

SKRIPSI

**PERANCANGAN DAN PEMBUATAN KWH METER DIGITAL
DENGAN PENGIRIMAN INFORMASI TAGIHAN DAN
PEMUTUSAN HUBUNGAN LISTRIK MELALUI SMS
DENGAN KENDALI PC**



Disusun Oleh :

MOCH. ASDIYUDIN

NIM: 02.17.136



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

SECRET

ATTENTION: SYSTEMS GROUP DATA CENTER MANAGER
FOR SYSTEMS GROUP DATA CENTER MANAGER
AND SYSTEMS GROUP DATA CENTER MANAGER
BY SYSTEMS GROUP

SECRET
SYSTEMS GROUP
NOV 17 1950

AMPHIBIOUS MARINE BATTALION
1-8 DISTRICT MARINE BATTALION
CENTRAL DISTRICT BATTALION
MARINE BATTALION (RESERVE) DISTRICT
SECRET

LEMBAR PERSETUJUAN



PERANCANGAN DAN PEMBUATAN KWH METER DIGITAL DENGAN PENGIRIMAN INFORMASI TAGIHAN DAN PEMUTUSAN HUBUNGAN LISTRIK MELALUI SMS DENGAN KENDALI PC PADA SERVER

SKRIPSI

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

Disusun Oleh :

MOCH. ASDIYUDIN
NIM : 02. 17. 136

Diperiksa dan Disetujui

Dosen Pembimbing I

A handwritten signature in black ink, appearing to read "M. Herbasuki", written over a horizontal line.

Ir. Teguh Herbasuki, MT
NIP. Y. 1038900209

Dosen Pembimbing II

A handwritten signature in black ink, appearing to read "I Komang Somawirata", written over a horizontal line.

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

Mengetahui

Ketua Jurusan Teknik Elektro S-1



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

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



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

**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : Moch. Asdiyudin
NIM : 02.17.136
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik Elektronika
Judul Skripsi : "PERANCANGAN DAN PEMBUATAN KWH METER
DIGITAL DENGAN PENGIRIMAN INFORMASI
TAGIHAN DAN PEMUTUSAN HUBUNGAN LISTRIK
MELALUI SMS DENGAN KENDALI PC"

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

Hari : Selasa

Tanggal : 18 Maret 2008

Dengan Nilai : 78,55 (B+) *Ref*

Panitia Ujian Skripsi



Ketua

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

Sekretaris

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

Anggota Penguji

Penguji Pertama

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

Penguji Kedua

(DR. Cahyo Crysdian, Msc)
NIP. 1030400412

PERANCANGAN DAN PEMBUATAN KWH METER DIGITAL DENGAN PENGIRIMAN INFORMASI TAGIHAN DAN PENUTUSAN HUBUNGAN LISTRIK MELALUI SMS DENGAN KENDALI PC

MOCH. ASDIYUDIN

Fakultas Teknologi Industri, Jurusan, Teknik Elektro, Konsentrasi
Teknik Elektronika S-1, Institut Teknologi Nasional, Jl. Raya
Karanglo KM. 2 Malang. Ipensk_02@yahoo.Com

Abstrak

Sampai pada saat ini, PT. PLN. Dalam mengukur konsumsi energi listrik pada sebagian besar pelanggannya, masih menggunakan kWh meter analog atau biasa disebut kWh meter elektromekanik. Salah satu kelemahannya adalah dalam pembacaan hasil pengukuran energi, karena petugas pembaca meter harus mendatangi satu persatu pada alat tersebut dan mencatatnya secara manual. Hal ini akan berdampak pada pelayanan pelanggan, karena seringnya terjadi kesalahan data.

Dengan melihat dan mengamati fenomena yang terjadi di atas tadi maka penulis berkeinginan dan berinisiatif "Bagaimana membuat sebuah kWh meter digital yang bisa mengirimkan informasi tagihan dan pemutusan listrik melalui SMS dengan kendali PC pada server", selain itu perancangan ini merupakan salah satu cara untuk mengantisipasi adanya manipulasi biaya tarif listrik yang dilakukan secara sepihak oleh PLN yang menyebabkan salah pengertian antara pelanggan dengan pihak PLN.

Kata Kunci : Mikrokontroler ATmega8535, Trafo Arus, Trafo Tegangan, ,Max232, LCD.

Abstraction

So while this time PT. PLN In a measurement of electrical very often at meet a applications of electronics system but also still there using old system that is mechanical system and temporary superiority in mechanical system itself is a reliability of operational that durable but the weakness lays is the accurateness of it sometimes hesitated and also the performance of speed and precision, so that produce a points goes to still less ideal or accurate compared with electronics system which using software to increase performance from a system of device electronics.

By seeing the closely phenomenon above so that the author want to take the initiative " How to make a Digital kWh meter that send the claim information and severance of electricity by using sms with PC control in server ". Besides this planning is one manner to anticipate existence of cost manipulation of electricity tariff that done unilaterally by PLN causes misapprehension between customer with PLN.

Key word: Mikrokontroler ATmega8535, Trafo Arus, Trafo Tegangan, ,Max232, LCD.

KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya, sehingga dapat menyelesaikan skripsi yang berjudul “Perancangan dan Pembuatan kWh Meter Digital dengan Pengiriman Informasi Tagihan dan Pemutusan Hubungan Listrik melalui SMS dengan Kendali PC” ini dengan lancar. Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Elektronika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

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

1. Bapak Prof. DR. Ir. Abraham Lomi, MSEE selaku Rektor ITN Malang.
2. Bapak Ir. F. Yudi Limpraptono, MT selaku Ketua Jurusan Teknik Elektro S-1.
3. Bapak Ir. Teguh Herbasuki, MT selaku Dosen Pembimbing I.
4. Bapak I Komang Somawirata, ST, MT selaku Dosen Pembimbing II.
5. Ayah dan Ibu serta keluarga besar yang telah memberikan do’a restu, dorongan, semangat, dan biaya.
6. Rekan-rekan mahasiswa/i Elektronika 2002 yang telah memberikan semangat dan motivasi dalam penyelesaian skripsi ini.
7. Semua pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini.

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

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

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

Malang, Maret 2008

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
ABSTRAKSI	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xiii
DAFTAR GRAFIK	xiv

BAB I. PENDAHULUAN

1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Tujuan Pembahasan.....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi	3
1.6. Sistematika Penulisan.....	5

BAB II. LANDASAN TEORI

2.1. Pengertian Daya dalam Rangkaian Arus Bolak-Balik	6
2.2. <i>Transformator</i>	10
2.2.1. Sensor Arus (<i>Transformator Arus</i>)	12
2.2.2. Sensor Tegangan (<i>Transformator Tegangan</i>).....	12

2.3. <i>Phase Detector</i>	13
2.4. Penguat Operasional	14
2.4.1. Penguat Tak Membalik	14
2.5. Mikrokontroler ATmega8535	15
2.5.1. Fungsi Pin Mikrokontroler ATmega8535	16
2.5.2. Arsitektur Mikrokontroler ATmega8535	17
2.5.3. Organisasi Memori	19
2.5.4. SRAM	20
2.5.5. Register I/O	20
2.5.6. <i>Reset dan Characteristics</i>	22
2.5.7. EEPROM	23
2.5.8. <i>Chanel Multiplexer Input ADC</i>	24
2.6. <i>IC MAX 232</i>	25
2.6.1. <i>Interface</i>	27
2.7. LCD (<i>Liquid Crystal Display</i>) M1632	28
2.7.1. <i>Sinyal Interface M1632</i>	32
2.8. Rangkaian <i>Swicth</i>	34
2.9. Telepon Seluler	34
2.9.1. Siemen M-35	35
2.9.2. Format Data SMS	36
2.9.3. Prinsip Kerja SMS (<i>Short Message Service</i>)	36
2.9.4. PDU untuk Kirim SMS ke <i>SMS-Centre</i>	37
2.9.5. PDU untuk Terima SMS dari <i>SMS-Centre</i>	43
2.9.6. <i>AT-Commend</i>	45

BAB III. PERANCANGAN DAN PEMBATAN ALAT

3.1. Pendahuluan	46
3.1.1. Spesifikasi Alat.....	46
3.1.2. Blok Diagram Keseluruhan Sistem	47
3.1.3. Proses Kerja Alat.....	49
3.2. Perancangan Perangkat Keras (<i>Hardware</i>)	51
3.2.1. Perancangan Sensor.....	51
3.2.1.1. Perancangan Sensor Arus	51
3.2.1.2. Perancangan Sensor Tegangan	52
3.2.1.3. Perancangan Sensor $\text{Cos}\phi$	53
3.2.2. Rangkaian Pengkondisi Sinyal (Pengubah AC ke DC).....	54
3.2.3. Rangkaian <i>Zero Crossing Detector</i>	56
3.2.4. Perancangan Mikrokontroler pada kWh Meter	58
3.2.4.1. Perancangan Penggunaan Port pada Mikrokontroler ATMega8535	58
3.2.4.2. Internal ADC pada Mikrokontroler ATMega8535	59
3.2.4.3. Perancangan Rangkaian Reset	60
3.2.5. Perancangan Rangkaian LCD	61
3.2.6. Rangkaian <i>Switch</i>	64
3.2.7. Perancangan Komunikasi Data <i>Handphone</i> dengan Mikrokontroler	66
3.2.8. <i>Push Button</i>	67
3.3. Perancangan Perangkat lunak (<i>Software</i>)	68
3.3.1. <i>Flowchart</i> pada kWh Meter.....	69

3.3.2. <i>Flowchart</i> untuk Program Delphi.....	70
---	----

BAB IV. PENGUJIAN DAN ANALISA SISTEM

4.1. Pengujian Rangkaian Sensor Tegangan dan Pengkondisi Sinyal.....	71
4.1.1. Tujuan.....	71
4.1.2. Alat dan Bahan	71
4.1.3. Langkah Pengujian	71
4.1.4. Hasil Pengujian.....	72
4.2. Pengujian Rangkaian Sensor Arus dan Pengkondisi Sinyal.....	74
4.2.1. Tujuan.....	74
4.2.2. Alat dan Bahan	74
4.2.3. Langkah Pengujian	74
4.2.4. Hasil Pengujian	75
4.3. Pengujian $\text{Cos}\phi$	78
4.3.1. Tujuan.....	78
4.3.2. Alat dan Bahan	78
4.3.3. Langkah Pengujian	78
4.3.4. Hasil Pengujian.....	78
4.4. Pengujian Perbandingan antara Kwh Meter Digital yang dirancang dengan Kwh Meter Digital Standart PLN Tipe DSM-IC030A Merk SAINT	79
4.4.1. Tujuan.....	79
4.4.2. Alat dan Bahan	80
4.4.3. Langkah Pengujian	80

4.4.4. Hasil Pengujian.....	80
4.5. Pengujian <i>Handphone</i>	82
4.5.1. Pembahasan	83
4.6. Pengujian Keseluruhan Sisitem.....	84
4.6.1. Pengujian Pengiriman Pemakaian Energi Listrik dan Biaya....	84
4.6.1.1. Tujuan	84
4.6.1.2. Peralatan yang Digunakan	85
4.6.1.3. Langkah Pengujian.....	85
4.6.2. Pengujian Penonaktifan dan Pengaktifan Beban Listrik	86
4.2.1. Langkah Pengujian.....	86

BAB V. PENUTUP

5.1. Kesimpulan.....	88
5.2. Saran-saran	89

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1.	Segitiga Daya	7
Gambar 2.2.	<i>Transformator</i>	10
Gambar 2.3.	IC TC8051BP	13
Gambar 2.4.	Polaritas Tegangan dan Arus untuk Penguat Tak Membalik	15
Gambar 2.5.	Konfigurasi Pin ATmega8535.....	16
Gambar 2.6.	Blok Diagram Mikrokontroler ATmega8535.....	18
Gambar 2.7.	Peta Memori ATmega8535	19
Gambar 2.8.	SDRAM dalam Organisasi Memori ATmega8535	20
Gambar 2.9.	<i>ADC Internal</i> Mikrokontroler ATmega8535	25
Gambar 2.10.	RS 232 (<i>MAXIM</i>).....	26
Gambar 2.11.	Skematik DB9.....	27
Gambar 2.12.	Modul LCD 2x16 Karakter.....	29
Gambar 2.13.	Mengirim dan Mengambil Data dari atau ke M1632	33
Gambar 2.14.	Konektor Siemen M-35	35
Gambar 3.1.	Blok Diagram Keseluruhan Sistem.....	47
Gambar 3.3.	Rangkaian Penyearah Gelombang Penuh	55
Gambar 3.4.	Rangkaian <i>Zero Crossing Detector</i>	57
Gambar 3.5.	Minimum Sistem ATmega8535	58
Gambar 3.6.	Perancangan Rangkaian <i>Power On Reset</i> ATmega8535	61
Gambar 3.7.	Rangkaian LCD	63
Gambar 3.8.	Rangkaian <i>Switch</i>	64

Gambar 3.9. Rangkaian <i>Interface</i> antara Handphone dengan Mikrokontroler ATMega8535	66
Gambar 3.10. Rangkaian <i>Push Button</i>	68
Gambar 3.11. <i>Flowchart</i> pada kWh Meter	69
Gambar 3.12. <i>Flowchart</i> pada Pusat untuk Data Pelanggan Listrik.....	70
Gambar 4.1. Rangkaian Pengujian Sensor Tegangan dan Pengkondisi Sinyal	72
Gambar 4.2. Mengukur Sensor Tegangan	73
Gambar 4.3. Tampilan pada <i>Voltmeter</i>	73
Gambar 4.4. Rangkaian Pengujian Sensor Arus dan Pengkondisi Sinyal	75
Gambar 4.5. Tampilan pada Ammeter.....	77
Gambar 4.6. Menguji Sensor Arus	77
Gambar 4.7. Tampilan pada Sensor Arus	77
Gambar 4.8. Tampilan pada <i>Oscilloscop</i>	79
Gambar 4.9. kWh Meter Digital Merk SAINT dengan Tipe DSM-IC030A.....	81
Gambar 4.10. kWh Meter Digital dalam Perancangan.....	82
Gambar 4.11. Hasil Pengujian Handphone.....	82
Gambar 4.12. Proses Penerima Pemakaian Energi Listrik dan Biaya.....	85
Gambar 4.13. Proses Pemutusan Aliran Listrik pada Beban.....	86
Gambar 4.14. Proses Menghidupkan Aliran Listrik pada Beban	87

DAFTAR TABEL

Tabel 2.1. Keterangan Pin IC TC5081BP	14
Tabel 2.2. <i>Reset dan Characteristics</i>	23
Tabel 2.3. Fungsi Pin-Pin RS232	27
Tabel 2.4. Pemilihan Register pada LCD M1632	28
Tabel 2.5. Fungsi Pin-Pin LCD	30
Tabel 2.6. Keterangan Konektor Siemen M-35.....	35
Tabel 2.7. Berbarapa Nomor SMS- <i>Centre National Code</i>	38
Tabel 2.8. Berbarapa Nomor SMS- <i>Centre International Code</i>	38
Tabel 2.9. Rumus Menghitung Waktu Validitas SMS	40
Tabel 2.10. Format SMS Kirim dalam PDU	43
Tabel 2.11. Format SMS Terima dalam PDU	44
Tabel 2.12. <i>AT Command</i> pada SMS	45
Tabel 4.1. Data Percobaan Sensor Tegangan	72
Tabel 4.2. Data Percobaan Sensor Arus	75
Tabel 4.3. Data Pengukuran $\cos\phi$	78
Tabel 4.4. Hasil Pengamatan	87

DAFTAR GRAFIK

Grafik 2.1. Faktor Daya Leading	8
Grafik 2.2. Faktor Daya Lagging	9
Grafik 3.1. Beda Fasa antara Tegangan (V) dan Arus (I)	53
Grafik 4.1. Pengujian Tegangan Sumber	73
Grafik 4.2. Pengujian Sensor Tegangan	74
Grafik 4.3. Pengujian Sensor Arus untuk Lampu Pijar.....	76
Grafik 4.4. Pengujian Sensor Arus untuk Lampu TL dan Pijar	76

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan ilmu pengetahuan dan teknologi yang semakin maju saat ini, seiring itu pula dengan perkembangan dunia industri terutama dibidang teknologi informasi dan elektronika membawa peranan besar bagi kehidupan manusia yaitu dengan berbagai macam penyempurnaan untuk mendapatkan hasil yang semaksimal mungkin, dari penekanan pada faktor kemudahan operasional serta kepraktisan membuat peralihan teknologi dari sistem analog ke sistem digital semakin meningkat, hal ini merupakan fenomena yang sangat menarik untuk terus dikaji dan dikembangkan sehingga nantinya dapat bermanfaat bagi khalayak umum sesuai dengan tuntutan pada bidang atau profesinya.

Pada saat ini PT PLN menggunakan kwh meter analog dan untuk pengambilan data penggunaan listrik untuk tiap bulannya harus mengandalkan tenaga manusia sebagai petugas pencatat kwh meter yang harus datang setiap bulan.

Dengan melihat dan mengamati fenomena yang terjadi di atas tadi maka penulis berkeinginan dan berinisiatif **“Bagaimana membuat sebuah kWh meter digital yang bisa mengirimkan informasi tagihan dan pemutusan listrik melalui SMS dengan kendali PC pada server”**, untuk pengiriman data pada alat ini yaitu dengan komunikasi melalui sms sehingga pengirimannya secara langsung dapat diterima oleh *server*. Setelah data dikirim oleh masing-masing

kwh meter dengan nomor yang berbeda maka secara langsung data akan disimpan di *database*.

Apabila pelanggan yang bersangkutan belum membayar tagihan listriknya sampai dengan tanggal yang sudah ditentukan, maka server akan mengirimkan SMS ke kwh meter pelanggan tersebut untuk memutuskan aliran listriknya. Apabila tagihan sudah dibayar maka aliran listrik akan disambung kembali.

Selain itu perancangan ini merupakan salah satu cara untuk mengantisipasi adanya manipulasi biaya tarif listrik yang dilakukan secara sepihak oleh PLN yang menyebabkan salah pengertian antara pelanggan dengan pihak PLN.

1.2. Rumusan Masalah

Mengacu pada permasalahan yang ada, maka dapat dibuat rumusan masalah sebagai berikut :

1. Bagaimana merancang dan membuat kWh meter digital yang bisa mengirimkan informasi tagihan dan pemutusan listrik melalui SMS dengan kendali PC pada server ?
2. Bagaimana membuat tampilan biaya tarif listrik pada kWh meter digital yang memudahkan pelanggan tahu berapa biaya yang harus dibayar pada bulan ini ?
3. Bagaimana memproses pengiriman informasi tagihan dan pemutusan listrik melalui SMS dengan kendali PC pada *server* ?

1.3. Tujuan Pembahasan

Adapun tujuan dari perancangan skripsi ini adalah : Merencanakan dan membuat kWh meter digital dengan pengiriman informasi tagihan dan pemutusan hubungan listrik melalui SMS dengan kendali PC.

1.4. Batasan Masalah

Dalam perancangan dan pembuatan alat ini maka perlu adanya pembatasan masalah, antara lain :

1. Dalam perancangan alat ini akan dibahas tentang perangkat keras (*hardware*), sedangkan untuk perangkat lunaknya (*software*) dibahas secara garis besarnya saja.
2. Alat ini dirancang untuk digunakan pada kWh meter 1 fasa dengan fasa pencatat tarif tunggal pada rumah tangga dengan batas daya 450 VA.
3. *Handphone* sebagai sarana komunikasi data antara *Client* dan *Server* dan dianggap jaringan GSM dalam kondisi stabil.
4. Tidak membahas *Catu Daya*.

1.5. Metodologi Pembahasan

Metodologi yang dipakai dalam pembuatan skripsi ini adalah:

1. Studi Literatur

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

2. *Field Research*

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

3. Perencanaan Dan Pembuatan Alat

Dalam pembuatan alat ini menggunakan konsep sebagai berikut :

- Perencanaan sistem secara keseluruhan (pembuatan blok diagram sistem).
- Mendeskripsikan fungsi dari masing-masing blok diagram.
- Membuat perangkat keras (*hardware*) dan perangkat lunaknya (*software*).

4. Pengujian Alat

Dengan melakukan pengujian perblok rangkaian dan kerja seluruh sistem pada alat tersebut.

5. Penyusunan Laporan Skripsi

Membuat laporan yang terdiri dari: Pendahuluan, Landasan Teori, Perencanaan dan Pembuatan Alat, Pengujian Alat dan Penutup.

1.6. Sistematika Pembahasan

Sistematika penyusunan skripsi dibagi menjadi beberapa bagian utama, yaitu:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi pembahasan, dan sistematika pembahasan.

BAB II Teori Penunjang

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat.

BAB III Perencanaan dan Pembuatan Alat

Perancangan dan perealisasiian alat yang meliputi spesifikasi, perencanaan blok diagram, prinsip kerja dan realisasi alat.

BAB IV Pengujian Alat

Memuat hasil pengujian terhadap alat yang telah direalisasikan.

BAB V Kesimpulan dan Saran

Memuat kesimpulan dan saran-saran.

BAB II

LANDASAN TEORI

2.1. Pengertian Daya Dalam Rangkaian Arus Bolak-Balik

Daya didalam rangkaian arus bolak-balik merupakan hasil perkalian tegangan dengan arus. Daya ini merupakan daya sesaat yang besarnya akan berubah sesuai dengan perubahan besar tegangan dan arus. Secara matematis daya sesaat dapat ditulis sebagai berikut : $P_{(t)} = V_{(t)} \times I_{(t)}$

Dimana:

$$P_{(t)} = \text{Daya Sesaat (Watt)}$$
$$V_{(t)} = \text{Tegangan Sesaat (Volt)}$$
$$I_{(t)} = \text{Arus Sesaat (Ampere)}$$

Dalam rangkaian arus bolak balik terdapat beberapa besaran daya yaitu :

- Daya aktif atau daya nyata adalah daya rata-rata dari hasil perkalian antara tegangan sesaat dan arus sesaat, merupakan daya yang diserap oleh beban yang besarnya $V.I.\cos \varphi$.

$$P = \text{daya nyata (kW)}$$
$$= V \cdot I \cdot \cos \varphi$$

- Daya reaktif (Q) merupakan daya yang diserap oleh beban yang mempunyai reaktansi (reaktansi induktif dan reaktansi kapasitif) yang besarnya $V.I.\sin \varphi$.

$$Q = \text{daya reaktif (kVAR)}$$
$$= V \cdot I \cdot \sin \varphi$$

- Daya semu adalah hasil kali tegangan dan arus.

$$S = \text{daya semu (kVA)}$$

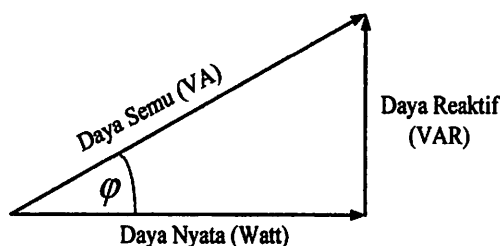
$$= V \cdot I$$

- Faktor daya adalah Perbandingan antara daya nyata dan daya semu.

$$\text{Faktor Daya} = \frac{\text{Daya Nyata (kW)}}{\text{Daya Semu (kVA)}}$$

- ϕ = sudut *phase*

Hubungan ini dapat digambarkan sebagai berikut :



Gambar 2.1
Segitiga Daya

Pada sebagian besar peralatan mengandung dua unsur / jenis beban yaitu beban resistif dan beban reaktif. Dalam hal ini maka akan membutuhkan pula komponen arus yang disebut arus I_r (arus beban resistif) dan arus I_x (arus beban reaktif), kedua komponen arus tersebut adalah :

- Arus beban resistif adalah arus yang dikonversikan menjadi kerja, biasanya dalam bentuk panas, kerja mekanik, cahaya dan bentuk energi lain. Daya yang dihasilkan dari adanya arus ini adalah daya kerja dengan satuan Watt, Kilo Watt, dsb.
- Arus beban reaktif mengalir pada komponen beban yang tidak dapat dikonversikan menjadi bentuk penggunaan energi lain secara langsung,

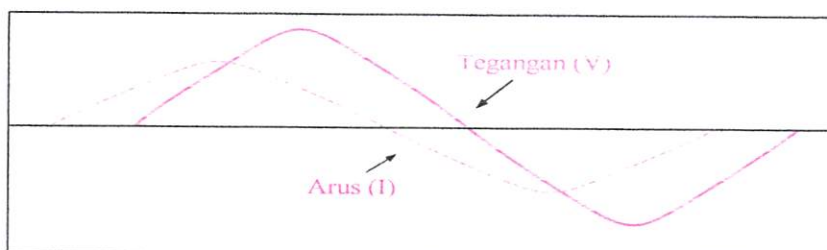
tetapi keberadaannya tidak dapat dipisahkan dari kebutuhan beban antara lain untuk menghasilkan fluks, dalam pengoperasian peralatan elektromagnetis (misalnya : trafo, motor induksi, dsb). Tanpa arus ini maka tidak ada arus magnetisasi dan energi tidak mengalir melalui trafo atau menembus celah udara pada motor induksi.

Suatu beban akan membutuhkan suplai daya aktif jika beban tersebut bersifat induktif dan suatu beban membutuhkan suplai daya reaktif jika beban tersebut bersifat kapasitif. Jadi faktor daya dapat dilihat dari hubungan antara arus aktif, arus magnetisasi dan arus total.

- Arus nyata (I_a) adalah arus yang dibeban dan kedalam energi panas.
- Arus magnetisasi (I_m) adalah arus yang mengalir dibeban untuk menimbulkan medan magnet.
- Arus total (I) adalah arus yang mengalir di jaringan dan merupakan penjumlahan vektor dari arus nyata dan arus magnetisasi.

Dalam bentuk hubungan tersebut digambarkan sebagai berikut :

Apabila arus mendahului tegangan, maka faktor daya itu dikatakan *leading*. Faktor daya *leading* ini terjadi apabila bebannya kapasitif, seperti kapasitor, generator sinkron dan motor sinkron.



Grafik 2.1
Faktor Daya “Leading”

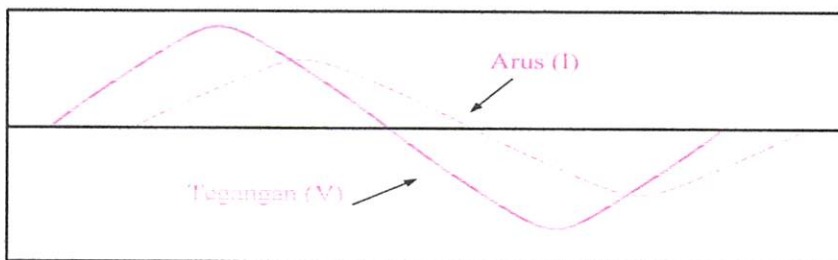
$$i = I_M \cdot \sin \omega t$$

$$\begin{aligned} V_C &= \frac{1}{C} \int_0^T i \, dt \\ &= \frac{1}{C} \int_0^T I_M \cdot \sin \omega t \cdot dt \\ &= \frac{I_M}{\omega \cdot C} \int_0^T \sin \omega t \cdot d\omega t \\ &= (-) \frac{I_M}{\omega \cdot C} \cos \omega t \end{aligned}$$

$$V_C = \frac{I_M}{X_C} (-) \cdot \sin (90^\circ - \omega t)$$

$$V_C = \frac{I_M}{X_C} \cdot \sin (\omega t - 90^\circ)$$

Apabila arus tertinggal dari tegangan, maka faktor daya itu dikatakan *lagging*. Faktor daya *lagging* ini terjadi apabila bebannya induktif, seperti motor induksi (AC) dan transformator.



Grafik 2.2

Faktor Daya “Lagging”

$$i = I_M \cdot \sin \omega t$$

$$\begin{aligned} V_L &= L \cdot \frac{di}{dt} \\ &= L \cdot \frac{dI_M \cdot \sin \omega t}{dt} \end{aligned}$$

$$= \omega \cdot L \cdot I_M \cdot \frac{d \sin \omega t}{d \omega t}$$

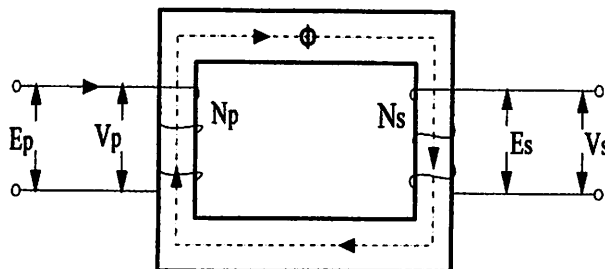
$$= \omega \cdot L \cdot I_M \cdot \cos \omega t$$

$$V_L = \omega \cdot L \cdot I_M \cdot \sin (90^\circ - \omega t)$$

$$V_L = V_M \cdot \sin (90^\circ + \omega t)$$

2.2. Transformator

Transformator adalah suatu alat listrik yang dapat memindahkan dan mengubah energi listrik dari satu atau lebih rangkaian listrik ke rangkaian listrik yang lain melalui suatu gandingan magnet dan berdasarkan prinsip induksi elektromagnet. Prinsip Prinsip kerja Transformator berdasarkan induksi magnet antara rangkaian primer dan rangkaian sekunder. Gandingan magnet berupa inti besi tempat melakukan fluks bersama-sama. Seperti ditunjukkan pada gambar berikut ini :



Gambar 2.2
Transformator

Dalam bidang tenaga listrik pemakaian Transformator dikelompokkan menjadi :

1. Transformator Daya.
2. Transformator Distribusi.

3. Transformator Pengukuran, yaitu terdiri dari : transformator arus dan transformator tegangan.

Cara kerja transformator berdasarkan induksi elektromagnetik, yaitu :

Apabila kumparan primer dihubungkan dengan tegangan bolak-balik, maka akan mengalir arus bolak-balik (I_1) pada kumparan tersebut. Oleh karena kumparan mempunyai inti, maka arus I_1 akan menimbulkan fluks magnet yang berubah-ubah pada intinya. Sehingga kumparan primer akan timbul GGL induksi E_p . Besarnya GGL induksi pada kumparan primer (E_p) adalah :

$$E_p = -N_p \frac{d\phi}{dt} \text{ volt}$$

$$E_p = -N_p \frac{d(\phi_{\max} \sin \omega t)}{dt} = -N_p \cdot \omega \phi_{\max} \cdot \cos \omega t$$

Harga Efektifnya :

$$E_p = \frac{N_p \cdot 2\pi f \phi_{\max}}{\sqrt{2}} = 4,44 N_p \cdot f \cdot \phi_{\max}$$

Pada rangkaian sekunder, fluks (ϕ) bersama tadi menimbulkan :

$$E_p = -N_s \frac{d\phi}{dt} \text{ volt}$$

$$E_p = -N_s \frac{d(\phi_{\max} \sin \omega t)}{dt} = -N_s \cdot \omega \phi_{\max} \cdot \cos \omega t$$

$$E_s = \frac{N_s \cdot 2\pi f \phi_{\max}}{\sqrt{2}} = 4,44 N_s \cdot f \cdot \phi_{\max}$$

Apabila transformator dianggap ideal, maka dianggap tidak terdapat kerugian daya. Perbandingan antara EMF pada primer dan sekunder adalah :

$$A = \frac{E_p}{E_s} = \frac{V_p}{V_s} = \frac{N_p}{N_s} = \frac{I_p}{I_s}$$

2.2.1. Sensor Arus (*Transformator Arus*)

Transformator arus berguna untuk menyediakan cara pengurangan arus saluran keharga-harga yang dapat digunakan untuk mengoperasikan alat pengukur arus rendah atau alat kendali baku, alat ini benar terisolasi dari rangkaian utama, karena *transformator* arus ini digunakan berkaitan dengan alat pengukur arus, maka lilitan primernya dirancang untuk dihubungkan secara seri dengan saluran (beban). Oleh karena itu, impedansi lilitan primer dibuat serendah mungkin. Perbandingan arus primer dan skunder berbanding terbalik dengan perbandingan jumlah primer dan skunder, atau dapat dituliskan :

$$\frac{N_1}{N_2} = \frac{I_2}{I_1} \dots\dots\dots[1]$$

Dimana :

N_1 = Jumlah lilitan primer

I_1 = Arus sisi primer

N_2 = Jumlah lilitan skunder

I_2 = Arus sisi skunder

Dengan mengetahui perbandingan transformasi $\frac{N_1}{N_2}$ dan pembacaan

Ammeter (I_2), arus beban $I_1 = \frac{N_2}{N_1} \times I_2$. Operasi sebuah transformator disebut dalam keadaan ideal, yaitu jika rangkaian sekunder mempunyai impedansi yang rendah pada saat digunakan atau dalam keadaan dihubung singkat.

2.2.2. Sensor Tegangan (*Transformator Tegangan*)

Transformator tegangan berguna untuk menurunkan tegangan dari jala-jala menjadi tegangan yang dibutuhkan oleh ADC yang ada didalam mikrokontroller ATmega8535 untuk bisa bekerja, yaitu sebesar 0-5 Volt. Karena

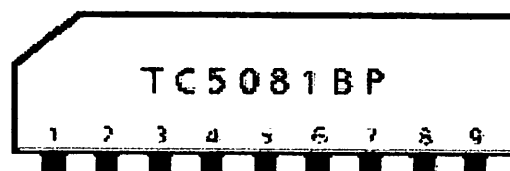
tegangan yang diukur adalah tegangan ± 220 Volt, maka diperlukan adanya *transformator* penurun tegangan (*Transformator Step-Down*). Tegangan ini kemudian disearahkan atau diubah menjadi tegangan DC agar dapat diterjemahkan menjadi tegangan digital oleh ADC tersebut. Dengan mengetahui N_1 dan N_2 , membaca tegangan V_2 , serta menganggap transformator ideal maka tegangan V_1 adalah :

$$V_1 = \frac{N_1}{N_2} \times V_2$$

Pentanahan rangkaian sekunder diperlukan untuk mencegah adanya beda potensial yang besar antara kumparan primer dan kumparan sekunder.

