltage	$V_{CC} = 5V$ $V_{in} = V_{CC}/2$			40	mV
Analog Comparator Input Leakage Current	$V_{CC} = 5V$ $V_{in} = V_{CC}/2$	-50		50	nA
Analog Comparator Propagation Delay	$V_{CC} = 2.7V$		750		ns
	$V_{CC} = 4.0V$		500		

"Max" means the highest value where the pin is guaranteed to be read as low

"Min" means the lowest value where the pin is guaranteed to be read as high

Although each I/O port can sink more than the test conditions (20 mA at  $V_{CC} = 5V$ , 10 mA at  $V_{CC} = 3V$ ) under steady state conditions (non-transient), the following must be observed:

PDIP Package:

- 1) The sum of all IOL, for all ports, should not exceed 200 mA.
- 2) The sum of all IOL, for port A0 - A7, should not exceed 100 mA.
- 3) The sum of all IOL, for ports B0 - B7, C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.

TQFP and QFN/MLF Package:

- 1) The sum of all IOL, for all ports, should not exceed 400 mA.
- 2) The sum of all IOL, for ports A0 - A7, should not exceed 100 mA.
- 3) The sum of all IOL, for ports B0 - B4, should not exceed 100 mA.
- 4) The sum of all IOL, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.
- 5) The sum of all IOL, for ports D3 - D7, should not exceed 100 mA.
- 6) The sum of all IOL, for ports C0 - C7, should not exceed 100 mA.

If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.

Although each I/O port can source more than the test conditions (20 mA at  $V_{CC} = 5V$ , 10 mA at  $V_{CC} = 3V$ ) under steady state conditions (non-transient), the following must be observed:

PDIP Package:

- 1) The sum of all IOH, for all ports, should not exceed 200 mA.
- 2) The sum of all IOH, for port A0 - A7, should not exceed 100 mA.
- 3) The sum of all IOH, for ports B0 - B7, C0 - C7, D0 - D7 and XTAL2, should not exceed 100 mA.

TQFP and QFN/MLF Package:

- 1) The sum of all IOH, for all ports, should not exceed 400 mA.
- 2) The sum of all IOH, for ports A0 - A7, should not exceed 100 mA.
- 3) The sum of all IOH, for ports B0 - B4, should not exceed 100 mA.
- 4) The sum of all IOH, for ports B3 - B7, XTAL2, D0 - D2, should not exceed 100 mA.



# **AT – Command Set For Siemens**

---



**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 6.0.0
Test command <b>AT+CSMS=?</b>	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 +CNM=1,2 or +CNM=1,3 is not activated. If necessary, the latter should be deactivated first.
Read command <b>AT+CSMS?</b>	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> 0 Broadcast type messages Type not supported
Write command <b>AT+CSMS=&lt;service&gt;</b>	Parameter <service> 0 GSM 3.40 and 3.41  Response +CSMS: <mt>,<mo>,<bm> OK/ERROR/+CMS ERROR

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

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

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

<b>AT+CNMI</b>	<b>Display new incoming SMS Revision according to GSM 07.05 Version 4.7.0</b>
Test command <b>AT+CNMI=?</b>	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 <bf>s) Parameter <mode> 0       Buffers unexpected messages (but is equivalent to rejecting; see <bit>) 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 S25f) 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 +CMTI: <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.  NOTE: <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>
<p>Read command</p> <p>AT+CNMI?</p>	<p>Response</p> <p>+CNMI: &lt;mode&gt;, &lt;mt&gt;, &lt;bm&gt;, &lt;ds&gt;, &lt;bfr&gt;</p> <p>Parameter</p> <p>&lt;mode&gt; See Test command</p> <p>&lt;mt&gt; See Test command</p> <p>&lt;bm&gt; See Test command</p> <p>&lt;ds&gt; See Test command</p> <p>&lt;bfr&gt; See Test command</p>
<p>Write command</p> <p>AT+CNMI= [&lt;mode&gt; [, &lt;mt&gt;[, &lt;bm&gt; [, &lt;ds&gt;[, &lt;bfr&gt;]]]] ]</p>	<p>Parameter</p> <p>&lt;mode&gt; See Test command</p> <p>&lt;mt&gt; See Test command</p> <p>&lt;bm&gt; See Test command</p> <p>&lt;ds&gt; See Test command</p> <p>&lt;bfr&gt; See Test command</p> <p>Response</p> <p>OK/ERROR/+CMS ERROR</p>
	<p>Unexpected message</p> <p>+CMT: &lt;mem&gt; &lt;index&gt; Indication that new message has arrived</p> <p>+CMT: ., &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>

AT+CNMA	<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>
<p>Test command</p> <p>AT+CNMA=?</p>	<p>Response</p> <p>+CNMA: (list of supported &lt;n&gt;s)</p> <p>Parameter</p> <p>&lt;n&gt; 0 Mode of functioning analogous to GSM 07.05 text mode</p>
<p>Write command</p> <p>AT+CNMA[=&lt;n&gt;]</p>	<p>Parameter</p> <p>&lt;n&gt; See Test command</p> <p>Response</p> <p>OK/ERROR/+CMS ERROR: &lt;err&gt;</p>

<p><b>AT+CMGL</b></p>	<p><b>List SMS</b>  <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;e)</p> <p>Parameter  &lt;stat&gt;</p> <table border="0"> <tr> <td style="padding-right: 20px;">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</b>  <b>[=&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;,&lt;alpha&gt;,&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 <b>&lt;index&gt;</b> Index of message in selected memory <mem1>  Response <b>If PDU mode (+CMGF=0) and command are successful:</b> <b>+CMGR: &lt;stat&gt;,[&lt;alpha&gt;],&lt;length&gt;&lt;CR&gt;&lt;LF&gt;&lt;pdu&gt;</b>  Parameter <b>&lt;pdu&gt;</b> Siehe "AT+CMGL" otherwise: <b>+CMS ERROR: &lt;err&gt;</b>

<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 <b>&lt;length&gt;</b> Length of PDU <b>&lt;pdu&gt;</b> See "AT+CMGL" <b>&lt;mr&gt;</b> Message reference  Response <b>If sending is successful:</b> <b>+CMGS: &lt;mr&gt;</b> <b>If sending is not successful:</b> <b>+CMS ERROR: &lt;err&gt;</b>

<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;toda&gt;]]</b>	Parameter <b>&lt;index&gt;</b> Index of message in selected memory <mem1> <b>&lt;da&gt;</b> Destination address in string format <b>&lt;toda&gt;</b> Format of destination address  <b>&lt;mr&gt;</b> Message reference Response <b>If sending is successful:</b> <b>+CMSS: &lt;mr&gt;</b> <b>If sending is not successful:</b> <b>+CMS ERROR: &lt;err&gt;</b>

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

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

<b>AT+CSCB</b>		<b>Select cell broadcast messages</b>	
Test command <b>AT+CSCB=?</b>	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>
Read command <b>AT+CSCB?</b>	Response +CSCB: <mode>,<mids>,<dcss>		
	Parameter <mode> <mids> <dcss>	See Test command String type; combinations of CBM message IDs String type; combinations of CBM data coding schemes	
Write command <b>AT+CSCB=[&lt; mode&gt;[,&lt;mids &gt;[,&lt;dcss&gt;]]]</b>			

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