2.3. *Phase Detektor*

Untuk mendeteksi adanya perbedaan fasa antara arus dan tegangan, maka dalam perencanaan dan pembuatan alat ini menggunakan IC TC5081BP, buatan Toshiba. IC TC5081BP berupa *IC Phase Comparator* yang akan mendeteksi 2 *input* pulsa yang berbeda, maka *PD-Out* akan mendeteksi jika saat input pulsa berupa beban kapasitif (*Leading*) dan berupa beban induktif (*Lagging*) dengan impedansi yang tinggi, maka hasil dari data *PD-Out* tadi akan dikeluarkan melalui Pin 4 pada IC TC5081BP yang berupa *phase output*. IC ini terdiri dari 9 pin seperti yang terlihat pada gambar dibawah ini :



Gambar 2.3
IC TC5081BP

Tabel 2.1 Keterangan Pin IC TC5081BP

Pin	Nama	Keterangan
1	<i>A Out</i>	<i>Amplifier Output</i>
2	<i>A In</i>	<i>Amplifier Input</i>
3	<i>PD Out</i>	<i>Phase detektor</i>
4	<i>Phase Output</i>	<i>Phase detector Output</i>
5	<i>VDD</i>	<i>Supplay Voltage</i>
6	<i>NC</i>	<i>No Connection</i>
7	<i>S</i>	<i>Reference Frekuensi Input</i>
8	<i>R</i>	<i>VCO Frekuensi Input</i>
9	<i>GND</i>	<i>Ground</i>

Karena dalam satu gelombang *sinusoida* mempunyai nilai 360^0 dalam frekuensi jala-jala, maka dapat disimpulkan bahwa :

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

$$T = \frac{1}{50} = 0,02 \text{ detik} = 20 \text{ mS}$$

Maka :

$$\varphi = \frac{t}{T} \times 360^0$$

2.4. Penguat Operasional

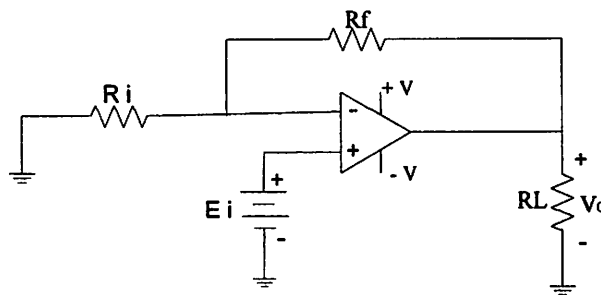
2.4.1. Penguat Tak Membalik

Penguat Tak Membalik (*Non-Inverting*) adalah penguat yang tegangan keluaran V_0 mempunyai polaritas yang sama seperti tegangan masukan E_i , yang diperlihatkan pada gambar 2.4. Persamaan tegangan keluaran dari penguat tak membalik adalah sebagai berikut :

$$V_0 = \left[1 + \frac{R_f}{R_i} \right] \times E_i \text{ Volt}$$

Sehingga diperoleh gain tegangannya :

$$A_{cl} = \frac{V_0}{E_i} = \left[1 + \frac{R_f}{R_i} \right] \text{ Kali}$$



Gambar 2.4

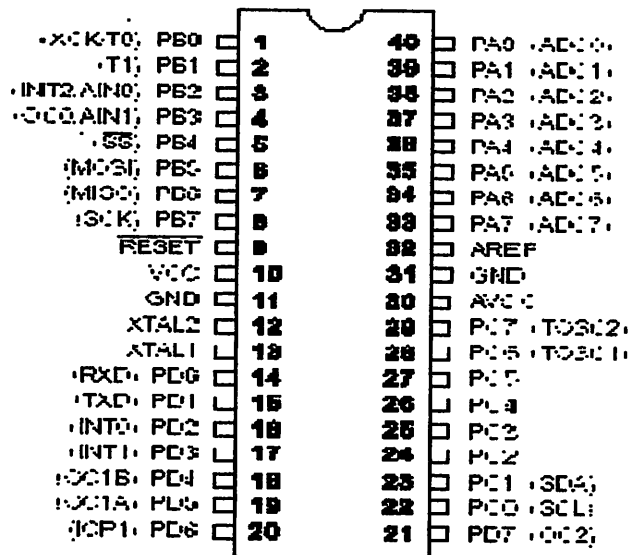
Polaritas Tegangan dan Arah Arus untuk Penguat Tak Membalik

2.5. Mikrokontroler ATmega8535

AVR merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir semua instruksi dieksekusi dalam satu siklus clock. AVR mempunyai 32 register *general-purpose*, *timer/counter* fleksibel dengan *mode compare*, *interrupt internal* dan *eksternal*, serial USART, *Programmable Watchdog Timer*, dan *mode power saving*. Beberapa diantaranya mempunyai ADC dan PWM *internal*. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI.

2.5.1. Fungsi Pin Mikrokontroler ATmega8535

Susunan pin-pin dari IC mikrokontroler ATmega8535 diperlihatkan pada gambar 2.5. IC ini tersusun dari 40 pin yang memiliki beberapa fungsi tertentu.



Gambar 2.5

Konfigurasi Pin ATmega8535

Penjelasan dari masing-masing pin adalah sebagai berikut :

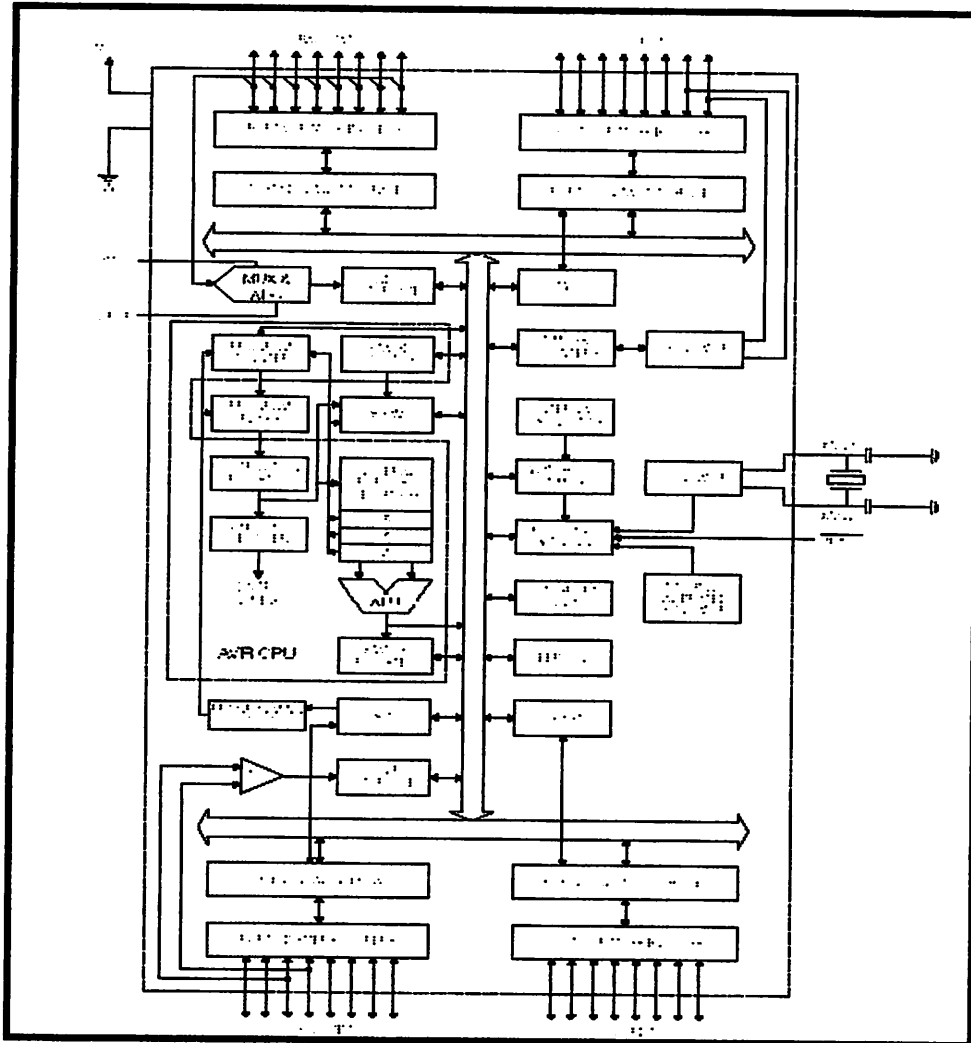
1. **VCC** Merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. **GND** Merupakan pin *ground*.
3. **Port A (PA0..PA7)** Merupakan input analog untuk ADC, jika tidak digunakan port A dapat berfungsi sebagai 8-bit *be-directional I/O* port.
4. **Port B (PB0..PB7)** Port B merupakan port I/O 8-bit bi-directional. pin-pin pada port ini dapat diberi resistor *pull-up internal* secara individual.
5. **Port C (PC0..PC7)** Port C merupakan I/O 8 bit dua arah dengan resistor *pull up internal*. Port ini mampu menyediakan arus hingga 20mA sehingga

dapat langsung menggerakkan LED dari mikrokontroler ini. Pin ini juga dapat digunakan sebagai *oscillator* dari *Timer/Counter*.

6. **Port D (PD0..PD7)** Port D merupakan I/O 8 bit dua arah. Port ini akan pada kondisi *tristate* apabila reset diaktifkan. Port ini juga digunakan sebagai register fungsi khusus, yaitu : komparator analog, *interupsi eksternal*, dan komunikasi serial.
7. **RESET** Merupakan pin input yang aktif rendah, jika pin ini aktif rendah selama minimal 1ns ketika *oscillator* bekerja akan mereset peralatan.
8. **XTAL1** *Input* bagi *inverting oscillator amplifier* dan *input* bagi *clock internal*.
9. **XTAL2** *Output inverting oscillator amplifier*
10. **AVCC** AVCC adalah sumber tegangan +5Volt untuk Port A dan untuk ADC *internal*. Meskipun ADC *internal* tidak digunakan pin ini tetap harus dihubungkan dengan sumber tegangan +5 Volt. Jika ADC *internal* digunakan, pin ini harus dihubungkan dengan +5 Volt dengan tambahan *filter lowpass* pada keluarannya.
11. **AREF** Merupakan masukan untuk tegangan *referensi* ADC.

2.5.2. Arsitektur Mikrokontroler ATmega8535

Seperti yang diungkapkan diatas bahwa mikrokontroler ATmega8535 merupakan mikrokontroler CMOS dengan daya rendah yang memiliki arsitektur AVR RISC 8-bit. Arsitektur ini mendukung kemampuan untuk melaksanakan *eksekusi instruksi* hanya dalam siklus *clock oscillator*.



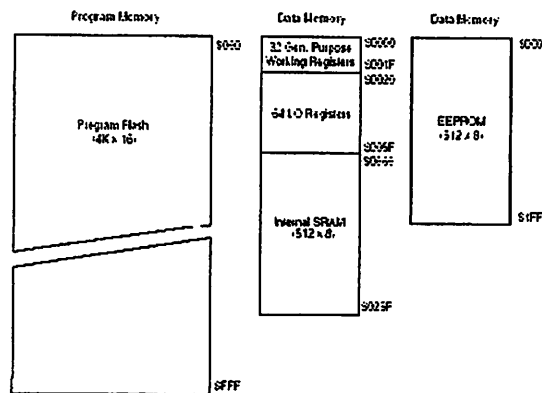
Gambar 2.6
Blok Diagram Mikrokontroler ATmega8535

Dari gambar tersebut dilihat bahwa ATmega8535 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 512 byte.
7. Memori *Flash* sebesar 8kb dengan kemampuan *Read While Write*.
8. Unit interupsi internal dan eksternal.
9. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
10. Port USART untuk komunikasi serial.

2.5.3. Organisasi Memori



Gambar 2.7

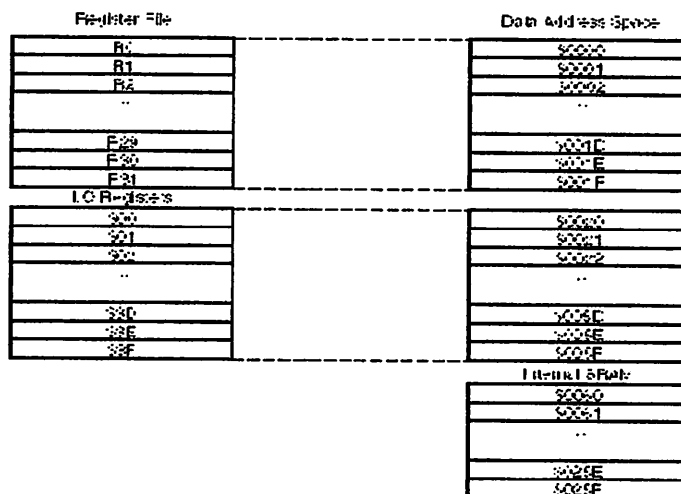
Peta Memori ATmega8535

Dalam organisasi memori AVR, 32 register keperluan umum (GPR) menempati *space* data pada alamat terbawah, yaitu \$0000 sampai \$001F. Sedangkan register-register khusus, untuk penanganan I/O dan kontrol terhadap mikrokontroler, menempati 64 alamat berikutnya, yaitu mulai dari \$0020 hingga \$005F. Register-register ini merupakan register-register yang khusus digunakan untuk melakukan pengaturan fungsi terhadap berbagai peripheral mikrokontroler

semacam kontrol register, *timer/counter*, fungsi-fungsi I/O, dan sebagainya. Kelompok register ini dinamakan register I/O. alamat memori berikutnya digunakan untuk SRAM 512 byte, yaitu pada lokasi \$0060 sampai dengan \$025F. Gambaran peta memori untuk AVR ATmega8535 ditunjukkan oleh gambar 2.7.

2.5.4. SRAM

Mikrokontroler ATmega8535 memiliki memori yang berupa SRAM (*Static RAM*) yang berkapasitas 512 byte. Lokasi memori SRAM dalam peta memori AVR ATmega8535 ditunjukkan oleh gambar berikut.

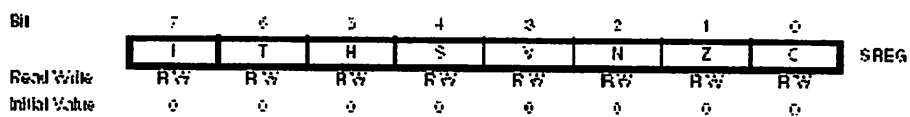


Gambar 2.8
SDRAM dalam Organisasi Memori ATmega8535

2.5.5. Register I/O

Register I/O merupakan register yang sangat penting karena register ini berfungsi untuk mengatur dan melakukan kontrol terhadap berbagai peripheral

mikrokontroler. Definisi masing-masing register I/O dapat dilihat pada Data Sheet. Meskipun semua register I/O penting, namun register I/O yang paling banyak digunakan adalah register STATUS (SREG) yang beralamat pada \$3F(\$5F). Register ini berisi status yang dihasilkan setiap operasi yang dilakukan ketika suatu instruksi dieksekusi.



1. Bit7 – 1: *Global Interrupt Enable*.

Bit ini harus diset untuk mengenabelkan interupsi. Setelah itu, kita dapat mengaktifkan instrupsi mana yang akan kita gunakan dengan cara mengenabel bit kontrol register yang bersangkutan secara individual. Bit ini akan di-clear apabila terjadi suatu instrupsi yang dipicu oleh hardware dan tidak akan mengijinkan terjadinya interupsi serta akan diset kembali oleh instruksi RETI.

2. Bit6 – T: *Bit Copy Storage*

Instruksi BLD dan BST menggunakan bit-T sebagai sumber atau tujuan dalam operasi bit yang dilakukan. Suatu dalam sebuah register GPR dapat disalin ke bit-T ini menggunakan instruksi BST dan sebaliknya bit-T dapat disalin kembali ke suatu bit dalam register GPR dengan menggunakaninstruksi BLD.

3. Bit5 – H: *Half Carry Flag*

Bit ini mengindikasikan *Half-carry* pada beberapa operasi aritmatika.

4. Bit4 – S: *stop Bit*

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

5. Bit3 – V: *Two's Complement Overflow Flag*

Bit ini berguna untuk mendukung operasi aritmatika.

6. Bit2 – N: *Negative Flag*

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

7. Bit1 – Z: *Zero Flag*

Bit ini akan diset bila hasil operasi yang diperoleh adalah nol.

8. Bit0 – C: *Carry Flag*

Bila suatu operasi menghasilkan *carry*, maka bit ini akan diset.

2.5.6. *Reset dan Characteristics*

Pin reset pada mikrokontroler merupakan masukan aktif low (0). Pulsa transisi dari tinggi (1) ke rendah (0) akan mereset mikrokontroler menuju alamat 0000H. Pin reset dihubungkan dengan rangkaian power on reset, sehingga arus dari vcc yang mengalir melewati resistor menuju ke kapasitor, kemudian dibuang ke pin 9 pada (reset) MCU ATmega8535 akan mengakibatkan aktif *low* atau logika (0). Sementara MCU ATMEGA8535 akan ter-reset jika diberi logika *low* selama sekurang 1 siklus mesin, dengan rumus didapat bahwa :

$$F_o = \frac{1}{RC}$$

Jadi dalam periode *Clock* (t) = 1/ F_o

Tabel 2-2 Reset dan Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling) ²			1.3	2.3	V
V_{RST}	RESE [™] Pin Threshold Voltage		0.1		0.9	V
t_{RS}	Minimum pulse width on RESE [™] Pin			50		ns
V_{BOD}	Brown-out Reset Threshold Voltage ³	BODLEVEL = 1	2.5	2.7	3.2	V
		BODLEVEL = 0	3.7	4.0	4.2	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		μ s
		BODLEVEL = 0		2		μ s
V_{HYS}	Brown-out Detector hysteresis			130		mV

Power-On Reset, merupakan sebuah intruksi bahwa MCU akan di reset secara otomatis jika tegangan catu daya lebih rendah dari tegangan ambang batas reset V_{POT} (*Power-On Reset Threshold*). Tegangan V_{POT} berkisar kurang lebih antara 0,1 sampai 0,9.

2.5.7. EEPROM

ATmega8535 memiliki kapasitas EEPROM 512 byte. Untuk melakukan akses terhadap EEPROM, terdapat tiga *register* penting yang berhubungan dengan EEPROM. Register tersebut adalah EEAR (*EEPROM Address Register*), EEDR (*EEPROM Data Register*) dan EECR (*EEPROM Control Register*).

Berikut ini adalah format dan cara pengaturan bit-bit dalam register tersebut.

- EEAR (*EEPROM Address Register*)

Bit	15	14	13	12	11	10	9	8	
	-	-	-	-	-	-	-	EEAR8	EEARH
	FFAR7	FFAR6	FFAR5	FFAR4	FFAR3	FFAR2	FFAR1	FFAR0	FFARH
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	

- **EEDR (EEPROM Data Register)**

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	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	

- **EECR (EEPROM Control Register)**

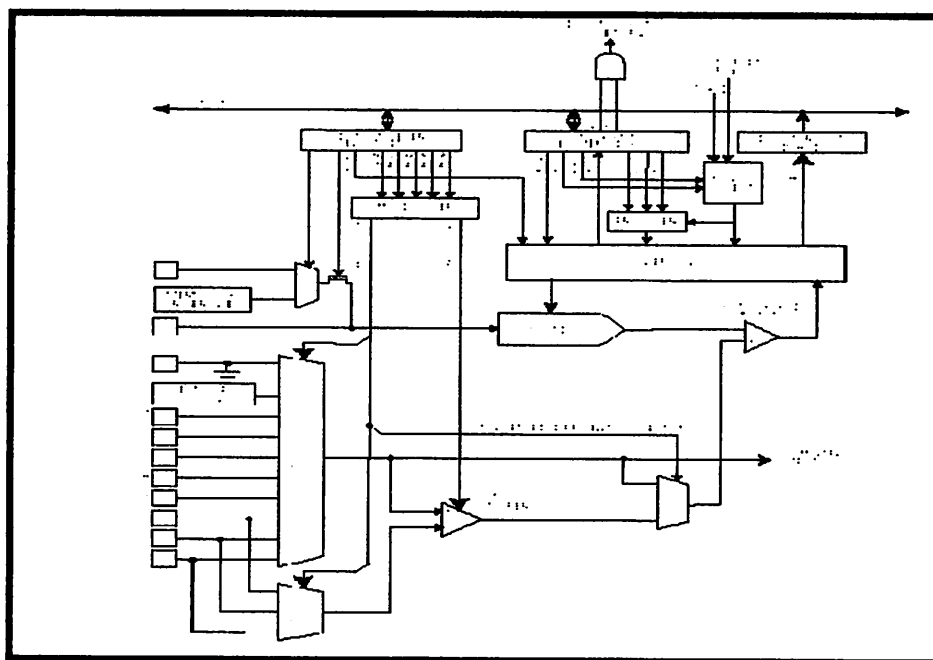
Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	EERIE	EEMWE	EEWE	EEPE	EECR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	X	0	

EEAR merupakan register yang digunakan untuk menempatkan alamat memori EEPROM yang akan diakses. Sedangkan EEDR merupakan register untuk menempatkan data sementara pada pengaksesan EEPROM. Pada operasi penulisan EEPROM yang ditunjuk dalam register EEAR, sedangkan pada operasi pembacaan EEPROM, register ini berisi data yang dibaca dari alamat EEPROM yang ditunjuk dalam EEAR.

2.5.8. Channel Multiplexer Input ADC

Mikrokontroler ATmega8535 memiliki *Internal* ADC 10 bit. Masukan ADC ini dimultiplex menjadi 8 *channel input* yang terhubung dengan Port A. ADC ini memiliki rangkaian cuplik dan genggam untuk menstabilkan tegangan Input ADC pada saat proses konversi sedang berlangsung.

Tegangan Referensi *Internal* diset pada 2,56 Volt, tetapi dapat juga dirubah-rubah sesuai keinginan user melalui pin AREF. Diagram Blok *Internal* ADC pada Mikorokontroller AVR ATmega8535 dapat dilihat dalam Gambar 2.9.



Gambar 2.9

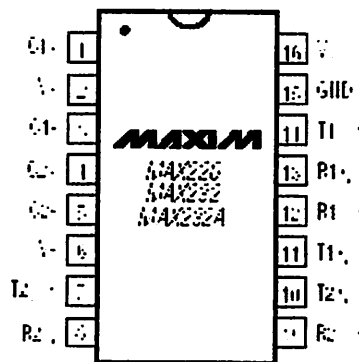
ADC *Internal* Mikrokontroller ATmega8535

2.6. IC MAX 232

Secara umum, RS-232 didefinisikan sebagai antar muka antara peralatan terminal data dengan peralatan komunikasi data menggunakan data biner secara serial. Pada prinsipnya proses transfer data dengan menggunakan *serial interface* ini sangat sederhana, data yang ditransfer suatu terminal akan diterima oleh terminal lain demikian juga sebaliknya. Komunikasi serial merupakan komunikasi data dengan pengiriman data secara satu per satu dengan menggunakan satu jalur kabel data. Sehingga komunikasi serial hanya menggunakan 2 kabel data yaitu

kabel data untuk pengiriman yang disebut *transmit* (Tx) dan kabel data untuk penerimaan yang disebut *receive* (Rx).

Untuk spesifikasi antarmuka RS-232 pada sinyal yang dikirim, level tegangannya adalah +5 Vdc sampai +15 Vdc untuk biner 0 (disebut *space* atau *on*) dan -5 Vdc sampai -15 Vdc untuk biner 1 (disebut *mark* atau *off*). Sedangkan untuk data yang diterima, level tegangannya adalah +3 Vdc sampai +13 Vdc untuk biner 0 dan -3 Vdc sampai -13 Vdc untuk biner 1. Kecepatan pengiriman data yaitu 50 bps sampai 19200 bps (*bit per second* atau bit per detik).



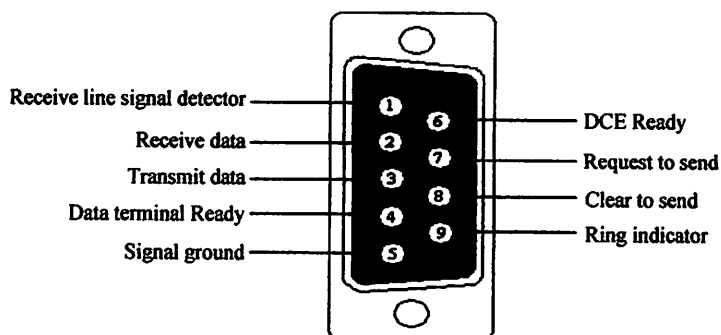
Gambar 2.10

RS 232 (MAXIM)

RS MAX232 tersusun dari 2 bagian yaitu *RS232 Line Driver* yang berfungsi mengubah level tegangan TTL ke level tegangan RS232 dan *RS232 Line Receiver* yang berfungsi mengubah level tegangan RS232 ke level tagangan TTL.

2.6.1. Interface

Istilah *interface* kalau diterjemahkan, mengandung arti sebagai penghubung. Namun bagi mereka yang awam dalam dunia komputer arti diatas mungkin sulit dicerna. Dalam suatu sistem komputer, kejadian dialat diluar komputer yang sedang dihubungkan dengan komputer mungkin dapat mengganggu komputer itu sendiri. Sinyal-sinyal yang tak dikenal seperti itu tidak diperkenankan untuk merusak rangkaian komputer yang ada. Untuk itu dibutuhkan alat perantara yang berfungsi sebagai penghubung dua lingkungan yang berbeda. Alat tersebut dinamakan *interface*. Karena interface merupakan semacam pintu gerbang maka interface sering disebut port I/O.



Gambar 2.11

Skematik DB 9.

Tabel 2-3 Fungsi Pin-Pin RS-232

D-Type-25 Pin no.	D-Type-9 Pin no.	Abbreviation	Full Name
Pin 2	Pin 3	TD	<i>Transmit Date</i>
Pin 3	Pin 2	RD	<i>Receive Date</i>
Pin 4	Pin 7	RTS	<i>Request To Send</i>
Pin 5	Pin 8	CST	<i>Clear To Send</i>
Pin 6	Pin 6	DSR	<i>Data Set ready</i>

Pin 7	Pin 5	SG	<i>Signal Ground</i>
Pin 8	Pin 1	CD	<i>Carrier Detect</i>
Pin 20	Pin 4	DTR	<i>Data Terminal ready</i>
Pin 22	Pin 9	RI	<i>Ring Indikator</i>

2.7. LCD (*Liquid Crystal Display*) M1632

LCD (*liquid Crystal Display*) adalah suatu jenis piranti *output* yang menggunakan daya rendah dengan pengontrol kontras dan kecerahan. Pengontrol utamanya dan karakter ada pada ROM (*Read Only Memory*) generator dan display data RAM (*Random Access Memory*) yang akan menghasilkan extended key codes (kode tombol/*keyboard* standart internasional dalam Hexsa) jika padanya diberikan inputan. Untuk mendapatkan fungsi dengan baik maka perlu diperhatikan proses inialisasi yang telah ditentukan oleh pabrik pembuatnya. Timming penginisialisasian sangat perlu dipertimbangkan, karena jika meleset sampai orde millisecond, maka dapat dipastikan LCD itu tidak dapat berfungsi.

Ada dua jenis register yang terdapat dalam LCD M1632 ini, yaitu data register dan instruction register. Dengan menggunakan pin RS (*Register Select*) pada LCD, pemakaian kedua register dapat dipilih. Pemilihan register dapat dipilih. Pemilihan register pada LCD ditunjukkan dalam tabel berikut ini :

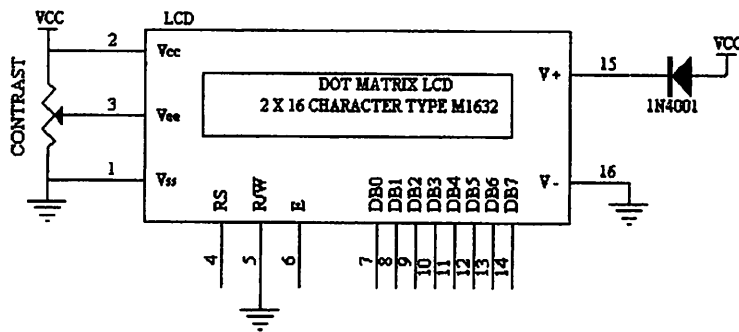
Tabel 2-4 Pemilihan Register pada LCD M1632

Nama Sinyal	No. Terminal	I/O	Tujuan	Keterangan Sinyal
RS	4	Input	MPU	0 : <i>Instruction Register</i> 0 : <i>Data Register</i>

Jika bagian yang dipilih adalah instruction register maka output yang

dihasilkan adalah meliputi operasional dari LCD, misalnya fungsi *display clear*, *cursor home*, *entry mode set*, *display on/off*, *cursor shift*, dan sejenisnya. Sebaliknya, jika bagian yang dipilih adalah data register, output yang dihasilkan adalah meliputi karakter yang tabelnya terdapat pada lampiran data sheet LCD.

Berikut adalah gambar dari LCD dengan pin- pin yang terhubung dengan mikrokontroler :



Gambar 2.12
Modul LCD 2 ×16 karakter

LCD M1632 mempunyai spesifikasi sebagai berikut :

- 16 karakter 2 baris dalam bentuk dot matrik 5×7 dan kursor.
- *Duty ratio* 1/16.
- Memiliki ROM pembangkit karakter untuk 192 jenis karakter.
- RAM untuk data *display* sebanyak 80×8 bit (80 karakter maksimum).
- Dapat dirangkai dengan MPU (*Mikroprocessor Unit*) 8 bit atau 4 bit.
- RAM data *display* dan RAM pembangkit karakter dibaca oleh MPU.
- Memiliki fungsi intruksi : *display ON/OFF*, *cursor ON/OFF*, *display character blink*, *cursor shift* dan *display shift*.

- Memiliki rangkaian *oscillator* sendiri.
- Sumber tegangan tunggal +5 volt.
- Memiliki rangkaian reset otomatis pada catu daya dihidupkan.
- Temperature operasi 0°-50° C.

LCD modul M1632 mempunyai 16 pin dengan fungsi sebagai berikut :

Tabel 2-5 Fungsi Pin – Pin LCD

No	Symbol	Level	Fungsi	
1	Vss	-	Power Supply	0V
2	Vcc	-		5V ± 10%
3	Vee	-		Kontras LCD
4	RS	H/L	H = data input L = <i>instruction input</i>	
5	R/W	H/L	H = <i>read</i> L = <i>read</i>	
6	E	H to L	<i>Enable signal</i>	
7	DB0	H/L	Data bus	
8	DB1	H/L		
9	DB2	H/L		
10	DB3	H/L		
11	DB4	H/L		
12	DB5	H/L		
13	DB6	H/L		
14	DB7	H/L		
15	V+BL	-	<i>Back light</i>	±5V 50-200mA
16	V-BL	-		0V (GND)

Pada LCD juga terdapat instruksi – instruksi sebagai berikut :

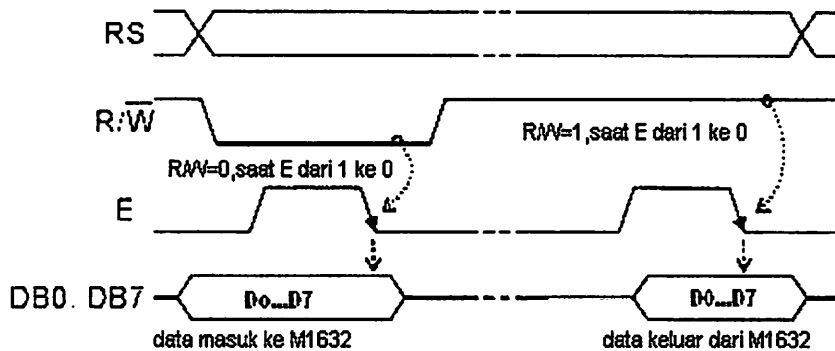
- *Display clear* : membersihkan tampilan yang ada pada LCD serta menyimpan, sedangkan kursor kembali ke posisi semula.
- *Cursor home* : hanya membersihkan tampilan dan kursor kembali ke semula.

- *Empty mode Set* : layar beraksi sebagai tampilan tulis.
 - S : 1/0 = menggeser layar.
 - 1/0 : 1 = kursor bergerak ke kanan dan layar bergerak ke kiri.
 - 1/0 : 0 = kursor bergerak ke kiri dan layar bergerak ke kanan
- *Display On/Off kontrol.*
 - D : 1 = layar *on*
 - D : 0 = layar *off*
 - C : 1 = kursor *on*
 - C : 0 = kursor *off*
 - B : 1 = kursor berkedip-kedip
 - B : 0 = kursor tidak berkedip-kedip
- *Cursor Display Shift*
 - S/C : 1 = LCD diidentifikasi sebagai layar
 - S/C : 0 = LCD diidentifikasi sebagai kursor
 - R/L : 1 = menggeser satu spasi ke kanan
 - R/L : 0 = menggeser satu spasi ke kiri
- *Fuction Set*
 - DL : 1 = panjang data LCD pada 8 bit
 - DL : 0 = panjang data LCD pada 4 bit
 - Bit upper ditransfer terlebih dahulu kemudian diikuti dengan 4 bit *lower*.
 - N : 1/0 = LCD menggunakan 2 atau 1 baris karakter
 - P : 1/0 = LCD menggunakan 5 x 10 dot matrik
- *CG RAM address set* : menulis alamat RAM ke karakter

- *DD RAM address set* : menulis alamat RAM ke tampilan
- *BF/address set* : BF = 1/0, LCD dalam keadaan sibuk atau tidak sibuk.
- *Data write to CG RAM or DD RAM* : membaca byte dari alamat terakhir RAM yang dipilih.

2.7.1. Sinyal *Interface M1632*

Untuk berhubungan dengan mikrokontroler pemakai M1632 dilengkapi dengan 8 jalur pada data (DB0....DB7) yang dipakai untuk menyalurkan kode ASCII maupun perintah pengatur kerja M1632. Selain itu dilengkapi pula dengan E, R/W, RS seperti layaknya komponen yang *kompatibel* dengan mikroprosesor. Kombinasi sinyal E dan R/W merupakan sinyal standar pada komponen buatan *Motorolla*. Sebaliknya sinyal-sinyal dari *Mikrokontroller* merupakan sinyal khas *Intel* dengan kombinasi sinyal WR dan RD. RS singkatan dari *register Select*, dipakai untuk membedakan jenis data yang dikirim ke M1632. Jika RS = 0 maka data yang dikirim adalah perintah untuk mengatur kerja M1632. Sebaliknya jika RS = 1 maka data yang dikirim oleh kode ASCII yang ditampilkan. Demikian pula saat pengambilan data, saat RS = 0, data yang diambil dari M1632 merupakan data status yang mewakili *aktivitas* M1632. Saat RS = 1, maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.



Gambar 2.13
Mengirim dan Mengambil Data dari atau Ke M1632

Proses mengirim atau mengambil data dari atau ke M1632 adalah sebagai berikut:

1. **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
2. **R/W** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **DB0..DB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **DB0 .. DB7** diterima oleh M1632.
3. Untuk mengambil data dari M1632 sinyal **R/W** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saat **E** menjadi 1, M1632 akan meletakkan datanya di **DB0 .. DB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

2.8. Rangkaian *Switch*

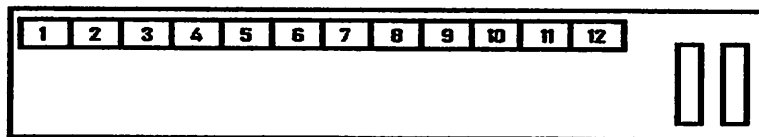
Untuk perancangan *switch* digunakan rangkaian *optotriac* karena rangkaian disini memakai *supply* tegangan 220 VAC. Rangkaian ini untuk mengaktifkan *triac* yang terhubung ke arus beban, resistor R1 berfungsi sebagai pembatas arus ke led di dalam *optotriac* dan resistor R2 sebagai pembatas arus *gate*. Maka ketika mikrokontroler berlogika high (0) maka *triac* ini akan aktif dan menyambungkan arus pada beban atau menhidupkan peralatan listrik begitu juga sebaliknya.

2.9 Telepon Seluler

Telepon Seluler (Ponsel) adalah suatu jenis telepon bergerak yang menggunakan teknologi sel sebagai akses komunikasinya sehingga memudahkan seseorang untuk berkomunikasi dimanapun dan dalam kondisi apapun. Sebuah ponsel dapat mengirim dan menerima data suara menggunakan pemancar RF. Dengan adanya ponsel, maka komunikasi lebih mudah dan lebih efisien tetapi sedikit lebih mahal. Seiring perkembangan teknologi di bidang *mobile* sistem, ponsel tidak hanya mengirim data suara tetapi juga data karakter yang biasa disebut SMS (*Short Message Service*). Disamping itu, pesatnya perkembangan teknologi saat ini menyebabkan ponsel tidak hanya dapat berkomunikasi antara sesama ponsel, tetapi juga dapat berkomunikasi dengan mikrokontroler. Hal ini menyebabkan kita dapat melakukan pentransferan data antara ponsel dan mikrokontroler.

2.9.1 Siemens M-35

Siemens merupakan salah satu merek ponsel yang dapat berkomunikasi dengan mikrokontroler melalui port serial, sehingga suatu pentransferan data dapat terjadi antara mikrokontroler dengan ponsel merk Siemens tersebut. Melalui pentransferan data ini pengguna ponsel dapat mengirim atau menerima suatu pesan singkat (SMS). Untuk dapat berkomunikasi dengan mikrokontroler, Siemens dilengkapi dengan *internal* modem yang dapat mengenali *AT Command*.



Gambar 2.14
Konektor Siemens M-35

Tabel 2-6 Keterangan Konektor Siemens M-35

Nama	Fungsi	In/Out
GND	<i>Ground</i>	
SELF SERVICE	<i>Recognition / control battery charger</i>	In/Out
LOAD	<i>Charging Voltage</i>	In
BATTERY	<i>Battery</i>	Out
DATA OUT	<i>Data Sent</i>	Out
DATA IN	<i>Data eceived</i>	In
Z CLK	<i>Recognition / control accessories</i>	
Z DATA	<i>Recognition / control accessories</i>	
MIC3	<i>Ground for microphone</i>	In
MIC	<i>Microphone input</i>	
AUD	<i>Loudspeaker</i>	Out
AUDG	<i>Ground for eksternal speaker</i>	

2.9.2 Format Data SMS

SMS (*Short Message Service*) merupakan sebuah mekanisme pengiriman pesan singkat melalui jaringan bergerak (*mobile network*). Panjang maksimum dari sebuah pesan singkat adalah 160 karakter, fasilitas ini disediakan oleh jaringan telepon seluler. Sebenarnya panjang pesan maksimum yang dapat dikirimkan melalui SMS adalah 140 karakter. Teknik ini bertumpu pada keadaan bahwa kode karakter ASCII *alfanumerik* yang mempunyai lebar data 7 bit (bit ke-7 selalu bernilai 0 sehingga bisa diabaikan). Teknik kompresi *septet to oktet* dilakukan dengan menyisipkan bit-bit LSB karakter selanjutnya ke dalam bit-bit MSB dari data sebelumnya secara berkesinambungan.

Ada dua Cara pengiriman dan penerimaan SMS, yaitu dengan menggunakan mode teks dan mode PDU (*Protocol Data Unit*). Mode teks tidak terdapat pada beberapa telepon seluler, maka dalam laporan skripsi ini digunakan pesan SMS dengan mode PDU dan tidak membahas mode yang lain. PDU berisi bilangan-bilangan heksadesimal yang terdiri atas: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

2.9.3 Prinsip Kerja SMS (*Short Message Service*)

Dibalik tampilan menu-menu SMS dalam telepon seluler sebenarnya adalah *AT Command* yang bertugas mengirim, menerima dan menghapus data dari dan ke *SMS centre*. *AT Command* tiap-tiap perangkat SMS bisa berbeda-beda tergantung dari jenis telepon seluler, akan tetapi pada dasarnya sama berdasarkan fungsinya. Data-data yang mengalir ke dan dari *SMS-centre* berbentuk PDU (*Protocol Data Unit*). PDU ini berisi bilangan-bilangan heksadesimal yang

mencerminkan bahasa input output. PDU tersebut terdiri dari beberapa *header*. *Header* untuk mengirim SMS ke *SMS-centre* berbeda dengan SMS yang diterima dari *SMS-centre*.

2.9.4 PDU untuk Kirim SMS ke SMS-Centre

Fomat PDU untuk mengirim SMS terdiri atas delapan *header*, yaitu sebagai berikut :

1. Nomor *SMS-Centre*

Header pertama ini terbagi atas tiga *subheader*, yaitu :

a. Jumlah pasangan heksadesimal *SMS-Centre* dalam bilangan heksa.

b. Kode *National/International*.

- Untuk *National*, kode *subheader-nya* yaitu 81
- Untuk *International*, kode *subheader-nya* yaitu 91

c. Nomor *SMS-Centre-nya* sendiri, dalam pasangan heksa dibalik- balik.

Jika tertinggal satu angka heksa yang tidak memiliki pasangan, angka tersebut akan dipasangkan dengan huruf F di depannya.

Contoh: untuk nomor *SMS-Centre* Indosat-M3 dapat ditulis dengan 2 cara yaitu :

Cara 1, (National) : 0855000000 diubah menjadi:

1) 06 → ada 6 pasang

2) 81 → 1 pasang

3) 80-55-00-00-00 → 5 pasang

Total ada 6 pasang. Digabung menjadi : **06818055000000**

Cara 2, (International) : 62855000000 diubah menjadi :

- 1) 07 → ada 7 pasang
- 2) 91 → 1 pasang
- 3) 26-58-05-00-00-F0 → 6 pasang

Total ada 7 pasang. Digabung menjadi : **07912658050000F0**

Berikut ini beberapa nomor SMS-Centre operator seluler di Indonesia :

Tabel 2-7 Beberapa Nomor SMS-Centre National Code

No	Operator Seluler	SMS-Centre No	Kode PDU
1.	Telkomsel	0811000000	06818011000000
2.	Satelindo	0816124	0581806121F4
3.	Excelcom	0818445009	06818081440590
4.	Indosat-M3	0855000000	0855000000

Tabel 2-8 Beberapa Nomor SMS-Centre International Code

No	Operator Seluler	SMS-Centre No	Kode PDU
1.	Telkomsel	62811000000	07912618010000F0
2.	Satelindo	62816124	059126181642
3.	Excelcom	62818445009	07912618485400F9
4.	Indosat-M3	62855000000	07912658050000F0

2. Tipe SMS

Untuk tipe SMS kirim = 1, jadi bilangan heksanya adalah 01 atau 11

3. Nomor Referensi SMS

Nomor referensi ini dibiarkan dulu 0. Jadi, bilangan heksanya adalah 00.

Nomor referensi ini akan diberikan secara otomatis oleh ponsel tersebut.

4. Nomor Ponsel Penerima

Sama seperti cara menulis PDU *Header* untuk *SMS-Center*, *header* ini juga terbagi atas tiga bagian, sebagai berikut :

- a. Jumlah bilangan desimal nomor ponsel yang dituju dalam bilangan heksa.
- b. *National/International Code*.
 - Untuk *National*, kode *subheader-nya* yaitu 81
 - Untuk *International*, kode *subheader-nya* yaitu 91
- c. Nomor ponsel yang dituju, dalam pasangan heksa dibalik. Jika tertinggal satu angka heksa atau tidak memiliki pasangan, angka tersebut dipasangkan dengan huruf 'F' di depannya.

Contoh :

Untuk nomor ponsel yang dituju 628129573337 dapat ditulis dengan dua cara, yaitu sebagai berikut :

- Cara 1, 08129573337 diubah menjadi :
 - a. 0B → ada 11 angka
 - b. 81
 - c. 80-21-59-37-33-F7

Sehingga digabung menjadi : **0B818021593733F7**

- Cara 2, 628129573337 diubah menjadi :
 - a. 0C → ada 12 angka
 - b. 91
 - c. 26-18-92-75-33-73

Sehingga digabungkan menjadi : **0C91261892753373**

5. Bentuk SMS, antara lain :

0 → 00 → dikirim sebagai SMS.

1 → 01 → dikirim sebagai *telex*.

2 → 02 → dikirim sebagai *fax*.

Dalam hal ini, untuk mengirim dalam bentuk SMS tentu saja harus memakai kode heksa 00.

6. Skema *Encoding Data I/O*

Ada dua skema, yaitu :

a. Skema 7 bit, ditandai dengan angka → 00

b. Skema 8 bit, ditandai dengan angka lebih besar dari 0, kemudian diubah ke heksa.

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

7. Jangka waktu/waktu validitas

Jika jangka waktu validitas diabaikan (kode 00), itu berarti tidak membatasi waktu berlakunya SMS. Sedangkan jika diisi dengan suatu bilangan integer yang kemudian diubah ke pasangan heksa tertentu, bilangan yang diberikan tersebut akan mewakili jumlah waktu validitas SMS tersebut. Rumus untuk menghitung jangka waktu validitas SMS adalah sebagai berikut:

Tabel 2-9 Rumus untuk Menghitung Jangka Waktu Validitas SMS

Integer (INT)	Jangka Waktu Validitas SMS
0 – 143	$(INT + 1) \times 5$ menit (berarti : 5 menit s/d 12 jam)
144 – 167	12 jam + $((INT - 143) \times 30)$ menit

168 – 196	(INT – 166) x 1 hari
197 – 255	(INT – 192) x 1 minggu

8. Isi SMS

Header ini terdiri atas dua *subheader*, yaitu :

a. Panjang isi (jumlah huruf dari isi)

Misalnya : untuk kata "bahaya", ada 6 huruf maka penulisannya 06

b. Isi berupa pasangan bilangan heksa

Untuk ponsel/SMS *gateway* berskema *encoding* 7 bit, jika mengetik suatu huruf dari keypadnya, berarti telah membuat 7 angka I/O berturutan.

Ada 2 langkah yang harus dilakukan untuk mengkonversikan isi SMS, yaitu :

Langkah pertama : mengubah menjadi kode 7 bit.

Langkah kedua : mengubah kode 7 bit menjadi 8 bit, yang diwakili oleh pasangan heksa.

Contoh : untuk kata "bahaya"

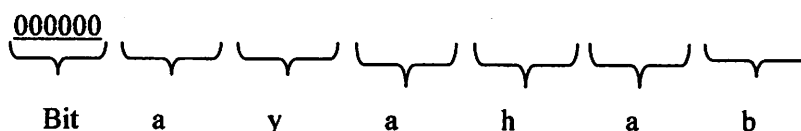
Langkah pertama :

Isi	Bit ke						
	7	6	5	4	3	2	1
B	1	1	0	0	0	1	0
A	1	1	0	0	0	0	1
H	1	1	0	1	0	0	0
A	1	1	0	0	0	0	1
Y	1	1	1	1	0	0	1
A	1	1	0	0	0	0	1

Langkah Kedua:

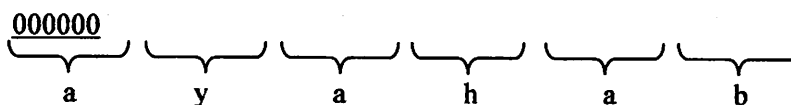
Mengubah kode 7 bit menjadi 8 bit, yaitu oleh karena total 7 bit x 6 huruf = 42 bit, sedangkan yang diperlukan adalah 8 bit x 6 huruf = 48 bit, maka diperlukan 6 bit *dummy* yang diisi dengan bilangan 0, ditambahkan pada MSB huruf terakhir. Setiap 8 bit mewakili suatu pasangan heksa. Tiap 4 bit mewakili suatu angka heksa, tentu saja karena secara logika $2^4 = 16$

Susunan menjadi :



Dummy

Susunan 4 bit dimulai dari MSB menjadi :



Isi	Bit ke							
	8	7	6	5	4	3	2	1
B	1	1	1	0	0	0	1	0
	E				2			
A	0	0	1	1	0	0	0	0
	3				0			
H	0	0	1	1	1	0	1	0
	3				A			
A	1	0	0	1	1	1	0	0
	9				C			
y	0	0	0	0	1	1	1	1
	0				F			
a	0	0	0	0	0	0	1	1
	0				3			

Dengan demikian kata "bahaya" hasil konversinya menjadi: **E2303A9C0F03**

Contoh: Untuk mengirimkan kata "bahaya" ke handphone nomor 6285664408185 lewat SMS-center Indosat-M3, tanpa membatasi jangka waktu valid, maka PDU lengkapnya adalah:

07	91	2658050000F0	01	00	0D	91	265866448081F5	00	00	06	E2303A9C0F03
----	----	--------------	----	----	----	----	----------------	----	----	----	--------------

Penjelasan dari format tersebut dapat dijelaskan pada tabel berikut :

Tabel 2-10 Format SMS Kirim dalam PDU

Heksa	Penjelasan
07	Panjang pasangan nomor SMS-Center termasuk tipe Kode
91	Tipe kode Nasional atau Internasional (91 = kode internasional)
2658050000F0	Service-Center (6285500000)
01	Tipe SMS (01 = tipe untuk pengiriman)
00	Nomor referensi
0D	Panjang nomor handphone penerima
91	Tipe kode Nasional atau internasional
265866448081F5	Nomor Handphone penerima (6285664408185)
00	Tipe bentuk SMS (00 = dikirim sebagai SMS)
00	Tipe data coding
06	Panjang pasangan dari isi SMS
E2303A9C0F03	Isi SMS (bahaya)

2.9.5. PDU untuk SMS Terima dari SMS-Centre

Terdapat 8 header untuk menerima SMS yang pada prinsipnya hampir sama dengan 8 header untuk mengirim SMS. Header tersebut meliputi:

1. Nomor SMS-Centre
2. Tipe SMS untuk SMS-terima = 4 menjadi **04**

3. Nomor ponsel pengirim
4. Bentuk SMS
5. Skema *Encoding*
6. Tanggal dan waktu SMS di *SMS-Centre*

Diwakili oleh 12 bilangan heksa (6 pasangan) yang berarti : yy/mm/dd
hh:mm:ss.

Misalnya : 207022512380 yang berarti 02/07/22 15:32:08 atau 22 Juli
2002 15:32:08 WIB.

7. Batas waktu validitas, jika tidak dibatasi dilambangkan dengan 00.
8. Isi SMS.

Isi SMS yang berupa tulisan disini nantinya dilakukan pengkonversian
yaitu mengubahnya menjadi 8 bit dan kemudian menjadi 7 bit.

Contoh: Format SMS Terima dalam PDU :

07	91	2658050000F0	01	00	0D	91	265866448081F5	00	00	06	E2303A9C0F03
----	----	--------------	----	----	----	----	----------------	----	----	----	--------------

Penjelasan dari format tersebut dapat dijelaskan pada tabel berikut :

Tabel 2-11 Format SMS Terima dalam PDU

Heksa	Penjelasan
07	Panjang pasangan nomor SMS-Center termasuk tipe Kode
91	Tipe kode Nasional atau Internasional (91 = kode internasional)
2658050000F0	Service-Center (62855000000)
04	Tipe SMS (04= tipe untuk terima)
0D	Panjang nomor handphone pengirim
91	Tipe kode Nasional atau internasional
265866448081F4	Nomor Handphone pengirim (6285664408184)

00	Tipe bentuk SMS (00 = diterima sebagai SMS)
00	Tipe data coding
701022512380	Waktu SMS sampai di SMS center yaitu tanggal 22-01-07, pukul 15:32:08 WIB.
00	Jangka waktu SMS expired (00 = tidak memiliki batas)
06	Panjang pasangan dari isi SMS
E2303A9C0F03	Isi SMS (bahaya)

2.9.6. AT - Command

Komunikasi data antara telepon seluler dengan periperal lain seperti mikrokontroler dilakukan secara serial menggunakan perintah-perintah AT (*AT Command*). Dengan mengirimkan perintah-perintah AT yang spesifik dapat memerintahkan telepon seluler untuk melakukan apa yang kita inginkan.

Tabel 2-12 AT Command pada SMS

Perintah	Fungsi
AT+CMGC	Mengirim sebuah perintah SMS
AT+CMGD	Menghapus sebuah SMS dalam SMS memori
AT+CMGF	SMS Format
AT+CMGL	Daftar SMS
AT+CMGR	Membaca dalam sebuah SMS
AT+CMGS	Mengirim sebuah SMS
AT+CSCA	Alamat dari pusat SMS servis

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1. Pendahuluan

Dalam bab ini akan dibahas perancangan dan pembuatan alat. Pembahasan akan dilakukan pada setiap blok rangkaian, cara kerja masing-masing blok rangkaian, perhitungan dan juga 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 *peripheral-peripheral* yang digunakan pada sistem ini. Sedangkan pada perancangan perangkat lunak akan meliputi diagram alir dan *software* secara umum. Akan tetapi kedua perangkat ini dalam kerjanya saling menunjang satu sama lain.

3.1.1. Spesifikasi Alat

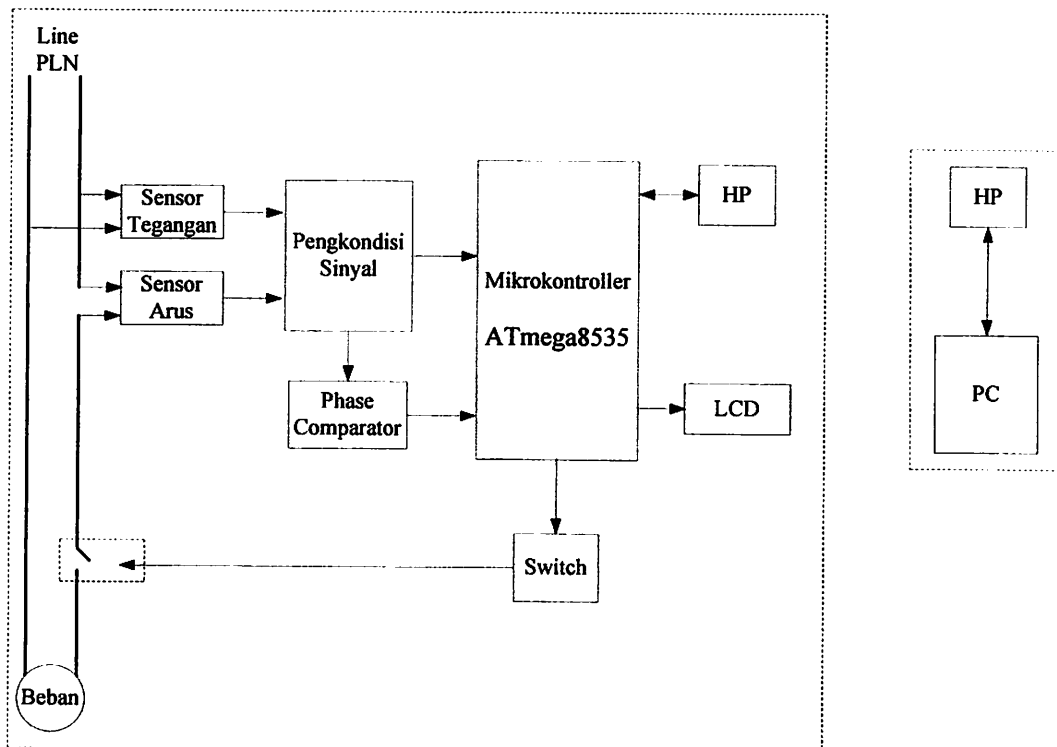
Spesifikasi “Perancangan dan Pembuatan kWh Meter Digital dengan Pengiriman Informasi Tagihan dan Pemutusan Hubungan Listrik Melalui SMS dengan Kendali PC pada *Server*” yang dirancang adalah sebagai berikut:

1. Alat ini menggunakan sistem minimum Mikrokontroler ATmega8535 sebagai pengontrol utama.
2. Menggunakan telepon seluler Siemens M35 dan Nokia 8250 sebagai media pengirim/penerima informasi.

3. Komunikasi telepon selular dengan unit mikrokontroler menggunakan komunikasi serial RS232.
4. Menggunakan *triac* untuk menonaktifkan/mengaktifkan kWh meter pada pelanggan yang bersangkutan.

3.1.2. Blok Diagram Keseluruhan Sistem

Perancangan dan pembuatan alat ini agar dapat dilakukan secara sistematis dan terstruktur maka perlu dibuat blok diagram yang menjelaskan dari sistem yang dirancang. Secara garis besar sistem perancangan ditunjukkan pada diagram blok dari Gambar 3.1.



Gambar 3.1

Blok Diagram Keseluruhan Sistem

Keterangan fungsi dari masing-masing blok diagram di atas adalah sebagai berikut :

1. Handphone (Siemens M35)

Berfungsi sebagai pengirim data informasi pemakaian energi dan tagihan listrik (Rp) berupa SMS dari kWh meter ke pusat selain itu juga sebagai penerima informasi perintah dari pusat.

2. Handphone (Nokia 8250)

Berfungsi sebagai pengirim perintah mengaktifkan dan menonaktifkan berupa SMS ke kWh meter dan menerima informasi pemakaian energi serta tagihan listrik (Rp) yang berupa SMS dari kWh meter.

3. LCD M1632

Berfungsi untuk menampilkan pesan yang berisi informasi tentang berapa pemakaian energi berupa angka pada kWh serta tagihan listrik yang harus di bayar pada bulan ini.

4. Mikrokontroler ATmega8535

Mikrokontroler ini berfungsi sebagai pengontrol utama semua system.

5. *Swicth*

Rangkaian ini menggunakan rangkaian *optotriac* untuk mengaktifkan *triac moc*, sesuai dengan data perintah dari mikrokontroler yang berfungsi untuk memutuskan aliran listrik .

6. PC (*Personal Computer*)

Berfungsi sebagai penampil data pelanggan listrik yang dikirim oleh kWh meter untuk menampilkan berapa pemakaian energi dan tagihan listrik yang harus di bayar pada bulan ini.

3.1.3. Proses Kerja Alat

Pada perancangan dan pembuatan alat ini, maka alat ini akan siap bekerja bila keseluruhan komponen pembentuknya yang meliputi *software* aplikasi utama, *handphone* dan Mikrokontroler sudah diaktifkan semua. Berdasarkan gambar 3.1, maka didapatkan suatu langkah kerja sistem yaitu :

a. Sistem Pada Pusat

Sistem pada pusat ini terdiri dari sebuah PC dan *Handphone* yang dihubungkan dengan komunikasi *Infra Red*. PC dioperasikan oleh operator dengan sebuah program *interface* yang bisa mengirimkan perintah-perintah ke sistem kWh melalui *handphone*. Perintah-perintah yang dapat dikirimkan ke kWh meter ada dua macam perintah yaitu:

1. Perintah Untuk Memutuskan Aliran Listrik Pada kWh Meter.

Pusat dapat mengirimkan perintah berupa SMS dengan format (Off) ke sistem kWh yang berisi perintah untuk memutuskan aliran listrik.

2. Perintah Untuk Mengaktifkan Aliran Listrik Pada kWh Meter.

Pusat dapat mengirimkan perintah berupa SMS dengan format (on) ke sistem kWh meter yang berisi perintah untuk mengaktifkan kembali aliran listrik.

Format penulisan SMS harus benar, penulisan menggunakan huruf kecil atau besar dibedakan. Penulisan SMS yang tidak benar menyebabkan perintah tidak dapat dikenali oleh mikrokontroller.

Selain dapat mengirimkan perintah, *server* juga dapat menerima data yang dikirimkan oleh kWh meter. Data yang dikirimkan oleh kWh meter tersebut berupa SMS yang berisi informasi pemakaian energi dan biaya tagihan listrik pada bulan ini, data tersebut kemudian diterima oleh PC yang ada di pusat untuk diproses selanjutnya.

b. Sistem Pada kWh Meter

Sistem kWh ini terdiri dari: (1) sensor tegangan digunakan untuk mengukur tegangan, (2) sensor arus digunakan untuk mengukur arus yang mengalir, (3) pendeteksi $\text{Cos}\phi$ berfungsi untuk mendeteksi sinyal listrik dengan melihat lebar pulsa histerisnya antara V dengan I dalam satu siklus, (4) pengkondisi sinyal berfungsi untuk menstabilkan tegangan AC yang akan dikonversikan ke ADC *internal* MCU ATmega8535, (5) minimum sistem Mikrokontroler ATmega8535, sebagai pemroses utama, (6) handphone Siemens M35, digunakan untuk menerima perintah dan mengirimkan data kWh meter, (7) RS 232 berfungsi sebagai pengkonversi level TTL ke level tegangan RS 232 agar bisa dibaca oleh kabel data HP Siemen M35, (8) LCD M1632, sebagai penampil pesan secara visual, (9) Rangkaian *Swicth*, dihubungkan dengan arus listrik pada beban yang digunakan untuk memutus dan menyambungkan kembali aliran listrik.

3.2. Perancangan Perangkat Keras (*Hardware*)

3.2.1. Perencanaan Sensor

3.2.1.1. Perencanaan Sensor Arus

Untuk mengukur arus listrik yang mengalir ke beban digunakan transformator arus CT (*Current Transformer*). CT digunakan sebagai media pengukur arus listrik secara tidak langsung. Dalam perancangan ini, transformator arus dapat diukur dengan cara sistem induksi, yaitu melewati besi atau kabel yang dililitkan pada transformator arus, dimana perbandingan lilitan yang terjadi pada lilitan primer dan sekunder sebesar 10 : 50. Dalam mengukur arus besaran arus harus dikonversikan menjadi tegangan AC terlebih dahulu dan untuk mengkonversikan arus menjadi tegangan digunakan resistor sebesar 1 Ω . Untuk mengetahui perbandingan transformasi, maka kondisi ideal pada transformator arus berlaku rumus :

$$\frac{N_1}{N_2} = \frac{I_2}{I_1}$$

Dimana:

N_1 = Jumlah lilitan primer

N_2 = Jumlah lilitan sekunder

I_1 = Arus sisi primer

I_2 = Arus sisi sekunder

Dengan mengetahui perbandingan transformasi $\frac{N_1}{N_2}$ dan pembacaan ammeter $I_1 = \frac{N_1}{N_2} \times I_2$. Operasi sebuah transformator disebut dalam keadaan ideal, yaitu jika rangkaian sekunder mempunyai impedansi yang rendah pada saat digunakan atau dalam keadaan dihubung singkat.

3.2.1.2. Perencanaan Sensor Tegangan

Transformator tegangan digunakan untuk mengukur tegangan, karena tegangan yang diukur adalah ± 220 Volt, maka digunakan transformator penurun tegangan, yaitu transformator *stepdown* yang diproduksi oleh *universal* yang banyak di jual dipasaran. Dimana tegangan ini kemudian disearahkan atau diubah terlebih dahulu menjadi tegangan ADC agar dapat diterjemahkan menjadi tegangan digital untuk mendapatkan tegangan yang berkisar antara 0 – 5 Volt.

Spesifikasi dari sensor tegangan yang digunakan adalah sebagai berikut :

$$V_{\text{Primer}} = 0 \text{ Volt, } 110 \text{ Volt, } 220 \text{ Volt.}$$

$$V_{\text{Sekunder}} = 0 \text{ Volt (CT), } +15 \text{ Volt, } 7,5 \text{ Volt, } -15 \text{ Volt.}$$

$$I = 2 \text{ A}$$

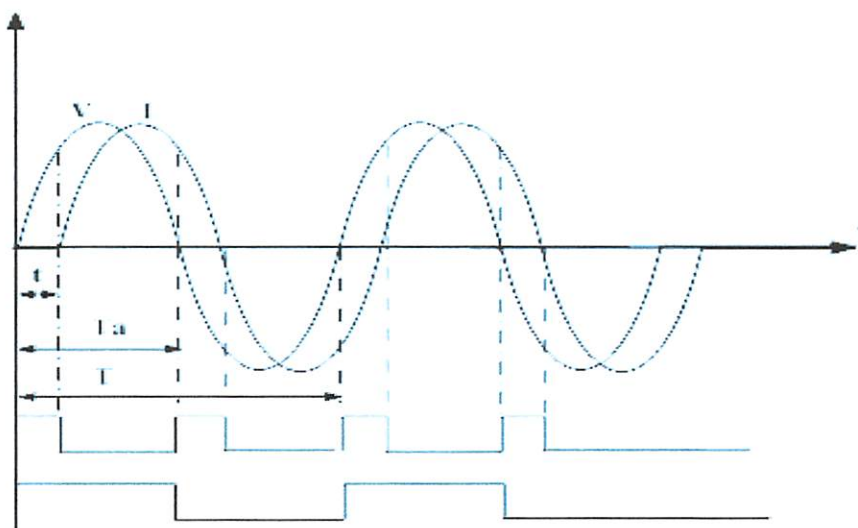
Dengan mengetahui N_1 dan N_2 , membaca tegangan V_2 , serta menganggap transformator ideal maka tegangan V_1 adalah:

$$V_1 = \frac{N_1}{N_2} \times V_2$$

Dalam perancangan ini output dari tegangan tersebut dipilih sebesar 7,5 Volt. Hal ini dimaksudkan agar tegangan tersebut dapat masuk atau di baca oleh mikrokontroler setelah disearahkan tegangannya untuk diproses lebih lanjut.

3.2.1.3. Perancangan Sensor $\cos \phi$

Untuk menghitung daya yang bersatuan watt, maka diharuskan mengetahui nilai $\cos \phi$. Nilai $\cos \phi$ ini merupakan beda fasa antara tegangan dan arus. Dalam perancangan sensor $\cos \phi$ digunakan IC TC5081, dimana IC ini dilengkapi dengan fasilitas *Phase Comparator* yang digunakan untuk mendeteksi beda fasa antara tegangan dan arus, dengan meninjau dari lebar tegangan histerisnya antara nilai V dan I dalam 1 siklus seperti ditunjukkan pada gambar grafik 3.2 dibawah ini :



Grafik 3.1

Grafik Beda Fasa Antara V dan I

Untuk mengetahui Nilai Cos ϕ maka digunakan rumus sebagai berikut :

$$\phi = \frac{1}{\Delta t} \times 360^0 \dots\dots\dots(1)$$

$$T = \frac{1}{f} \dots\dots\dots(2)$$

Keterangan :

t = Panjang denyut

T = Periode

f = Frekuensi listrik (50Hz)

Karena dalam satu gelombang *sinusoida* mempunyai nilai 360^0 dalam frekuensi jala-jala, maka dapat diketahui bahwa :

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

$$T = \frac{1}{50} = 0,02 \text{detik, atau}$$

$$= 20 \text{ ms}$$

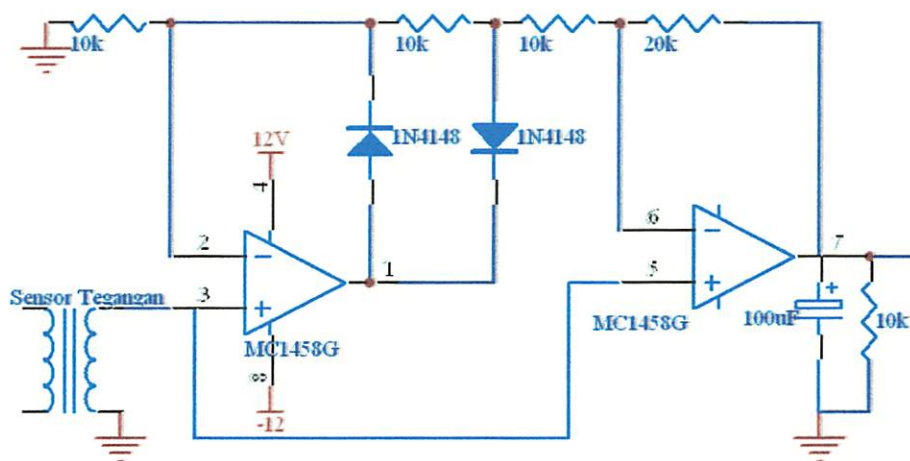
3.2.2. Rangkaian Pengkondisi Sinyal (Pengubah AC ke DC)

Tegangan yang keluar dari sensor arus dan sensor tegangan merupakan tegangan AC. Untuk mendapatkan tegangan yang stabil sebelum tegangan

tersebut diubah menjadi data digital maka tegangan tersebut harus disearahkan terlebih dahulu menjadi tegangan DC.

Dalam perancangan rangkaian penyearah dimaksudkan agar arus yang diserap lebih kecil sehingga tidak akan mempengaruhi tegangan pada sensor. Untuk itu digunakan penyearah yang dapat berfungsi sebagai diode ideal yang aktif sehingga tidak ada sinyal yang terpotong.

Mekanisme terjadinya konduksi pada masing-masing diode tergantung pada polaritas tegangan yang terjadi pada masukan. Keadaan positif atau negatif dari masukan didasarkan pada referensi CT (*Center-tap*). Pada gambar 3.3



Gambar 3.2

Rangkaian Penyearah Gelombang Penuh

Nampak bahwa pada setengah periode pertama misalnya, V_1 berharga positif dan V_2 berharga negatif, ini menyebabkan D1 berkonduksi (berpanjar maju) dan D2 tidak berkonduksi (berpanjar mundur). Pada setengah periode ini arus i D1 mengalir dan menghasilkan keluaran yang akan nampak pada hambatan beban.

Pada setengah periode berikutnya, V_2 berharga positif dan V_1 berharga negatif, menyebabkan $D2$ berkonduksi dan $D1$ tidak berkonduksi. Pada setengah periode ini mengalir arus i_{D2} dan menghasilkan keluaran yang akan nampak pada hambatan beban. Dengan demikian selama satu periode penuh hambatan beban akan dilewati arus i_{D1} dan i_{D2} secara bergantian dan menghasilkan tegangan keluaran DC yang keluarannya difilter terlebih dahulu sebelum diproses oleh ADC internal pada mikrokontroler.

Dengan persamaan :

$$V_o = - (V_1 + 2V_2) \dots\dots\dots(2)$$

Sehingga untuk daur negatif gelombang di dapat :

$$V_o = - (V_1 + 0) = -V_1$$

Sedangkan untuk daur positif gelombang didapat :

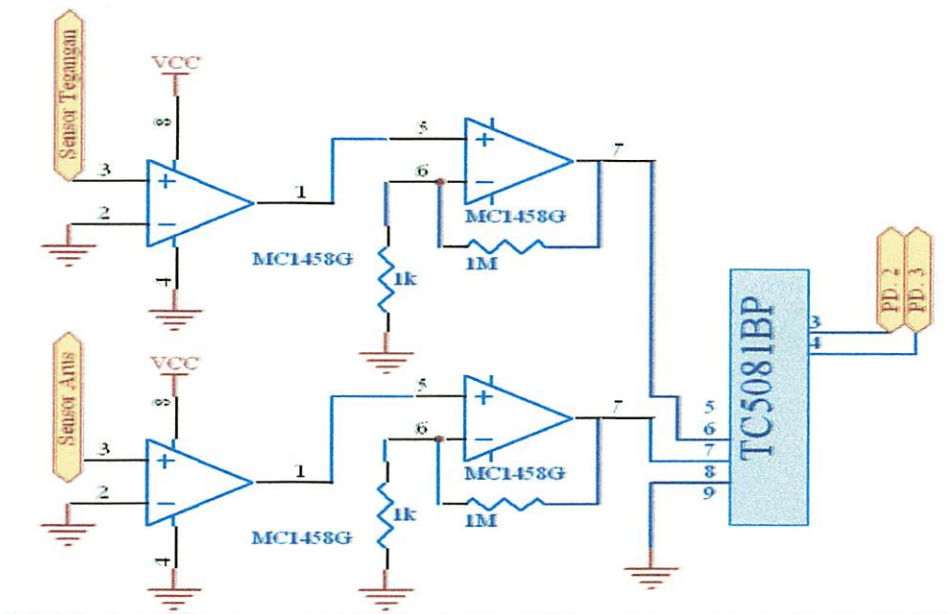
$$V_o = - (V_1 + 2V_2) = -V_1$$

Begitu pula untuk penyearah gelombang penuh yang inputannya untuk sensor arus sama seperti inputan pada sensor tegangan.

3.2.3. Rangkaian *Zero Crossing Detector*

Rangkaian *zero crossing detector* terbentuk oleh komponen utama dengan menggunakan penguat IC MC4158 (Op-Amp) dan rangkaian pembanding menggunakan IC TC5081. Dengan menghubungkan keluaran dari sensor arus dan sensor tegangan yang kemudian dikuatkan sinyalnya. Dalam penguatan sinyal ini

kurang lebih sebesar 1000 kali penguatan setelah di *voltage follower*, kalo bisa dikuatkan sekuat-kuatnya dikarenakan takut terjadi sinyal keluaran yang sangat kecil yang tidak sesuai dengan kebutuhan. Seperti terlampir pada gambar dibawah ini :



Gambar 3.3

Rangkaian *Zero Crossing Detector*

Maka dengan hasil perhitungan diketahui bahwa penguatan dapat dihitung :

Jika : $R_f = 1M\Omega$

$R_i = 1K\Omega$

Maka :

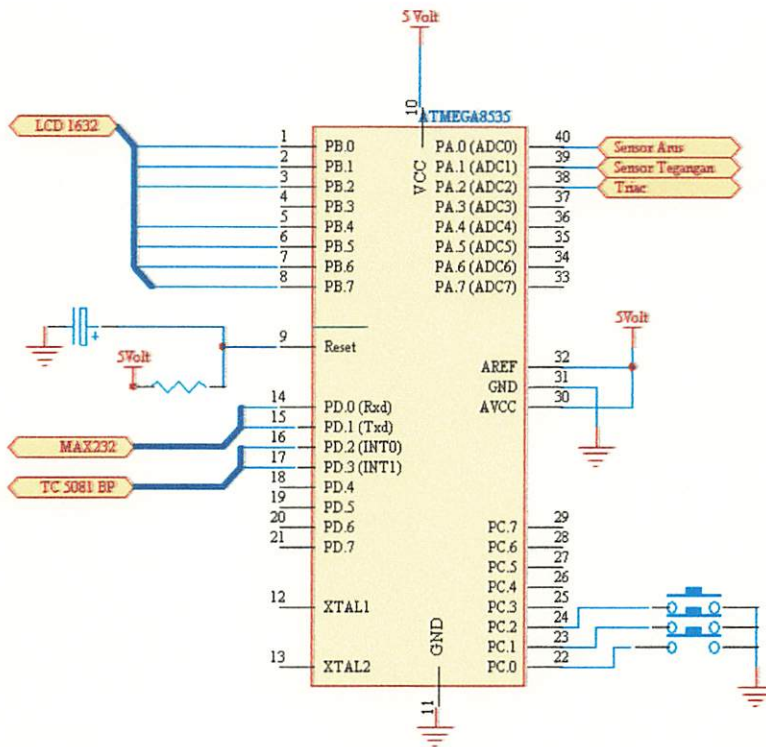
$$A = \left(\frac{R_f}{R_i} \right)$$

$$A = \left(\frac{1.10^6}{1.10^3} \right) = 1000 \text{ kali}$$

3.2.4. Perancangan Mikrokontroler Pada kWh Meter

3.2.4.1. Perancangan Penggunaan Port pada Mikrokontroler ATmega8535

Dengan rangkaian yang sederhana ini mikrokontroler dibuat sebagai sistem minimum menjadi pengontrol alat, disamping itu rangkaian ini dapat dibuat bermacam-macam alat dengan menambah sedikit komponen tambahan lainnya. Adapun rangkaiannya ditunjukkan seperti pada Gambar 3.5.



Gambar 3.4

Minimum Sistem ATmega8535

Agar sebuah mikrokontroler dapat bekerja sebagai pengontrol, maka kaki-kaki/*port* mikrokontroler dihubungkan dalam rangkaian-rangkaian eksternal. Dalam perancangan ini, *port* yang digunakan adalah sebagai berikut:

1. PortB.0-B.7 sebagai output ke LCD M1632 kecuali PortB.3 tidak digunakan.
2. PortA.0 digunakan sebagai input dari Sensor Arus.
3. PortA.1 digunakan sebagai input dari Sensor Tegangan.
4. PortA.2 digunakan sebagai output *Driver Relay*.
5. PortD.0 digunakan sebagai input dari data serial RS232.
6. PortD.1 digunakan sebagai output dari data serial RS232.
7. PortD.2 dan D.3 digunakan sebagai input dari data IC TC5081BP.
8. PortC.0-2 sebagai masukan perintah untuk *push button*.
9. VCC dihubungkan dengan tegangan sebesar +5V
10. GND dihubungkan ke *ground* catu daya.

3.2.4.2. Internal ADC pada Mikrokontroler ATmega 8535

Agar tegangan yang dikeluarkan oleh rangkaian penyearah dapat dibaca oleh mikrokontroler ATmega8535, maka nilai tegangan tersebut harus diubah menjadi bentuk data digital. Untuk itu digunakan *Converter Analog ke Digital* (ADC). ADC yang digunakan dalam perancangan ini adalah ADC *internal* dari mikrokontroler ATmega8535 yang merupakan ADC dengan 8 masukan analog yang dimultipleks menjadi data digital 10 bit. Dalam perancangan alat ini digunakan dua buah input analog yaitu IN_0 dan IN_1 . Untuk itu maka *address* yang tidak diperlukan pada ADC dihubungkan dengan *ground*.

Dikarenakan tegangan yang terukur cukup kecil maka tingkat resolusi ADC *internal* mikrokontroler yang diharapkan cukup besar, sehingga pada ADC *internal* ATmega8535 menggunakan V_{ref} sebesar 5V, dimana tingkat resolusi ADC *internal* ini adalah :

$$\text{Resolusi} = \frac{V_{\text{referensi}}}{(2^{10} - 1)}$$

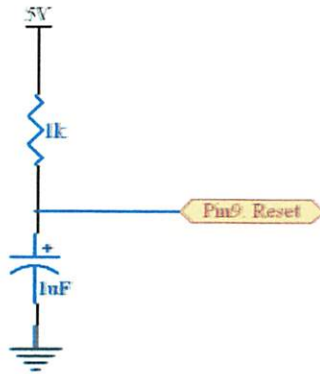
Jadi besarnya resolusi adalah sebesar :

$$\begin{aligned}\text{Resolusi} &= \frac{5 \text{ V}}{(2^{10} - 1)} = 0,00488 \text{ V} \\ &= 4,88 \text{ mV}\end{aligned}$$

Sehingga dengan perubahan inputan analog 4,88 mV maka output dari ADC akan mengalami perubahan 1 bit.

3.2.4.3. Perencanaan Rangkaian Reset

Pin reset pada mikrokontroller merupakan masukan aktif *low* (0). Pulsa transisi dari tinggi (1) ke rendah (0) akan mereset mikrokontroler menuju alamat 0000H. Pin reset dihubungkan dengan rangkaian *power on reset* yang diperlihatkan pada gambar 3.6 berikut ini :



Gambar 3.5

Perencanaan Rangkaian *Power On Reset* ATmega8535

Untuk mereset ATmega8535 diberi logika *low* selama 1 siklus mesin:

$$\begin{aligned}
 F_o &= \frac{1}{RC} \\
 &= \frac{1}{1.10^3 \times 1.10^{-6}} \\
 &= 1000 \text{ kHz}
 \end{aligned}$$

Maka Periode *Clock* = $1/F_o$

$$\begin{aligned}
 T_{Rst} &= 1/1000 \\
 &= 1000\mu\text{s}
 \end{aligned}$$

3.2.5. Perencanaan Rangkaian LCD (*Liquid Crystal Display*)

Untuk berhubungan dengan mikrokontroler, memakai LCD M1632 menggunakan 4 jalur data (PB4...DB7) yang dipakai untuk menyalurkan kode

ASCII maupun perintah pengatur kerjanya M1632. Selain itu dilengkapi pula dengan **E**, **R/W*** dan **RS** seperti layaknya komponen yang kompatibel dengan mikroprosesor.

RS, singkatan dari *Register Select*, dipakai untuk membedakan jenis data yang dikirim ke M1632, kalau **RS=0** data yang dikirim adalah perintah untuk mengatur kerja M1632, sebaliknya kalau **RS=1** data yang dikirim adalah kode ASCII yang ditampilkan.

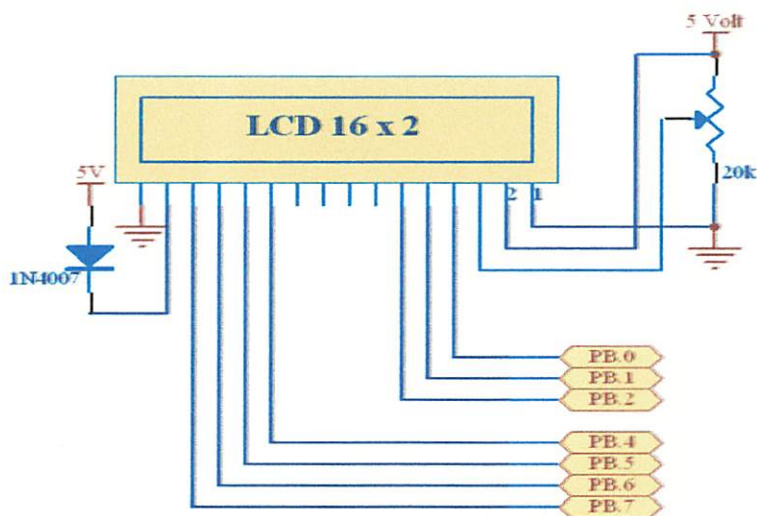
Demikian pula saat pengambilan data, saat **RS=0** data yang diambil dari M1632 merupakan data status yang mewakili aktivitas M1632, dan saat **RS=1** maka data yang diambil merupakan kode ASCII dari data yang ditampilkan.

Proses mengirim/mengambil data ke/dari M1632 bisa dijabarkan sebagai berikut :

- **RS** harus dipersiapkan dulu, untuk menentukan jenis data seperti yang telah dibicarakan di atas.
- **R/W*** di-nol-kan untuk menandakan akan diadakan pengiriman data ke M1632. Data yang akan dikirim disiapkan di **PB0..PB7**, sesaat kemudian sinyal **E** di-satu-kan dan di-nol-kan kembali. Sinyal **E** merupakan sinyal sinkronisasi, saat **E** berubah dari 1 menjadi 0 data di **PB0 .. PB7** diterima oleh M1632.
- Untuk mengambil data dari M1632 sinyal **R/W*** di-satu-kan, menyusul sinyal **E** di-satu-kan. Pada saat **E** menjadi 1, M1632 akan meletakkan

datanya di **PB0 .. PB7**, data ini harus diambil sebelum sinyal **E** di-nol-kan kembali.

Untuk tampilan dipergunakan LCD Dot Matrik 2 x 16 karakter. 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 1 dan *enable* berlogika 1. Instruksi dikirim pada LCD bila keadaan RS 0 dan *Enable* 1. Pin LCD ini untuk data terkoneksi pada PB.1 mikrokontoller. Kemudian untuk RS dihubungkan pada PB.0 tulis/baca (*Read/Write*) diberikan logika *low*, karena disini LCD bersifat menulis data, dan yang terakhir *Enable* (E) dikendalikan dengan PB.2. Gambar rangkaian LCD ditunjukkan pada gambar 3.7 sebagai berikut :



Gambar 3.6
Rangkaian LCD

VR pada pin 3 (VEE) digunakan untuk mengatur kontras dari karakter yang ditampilkan, sedangkan pin 15 (V+) diberi dioda gunanya agar tegangan yang masuk sesuai dengan data datasheet yaitu 4,5 V.

Jika diketahui : Tegangan Ambang Dioda = 0,6 V

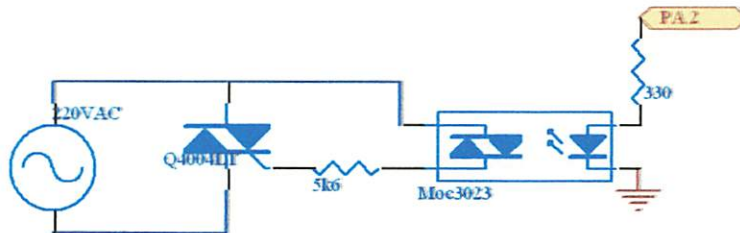
$$VCC = 5 V$$

Maka tegangan Input adalah = $5V - 0,6V$

$$= 4,4 V$$

3.2.6. Rangkaian *Switch*

Untuk perancangan *switch* digunakan rangkaian *triac* karena rangkaian disini memakai *supply* tegangan 220 VAC.



Gambar 3.7

Rangkaian *Switch*

Tegangan output maksimal dari mikrokontroller adalah 5 Volt sedangkan untuk rangkaian *triac* memerlukan *supply* 220VAC, maka diperlukan rangkaian *optotriac* untuk mengaktifkan *triac* MOC 3023, maka analisa data yang digunakan:

$$V_{in} = 5 \text{ Volt}$$

Data Sheet untuk mengaktifkan MOC 3023:

$$V_F \text{ (tegangan forward dioda)} = 1,5 \text{ Volt}$$

$$I_{FT} \text{ (arus forward Trigger)} = 10 \text{ mA}$$

Maka untuk mengaktifkan *triac*, resistor yang dipasang:

$$R = (V_{in} - V_F) / I_{FT}$$

$$R = (5V - 1,5V) / 10mA$$

$$R = 3,5V / 10mA$$

$$R = 350 \Omega$$

Karena nilai resistor 350 Ω tidak ada dipasaran, maka digunakan resistor yang mendekati yaitu 330 Ω .

Karena tegangan jaringan yang digunakan adalah 220V dan arus *gate* maksimum (I_{GTM}) adalah 25 mA, maka total nilai R adalah :

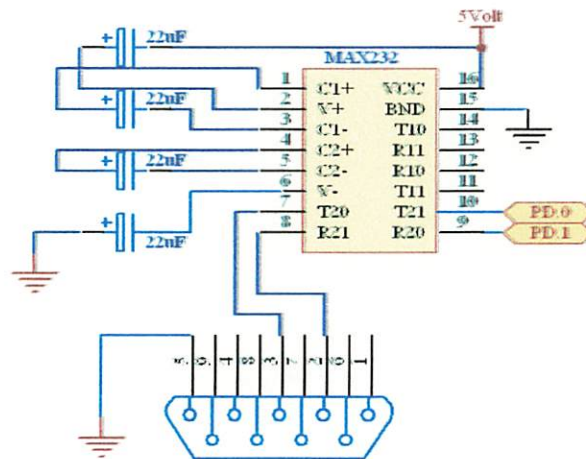
$$R = V_{jaringan} / I_{GTM} = 220 / 25 \cdot 10^{-3} = 8K8\Omega$$

Karena nilai resistor 8,8K Ω tidak ada dipasaran, maka digunakan resistor yang mendekati dibawahnya yaitu 5K6 Ω .

3.2.7. Perancangan Komunikasi data Handphone dengan Mikrokontroler

Pada sistem digunakan *handphone* Siemens tipe M35. *Handphone* ini berperan sebagai *konverter* yang menjadi perantara pertukaran data dalam bentuk SMS antara HP (*handphone*) dengan mikrokontroler maupun sebaliknya, sehingga data dapat dikomunikasikan melalui jaringan GSM dan dapat diolah oleh mikrokontroler.

Pada *handphone* ini memiliki suatu *pin-out* yang berfungsi sebagai *interface* ke perangkat lain seperti mikrokontroler. Untuk pengiriman dan penerimaan data pada HP Siemens M35 menggunakan RS232 yang ditampilkan gambar 3.9. dibawah ini.



Gambar 3.8

Rangkaian *Interface* antara Handphone dengan Mikrokontroler

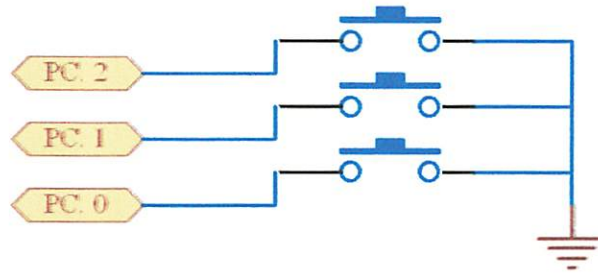
Serial *Control* (SCON) merupakan register khusus pengontrol kerja port serial, diset untuk mentransmisikan 8 bit data UART (*Universal Asynchronous*

Receiver Transmitter) yang merupakan standar komunikasi data dengan *baud rate* yang dapat diatur (*variable*). Penentuan *baud rate* tersebut dilakukan dengan pengesetan *timer/counter* 1 high bit. Yang diinginkan *baud rate* 19200 bps maka :

$$\begin{aligned} \text{UBRR} &= \frac{F_{\text{osc}}}{16 \times \text{Baudrate}} - 1 \\ &= \frac{4000000}{16 \times 19200} - 1 \\ &= 12,0208 \end{aligned}$$

3.2.8. *Push Button*

Dalam perancangan ini, mikrokontroler ATmega8535 sebenarnya telah diatur untuk mengirimkan datanya ke *server* melalui SMS secara otomatis dalam setiap bulannya yaitu pada bulan, tanggal dan jam yang sudah ditentukan. Oleh karena itu, untuk melakukan percobaan pengiriman tersebut maka dibutuhkan *push button* guna mempercepat pengaturan pengiriman, sehingga tidak harus menunggu lagi proses pengiriman selama satu bulan, pin-pin ini akan dihubungkan ke mikrokontroler pada port (PC.0....PC.2) seperti yang terlihat pada gambar 3.10 dibawah ini :



Gambar 3.9

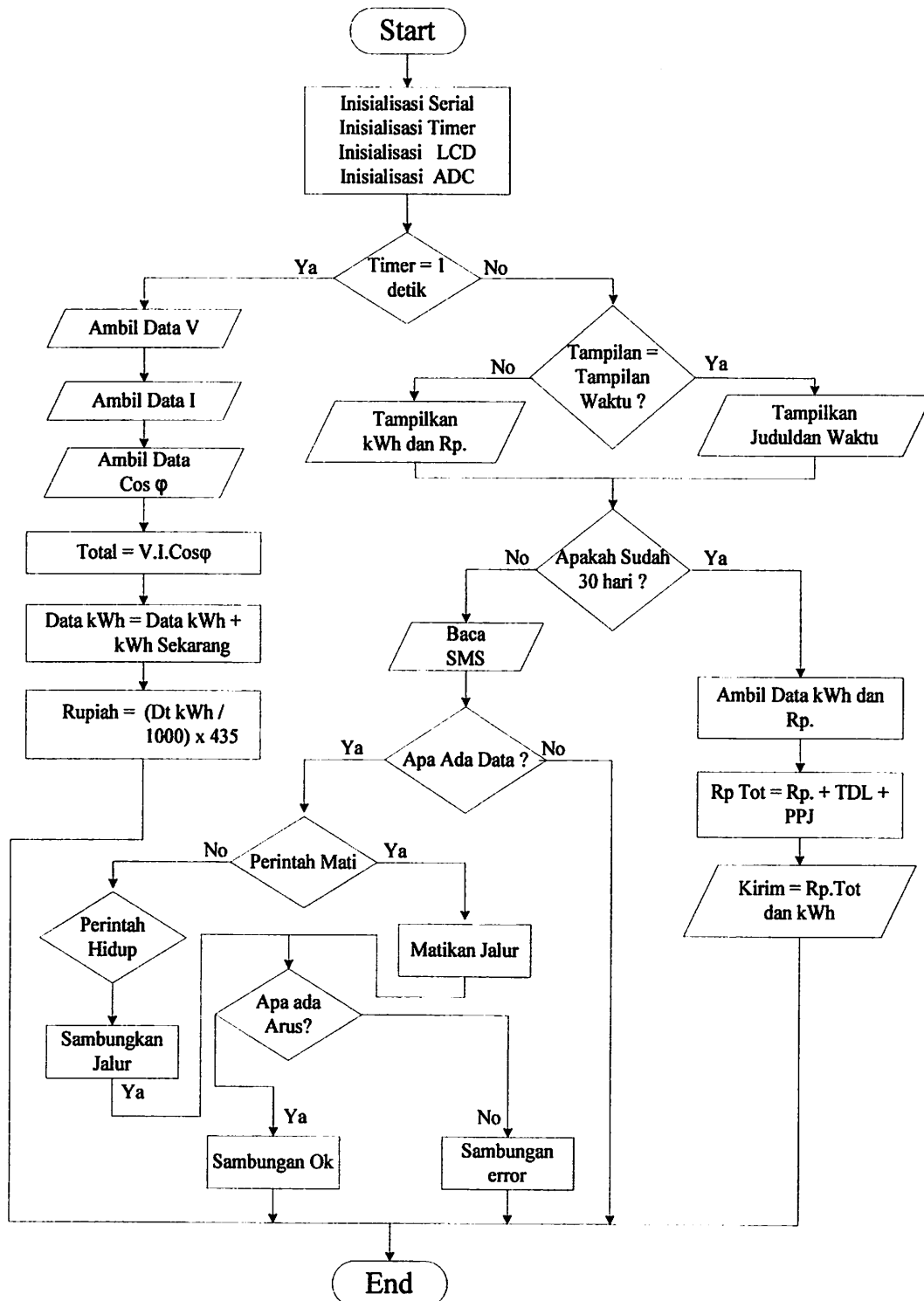
Rangkaian *Push Button*

3.3. Perancangan Perangkat Lunak (*Software*)

Untuk mendukung agar perangkat keras berfungsi sesuai dengan perencanaan, maka diperlukan perangkat lunak sebagai penunjangnya. Perangkat lunak ini sendiri maksudnya adalah suatu program yang kita buat yang nantinya akan ditanam kedalam mikrokontroller ATmega8535. Setelah mikrokontroller tersebut diprogram, maka akan diketahui apakah program yang telah kita buat bekerja sesuai dengan yang kita rencanakan atautkah masih memiliki kesalahan.

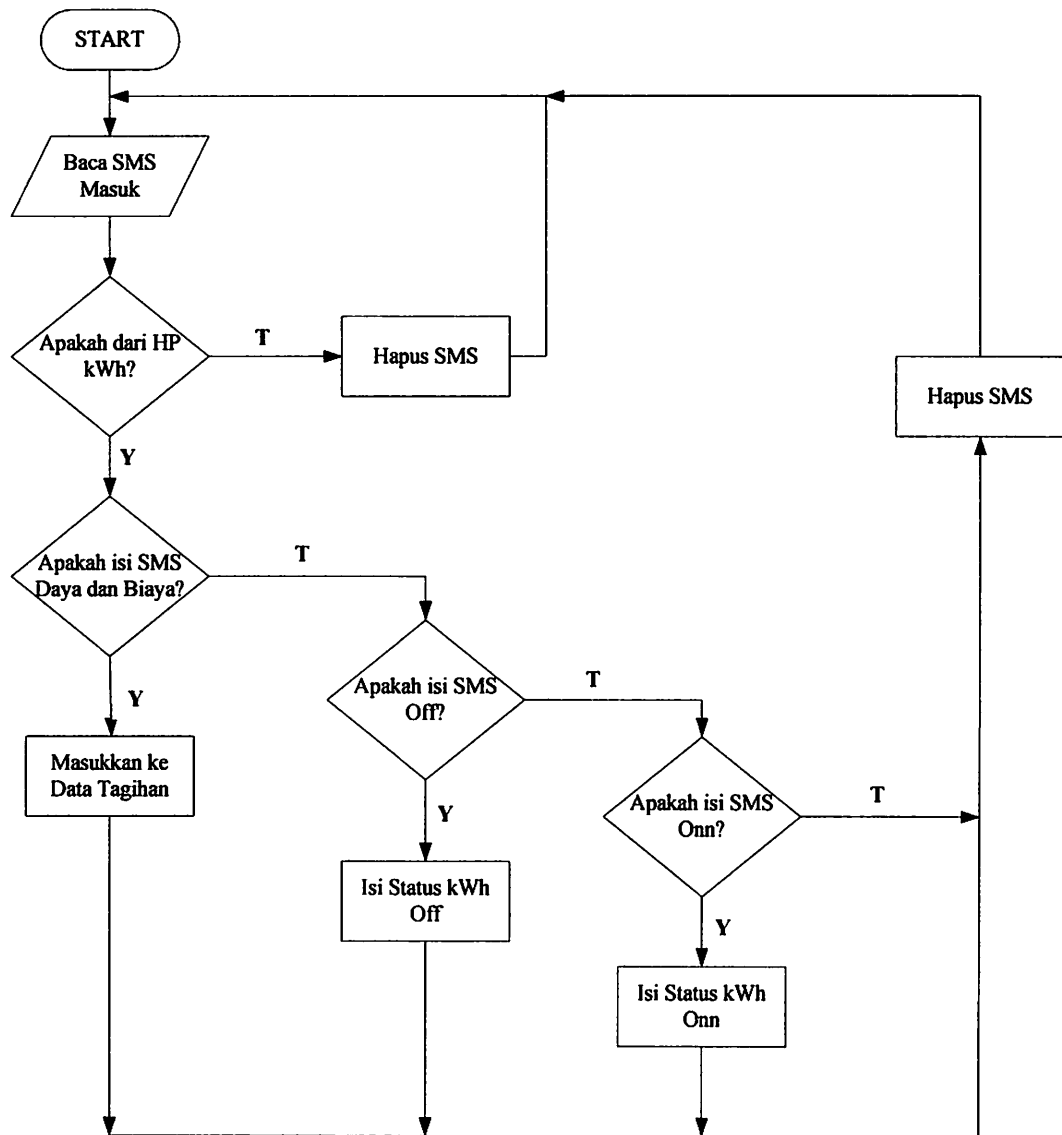
Sistem aplikasi mikrokontroller ATmega8535 ini dapat mengatur dan mengendalikan keseluruhan sistem apabila ada urutan instruksi yang mendefinisikan secara jelas urutan tugas yang harus dikerjakan. Urutan instruksi ini sangat penting untuk didefinisikan, karena mikrokontroller dapat bekerja secara pasti sesuai dengan instruksi yang telah dibuat.

3.3.1. Flowchart pada kWh Meter



Gambar 3.10. Flowchart pada kWh Meter

3.3.2. Flowchart untuk Program Delphi



Gambar 3.11. Flowchart pada Pusat untuk Data Pelanggan Listrik

BAB IV

HASIL PENGUJIAN ALAT

Sebagai cara untuk mengetahui unjuk kerja dari sistem yang dirancang dan dibuat, maka dilakukan pengujian alat. Pengujian ini meliputi pengujian sensor tegangan, sensor arus, pengujian handphone, dan pengujian keseluruhan sistem dengan beban.

4.1. Pengujian Rangkaian Sensor Tegangan dan Pengkondisi Sinyal

4.1.1. Tujuan

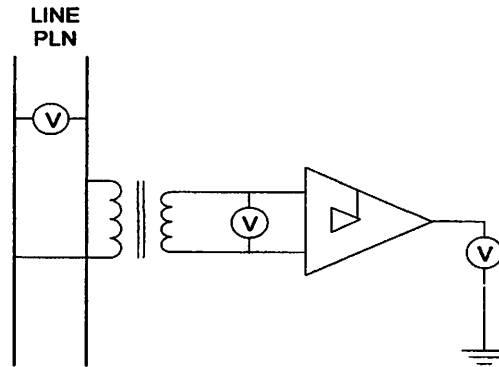
Untuk mengetahui tegangan keluaran rangkaian regulator dan rangkaian pengkondisi sinyal

4.1.2. Alat dan Bahan

1. Voltmeter Digital M-890C
2. Rangkaian Alat.

4.1.3. Langkah Pengujian

1. Persiapan untuk mengaktifkan peralatan
2. Mengatur tegangan jala-jala listrik dan mencatat hasilnya ke dalam tabel
3. Menguji tegangan pada sensor menggunakan voltmeter
4. Mengulangi kembali langkah 2, 3, dan 4 sesuai dengan beban yang digunakan.



Gambar 4.1

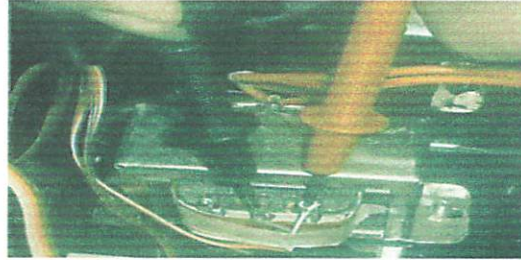
Rangkaian Pengujian Sensor Tegangan dan Pengkondisi Sinyal

4.1.4. Hasil Pengujian

Tabel 4.1 Data Percobaan Sensor Tegangan

Beban Listrik		V _(S) (Volt)	V _(AC) Sensor (Volt)	V _(DC) Penyearah (mV)
Lampu Pijar	Tanpa Beban	223	8.80	250
	60 Watt	222	8.74	246
	100 Watt	220	8.72	240
	160 Watt	219	8.67	230
	200 Watt	218	8.62	227
	260 Watt	217	8.58	217

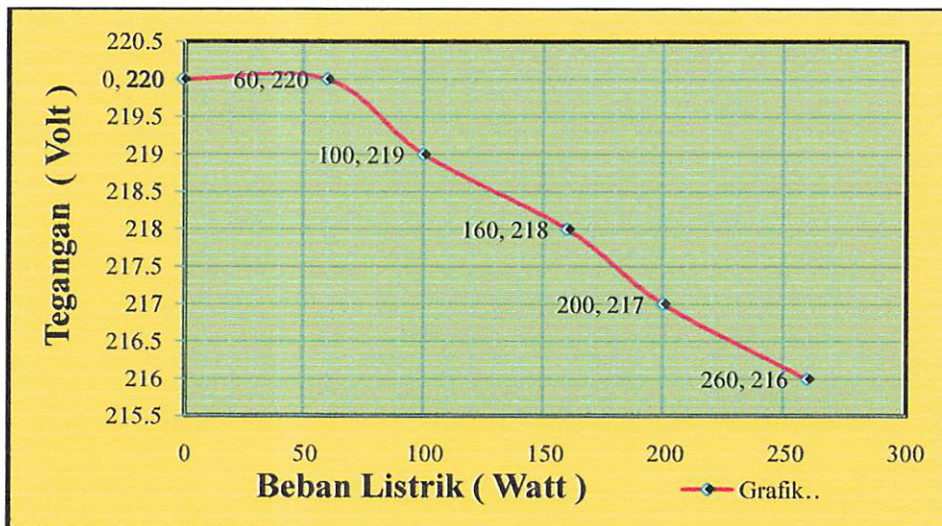
Dari tabel diatas dapat disimpulkan bahwa semakin besar tegangan sumber yang masuk, maka semakin besar keluaran hasil tegangan sensornya, hal ini disebabkan karena adanya drop tegangan yang terjadi pada sensor tegangan, dan semakin besar daya (beban) yang digunakan, maka *output* tegangan (V_{AC} Sensor) semakin kecil.



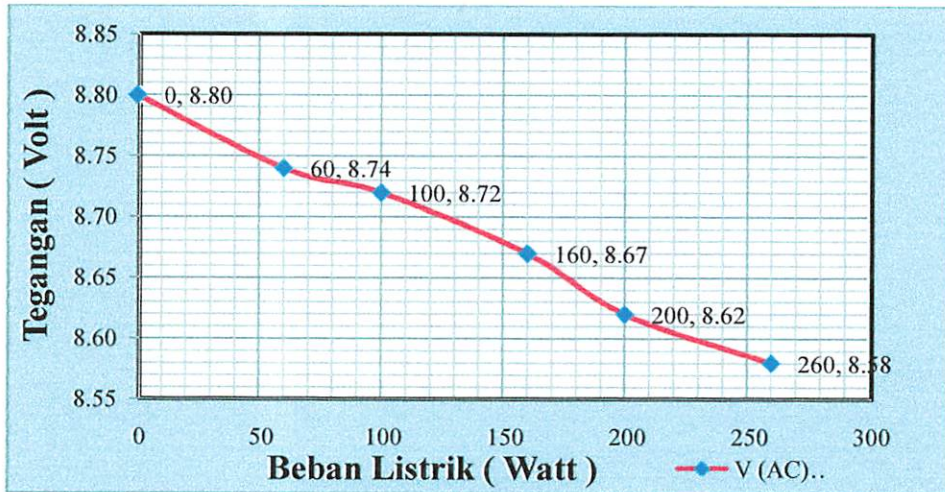
Gambar 4.2
Menguji Sensor Tegangan



Gambar 4.3
Tampilan pada Voltmeter



Grafik 4.1 Pengujian Tegangan Sumber



Grafik 4.2 Pengujian Sensor Tegangan

4.2 Pengujian Rangkaian Sensor Arus dan Pengkondisi Sinyal

4.2.1 Tujuan

Untuk mengetahui arus yang mengalir melewati sensor dan penguatan pada rangkaian pengkondisi sinyal.

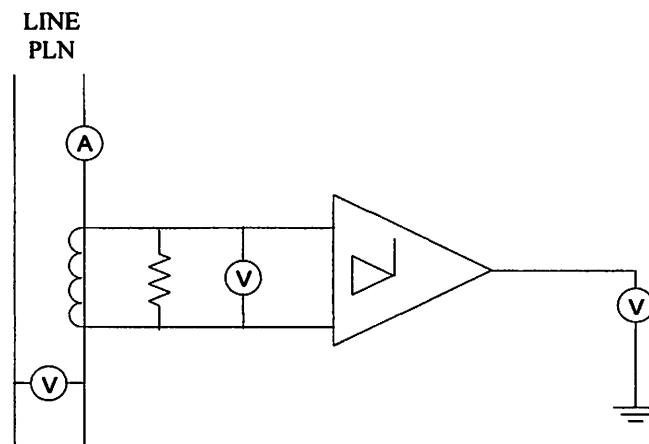
4.2.2 Alat dan Bahan

1. Ampere meter digital
2. Voltmeter digital M-890C
3. Rangkaian alat

4.2.3 Langkah Pengujian

1. Persiapan untuk pengaktifkan peralatan.
2. Menguji tegangan jala-jala dan mencatat hasilnya ke dalam tabel.

3. Menguji arus beban pada Ampere meter.
4. Menguji tegangan pada sensor menggunakan Voltmeter.
5. Mengulangi kembali langkah 2, 3 dan 4 sesuai dengan beban yang digunakan.



Gambar 4.4

Rangkaian Pengujian Sensor Arus dan Pengkondisi Sinyal

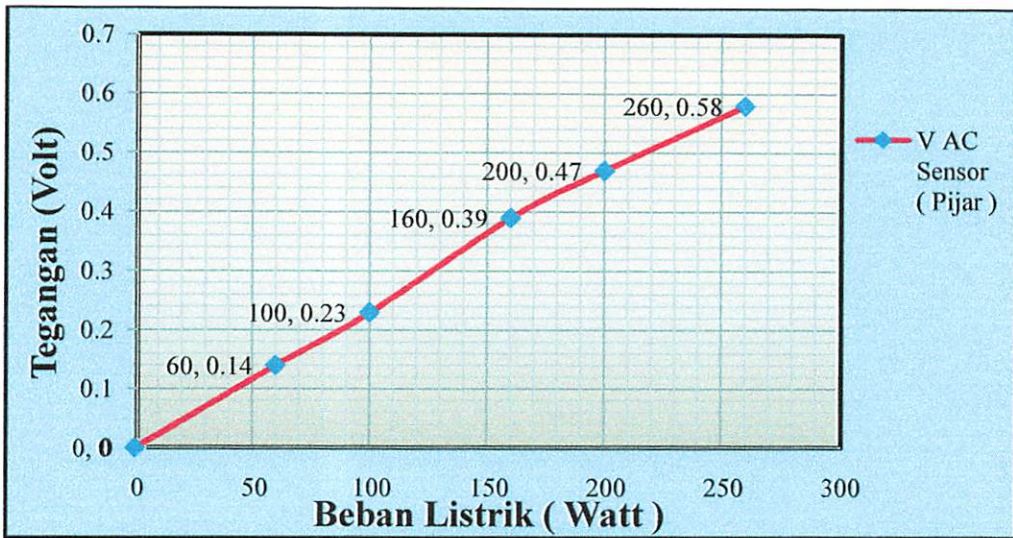
4.2.4 Hasil Pengujian

Tabel 4.2 Data Percobaan Sensor Arus

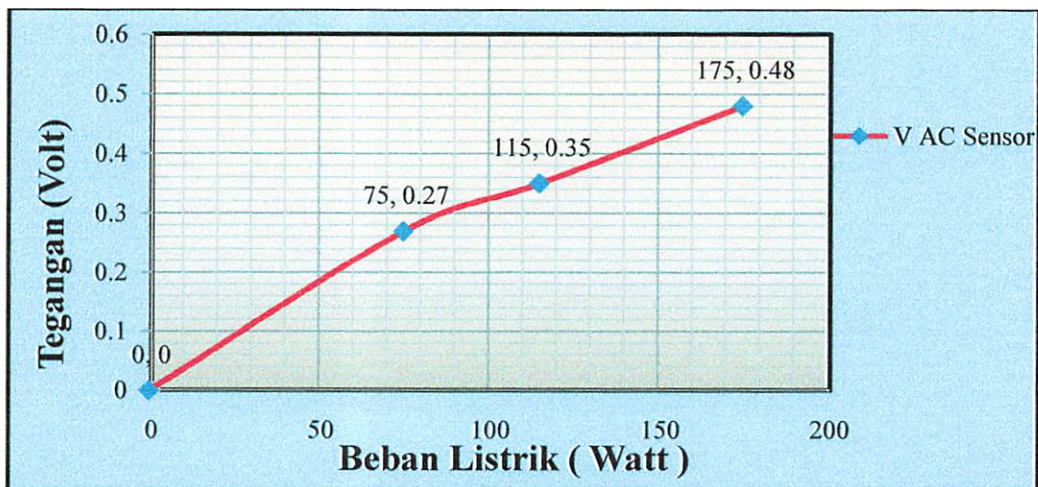
Beban Listrik		V (s) (Volt)	Ampere Meter (Ampere)	V (AC) Sensor (Volt)	V (DC) Penyearah (mV)
Lampu Pijar	Tanpa Beban	220	0	0	0
	60 Watt	220	0,28	0,14	156
	100 Watt	220	0,44	0,23	190
	160 Watt	220	0,72	0,39	258
	200 Watt	220	0,89	0,47	308
	260 Watt	220	1,17	0,58	382

Lampu TL dan Pijar	15 + 60 Watt	220	0,48	0,27	310
	15 + 100 Watt	220	0,62	0,35	344
	15 + 160 Watt	220	0,87	0,48	462

Dari tabel diatas dapat disimpulkan bahwa semakin besar arus yang masuk maka semakin besar pula tegangannya.



Grafik 4.3 Pengujian Sensor Arus untuk Lampu Pijar



Grafik 4.3 Pengujian Sensor Arus untuk Lampu TL dan Pijar



Gambar 4.5
Tampilan pada Amperemeter



Gambar 4.6
Menguji Sensor Arus



Gambar 4.7
Tampilan pada Sensor Arus

4.3 Pengujian Cos ϕ

4.3.1 Tujuan

Untuk mengetahui nilai Cos ϕ yang diperoleh dari lebar pulsa antara Arus dan Tegangan.

4.3.2 Alat dan Bahan

1. *Oscilloscop*
2. Rangkaian Alat yang digunakan

4.3.3 Langkah Pengujian

1. Menghidupkan *Oscilloscop* dan menjalankan start pengukuran rangkaian alat yang nilainya dihasilkan oleh sensor arus dan tegangan.
2. Mengamati dan mencatat logika hasil keluaran IC TC5081 yang ditandai dengan adanya pulsa antara tegangan dan arus.

Tabel 4.3 Data Pengujian Cos ϕ

Beban Listrik	T (ms)	t (ms)	Pengujian	
			(ϕ)	Cos ϕ
Lampu TL 15Watt	20	19,90	358,2	0.999

Nilai Cos ϕ ini dapat berubah-ubah sesuai dengan beban yang digunakan.

4.3.4 Hasil Pengujian

Dari tabel diatas diketahui :

$$\text{Lampu TL} = 15 \text{ Watt}$$

$$t = 19,90 \text{ ms}$$

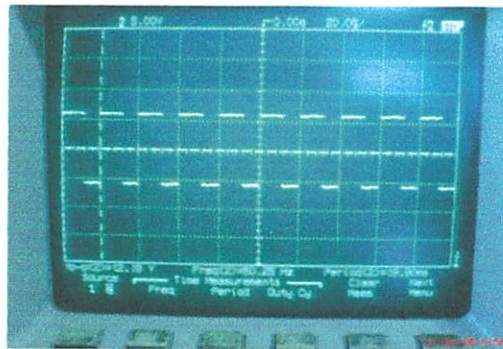
$$T = \frac{1}{\text{Frekuensi Listrik}}$$

$$T = \frac{1}{50 \text{ Hz}} = 0,02 \text{ s} = 20\text{ms}$$

Dengan perbandingan bahwa: $\frac{T}{t} = \frac{360^0}{\varphi}$ maka diperoleh rumus

sebagai berikut :
$$\varphi = \frac{t}{T} \times 360^0 = \frac{19,90\text{ms}}{20\text{ms}} \times 360^0$$

$$\varphi = 358,2^0 \quad \text{Cos } \varphi = 0,999$$



Gambar 4.8

Tampilan pada *Oscilloscop*

4.4 Pengujian Perbandingan antara kWh meter digital yang dirancang dengan kWh meter digital *Standart* PLN dengan tipe DSM-IC030A Merk SAINT.

4.4.1 Tujuan

Untuk mengetahui hasil perbandingan antara kWh meter alat dengan kWh meter digital tipe DSM-IC030A merk SAINT (*Standart* PLN), maka dapat diketahui besar *error* dari kWh meter digital yang dirancang.

4.4.2 Alat dan Bahan

1. kWh meter perancangan.
2. kWh meter digital *type* DSM-IC030A Milik PLN dengan Merk SAINT.
3. Beban yang digunakan.
4. *Stopwatch*.

4.4.3 Langkah Pengujian

1. Persiapan peralatan.
2. Menyalakan kedua kWh meter dengan menghubungkan tegangan sumber pada jala-jala listrik PLN.
3. Mencatat tampilan awal LCD pada masing-masing kWh meter.
4. Menghidupkan beban yang digunakan dan menjalankan *Stopwatch* selama 43 menit 12 detik.
5. Matikan beban yang digunakan.
6. Kemudian mencatat hasil yang ditampilkan pada LCD masing-masing kWh meter dan
7. Menghitung hasil dari pengujian tersebut.

4.4.4 Hasil Pengujian

Dengan mengamati dan mencatat hasil tampilan LCD pada masing-masing Kwh meter, dengan beban yang digunakan sebesar 260W selama 43 menit 12 detik, untuk tampilan LCD pada kWh meter perancangan menghasilkan energi

sebesar = 0.206704 Kwh dan tampilan LCD pada kWh meter tipe DSM-IC030A sebesar = 0.2 Kwh.

Sehingga dapat diketahui kesalahan relatifnya (*Error*) dari Kwh meter alat terhadap Kwh meter digital dengan tipe DSM-IC030A menggunakan rumus :

$$\%Error = \frac{(Pengukuran - Perhitungan)}{Perhitungan} \times 100\%$$

$$E = \frac{0.206704 - 0.2}{0.2} \times 100\%$$

$$= 3.352 \%$$



Gambar 4.9

Kwh Meter Digital Merk SAINT dengan Tipe DSM-IC030A

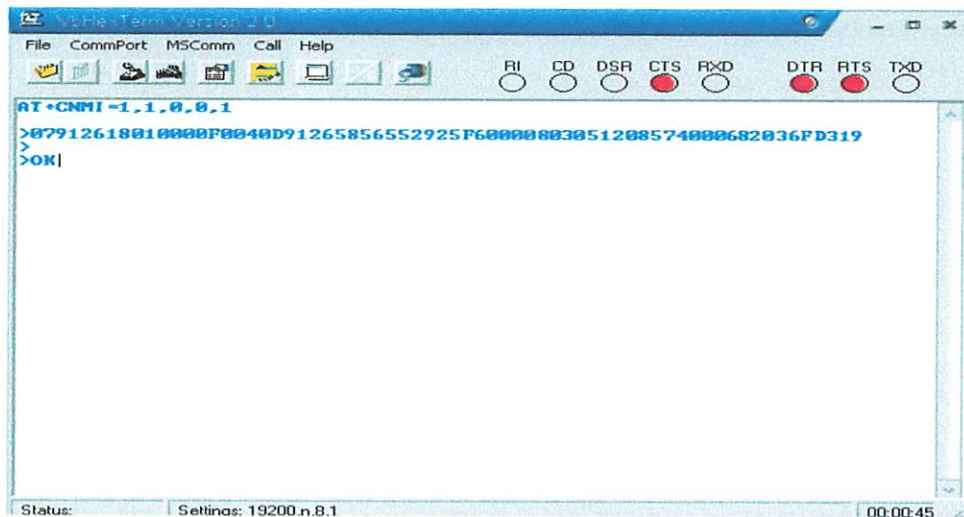


Gambar 4.10

Kwh Meter Digital Perancangan

4.5 Pengujian *Handphone*

Pengujian handphone dilakukan dengan cara menghubungkan handphone dengan PC dengan menggunakan kabel data. Hasil pengujian ditampilkan dengan program Serial tester.



Gambar 4.11

Hasil Pengujian *Handphone*

4.5.1 Pembahasan

Dari hasil pengujian didapat PDU adalah :

07912658050000F0, 04, 0D91, 265856552925F6, 00, 00, 803051208574, 00, 06,
82036FD319

Maka kita dapat mengartikan satu persatu dari kode hasil pengujian PDU tersebut bahwa:

1. (07912618010000F0)

Merupakan SMS-Centre untuk operator seluler termasuk tipe kode nasional atau internasional dan no *service centre*.

2. (04)

Merupakan tipe SMS untuk tipe terima

3. (0D91)

Merupakan panjang no *handpone* pengirim dan tipe kode nasional maupun internasional.

4. (265856552925F6)

Yang merupakan SMS tersebut dikirim dari no *handpone* 265856552925.

5. (00)

Merupakan SMS tersebut diterima dalam bentuk SMS (00).

6. (00)

Yang merupakan skema data *encoding*

7. (803051208574)

Merupakan waktu SMS tersebut sampai di SMS kWh Meter yaitu pada tanggal 08-03-15, pukul 20:58:47 WIB

8. (00)

Merupakan SMS tersebut tidak memiliki batas waktu valid

9. (06)

Merupakan panjang pasangan dari isi SMS

10. (82036FD319)

Merupakan isi SMS berupa off

4.6. Pengujian Keseluruhan Sistem

Dalam pengujian keseluruhan sistem ini dilakukan dalam tiga tahap yaitu pengujian untuk pengiriman pemakaian daya dan biaya, pengujian untuk menonaktifkan arus pada beban dan pengujian pengaktifan arus pada beban.

Sistem kWh meter menggunakan 1 fasa dengan fasa pencatat tarif tunggal pada rumah tangga. Sistem pada pusat menggunakan Laptop yang telah dilengkapi program Delphi dan telah dihubungkan dengan *Handphone*.

4.6.1. Pengujian Pengiriman Pemakaian Energi Listrik dan Biaya

4.6.1.1. Tujuan

Untuk mengetahui apakah *Listing* Program untuk mengirimkan SMS dari Mikrokontroler ke *handpone* penerima bisa dilakukan. Pengujian ini juga untuk mengetahui apakah Mikrokontroler ATmega8535 dapat digunakan dengan baik.

4.6.1.2. Peralatan yang Digunakan

1. Rangkaian Mikrokontroler sederhana yang telah di isi program.
2. Catu Daya.
3. *Handpone* tipe Siemen M35 untuk media pengiriman.
4. Ponsel Penerima yaitu Nokia 8250.
5. *Push Button* untuk mengatur tanggal pengiriman.

4.6.1.3. Langkah Pengujian

1. Membuat program kirim sms untuk di tuliskan ke mikrokontroller
2. Menghubungkan *handpone* pada alat
3. Memberikan pengaturan waktu kirim pada port PC.0, PC.1 dan PC.2 dengan menggunakan *Push Button*
4. Melihat hasil pada PC dengan program delphi yang sudah terhubung pada *handpone* penerima.



The screenshot shows the PT. PLN website interface. At the top, there is a header with the PT. PLN logo and the date 3/25/2008. Below the header is a table with the following data:

No Record	No Rekening	Energi Terpakai	Biaya	Bulan	Status
114	00001	000,80	050	03 2008	Sudah Membayar
112	00001	0000,00	000000	03 2008	Sudah Membayar
122	00001	0000,01	000005	03 2008	Belum Membayar
123	00001	0000,02	000011	03 2008	Belum Membayar
117	00001	0000,04	00001	03 2008	Belum Membayar
113	00001	0001,65	00420	03 2008	Sudah Membayar

Below the table, there are buttons for 'Cari', 'View All', 'Bayar', and 'Report'. To the right, there is a summary table:

No Rek.	Kich	Rp	Bulan
00001	0000,02	000011	03 2008

Below the summary table, the text 'Belum Membayar' is displayed in red. At the bottom of the page, there are icons for a person, a power plug, and a globe.

Gambar 4.12

Proses Penerima Pemakaian Energi Listrik dan Biaya

4.6.2. Pengujian Penonaktifan dan Pengaktifan Beban Listrik

Pada pengujian ini hanya dilakukan pengamatan rangkaian pemutus aliran listrik yang terpasang pada beban pelanggan

4.6.2.1. Langkah-langkah Pengujian

1. Buka form Menunggak.
2. Pilih tombol matikan kWh

Jika dipilih “Matikan kWh” maka akan ditampilkan status kWh pelanggan off.

3. Setelah itu pilih nomor rekening yang akan diputus.
4. Tekan tombol “Matikan kWh”.



Gambar 4.13

Proses Pemutusan Aliran Listrik pada Beban

5. Untuk mengaktifkan kembali tekan tombol “Hidupkan kWh”.



Gambar 4.14

Proses Menghidupkan Aliran Listrik pada Beban

Tabel 4.4 Hasil Pengamatan

No	Tombol Yang Ditekan	SMS Yang Dikirim	Keadaan Beban Lampu
1	Matikan kWh	off	Mati
2	Hidupkan kWh	on	Nyala
3	Matikan kWh	off	Mati
4	Hidupkan kWh	on	Nyala
5	Matikan kWh	off	Mati
6	Hidupkan kWh	on	Nyala

Dari hasil pengujian pada Tabel 4.4. dapat dilihat bahwa dalam 6 kali percobaan maka kondisi beban sesuai dengan yang diharapkan.

BAB V

PENUTUP

5.1. Kesimpulan

Pengiriman data berupa pemakaian daya dan biaya pada alat ini yaitu dengan komunikasi melalui sms sehingga pengirimannya secara langsung dapat diterima oleh server. Setelah data dikirim oleh masing-masing kwh meter dengan nomor hp kWh yang berbeda maka secara langsung data akan disimpan di *database server*.

Dari perancangan, pembuatan dan pengujian alat maka dapat diketahui kesimpulan sebagai berikut :

1. SMS akan dikirimkan hanya ke nomor yang sudah diset pada mikrokontroler untuk setiap bulannya.
2. Hasil percobaan dari pengujian perbandingan antara kwh alat dengan kWh digital Merek SAINT, dengan tipe DSM-IC030A maka dapat diambil kesimpulan *error* dari hasil perbandingan sebesar 3,352%.
3. Sistem ini bergantung pada jaringan GSM untuk transfer data kWh meter ke pusat.
4. Rata-rata selisih waktu antara pengiriman dan penerimaan SMS pada saat pengujian adalah 9,94 detik.
5. Kapasitas EEPROM 8-bit *internal* ATMEGA8535 sebesar 512 byte yang mampu tulis dalam 100.000 baca tulis.

5.2. Saran-saran

1. Perlu diperhatikan pada saat aliran listrik PLN padam / mati, biar pembacaan pemakaian energi tetap berjalan dan dapat disimpan tepat pada waktunya maka perlu ditambahi rangkaian *power* untuk model *batre charger*.
2. Dalam perkembangan selanjutnya perlu diperhatikan lagi untuk inputan sensor arus dan tegangan pada pengkalibrasian dengan alat yang sebenarnya, lebih baik menggunakan rangkaian atau komponen yang dilengkapi dengan IC ADE7755 atau persamaanya dan *software* yang handal untuk aplikasi tersebut.
3. Pada *server* untuk *software* delphinya lebih aplikatif ditambahi *print out*.

Daftar pustaka

- ATMega 8535 Datasheet* [http:// www. AVR. Com](http://www.AVR.Com). (Diakses tgl 09 September 2007).
- AT Command Set Reference Manual* [http:// www. Siemens. Com](http://www.Siemens.Com). (Diakses tgl 12 desember 2007).
- Gunawan, Malvino Hanafi. 1999. *Prinsip-Prinsip Elektronik*. Jakarta: Gava Media.
- Irawan, Josep Dedy dan Widodo, Budhi Romy. 2007. *Interfacing Paralel dan Serial menggunakan Delphi*. Malang: Graha Ilmu.
- ITN Malang. 2006. *Buku Panduan Belajar Borland Delphi*. Malang: Laboratorium Elektronika Digital.
- Khang, Bustam. 2002. *Trik Pemrograman Aplikasi Berbasis SMS*. Jakarta: P.T. Elex Media Komputindo.
- Malik, Jaja Jamaludin. 2006. *Membuat Form Cantik untuk Aplikasi Delphi*. Yogyakarta: Andi.
- Malik, Jaja Jamaludin. 2005. *Tip dan Trik Delphi*. Yogyakarta: Andi.
- MAX232 Datasheet*. [http:// www. RS232. com](http://www.RS232.com). (Diakses tgl 09 September 2007).
- Pranata, Antony. 1997. *Pemrograman Borland Delphi 6*. Yogyakarta: Andi Offset.
- Sapiie, Soedjana dan Nishino, Osamu. 2005. *Pengukuran dan Alat-alat Ukur listrik*. Jakarta : PT. Pradnya Paramita
- TC5081BP Datasheet*. [http://www. En. Toshiba. org. com](http://www.En.Toshiba.org.com). (Diakses tgl 09 September 2007).
- Wardana, Lingga. 2006. *Belajar Mikrokontroler AVR Seri ATMega 8535*. Yogyakarta: CV. Andi Offset.
- Zakaria, Marcus Teddy dan Widiadi, Josef. 2006. *Aplikasi SMS untuk Berbagai Keperluan*. Bandung: Informatika.

LAMPIRAN



FORM BIMBINGAN SKRIPSI

Nama : Moch. Asdiyudin
NIM : 02.17.136
Masa Bimbingan : 09 Nopember 2007 s/d 09 Mei 2008
Judul : Perancangan dan Pembuatan kWh Meter Digital dengan Pengiriman Informasi Tagihan dan Pemutusan Hubungan Listrik Melalui SMS dengan Kendali PC

No	Tanggal	Uraian	Paraf
1	15/11/07	Konsultasi labor blk & trj.	
2	12/12/07	Kons. pembuatan alat.	
3	06/02/08	Konsultasi bab I & II	
4	16/02/08	Konsultasi bab III	
5	20/02/08	Konsultasi Pengujian	
6	27/02/08	Konsultasi bab IV & V	
7	05/03/08	Kons. makalah seminar.	
8	08/03/08	Acc seminar	
9	12/03/08	Acc ujian	
10			

Malang,
Dosen Pembimbing

Ir. Teguh Herbasuki, MT
NIP. Y. 1038900209

Form S-4a



FORM BIMBINGAN SKRIPSI

Nama : Moch. Asdiyudin
NIM : 02.17.136
Masa Bimbingan : 09 Nopember 2007 s/d 09 Mei 2008
Judul : Perancangan dan Pembuatan kWh Meter Digital dengan Pengiriman Informasi Tagihan dan Pemutusan Hubungan Listrik Melalui SMS dengan Kendali PC

No	Tanggal	Uraian	Paraf
1	10/07/08	Proposuil	EB
2	12/08/08	Konsultasi skema / blok diagram system	EB
3	06/08/08	Konsultasi Prob I & II (di revisi)	EB
4	12/08/08	Kee Prob I & Prob 3 lanjutkan ke pengujian alat	EB
5	19/08/08	Rwisi Prob IV Pusat. masalah & seminar.	EB
6	07/08/08	Kee laporan skripsi	EB
7			
8			
9			
10			

Malang, 17-03-08
Dosen Pembimbing

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


Form S-4b



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

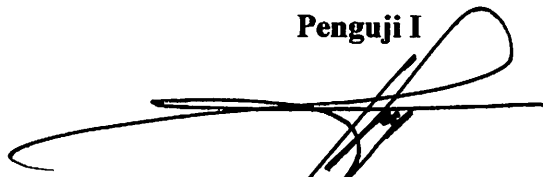
LEMBAR PERBAIKAN SKRIPSI

Nama : Moch. Asdiyudin
Nim : 02.17.136
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul : "PERANCANGAN DAN PEMBUATAN KWH METER
DIGITAL DENGAN PENGIRIMAN INFORMASI
TAGIHAN DAN PEMUTUSAN HUBUNGAN LISTRIK
MELALUI SMS DENGAN KENDALI PC"
Hari/Tgl Skripsi : Selasa, 18 Maret 2008

No.	Materi Perbaikan	Paraf
1.	Jelaskan cara komunikasi mikrokontroler ke PC lewat handphone.	
2.	Tes kembali pengujian Cos ϕ .	

Diperiksa / Disetujui

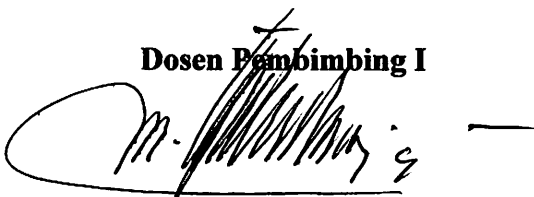
Penguji I



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

Mengetahui

Dosen Pembimbing I



Ir. Teguh Herbasuki, MT
NIP. Y. 1038900209

Dosen Pembimbing II



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



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

LEMBAR PERBAIKAN SKRIPSI

Nama : Moch. Asdiyudin
Nim : 02.17.136
Jurusan : Teknik Elektro S1
Konsentrasi : Teknik Elektronika
Judul : "PERANCANGAN DAN PEMBUATAN KWH METER
DIGITAL DENGAN PENGIRIMAN INFORMASI
TAGIHAN DAN PEMUTUSAN HUBUNGAN LISTRIK
MELALUI SMS DENGAN KENDALI PC"
Hari/Tgl Skripsi : Selasa, 18 Maret 2008

No.	Materi Perbaikan	Paraf
1.	Jelaskan cara komunikasi mikrokontroler ke PC lewat handphone.	
2.	Tes kembali pengujian Cosp.	

**Diperiksa / Disetujui
Penguji II**

DR. Cahyo Crysdiyan, Msc
NIP. 1030400412

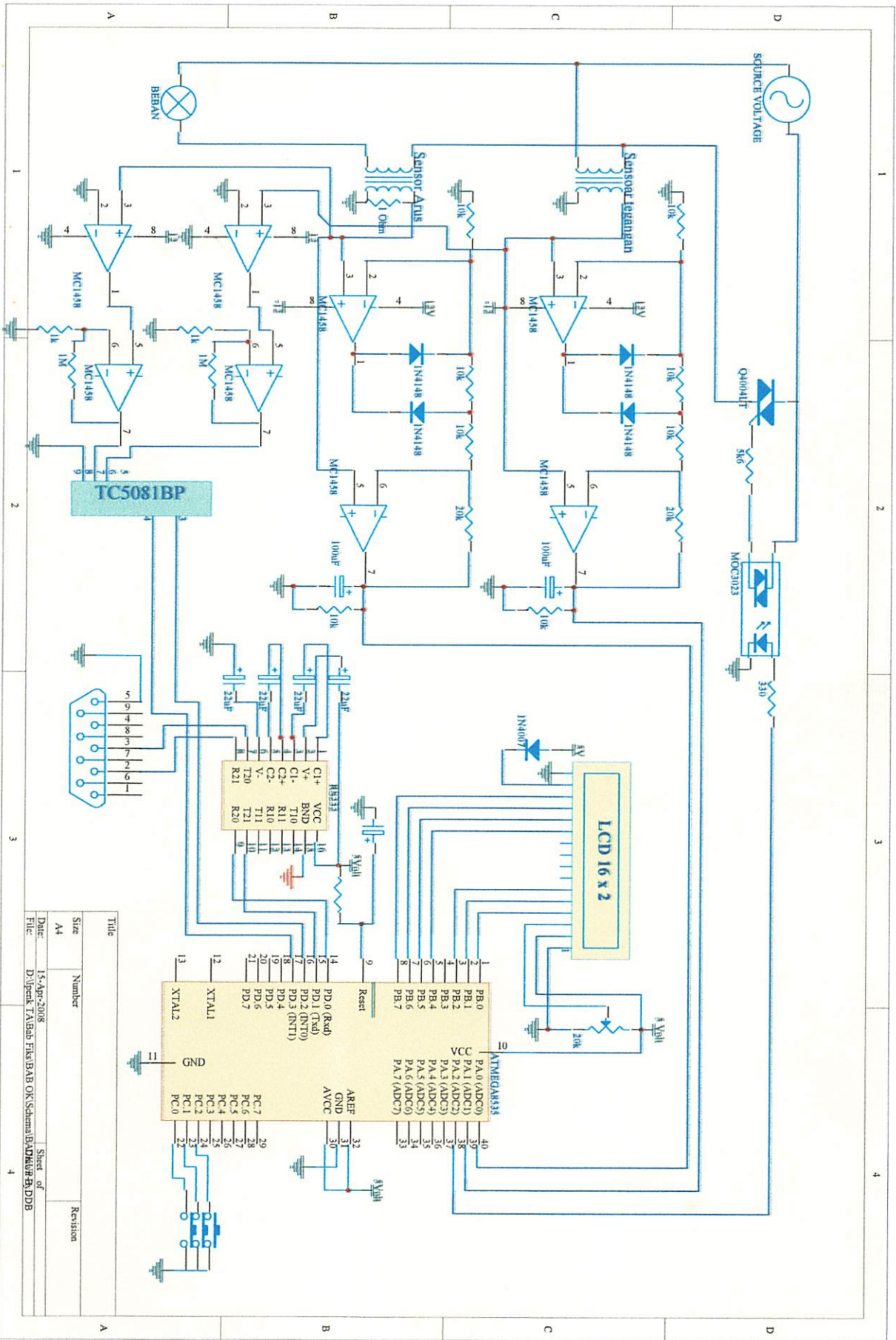
Mengetahui

Dosen Pembimbing I

Ir. Teguh Herbasuki, MT
NIP. Y. 1038900209

Dosen Pembimbing II

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



Title	Size	Number	Revision
A1	A1		

Date: 15-Apr-2008
 File: D:\ipenk TA.Bab Fils.BAB OK.Sahman.BA.DIRI.BA.DDB

Sheet of
4

software

```
/*
v x i x cp
220 x 45 x 99 /10000
*/

/*deklarasi file - file yang dibutuhkan untuk membangun aplikasi*/
#include <mega8535.h> //file yg berhubungan dengan register 2 ATMEGA8535
#include <delay.h> //file yg berhubungan dengan fungsi - fungsi penundaan
#include <string.h> //file yg berhubungan dengan operasi manipulasi string
#include <ctype.h> //file yg berhubungan dengan operasi manipulasi string
#include <stdlib.h>
#include <MyEEP.C>
#include "serial.c" //file yg berhubungan dengan operasi
//pengiriman dan penerimaan data secara serial
#include <pdu.h> //file yg berhubungan dengan penanganan PDU

#asm
.equ __lcd_port=0x18 ;PORTB
#endasm
#include <lcd.h> //file yg berhubungan dengan fungsi - fungsi LCD

/*deklarasi konstanta*/
#define ADC_VREF_TYPE 0xC0
#define TOIE1 2
#define RXEN 4
#define sbuf_size 125
#define text_size 15
#define PDU_size 30
#define delay_tombol 200
#define pemutus PORTA.2
#define tombol_menu PINC.0
#define tombol_naik PINC.1
#define tombol_turun PINC.2

unsigned char flash f_noSMSC[] = "6285500000"; //IM3
//unsigned char flash f_noSMSC[] = "62818445009"; //XL bebas

unsigned char flash f_notujuan[] = "6285655592526";
//unsigned char flash f_notujuan[] = "6285933109428";
//unsigned char flash f_notujuan[] = "623417309603";
unsigned int flash cos[]={1000,999,999,998,997,996,994,992,990,987,984,
981,978,974,970,965,961,956,951,945,939,
933,927,920,913,901,898,891,882,874,866,
857,848,838,829,819,809,798,788,777,766,
754,743,731,719,707,694,681,669,655,642,
629,615,601,587,573,559,544,529,515,500,
484,469,453,438,422,406,390,374,358,342,
325,309,292,275,258,241,224,207,190,173,
156,139,121,104,087,069,052,034,017,000
};

unsigned char
```

software

```
sbuf[sbuf_size], string5[5], string16[16], SMSsemu[PDU_size],  
serial_counter, tmp, panjang, tmpchar, mode_HP, longint[10],  
mode_menu, no_menu, waktu_tekan, phi, sudut;
```

```
unsigned int  
data_tegangan, data_arus, uitemp, uitick1000, hasil_kali, delay_HP,  
phi_count;
```

```
unsigned long int  
bit  
data_kwh, rupiah, selisih_kwh;  
waktunya_kwh, switch_tampilan, waktunya_kirim, masuk_menu;
```

```
#define notel string16  
#define noSMSC string16  
#define notujuan string16  
#define SMSasli sbuf
```

```
typedef struct
```

```
{  
    unsigned char second; //enter the current time, date, month, and year  
    unsigned char minute;  
    unsigned char hour;  
    unsigned char date;  
    unsigned char month;  
    unsigned int year;  
}
```

```
time;  
time RTC;  
#include <RTC.C>
```

```
/******PROTOTYPE FUNCTION*****/  
void set_port();  
void timer1_init(void);  
void adc_init();  
unsigned int read_adc(unsigned char adc_input);  
unsigned char baca_arus();  
unsigned char baca_tegangan();  
void prosedur_kwh();  
void prosedur_HP();  
void pecah_longint(long int data_pecah);  
void tampilan1();  
void tampilan2();  
void cari_sudut();  
void isi_buffer_kirim();  
void cek_waktunya_kirim();  
void simpan();  
void ambil_data_2bulan_yll();  
void prosedur_menu();  
void extract_SMS();  
void kirim(char *isi_SMS);
```

```
interrupt [TIM1_OVF] void timer1_ovf_isr(void)  
{  
    TCNT1H = 0xF0;  
    TCNT1L = 0x60;  
    phi_count++;  
    delay_HP++;  
    waktu_tekan++;  
    if (++uitick1000 >= 1000) //test apa sudah mencapai 1000x interrupt  
    (100ms)  
    {  
        uitick1000=0; //jk sudah terjadi 1000 x interrupt  
        tambah_waktu(); //reset nilai hitungan  
        //lakukan penambahan waktu  
        (dtk,mnt,jam,tgl,bln,thn)  
        waktunya_kwh=1;
```

software

```

}
}

interrupt [USART_RXC] void usart_rx_isr(void)
{
unsigned char ko;
ko=UDR;
sbuf[serial_counter]=ko;
if (++serial_counter>=sbuf_size) serial_counter=0;
}

interrupt [EXT_INT1] void ext_int1_isr(void)
{
    phi_count=0;
}

/*****P R O G R A M   U T A M A*****/
void main()
{
    timer1_init()    ;
    adc_init()       ;
    lcd_init(16)     ;
    usart_init()     ;
    set_port()       ;
    mode_HP=0        ;
    mode_menu=0      ;
    masuk_menu=0     ;
    PORTA.2=0;
    #asm("sei")      ;
    while(1)
    {
        cari_sudut()      ;
        prosedur_kwh()    ;
        prosedur_HP()     ;
        cek_waktunya_kirim();
        prosedur_menu()   ;
    }
}

/*****P R O G R A M   U T A M A*****/

/*=====*/
void set_port()
{
    DDRA.2    =1 ;
    DDRC.0    =0 ;
    PORTC.0   =1 ;
    DDRC.1    =0 ;
    PORTC.1   =1 ;
    DDRC.2    =0 ;
    PORTC.2   =1 ;
}

void timer1_init(void)
{
    TCCR1B = 0x00;    //stop
    TCNT1H = 0xF0;    //setup
    TCNT1L = 0x60;
    TCCR1A = 0x00;
    TCCR1B = 0x01;    //start Timer
    TIMSK  |= (1<<TOIE1); //set 8-bit Timer/Counter1 Overflow Interrupt Enable
}

/*=====*/
unsigned int read_adc(unsigned char adc_input)

```

software

```

{
ADMUX=adc_input|ADC_VREF_TYPE;
ADCSRA|=0x40; //Start the AD conversion
while ((ADCSRA & 0x10)==0); //Wait for the AD conversion to complete
ADCSRA|=0x10;
return ADCW;
}

```

```

void adc_init()
{
    ADMUX=ADC_VREF_TYPE;
    ADCSRA=0x85;
    SFIOR&=0xEF;
}

```

```

unsigned char baca_arus()
{
    char i;
    uitemp=0;
    for (i=0;i<20;i++)
    {
        uitemp=uitemp+read_adc(0);
    }
    return uitemp/20;
}

```

```

unsigned char baca_tegangan()
{
    char i;
    uitemp=0;
    for (i=0;i<20;i++)
    {
        uitemp=uitemp+read_adc(1);
    }
    return uitemp/20;
}

```

```

void pecah_longint(long int data_pecah)
{
    longint[0]=data_pecah/1000000000|0x30;
    longint[1]=data_pecah/100000000%10|0x30;
    longint[2]=data_pecah/10000000%10|0x30;
    longint[3]=data_pecah/1000000%10|0x30;
    longint[4]=data_pecah/100000%10|0x30;
    longint[5]=data_pecah/10000%10|0x30;
    longint[6]=data_pecah/1000%10|0x30;
    longint[7]=data_pecah/100%10|0x30;
    longint[8]=data_pecah/10%10|0x30;
    longint[9]=data_pecah%10|0x30;
}

```

```

void tampilan_waktu()
{
    lcd_gotoxy(0,1);
    lcd_putchar(RTC.hour/10%10|0x30);
    lcd_putchar(RTC.hour%10|0x30);
    lcd_putchar(':');
    lcd_putchar(RTC.minute/10%10|0x30);
    lcd_putchar(RTC.minute%10|0x30);
    lcd_putchar(' ');
    lcd_putchar(' ');
    lcd_putchar(' ');
    lcd_putchar(RTC.date/10%10|0x30);
    lcd_putchar(RTC.date%10|0x30);
    lcd_putchar('/');
    lcd_putchar(RTC.month/10%10|0x30);
    lcd_putchar(RTC.month%10|0x30);
}

```

software

```

lcd_putchar('/');
lcd_putchar(RTC.year/10%10|0x30);
lcd_putchar(RTC.year%10|0x30);
}

```

```

void tampilan1()
{
    lcd_clear();
    lcd_gotoxy(0,0);lcd_putsf("kwhMeter Digital");
    tampilan_waktu();
}

```

```

void tampilan2()
{
    lcd_clear();
    lcd_gotoxy(0,0);
    pecah_longint(data_kwh);
    lcd_putchar(longint[0]);
    lcd_putchar(longint[1]);
    lcd_putchar(longint[2]);
    lcd_putchar(longint[3]);
    lcd_putchar(',');
    lcd_putchar(longint[4]);
    lcd_putchar(longint[5]);
    lcd_putchar(longint[6]);
    lcd_putchar(longint[7]);
    lcd_putchar(longint[8]);
    lcd_putchar(longint[9]);
    lcd_putchar(' ');
    lcd_putchar(' ');
    lcd_putchar('k');
    lcd_putchar('w');
    lcd_putchar('h');
    lcd_gotoxy(0,1);
    strclr(longint,10);
    ltoa(rupiah,longint);
    lcd_putchar('R');
    lcd_putchar('p');
    lcd_putchar('.');
    lcd_puts(longint);
    lcd_putchar(',');
    lcd_putchar('-');
}

```

```

void prosedur_kwh()
{
    if (waktunya_kwh)
    {
        waktunya_kwh=0;
        data_tegangan=baca_tegangan();
        data_arus=baca_arus();
        hasil_kali = data_tegangan*((data_arus*2)/100)*(phi/1000)/3600;
        data_kwh = data_kwh+ hasil_kali;
        rupiah = ((data_kwh / 1000) * 435)/1000 ;
        if ((RTC.second%5)==0) switch_tampilan=!switch_tampilan;

        if (!masuk_menu)
        {
            if (switch_tampilan) tampilan1();
            else tampilan2();
        }
    }
}

```

```

void cek_waktunya_kirim()

```

software

```
{
if ((RTC.date==1)&&(RTC.hour==7)&&(RTC.minute==0)&&(RTC.second==0))
{
    waktunya_kirim=1;
}
}

void prosedur_HP()
{
unsigned char pjpg;
switch(mode_HP)
{
    case 0 : if (waktunya_kirim)
        {
            waktunya_kirim=0;
            ambil_data_2bulan_y11();
            if (selisih_kwh > data_kwh) selisih_kwh = data_kwh;
            else selisih_kwh = data_kwh - selisih_kwh;
            isi_buffer_kirim();
            kirim(SMSasli);
            rupiah=0;
            simpan();
            delay_HP=0;
            mode_HP=3;
        }
        else
        {
            strchr(sbuf,sbuf_size);
            serial_counter=0;
            send_textf("AT+CMGR=1\r\n");
            delay_HP=0;
            mode_HP=1;
        }
        break;
    case 1 : if(delay_HP>=1000)
        {
            pjpg=strlen(sbuf);
            strchr(string5,5);
            strcpy(string5,sbuf,pjpg-3,2);
            if ( (pjpg>50) && (strcmpf(string5,"OK")==0) )
            {
                extract_SMS();
                banding=strcmpf(notel,f_no_server);
                if(banding==0)
                {
                    if (strcmpf(SMSasli,"off")==0)
                    {
                        pemutus=1;
                        status=1;
                    }
                    else if (strcmpf(SMSasli,"on")==0)
                    {
                        pemutus=0;
                        status=0;
                    }
                }
                mode_HP=2;
            }
            else
            {
                mode_HP=2;
                status=3;
            }
            send_textf("AT+CMGD=1\r\n");
        }
        else
        {

```

```

                software
                mode_HP=2;
                status=4;
                //lcd_gotoxy(15,1);lcd_putsf("T");
                //lcd_clear();lcd_putsf("Tidak Ada SMS");
            }
            delay_HP=0;
        }
        break;
    case 2 : if( delay_HP >= 1000)
        {
            strchr(SMSasli,sbuf_size);
            if(status==1)
            {
                strcatf(SMSasli,"Beban OFF");
                kirim(SMSasli);
            }
            else if(status==0)
            {
                strcatf(SMSasli,"Beban ON");
                kirim(SMSasli);
            }
            //else if(status==3)
            //{
            //    strcatf(SMSasli,"AKSES DITOLAK");
            //    kirim(SMSasli);
            //}
            delay_HP=0;
            mode_HP=4;
        }
        break;
    case 3 : if( delay_HP >= 2000) mode_HP=0;
            break;
    case 4 : if( delay_HP >= 1000) mode_HP=0;
            break;
    default : mode_HP=0;
            break;
}
}

void isi_buffer_kirim()
{
    pecah_longint(selisih_kwh);
    strchr(SMSasli,sbuf_size);
    SMSasli[0]=longint[0];
    SMSasli[1]=longint[1];
    SMSasli[2]=longint[2];
    SMSasli[3]=longint[3];
    SMSasli[4]=',';
    SMSasli[5]=longint[4];
    SMSasli[6]=longint[5];
    SMSasli[7]='.';
    totalBayar=0;
    totalBayar=rupiah+27000;
    pecah_longint(totalBayar);
    SMSasli[8]=longint[4];
    SMSasli[9]=longint[5];
    SMSasli[10]=longint[6];
    SMSasli[11]=longint[7];
    SMSasli[12]=longint[8];
    SMSasli[13]=longint[9];
}

void simpan()
{
    unsigned char ttmp,alamat;

```

software

```
alamat=RTC.month-1;
if (alamat==0) alamat=12;
alamat=alamat*10;

ttmp=data_kwh>>0;
EEPROM_write(alamat,ttmp);
alamat++;
ttmp=data_kwh>>8;
EEPROM_write(alamat,ttmp);
alamat++;
ttmp=data_kwh>>16;
EEPROM_write(alamat,ttmp);
alamat++;
ttmp=data_kwh>>24;
EEPROM_write(alamat,ttmp);
}

void ambil_data_2bulan_y11()
{
unsigned char ttmp,alamat;

if (RTC.month==1) alamat = 11;
else if (RTC.month==2) alamat = 12;
else alamat = RTC.month-2;
alamat=(alamat*10)+3;
ttmp=EEPROM_read(alamat);
selisih_kwh=ttmp;
alamat--;
ttmp=EEPROM_read(alamat);
selisih_kwh=(selisih_kwh<<8)|ttmp;
alamat--;
ttmp=EEPROM_read(alamat);
selisih_kwh=(selisih_kwh<<8)|ttmp;
alamat--;
ttmp=EEPROM_read(alamat);
selisih_kwh=(selisih_kwh<<8)|ttmp;
}

void prosedur_menu()
{
switch(mode_menu)
{
case 0 : if(tombol_menu==0)
{
mode_menu=1;
if (++no_menu>=6) no_menu=0;
if (no_menu==0)
{
masuk_menu=0;
RTC.second=0;
}
else
{
masuk_menu=1;
lcd_clear();
tampilan_waktu();
}
if (no_menu==1)
{
lcd_gotoxy(1,0);
lcd_putsf("H");
}
else if (no_menu==2)
{
lcd_gotoxy(4,0);
lcd_putsf("M");
}
}
}
```

```

                                software
else if (no_menu==3)
{
    lcd_gotoxy(9,0);
    lcd_putsf("D");
}
else if (no_menu==4)
{
    lcd_gotoxy(12,0);
    lcd_putsf("M");
}
else if (no_menu==5)
{
    lcd_gotoxy(15,0);
    lcd_putsf("Y");
}
}
if (tombol_naik==0)
{
    if (masuk_menu)
    {
        if (no_menu==1)
        {
            RTC.hour++;
            if (RTC.hour>=24) RTC.hour=0;
        }
        else if (no_menu==2)
        {
            RTC.minute++;
            if (RTC.minute>=60) RTC.minute=0;
        }
        else if (no_menu==3)
        {
            RTC.date++;
            if (RTC.date>=31) RTC.date=1;
        }
        else if (no_menu==4)
        {
            RTC.month++;
            if (RTC.month>=13) RTC.month=1;
        }
        else if (no_menu==5)
        {
            RTC.year++;
            if (RTC.year>=100) RTC.year=0;
        }
        tampilan_waktu();
        mode_menu=3;
    }
    else
    {
        switch_tampilan=1;
    }
}
if (tombol_turun==0)
{
    if (masuk_menu)
    {
        if (no_menu==1)
        {
            RTC.hour--;
            if (RTC.hour==255) RTC.hour=23;
        }
        else if (no_menu==2)
        {
            RTC.minute--;
            if (RTC.minute==255) RTC.minute=59;
        }
    }
}

```

```

                software
            else if (no_menu==3)
            {
                RTC.date--;
                if (RTC.date<=0) RTC.date=30;
            }
            else if (no_menu==4)
            {
                RTC.month--;
                if (RTC.month<=0) RTC.month=12;
            }
            else if (no_menu==5)
            {
                RTC.year--;
                if (RTC.year==255) RTC.year=99;
            }
            tampilan_waktu();
            mode_menu=5;
        }
        else
        {
            switch_tampilan=0;
        }
    }
    break;
case 1 : if(tombol_menu==1)
        {
            waktu_tekan=0;
            mode_menu=2;
        }
        break;
case 2 : if (waktu_tekan>=delay_tombol) mode_menu=0;
        else
        {
            if(tombol_menu==0) mode_menu=1;
        }
        break;
case 3 : if (tombol_naik==1)
        {
            waktu_tekan=0;
            mode_menu=4;
        }
        break;
case 4 : if (waktu_tekan>=delay_tombol) mode_menu=0;
        else
        {
            if(tombol_naik==0) mode_menu=3;
        }
        break;
case 5 : if (tombol_turun==1)
        {
            waktu_tekan=0;
            mode_menu=6;
        }
        break;
case 6 : if (waktu_tekan>=delay_tombol) mode_menu=0;
        else
        {
            if(tombol_turun==0) mode_menu=5;
        }
        break;
default : mode_menu=0;
         no_menu=0;
         break;
}
}
}

```

software

```

//=====
//SMS RUTIN

void extract_SMS()
{
    strclr(string5,5);
    strcpy(string5,sbuf,27,2);
    panjang=(strtobyte(string5)*2)-2; //cari panjang pasangan SMSC
    strclr(notel,15); //bersihkan buffer penyimpan SMSC
    strcpy(notel,sbuf,31,panjang); //copy dari sbuf no SMSC
    panjang = panjang + 33;
    strclr(string5,5);
    strcpy(string5,sbuf,panjang,2); //copy buffer jumlah no pengirim
    panjang = panjang + 4;
    tmp=strtobyte(string5); //ubah dr string kr biner / byte
    tmpchar=tmp%2; //cek ganjil / genap
    if (tmpchar!=0) tmp=tmp+1; //if ganjil jml no pengirim + 1
    strcpy(notel,sbuf,panjang,tmp); //copy no pengirim dr sbuf
    baliktext(notel); //balik susunan no pengirim
    if (tmpchar!=0) //proses hilangkan sisipan 'F'
    {
        tmpchar=strlen(notel);
        notel[tmpchar-1]=0;
    }
    panjang = panjang + 20 + tmp ; //posisikan ke pointer awal isi SMS
    tmp=strlen(sbuf); //cari panjang total isi SMS
    strclr(SMSsemu,PDU_size);
    strcpy(SMSsemu,sbuf,panjang,tmp-panjang-7); //copy isi SMS
    ZIP_decompres(SMSsemu,SMSasli,text_size,sbuf_size); //ubah dr PDU ke text
}

void kirim(char *isi_SMS)
{
    char pjg_isi_SMS;

    pjg_isi_SMS=strlen(isi_SMS);
    ZIP_compres(SMSsemu,isi_SMS,text_size,sbuf_size);
    strclr(sbuf,sbuf_size); //bersihkan buffer string panjang gabungan
SMS
    strclr(noSMSC,16); //bersihkan buffer string panjang SMSC
    strcatf(noSMSC,c_noSMSC); //ambil no SMSC
    tmp=strlen(noSMSC); //cek panjangnya
    tmpchar=tmp%2; //cek is ganjil
    if (tmpchar!=0) //jika ganjil
    {
        tmp=tmp+1; //panjang = panjang + 1;
        strcatf(noSMSC,"F"); //tambah no SMSC dengan F
    }
    tmp=(tmp/2)+1; //hitung panjang pasangan SMSC
    strclr(string5,5);
    bytetostr(tmp,string5); //ubah ke string
    baliktext(noSMSC); //ubah posisi
    strcat(sbuf,string5); //masukkan jml pasangan SMSC ke frame
kirim
    strcatf(sbuf,"91"); //masukkan format no SMSC ke frame kirim
    strcat(sbuf,noSMSC); //masukkan no SMSC ke frame kirim
    strcatf(sbuf,"0100");

    strclr(notujuan,16); //bersihkan buffer string panjang no
tujuan
    strcatf(notujuan,c_notujuan); //ambil no tujuan
    tmp=strlen(notujuan); //cek panjangnya
    tmpchar=tmp%2; //cek is ganjil
    if (tmpchar!=0) //jika ganjil
    {
        strcatf(notujuan,"F"); //tambah no tujuan dengan F
    }
}

```

software

```
strclr(string5,5);
bytetostr(tmp,string5);
baliktext(notujuan);
strcat(sbuf,string5);
strcatf(sbuf,"91");
strcat(sbuf,notujuan);
strcatf(sbuf,"0000");
strclr(string5,5);
bytetostr(pjg_isi_SMS,string5);
strcat(sbuf,string5);
strcat(sbuf,SMSsemu);

tmp=strlen(sbuf);
strclr(string5,5);
if (tmp >=100 )
{
    string5[0]=tmp/100|0x30;
    string5[1]=tmp/10%10|0x30;
    string5[2]=tmp%10|0x30;
}
else
{
    string5[0]=tmp/10%10|0x30;
    string5[1]=tmp%10|0x30;
}
UCSRB &= ~(1<<RXEN);
send_textf("AT+CMGC=");
send_textf(string5);
delay_ms(30);
send_textf("\r\n");
delay_ms(200);
send_textf(sbuf);

usart_send(0x1A);
delay_ms(30);
send_textf("\r\n");
UCSRB |= (1<<RXEN);
```

```
//ubah ke string
//ubah posisi
//masukkan jml no tujuan ke frame kirim
//masukkan format no tujuan ke frame kirim
//masukkan no tujuan ke frame kirim

//ubah ke string

//cari total panjang sms
//bersihkan buffer string
//proses panjang sms ubah ke string

//matikan enable penerimaan serial
//kirim perintah kirim SMS
//kirim total panjang SMS

//kirim enter
//kasih kesempatan ponsel ngirin '>'
//kirim gabungan data SMS yg mau dikirim

//kirim CTRL Z

//kirim enter
//hidupkan enable penerimaan serial
```

```
void cari_sudut()
{
    if (PIND.2==0)
    {
        sudut= phi_count*180/10;
        phi=cos[sudut];
    }
}
```

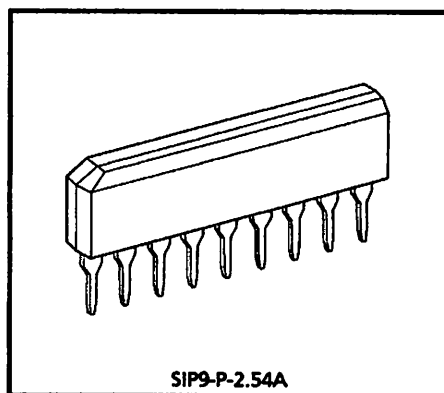
TC5081BP

PHASE COMPARATOR

The TC5081BP is phase comparator for PLL frequency synthesizer type, and consists of a digital phase comparator and an amplifier for active low pass filter.

FEATURES

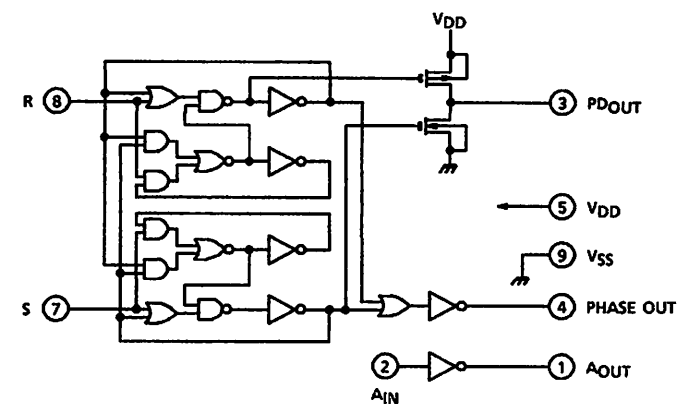
- The phase comparator detects two input pulse phase differences and outputs proportionate positive or negative pulses to PDOUT. When the input pulse phases are the same, PDOUT has high impedance.
- Because the IC is CMOS, the input impedance of the filter for the amp is extremely high and has excellent characteristics.
- TC5081BP comes in a SIP 9 PIN.



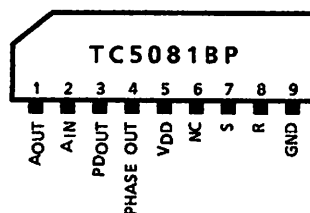
SIP9-P-2.54A

Weight : 0.92g (Typ.)

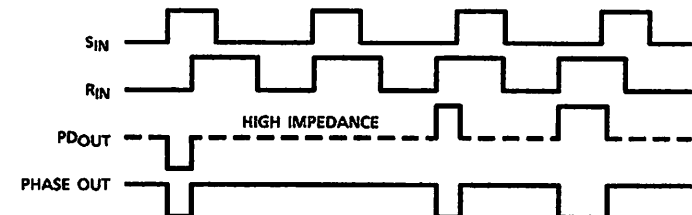
LOGIC DIAGRAM



PIN CONNECTION (SIDE VIEW)



PHASE COMPARATOR TIMING CHART

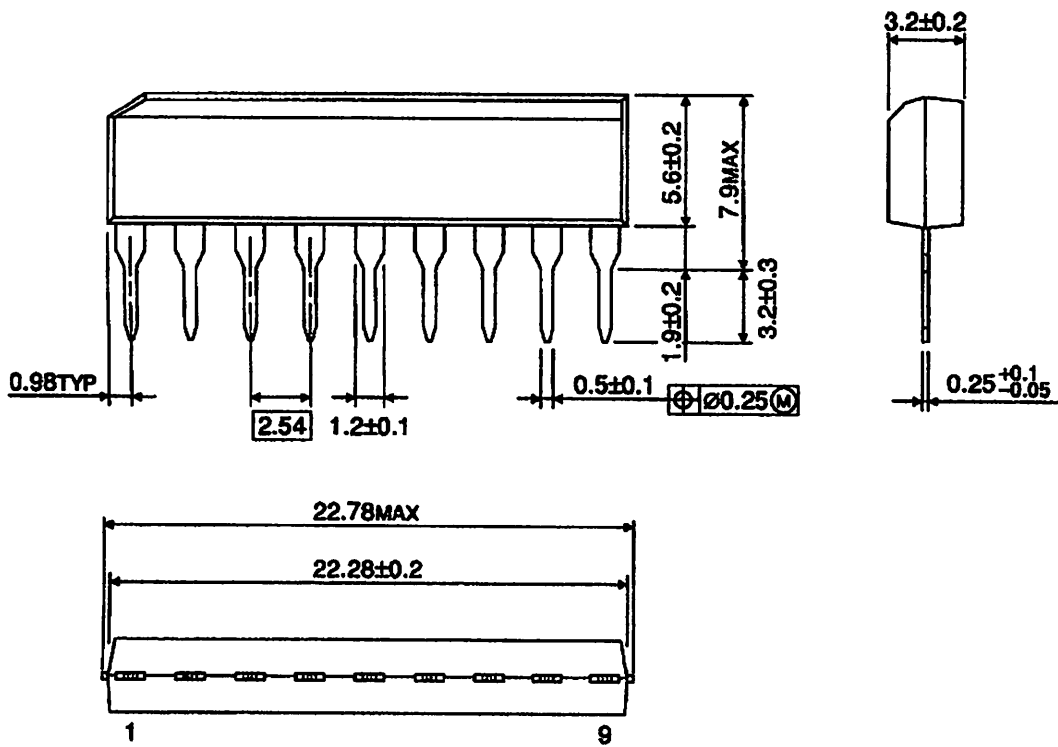


961001EBA2

● TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.

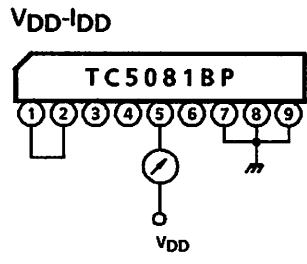
OUTLINE DRAWING
SIP9-P-2.54A

Unit : mm

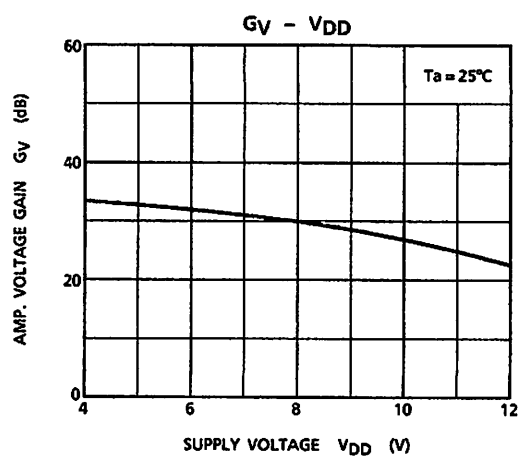
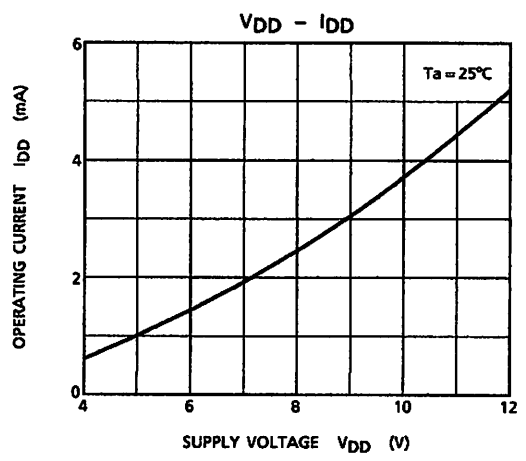
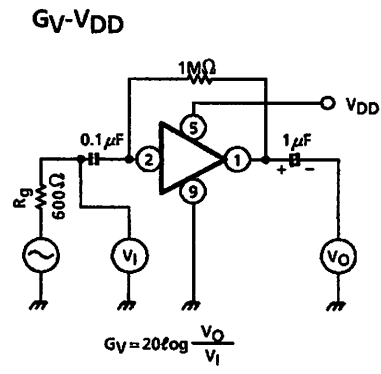


Weight : 0.92g (Typ.)

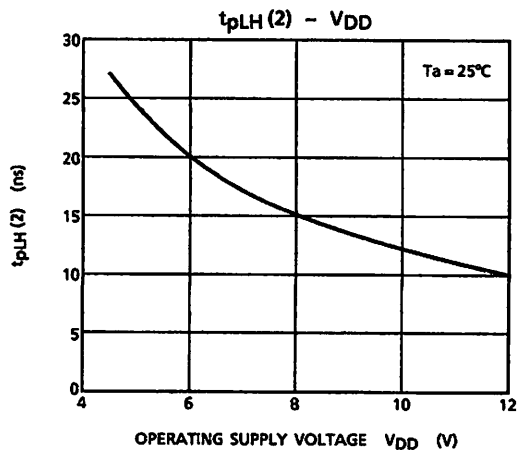
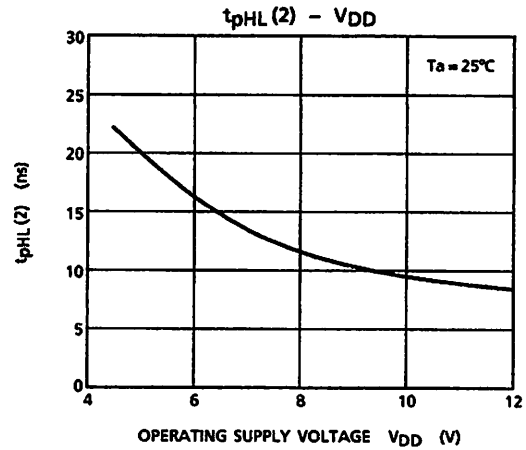
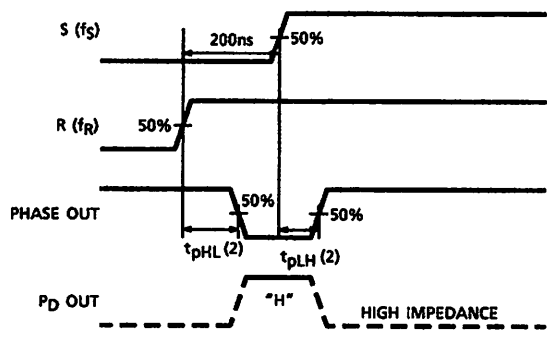
TEST CIRCUIT 2



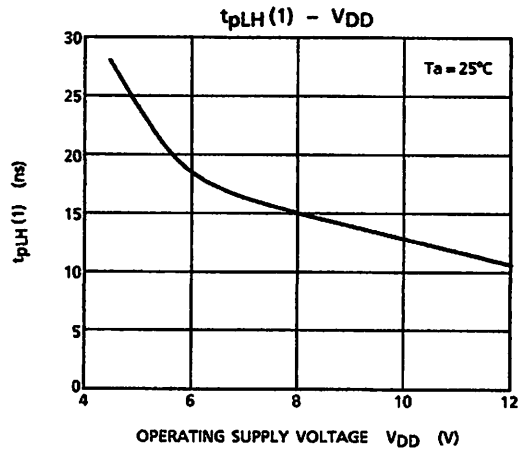
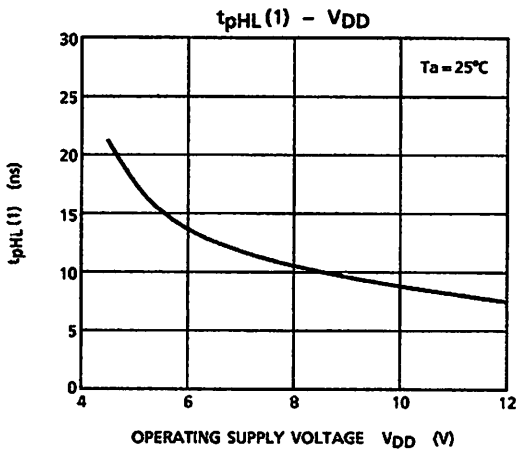
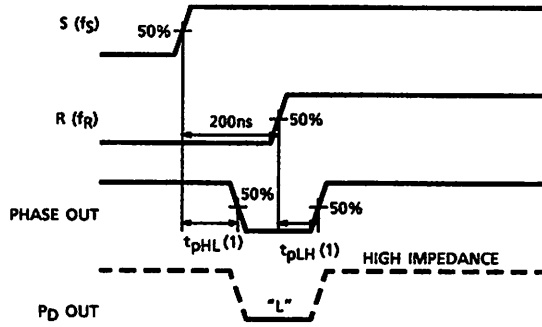
TEST CIRCUIT 3



WAVE FORM 2 (The lagging phase, $f_S < f_R$)



WAVE FORM 1 (The leading phase, $f_S > f_R$)



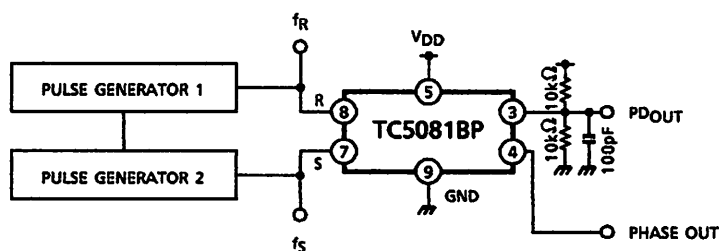
MAXIMUM RATINGS (Ta = 25°C)

CHARACTERISTIC	SYMBOL	RATING	UNIT
Supply Voltage	V _{DD}	15	V
Input Voltage	V _{IN}	-0.3~V _{DD} +0.3	V
Operating Temperature	T _{opr}	-30~75	°C
Storage Temperature	T _{stg}	-55~125	°C

ELECTRICAL CHARACTERISTICS (V_{DD} = 7.5V, Ta = -30~75°C)

CHARACTERISTIC	SYMBOL	TEST CIRCUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Operating Supply Voltage	V _{DD}	—	—	4.5	—	12	V
Output Voltage	"H" Level	V _{OH}	V _{IH} = 6.6V, I _{OH} = -50μA	7.3	—	—	V
	"L" Level	V _{OL}	V _{IL} = 1.6V, I _{OL} = 50μA	—	—	0.2	V
Quiescent Current	I _{DD}	—	V _{IH} = 7.5V, V _{IL} = 0V	—	—	200	μA
3 State Leak Current	"H" Level	I _{TLH}	—	—	—	500	nA
	"L" Level	I _{TLL}	—	—	—	-500	nA
Filter Amp. Voltage Gain	G _V	3	R _{①-②} = 1MΩ, f _{IN} = 1kHz R _g = 600Ω	—	30	—	dB

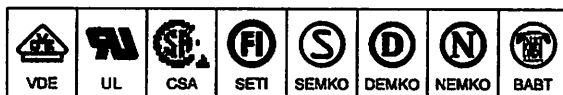
TEST CIRCUIT 1



The pulse generator-1 is synchronized with the pulse generator-2.
Then, the phase of f_R and f_S is variable.

961001EBA2'

- The products described in this document are subject to foreign exchange and foreign trade control laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.



6-Pin DIP Random-Phase Optoisolators Triac Driver Output (400 Volts Peak)

The MOC3020 Series consists of gallium arsenide infrared emitting diodes, optically coupled to a silicon bilateral switch.

- *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.*
- They are designed for applications requiring isolated triac triggering.

Recommended for 115/240 Vac(rms) Applications:

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115 Vac Peripherals
- Motor Controls
- Static ac Power Switch
- Solid State Relays
- Incandescent Lamp Dimmers

MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

INFRARED EMITTING DIODE

Reverse Voltage	V_R	3	Volts
Forward Current — Continuous	I_F	60	mA
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Negligible Power in Triac Driver Derate above 25°C	P_D	100 1.33	mW mW/°C

OUTPUT DRIVER

Off-State Output Terminal Voltage	V_{DRM}	400	Volts
Peak Repetitive Surge Current (PW = 1 ms, 120 pps)	I_{TSM}	1	A
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	300 4	mW mW/°C

TOTAL DEVICE

Isolation Surge Voltage ⁽¹⁾ (Peak ac Voltage, 60 Hz, 1 Second Duration)	V_{ISO}	7500	Vac(pk)
Total Power Dissipation @ $T_A = 25^\circ\text{C}$ Derate above 25°C	P_D	330 4.4	mW mW/°C
Junction Temperature Range	T_J	-40 to +100	°C
Ambient Operating Temperature Range ⁽²⁾	T_A	-40 to +85	°C
Storage Temperature Range ⁽²⁾	T_{stg}	-40 to +150	°C
Soldering Temperature (10 s)	T_L	260	°C

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

Preferred devices are Motorola recommended choices for future use and best overall value.
Global Optoisolator is a trademark of Motorola, Inc.

MOC3021
[IFT = 15 mA Max]
MOC3022
[IFT = 10 mA Max]
MOC3023*
[IFT = 5 mA Max]

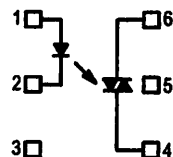
*Motorola Preferred Device

STYLE 6 PLASTIC



STANDARD THRU HOLE
CASE 730A-04

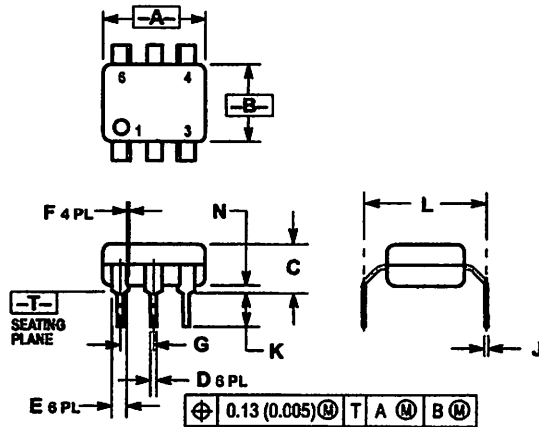
SCHEMATIC



1. ANODE
2. CATHODE
3. NC
4. MAIN TERMINAL
5. SUBSTRATE
DO NOT CONNECT
6. MAIN TERMINAL



MOC3021 MOC3022 MOC3023



- 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	0.240	0.260	6.10	6.60
C	0.115	0.200	2.93	5.08
D	0.015	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	0.008	0.012	0.21	0.30
K	0.100	0.150	2.54	3.81
L	0.400	0.425	10.16	10.80
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 the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended 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 (M) are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA / EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

MFAX: RMFA00@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com

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

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

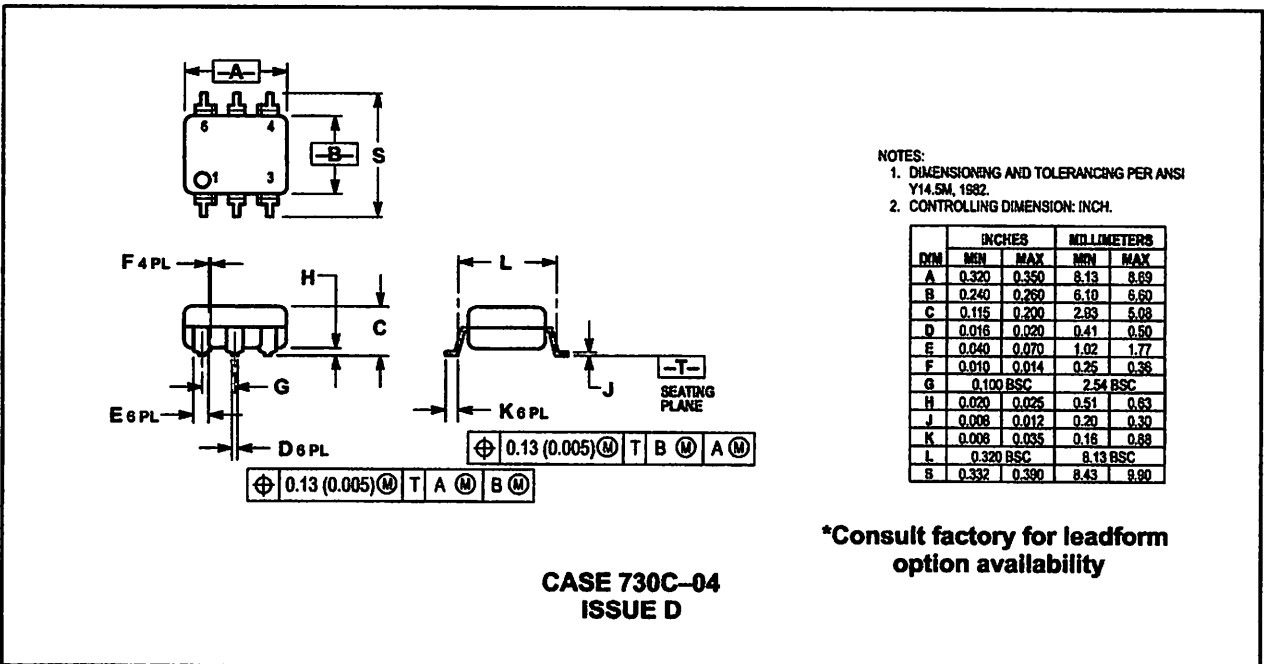
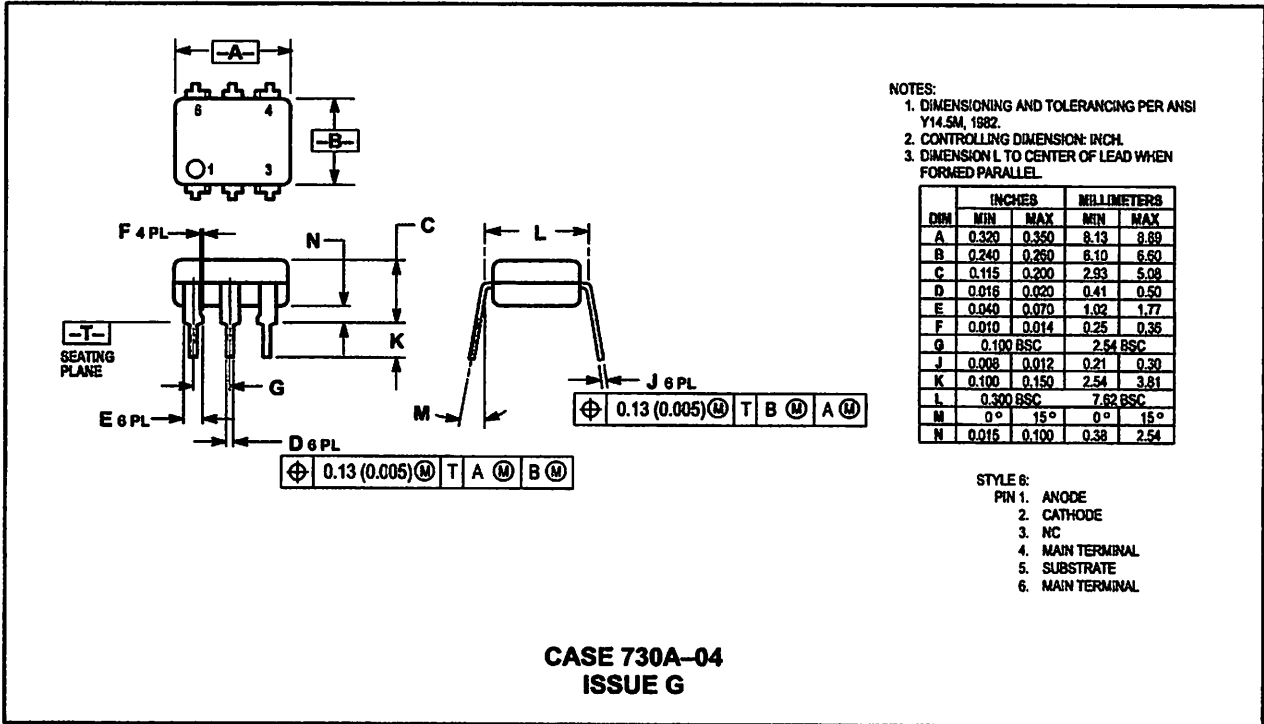
MOTOROLA

MOC3020/D

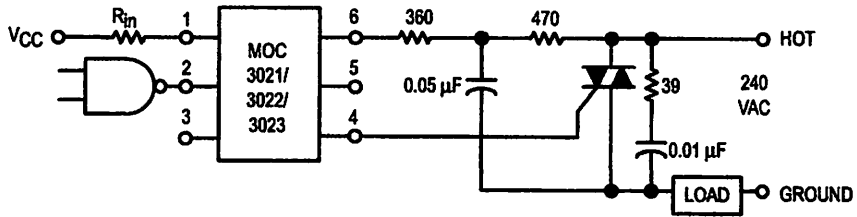


MOC3021 MOC3022 MOC3023

PACKAGE DIMENSIONS



MOC3021 MOC3022 MOC3023



* This optoisolator should not be used to drive a load directly. It is intended to be a trigger device only.

Additional information on the use of optically coupled triac drivers is available in Application Note AN-780A.

In this circuit the "hot" side of the line is switched and the load connected to the cold or ground side.

The 39 ohm resistor and 0.01 μF capacitor are for snubbing of the triac, and the 470 ohm resistor and 0.05 μF capacitor are for snubbing the coupler. These components may or may not be necessary depending upon the particular triac and load used.

Figure 8. Typical Application Circuit

MOC3021 MOC3022 MOC3023



Figure 3. Trigger Current versus Temperature

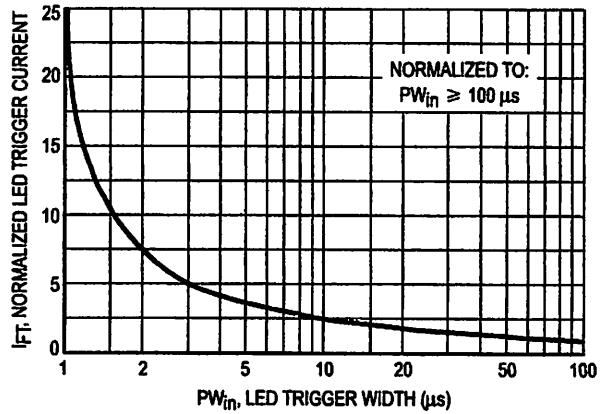


Figure 4. LED Current Required to Trigger versus LED Pulse Width

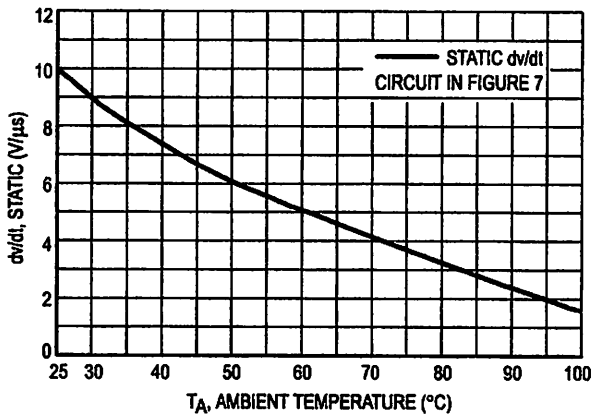


Figure 5. dv/dt versus Temperature

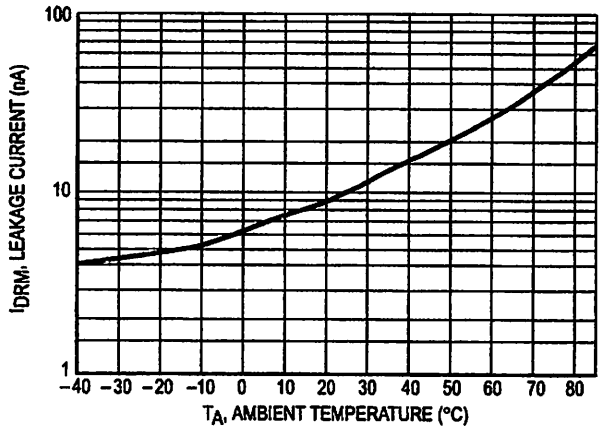
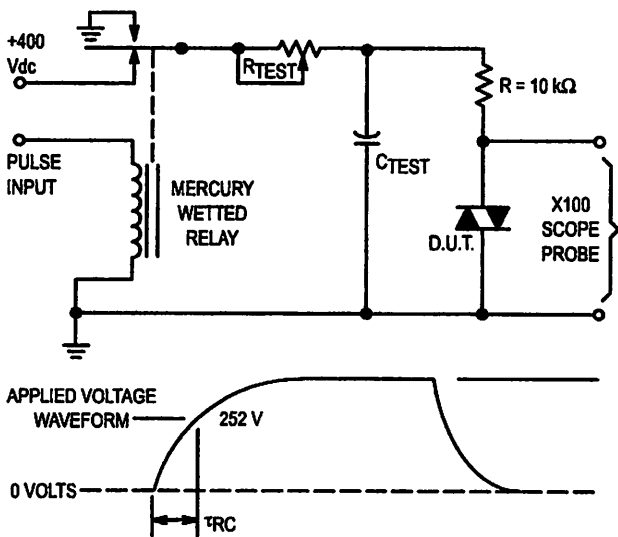


Figure 6. Leakage Current, I_{DRM} versus Temperature



1. The mercury wetted relay provides a high speed repeated pulse to the D.U.T.
2. 100x scope probes are used, to allow high speeds and voltages.
3. The worst-case condition for static dv/dt is established by triggering the D.U.T. with a normal LED input current, then removing the current. The variable R_{TEST} allows the dv/dt to be gradually increased until the D.U.T. continues to trigger in response to the applied voltage pulse, even after the LED current has been removed. The dv/dt is then decreased until the D.U.T. stops triggering. τ_{RC} is measured at this point and recorded.

$$dv/dt = \frac{0.63 V_{max}}{\tau_{RC}} = \frac{252}{\tau_{RC}}$$

Figure 7. Static dv/dt Test Circuit

MOC3021 MOC3022 MOC3023

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
----------------	--------	-----	-----	-----	------

INPUT LED

Reverse Leakage Current ($V_R = 3\text{ V}$)	I_R	—	0.05	100	μA
Forward Voltage ($I_F = 10\text{ mA}$)	V_F	—	1.15	1.5	Volts

OUTPUT DETECTOR ($I_F = 0$ unless otherwise noted)

Peak Blocking Current, Either Direction (Rated $V_{DRM}^{(1)}$)	I_{DRM}	—	10	100	nA
Peak On-State Voltage, Either Direction ($I_{TM} = 100\text{ mA Peak}$)	V_{TM}	—	1.8	3	Volts
Critical Rate of Rise of Off-State Voltage (Figure 7, Note 2)	dv/dt	—	10	—	$\text{V}/\mu\text{s}$

COUPLED

LED Trigger Current, Current Required to Latch Output (Main Terminal Voltage = $3\text{ V}^{(3)}$)	MOC3021	I_{FT}	—	8	15	mA
	MOC3022					
	MOC3023					
Holding Current, Either Direction	I_H	—	100	—	μA	

1. Test voltage must be applied within dv/dt rating.
2. This is static dv/dt. See Figure 7 for test circuit. Commutating dv/dt is a function of the load-driving thyristor(s) only.
3. All devices are guaranteed to trigger at an I_F value less than or equal to max I_{FT} . Therefore, recommended operating I_F lies between max I_{FT} (15 mA for MOC3021, 10 mA for MOC3022, 5 mA for MOC3023) and absolute max I_F (60 mA).

TYPICAL ELECTRICAL CHARACTERISTICS

$T_A = 25^\circ\text{C}$

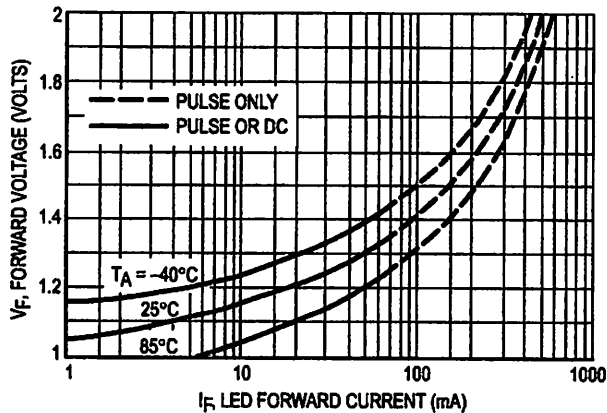


Figure 1. LED Forward Voltage versus Forward Current

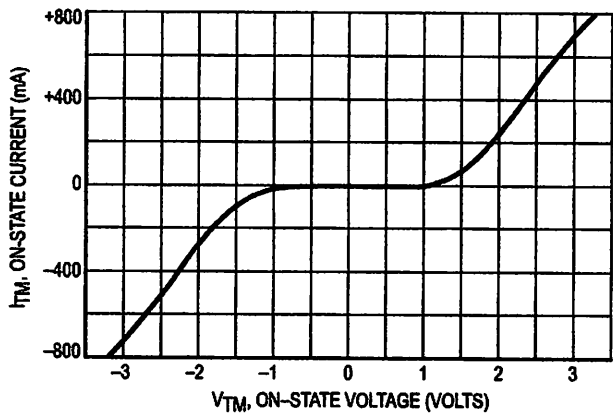


Figure 2. On-State Characteristics

Features

High-performance, Low-power AVR[®] 8-bit Microcontroller

Advanced RISC Architecture

- 130 Powerful Instructions – Most Single Clock Cycle Execution
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- Up to 16 MIPS Throughput at 16 MHz
- On-chip 2-cycle Multiplier

Nonvolatile Program and Data Memories

- 8K 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
- 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 512 Bytes Internal SRAM
- Programming Lock for Software Security

Peripheral 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
- 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels for TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
- Byte-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

Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

I/O and Packages

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad MUF

Operating Voltages

- 2.7 - 5.5V for ATmega8535L
- 4.5 - 5.5V for ATmega8535

Speed Grades

- 0 - 8 MHz for ATmega8535L
- 0 - 16 MHz for ATmega8535



8-bit AVR[®] Microcontroller with 8K Bytes In-System Programmable Flash

ATmega8535
ATmega8535L

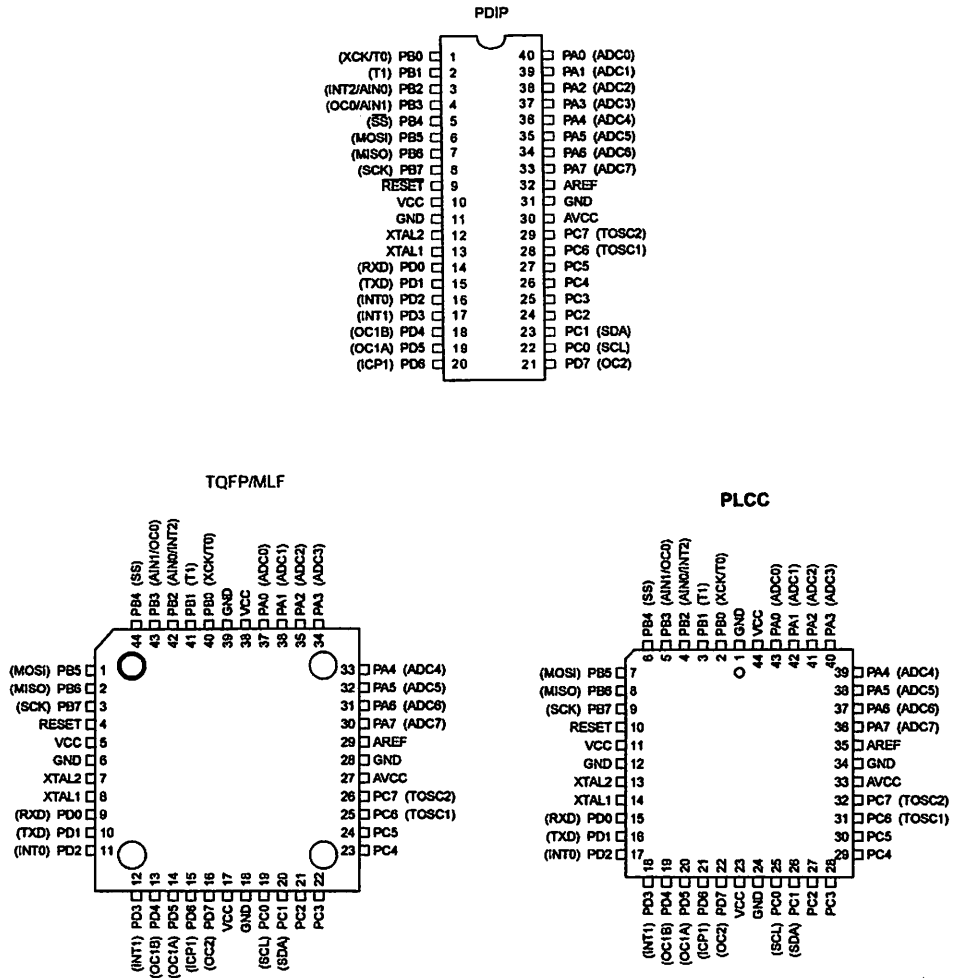
Advance
Information





Pin Configurations

Figure 1. Pinout ATmega8535



Disclaimer

Typical values contained in this data sheet 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.

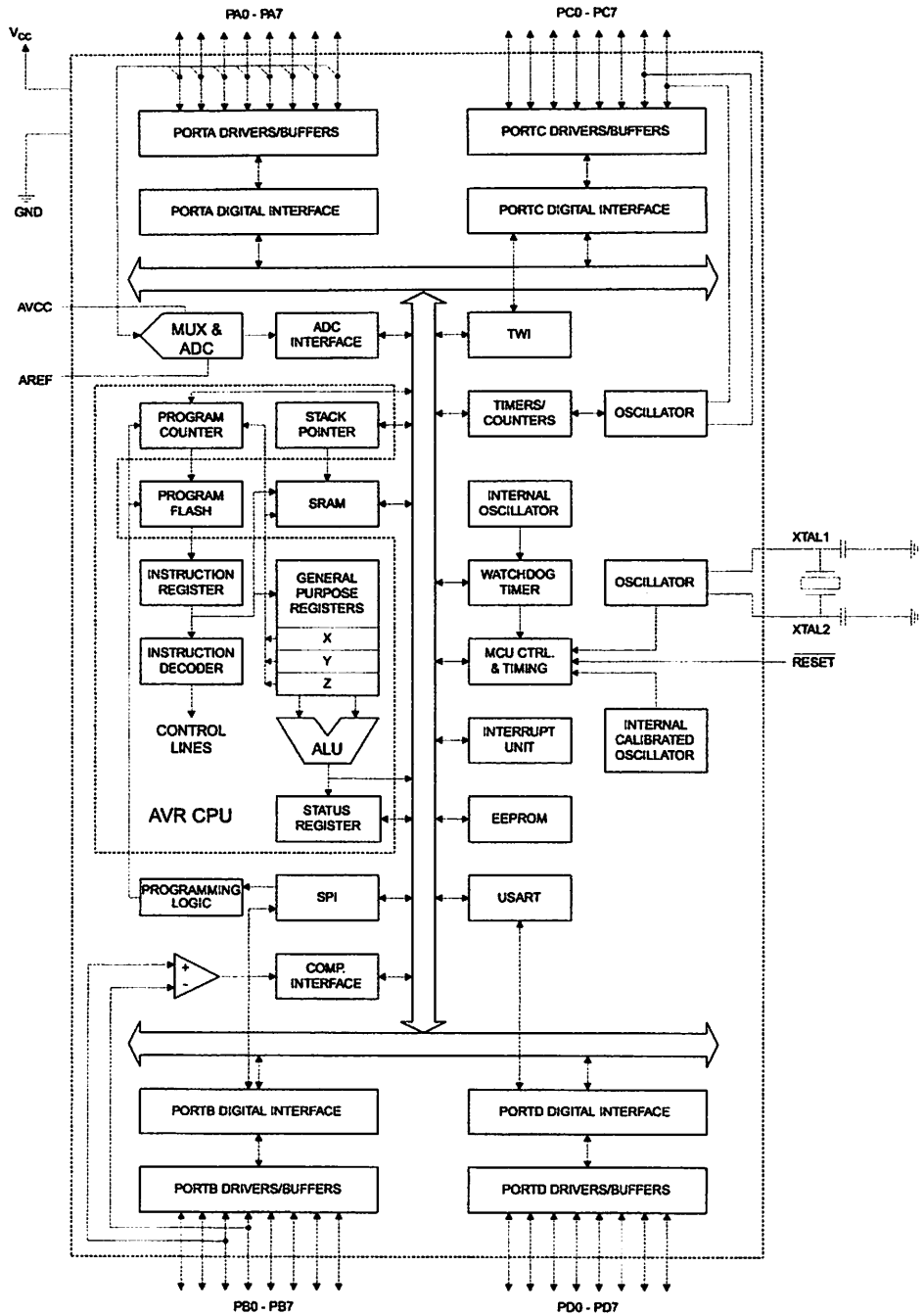
ATmega8535(L)

Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram





AVR ATmega8535 Memories

This section describes the different memories in the ATmega8535. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega8535 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

In-System Reprogrammable Flash Program Memory

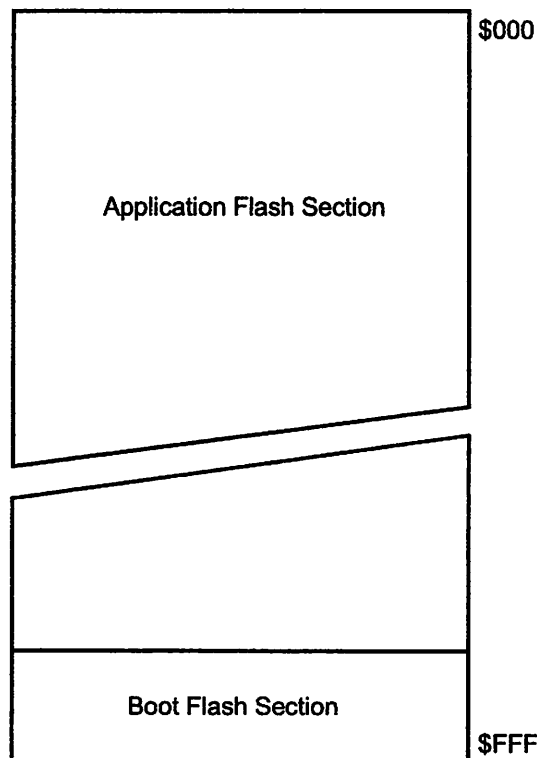
The ATmega8535 contains 8K bytes On-chip In-System Reprogrammable Flash memory for program storage. Since all AVR instructions are 16 or 32 bits wide, the Flash is organized as 4K x 16. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 10,000 write/erase cycles. The ATmega8535 Program Counter (PC) is 12 bits wide, thus addressing the 4K 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 220. "Memory Programming" on page 233 contains a detailed description on Flash Programming in SPI or Parallel Programming mode.

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



RAM Data Memory

Figure 9 shows how the ATmega8535 SRAM Memory is organized.

The 608 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 512 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.

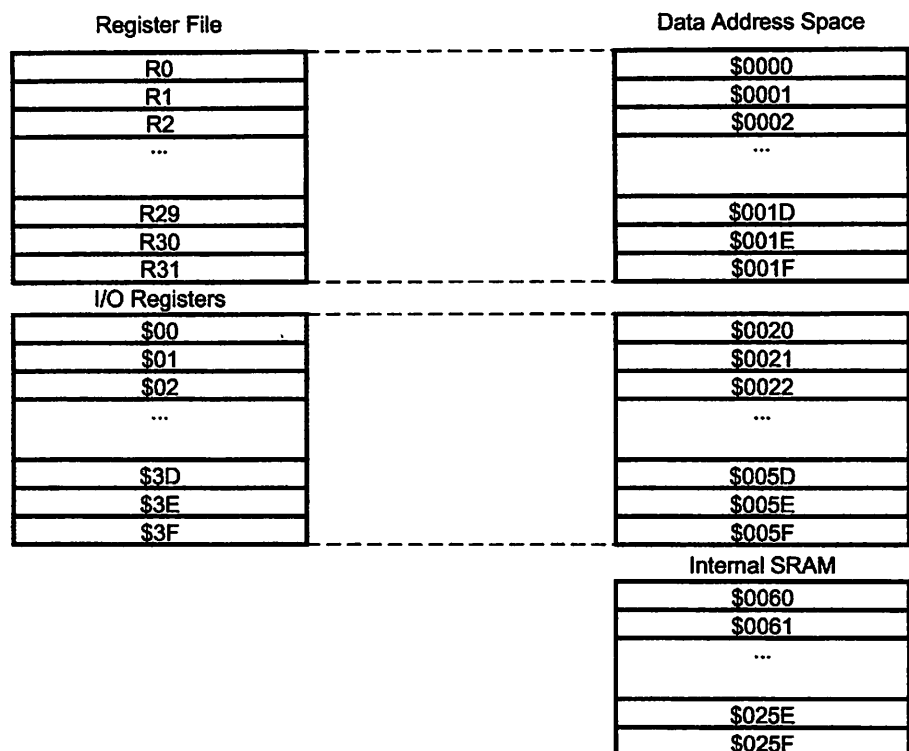
The direct addressing reaches the entire data space.

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 512 bytes of internal data SRAM in the ATmega8535 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

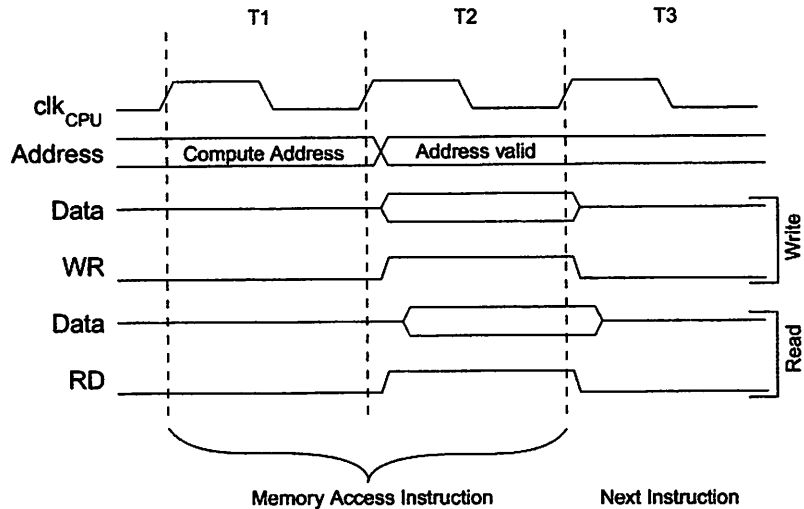




Data Memory Access Times

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 ATmega8535 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.

"Memory Programming" on page 233 contains a detailed description on EEPROM Programming in SPI or Parallel Programming mode.

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.

The EEPROM Address Register – 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	

- **Bits 15..9 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8535 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.

The EEPROM Data Register – EEDR

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	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.

The EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	EERIE	EEMWE	EEWE	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 ATmega8535 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 EEWE is cleared.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address. If EEMWE is zero, setting EEWE 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 EEWE bit for an EEPROM write procedure.





• **Bit 1 – EEW: EEPROM Write Enable**

The EEPROM Write Enable Signal EEW is the write strobe to the EEPROM. When address and data are correctly set up, the EEW 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 EEW, 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 EEW 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 EEW in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEW.

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 updated by the CPU, step 2 can be omitted. See “Boot Loader Support – Read-While-Write Self-Programming” on page 220 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 EEW bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEW 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 EEW 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 ⁽¹⁾	Typ Programming Time
EEPROM Write (from CPU)	8448	8.4 ms

Note: 1. Uses 1 MHz clock, independent of CKSEL Fuse settings.



System Control and Reset

Resetting the AVR

During Reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be an RJMP instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa. The circuit diagram in Figure 15 shows the reset logic. Table 15 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in "Clock Sources" on page 23.

Reset Sources

The ATmega8535 has four sources of Reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the \overline{RESET} pin for longer than the minimum pulse length.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.

Figure 15. Reset Logic

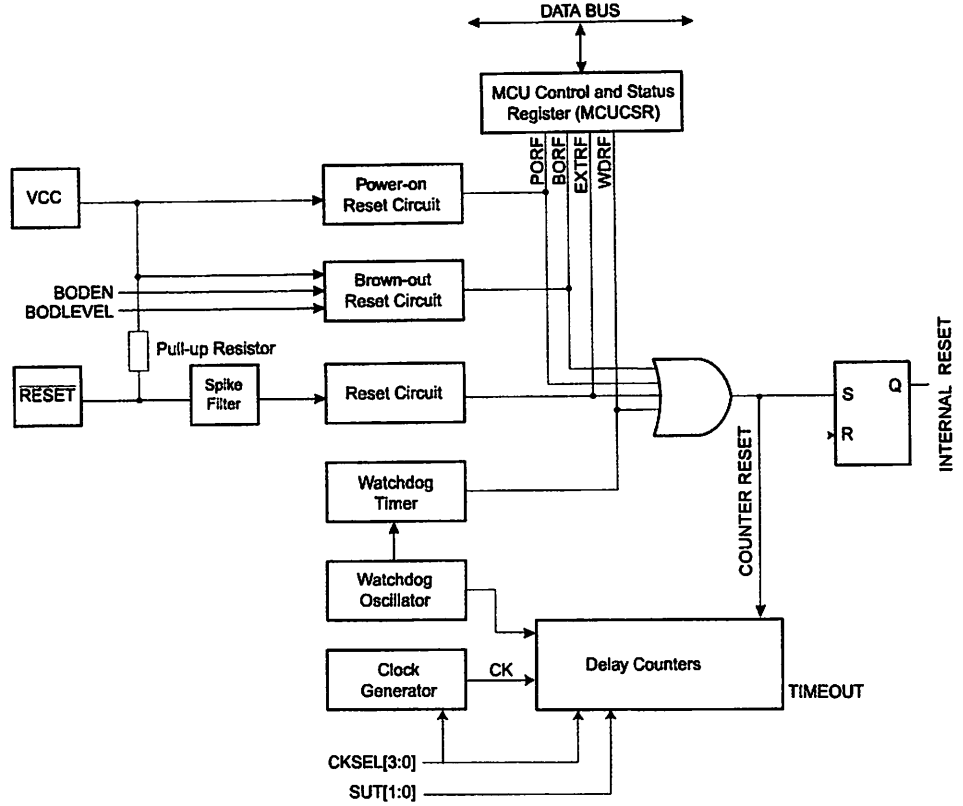


Table 15. Reset Characteristics⁽¹⁾

Symbol	Parameter	Condition	Min	Typ	Max	Units
V _{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling) ⁽²⁾			1.3	2.3	V
V _{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.1		0.9	V
t _{RST}	Minimum pulse width on $\overline{\text{RESET}}$ Pin			50		ns
V _{BOT}	Brown-out Reset Threshold Voltage ⁽³⁾	BODLEVEL = 1	2.5	2.7	3.2	V
		BODLEVEL = 0	3.7	4.0	4.2	
t _{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		μs
		BODLEVEL = 0		2		μs
V _{HYST}	Brown-out Detector hysteresis			130		mV

- Notes:
1. Values are guidelines only.
 2. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).
 3. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the



production test. This guarantees that a Brown-out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using $BODLEVEL = 1$ for ATmega8535L and $BODLEVEL = 0$ for ATmega8535. $BODLEVEL = 1$ is not applicable for ATmega8535.

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 15. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 16. MCU Start-up, \overline{RESET} Tied to V_{CC}

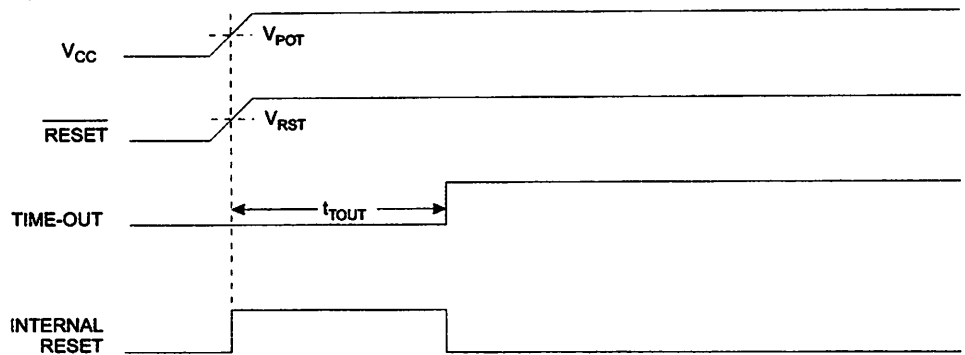
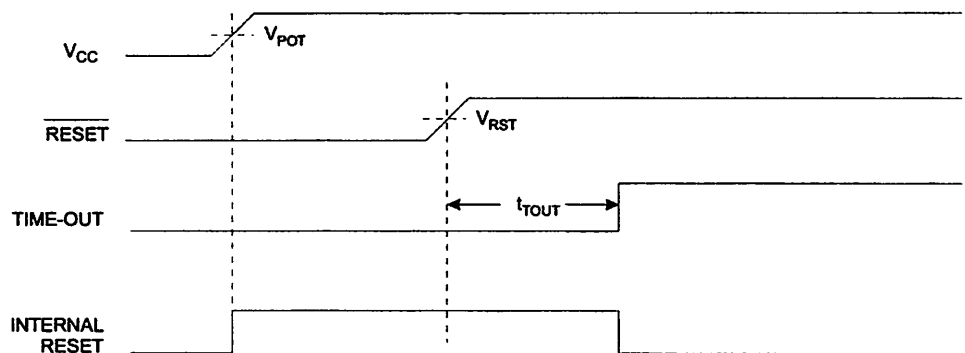


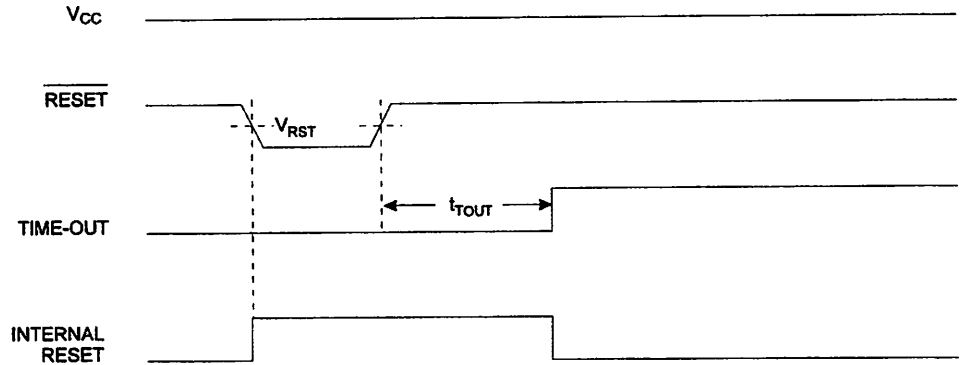
Figure 17. MCU Start-up, \overline{RESET} Extended Externally



External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 15) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} on its positive edge, the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 18. External Reset During Operation



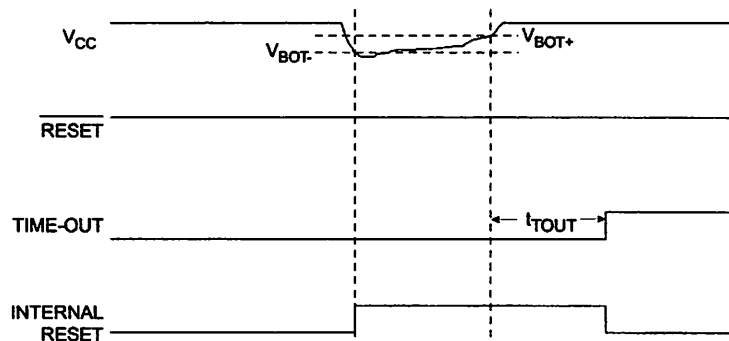
Brown-out Detection

ATmega8535 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{\text{BOT+}} = V_{\text{BOT}} + V_{\text{HYST}}/2$ and $V_{\text{BOT-}} = V_{\text{BOT}} - V_{\text{HYST}}/2$.

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level ($V_{\text{BOT-}}$ in Figure 19), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level ($V_{\text{BOT+}}$ in Figure 19), the delay counter starts the MCU after the time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 15.

Figure 19. Brown-out Reset During Operation

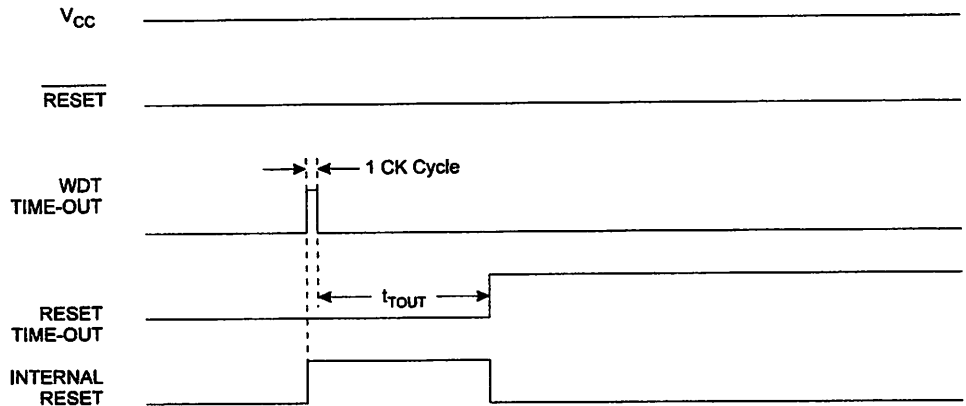




Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 39 for details on operation of the Watchdog Timer.

Figure 20. Watchdog Reset During Operation



MCU Control and Status Register – MCUCSR

The MCU Control and Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
	-	ISC2	-	-	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0		See Bit Description				

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUCSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

Internal Voltage Reference

ATmega8535 features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC. The 2.56V reference to the ADC is generated from the internal bandgap reference.

Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in Table 16. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODEN Fuse).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

Table 16. Internal Voltage Reference Characteristics⁽¹⁾

Symbol	Parameter	Min	Typ	Max	Units
V_{BG}	Bandgap reference voltage	1.15	1.23	1.35	V
t_{BG}	Bandgap reference start-up time		40	70	μ s
I_{BG}	Bandgap reference current consumption		10		μ A

Note: 1. Values are guidelines only.

Watchdog Timer

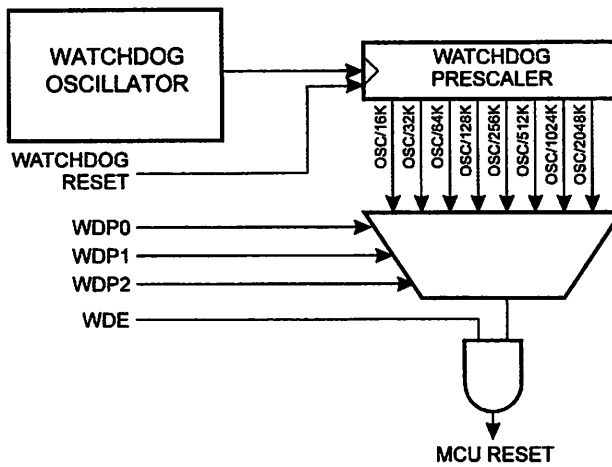
The Watchdog Timer is clocked from a separate On-chip Oscillator which runs at 1 MHz. This is the typical value at $V_{CC} = 5V$. See characterization data for typical values at other V_{CC} levels. By controlling the Watchdog Timer prescaler, the Watchdog Reset interval can be adjusted as shown in Table 18 on page 41. The WDR – Watchdog Reset – instruction resets the Watchdog Timer. The Watchdog Timer is also reset when it is disabled and when a Chip Reset occurs. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog Reset, the ATmega8535 resets and executes from the Reset Vector. For timing details on the Watchdog Reset, refer to page 38.

To prevent unintentional disabling of the Watchdog or unintentional change of Time-out period, three different safety levels are selected by the Fuses S8535C and WDTON as shown in Table 17. Safety level 0 corresponds to the setting in AT90S8535. There is no restriction on enabling the WDT in any of the safety levels.

Table 17. WDT Configuration as a Function of the Fuse Settings of M103C and WDTON

S8535C	WDTON	Safety Level	WDT Initial State	How to Disable the WDT	How to Change Time-out
Unprogrammed	Unprogrammed	1	Disabled	Timed sequence	Timed sequence
Unprogrammed	Programmed	2	Enabled	Always enabled	Timed sequence
Programmed	Unprogrammed	0	Disabled	Timed sequence	No restriction
Programmed	Programmed	2	Enabled	Always enabled	Timed sequence

Figure 21. Watchdog Timer



Watchdog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	WDTCR
	-	-	-	WDCE	WDE	WDP2	WDP1	WDP0	
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• **Bits 7..5 – Res: Reserved Bits**

These bits are reserved bits in the ATmega8535 and will always read as zero.

• **Bit 4 – WDCE: Watchdog Change Enable**

This bit must be set when the WDE bit is written to logic zero. Otherwise, the Watchdog will not be disabled. Once written to one, hardware will clear this bit after four clock cycles. Refer to the description of the WDE bit for a Watchdog disable procedure. In Safety Level 1 and 2, this bit must also be set when changing the prescaler bits. See “Timed Sequences for Changing the Configuration of the Watchdog Timer” on page 43.

8-bit Timer/Counter0 with PWM

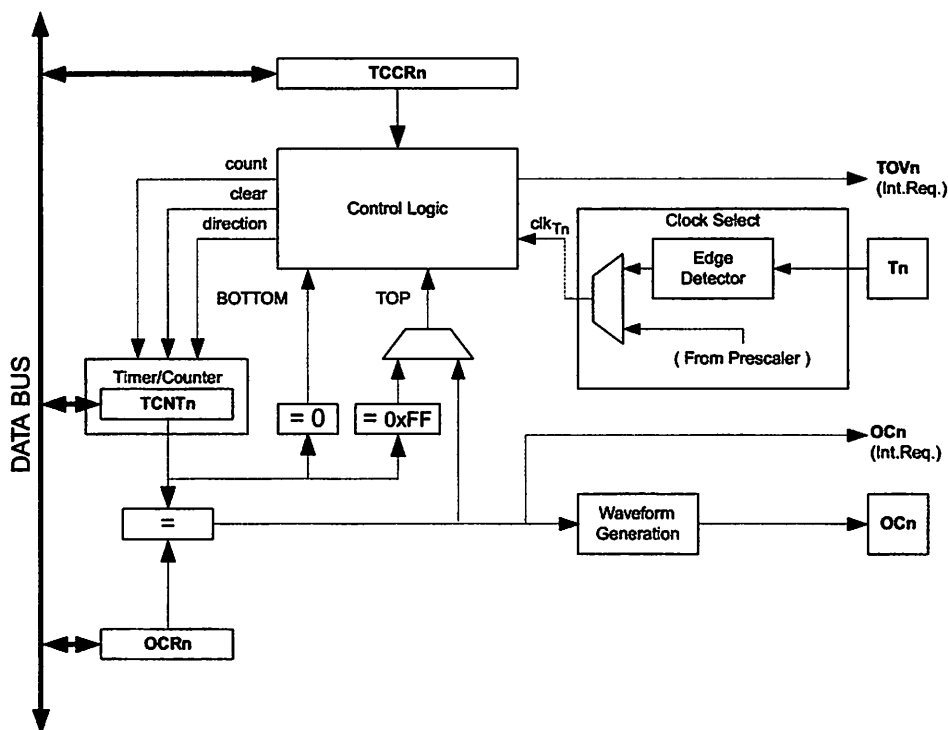
Timer/Counter0 is a general purpose, single channel, 8-bit Timer/Counter module. The main features are:

- Single Channel Counter
- Clear Timer on Compare Match (Auto Reload)
- Glitch-free, Phase Correct Pulse Width Modulator (PWM)
- Frequency Generator
- External Event Counter
- 10-bit Clock Prescaler
- Overflow and Compare Match Interrupt Sources (TOV0 and OCF0)

Overview

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 27. For the actual placement of I/O pins, refer to "Pinout ATmega8535" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "8-bit Timer/Counter Register Description" on page 80.

Figure 27. 8-bit Timer/Counter Block Diagram



Registers

The Timer/Counter (TCNT0) and Output Compare Register (OCR0) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in the figure) signals are all visible in the Timer Interrupt Flag Register (TIFR). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK). TIFR and TIMSK are not shown in the figure since these registers are shared by other timer units.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Register (OCR0) is compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC0). See "Output Compare Unit" on page 70. for details. The Compare Match event will also set the Compare Flag (OCF0) which can be used to generate an output compare interrupt request.

Definitions

Many register and bit references in this document are written in general form. A lower case "n" replaces the Timer/Counter number, in this case 0. However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 38 are also used extensively throughout the document.

Table 38. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXIMUM when it becomes 0xFF (decimal 255).
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF (MAX) or the value stored in the OCR0 Register. The assignment is dependent on the mode of operation.

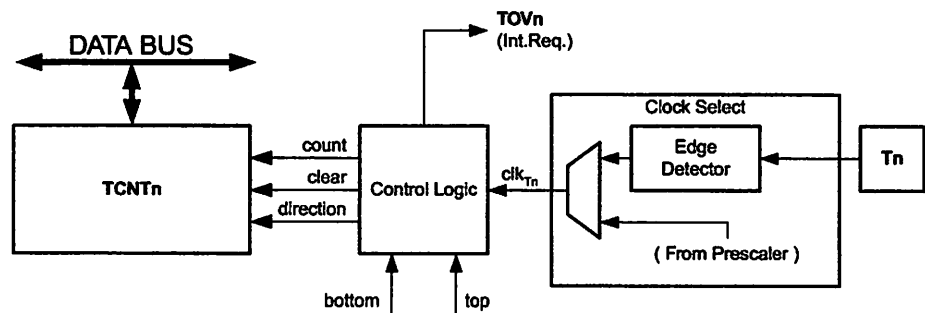
Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the clock select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 84.

Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 28 shows a block diagram of the counter and its surroundings.

Figure 28. Counter Unit Block Diagram



Signal description (internal signals):

- count** Increment or decrement TCNT0 by 1.
- direction** Select between increment and decrement.
- clear** Clear TCNT0 (set all bits to zero).
- clk_{Tn}** Timer/Counter clock, referred to as clk_{T0} in the following.



- top** Signalize that TCNT0 has reached maximum value.
- bottom** Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0). There are close connections between how the counter behaves (counts) and how waveforms are generated on the output compare output OC0. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 73.

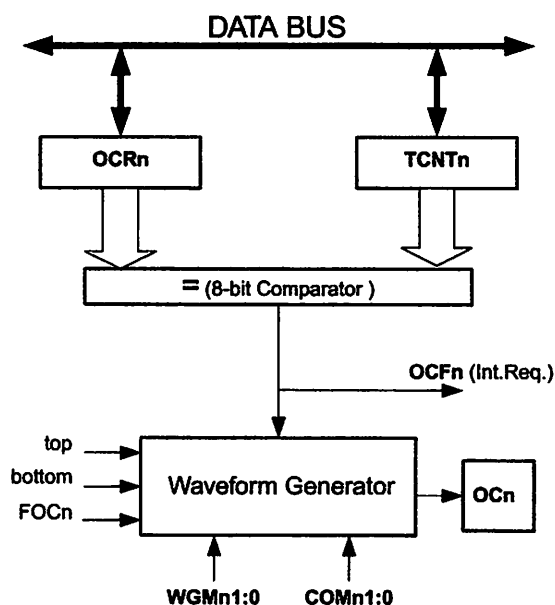
The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0). Whenever TCNT0 equals OCR0, the comparator signals a match. A match will set the Output Compare Flag (OCF0) at the next timer clock cycle. If enabled (OCIE0 = 1 and Global Interrupt Flag in SREG is set), the Output Compare Flag generates an output compare interrupt. The OCF0 Flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0 Flag can be cleared by software by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to operating mode set by the WGM01:0 bits and Compare Output mode (COM01:0) bits. The max and bottom signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (See "Modes of Operation" on page 73.).

Figure 29 shows a block diagram of the output compare unit.

Figure 29. Output Compare Unit, Block Diagram





8-bit Timer/Counter Register Description

Timer/Counter Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	TCCR0
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	
Read/Write	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 – FOC0: Force Output Compare**

The FOC0 bit is only active when the WGM00 bit specifies a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0 is written when operating in PWM mode. When writing a logical one to the FOC0 bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0 output is changed according to its COM01:0 bits setting. Note that the FOC0 bit is implemented as a strobe. Therefore it is the value present in the COM01:0 bits that determines the effect of the forced compare.

A FOC0 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0 as TOP.

The FOC0 bit is always read as zero.

- **Bit 6, 3 – WGM01:0: Waveform Generation Mode**

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of waveform generation to be used. Modes of operation supported by the Timer/Counter unit are: Normal mode, Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes. See Table 39 and “Modes of Operation” on page 73.

Table 39. Waveform Generation Mode Bit Description⁽¹⁾

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	TOP	MAX

Note: 1. The CTC0 and PWM0 bit definition names are now obsolete. Use the WGM01:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the Timer.

- **Bit 5:4 – COM01:0: Compare Match Output Mode**

These bits control the Output Compare pin (OC0) behavior. If one or both of the COM01:0 bits are set, the OC0 output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0 pin must be set in order to enable the output driver.

When OC0 is connected to the pin, the function of the COM01:0 bits depends on the WGM01:0 bit setting. Table 40 shows the COM01:0 bit functionality when the WGM01:0 bits are set to a normal or CTC mode (non-PWM).

Table 40. Compare Output Mode, non-PWM Mode

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on Compare Match
1	0	Clear OC0 on Compare Match
1	1	Set OC0 on Compare Match

Table 41 shows the COM01:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 41. Compare Output Mode, Fast PWM Mode⁽¹⁾

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on Compare Match, set OC0 at TOP
1	1	Set OC0 on Compare Match, clear OC0 at TOP

Note: 1. A special case occurs when OCR0 equals TOP and COM01 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 74 for more details.

Table 42 shows the COM01:0 bit functionality when the WGM01:0 bits are set to phase correct PWM mode.

Table 42. Compare Output Mode, Phase Correct PWM Mode⁽¹⁾

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on Compare Match when up-counting. Set OC0 on Compare Match when down-counting.
1	1	Set OC0 on Compare Match when up-counting. Clear OC0 on Compare Match when down-counting.

Note: 1. A special case occurs when OCR0 equals TOP and COM01 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Phase Correct PWM Mode" on page 76 for more details.



• **Bit 2:0 – CS02:0: Clock Select**

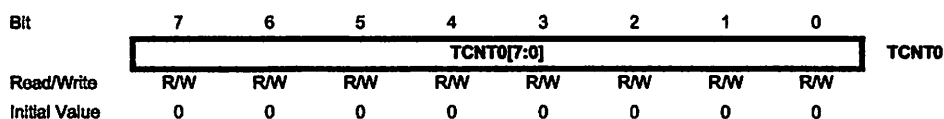
The three Clock Select bits select the clock source to be used by the Timer/Counter.

Table 43. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/counter stopped).
0	0	1	clk _{IO} /(No prescaling)
0	1	0	clk _{IO} /8 (From prescaler)
0	1	1	clk _{IO} /64 (From prescaler)
1	0	0	clk _{IO} /256 (From prescaler)
1	0	1	clk _{IO} /1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

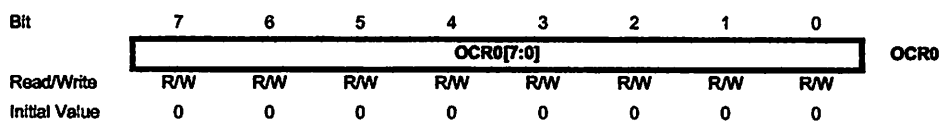
If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Timer/Counter Register – TCNT0



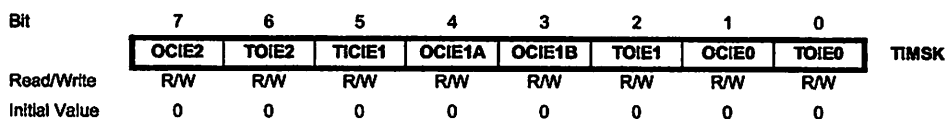
The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0 Register.

Output Compare Register – OCR0



The Output Compare Register contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0 pin.

Timer/Counter Interrupt Mask Register – TIMSK



• **Bit 1 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable**

When the OCIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs (i.e., when the OCF0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR).

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs (i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register – TIFR).

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
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 1 – OCF0: Output Compare Flag 0**

The OCF0 bit is set (one) when a Compare Match occurs between the Timer/Counter0 and the data in OCR0 – Output Compare Register0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0 is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0 (Timer/Counter0 Compare Match Interrupt Enable), and OCF0 are set (one), the Timer/Counter0 Compare Match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow Interrupt is executed. In phase correct PWM mode, this bit is set when Timer/Counter0 changes counting direction at 0x00.

Timer/Counter0 and Timer/Counter1 Prescalers

Timer/Counter1 and Timer/Counter0 share the same prescaler module, but the Timer/Counter0 can have different prescaler settings. The description below applies to both Timer/Counter1 and Timer/Counter0.

Internal Clock Source

The Timer/Counter can be clocked directly by the system clock (by setting the $CSn2:0 = 1$). This provides the fastest operation, with a maximum Timer/Counter clock frequency equal to system clock frequency ($f_{CLK_I/O}$). Alternatively, one of four taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either $f_{CLK_I/O}/8$, $f_{CLK_I/O}/64$, $f_{CLK_I/O}/256$, or $f_{CLK_I/O}/1024$.

Prescaler Reset

The prescaler is free running (i.e., operates independently of the clock select logic of the Timer/Counter) and it is shared by Timer/Counter1 and Timer/Counter0. Since the prescaler is not affected by the Timer/Counter's clock select, the state of the prescaler will have implications for situations where a prescaled clock is used. One example of prescaling artifacts occurs when the timer is enabled and clocked by the prescaler ($6 > CSn2:0 > 1$). The number of system clock cycles from when the timer is enabled to the first count occurs can be from 1 to $N+1$ system clock cycles, where N equals the prescaler divisor (8, 64, 256, or 1024).

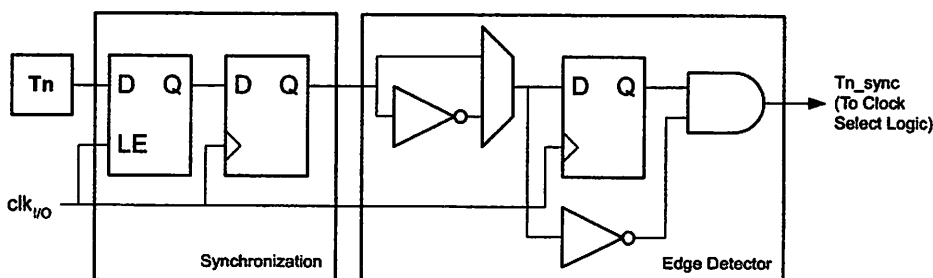
It is possible to use the Prescaler Reset for synchronizing the Timer/Counter to program execution. However, care must be taken if the other Timer/Counter that shares the same prescaler also uses prescaling. A prescaler reset will affect the prescaler period for all Timer/Counter0s it is connected to.

External Clock Source

An external clock source applied to the T1/T0 pin can be used as Timer/Counter clock (clk_{T1}/clk_{T0}). The T1/T0 pin is sampled once every system clock cycle by the pin synchronization logic. The synchronized (sampled) signal is then passed through the edge detector. Figure 38 shows a functional equivalent block diagram of the T1/T0 synchronization and edge detector logic. The registers are clocked at the positive edge of the internal system clock ($clk_{I/O}$). The latch is transparent in the high period of the internal system clock.

The edge detector generates one clk_{T1}/clk_{T0} pulse for each positive ($CSn2:0 = 7$) or negative ($CSn2:0 = 6$) edge it detects.

Figure 38. T1/T0 Pin Sampling



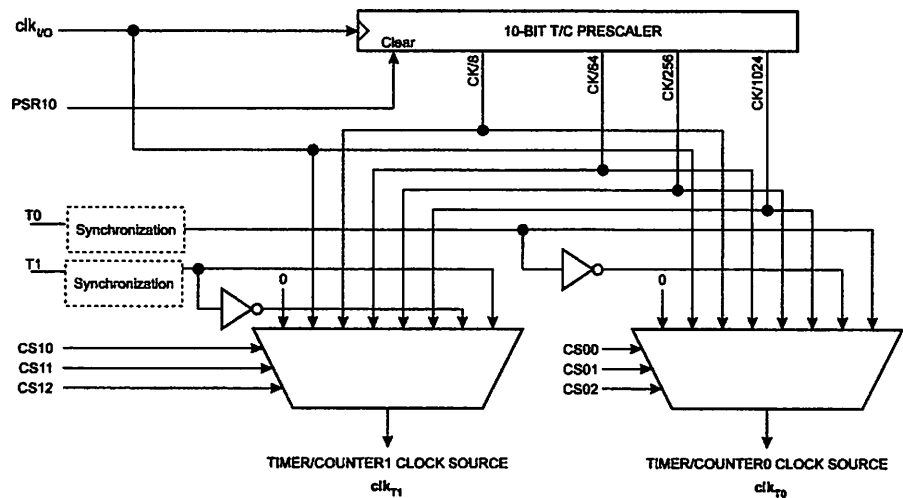
The synchronization and edge detector logic introduces a delay of 2.5 to 3.5 system clock cycles from an edge has been applied to the T1/T0 pin to the counter is updated.

Enabling and disabling of the clock input must be done when T1/T0 has been stable for at least one system clock cycle, otherwise it is a risk that a false Timer/Counter clock pulse is generated.

Each half period of the external clock applied must be longer than one system clock cycle to ensure correct sampling. The external clock must be guaranteed to have less than half the system clock frequency ($f_{ExtClk} < f_{clk_{I/O}}/2$) given a 50/50% duty cycle. Since the edge detector uses sampling, the maximum frequency of an external clock it can detect is half the sampling frequency (Nyquist sampling theorem). However, due to variation of the system clock frequency and duty cycle caused by Oscillator source (crystal, resonator, and capacitors) tolerances, it is recommended that maximum frequency of an external clock source is less than $f_{clk_{I/O}}/2.5$.

An external clock source can not be prescaled.

Figure 39. Prescaler for Timer/Counter0 and Timer/Counter1⁽¹⁾



Note: 1. The synchronization logic on the input pins (T1/T0) is shown in Figure 38.

Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	SFIOR
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• **Bit 0 – PSR10: Prescaler Reset Timer/Counter1 and Timer/Counter0**

When this bit is written to one, the Timer/Counter1 and Timer/Counter0 prescaler will be reset. The bit will be cleared by hardware after the operation is performed. Writing a zero to this bit will have no effect. Note that Timer/Counter1 and Timer/Counter0 share the same prescaler and a reset of this prescaler will affect both timers. This bit will always be read as zero.

Table 61. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

Note: 1. The baud rate is defined to be the transfer rate in bit per second (bps).

BAUD Baud rate (in bits per second, bps)

f_{osc} System Oscillator clock frequency

UBRR Contents of the UBRRH and UBRRL Registers, (0 - 4095)

Some examples of UBRR values for some system clock frequencies are found in Table 69 (see page 166).

Double Speed Operation (U2X)

The transfer rate can be doubled by setting the U2X bit in UCSRA. Setting this bit only has effect for the asynchronous operation. Set this bit to zero when using synchronous operation.

Setting this bit will reduce the divisor of the baud rate divider from 16 to 8, effectively doubling the transfer rate for asynchronous communication. Note however that the Receiver will in this case only use half the number of samples (reduced from 16 to 8) for data sampling and clock recovery, and therefore a more accurate baud rate setting and system clock are required when this mode is used. For the Transmitter, there are no downsides.

External Clock

External clocking is used by the synchronous slave modes of operation. The description in this section refers to Figure 70 for details.

External clock input from the XCK pin is sampled by a synchronization register to minimize the chance of meta-stability. The output from the synchronization register must then pass through an edge detector before it can be used by the Transmitter and Receiver. This process introduces a two CPU clock period delay and therefore the maximum external XCK clock frequency is limited by the following equation:

$$f_{XCK} < \frac{f_{osc}}{4}$$

Note that f_{osc} depends on the stability of the system clock source. It is therefore recommended to add some margin to avoid possible loss of data due to frequency variations.



Analog-to-Digital Converter

Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 65 - 260 μ s Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 8 Multiplexed Single Ended Input Channels
- 7 Differential Input Channels
- 2 Differential Input Channels with Optional Gain of 10x and 200x⁽¹⁾
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

Note: 1. The differential input channel are not tested for devices in PDIP and PLCC Package. This feature is only guaranteed to work for devices in TQFP and MLF Packages.

The ATmega8535 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The single-ended voltage inputs refer to 0V (GND).

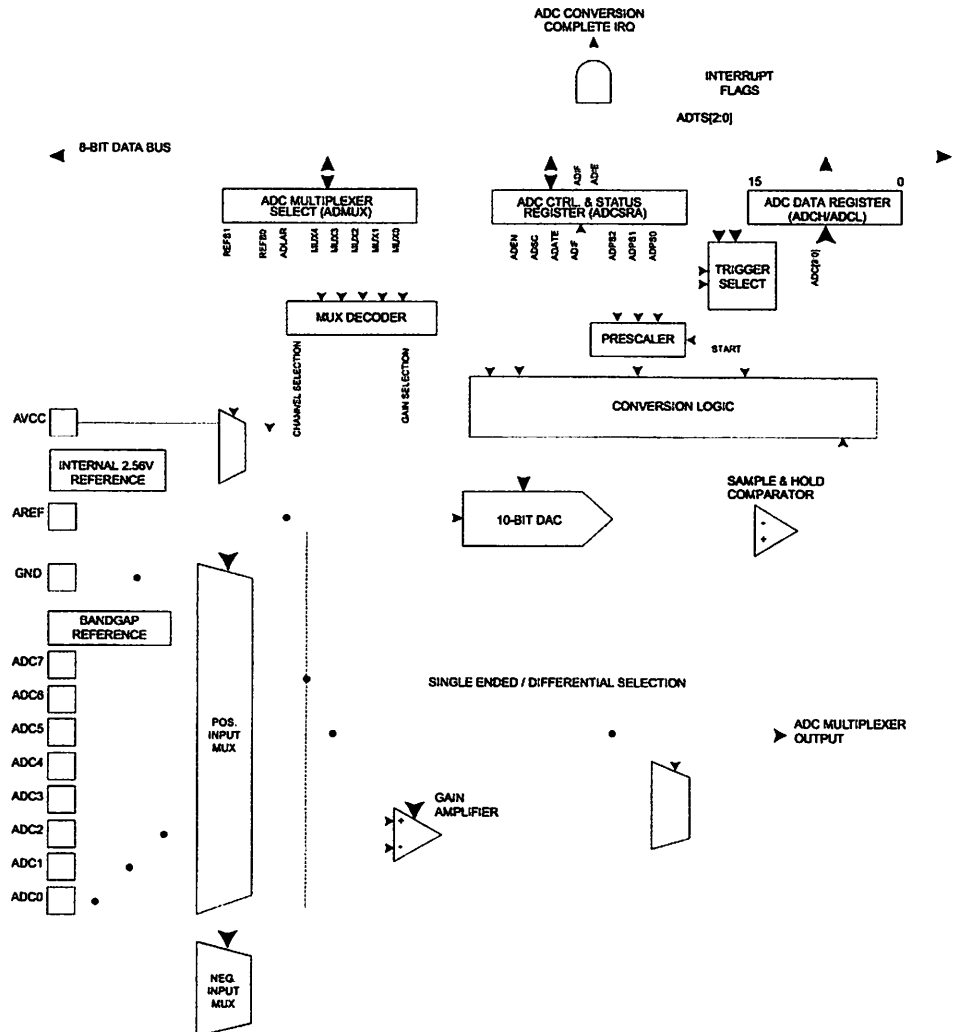
The device also supports 16 differential voltage input combinations. Two of the differential inputs (ADC1, ADC0 and ADC3, ADC2) are equipped with a programmable gain stage, providing amplification steps of 0 dB (1x), 20 dB (10x), or 46 dB (200x) on the differential input voltage before the A/D conversion. Seven differential analog input channels share a common negative terminal (ADC1), while any other ADC input can be selected as the positive input terminal. If 1x or 10x gain is used, 8-bit resolution can be expected. If 200x gain is used, 7-bit resolution can be expected.

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 98.

The ADC has a separate analog supply voltage pin, AVCC. AVCC must not differ more than ± 0.3 V from V_{CC} . See the paragraph "ADC Noise Canceler" on page 210 on how to connect this pin.

Internal reference voltages of nominally 2.56V or AVCC are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

Figure 98. Analog-to-Digital Converter Block Schematic



Operation

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF pin minus 1 LSB. Optionally, AVCC or an internal 2.56V reference voltage may be connected to the AREF pin by writing to the REFSn bits in the ADMUX Register. The internal voltage reference may thus be decoupled by an external capacitor at the AREF pin to improve noise immunity.

The analog input channel and differential gain are selected by writing to the MUX bits in ADMUX. Any of the ADC input pins, as well as GND and a fixed bandgap voltage reference, can be selected as single ended inputs to the ADC. A selection of ADC input pins can be selected as positive and negative inputs to the differential gain amplifier.

If differential channels are selected, the differential gain stage amplifies the voltage difference between the selected input channel pair by the selected gain factor. This amplified value then becomes the analog input to the ADC. If single ended channels are used, the gain amplifier is bypassed altogether.



The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSRA. Voltage reference and input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power saving sleep modes.

The ADC generates a 10-bit result which is presented in the ADC Data Registers, ADCH and ADCL. By default, the result is presented right adjusted, but can optionally be presented left adjusted by setting the ADLAR bit in ADMUX.

If the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH, to ensure that the content of the data registers belongs to the same conversion. Once ADCL is read, ADC access to data registers is blocked. This means that if ADCL has been read, and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. When ADCH is read, ADC access to the ADCH and ADCL Registers is re-enabled.

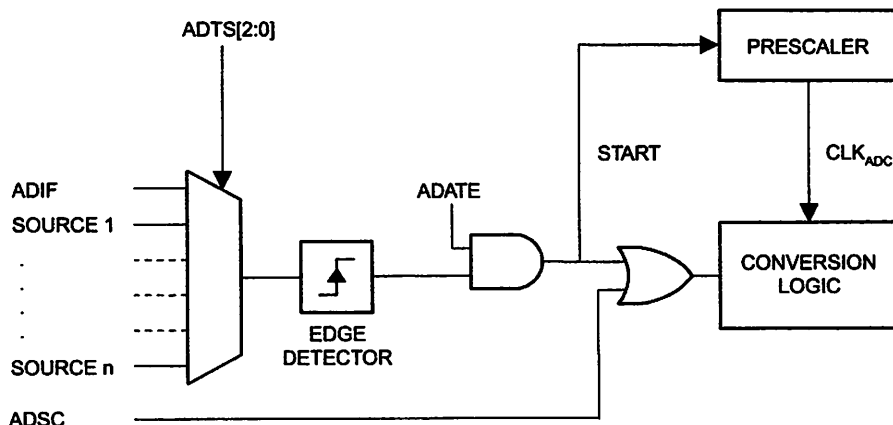
The ADC has its own interrupt which can be triggered when a conversion completes. When ADC access to the data registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

Starting a Conversion

A single conversion is started by writing a logical one to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

Alternatively, a conversion can be triggered automatically by various sources. Auto Triggering is enabled by setting the ADC Auto Trigger Enable bit, ADATE in ADCSRA. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in SFIOR (See description of the ADTS bits for a list of the trigger sources). When a positive edge occurs on the selected trigger signal, the ADC prescaler is reset and a conversion is started. This provides a method of starting conversions at fixed intervals. If the trigger signal still is set when the conversion completes, a new conversion will not be started. If another positive edge occurs on the trigger signal during conversion, the edge will be ignored. Note that an interrupt flag will be set even if the specific interrupt is disabled or the global interrupt enable bit in SREG is cleared. A conversion can thus be triggered without causing an interrupt. However, the interrupt flag must be cleared in order to trigger a new conversion at the next interrupt event.

Figure 99. ADC Auto Trigger Logic

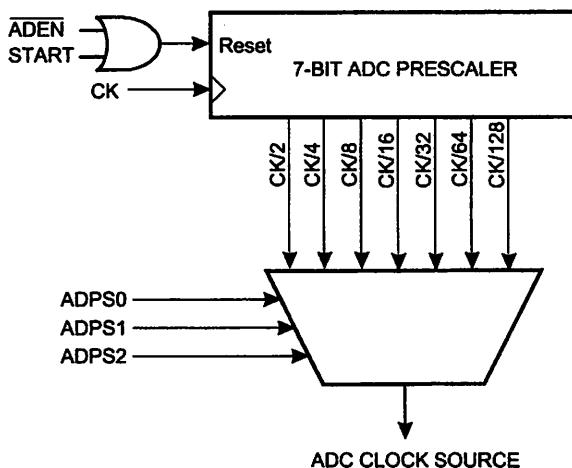


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

Prescaling and Conversion Timing

Figure 100. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200 kHz to get a higher sample rate.



The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100 kHz. The prescaling is set by the ADPS bits in ADCSRA. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA, the conversion starts at the following rising edge of the ADC clock cycle. See "Differential Gain Channels" on page 208 for details on differential conversion timing.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers, and ADIF is set. In Single Conversion mode, ADSC is cleared simultaneously. The software may then set ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

In Free Running mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. For a summary of conversion times, see Table 82.

Figure 101. ADC Timing Diagram, First Conversion (Single Conversion Mode)

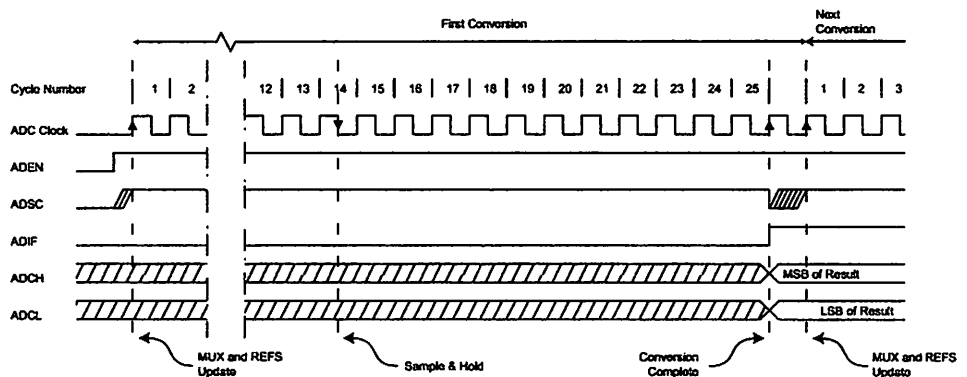


Figure 102. ADC Timing Diagram, Single Conversion

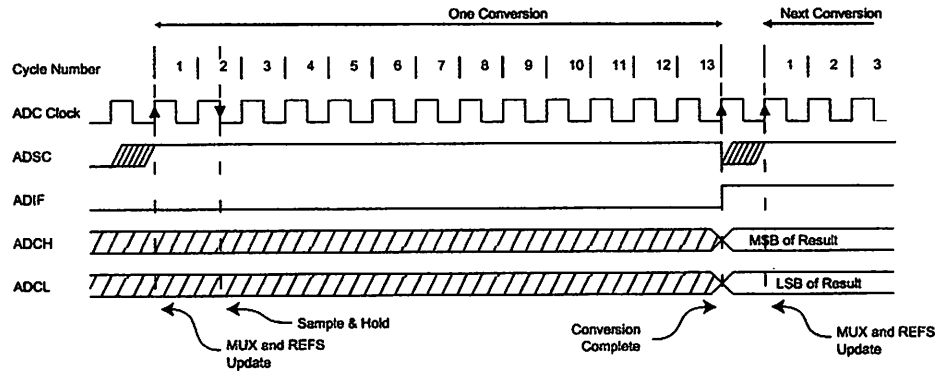


Figure 103. ADC Timing Diagram, Auto Triggered Conversion

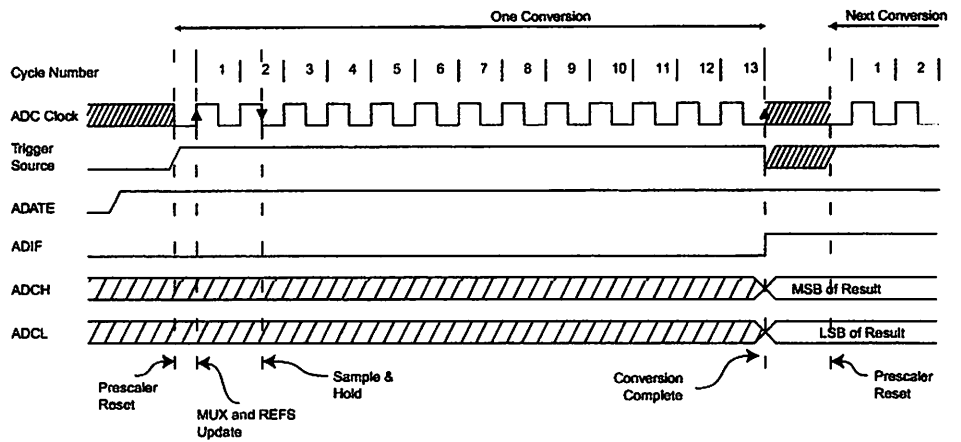


Figure 104. ADC Timing Diagram, Free Running Conversion

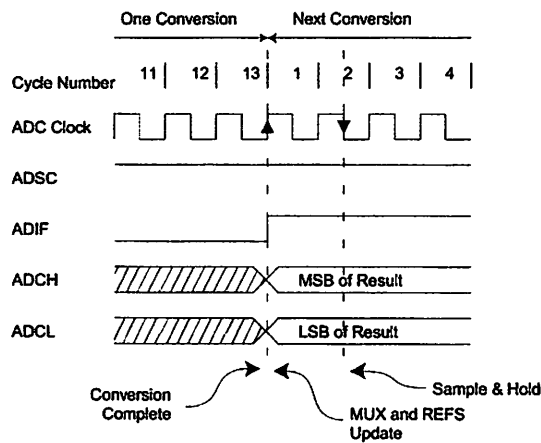




Table 82. ADC Conversion Time

Condition	Sample & Hold (Cycles from Start of Conversion)	Conversion Time (Cycles)
First conversion	14.5	25
Normal conversions, single ended	1.5	13
Auto Triggered conversions	2	13.5
Normal conversions, differential	1.5/2.5 ⁽¹⁾	13/14 ⁽¹⁾

Note: 1. Depending on the state of CK_{ADC2} .

Differential Gain Channels

When using differential gain channels, certain aspects of the conversion need to be taken into consideration.

Differential conversions are synchronized to the internal clock CK_{ADC2} equal to half the ADC clock. This synchronization is done automatically by the ADC interface in such a way that the sample-and-hold occurs at a specific phase of CK_{ADC2} . A conversion initiated by the user (i.e., all single conversions, and the first free running conversion) when CK_{ADC2} is low will take the same amount of time as a single ended conversion (13 ADC clock cycles from the next prescaled clock cycle). A conversion initiated by the user when CK_{ADC2} is high will take 14 ADC clock cycles due to the synchronization mechanism. In free running mode, a new conversion is initiated immediately after the previous conversion completes, and since CK_{ADC2} is high at this time, all automatically started (i.e., all but the first) free running conversions will take 14 ADC clock cycles.

The gain stage is optimized for a bandwidth of 4 kHz at all gain settings. Higher frequencies may be subjected to non-linear amplification. An external low-pass filter should be used if the input signal contains higher frequency components than the gain stage bandwidth. Note that the ADC clock frequency is independent of the gain stage bandwidth limitation. For example, the ADC clock period may be 6 μ s, allowing a channel to be sampled at 12 kSPS, regardless of the bandwidth of this channel.

If differential gain channels are used and conversions are started by Auto Triggering, the ADC must be switched off between conversions. When Auto Triggering is used, the ADC prescaler is reset before the conversion is started. Since the gain stage is dependent of a stable ADC clock prior to the conversion, this conversion will not be valid. By disabling and then re-enabling the ADC between each conversion (writing ADEN in ADCSRA to "0" then to "1"), only extended conversions are performed. The result from the extended conversions will be valid. See "Prescaling and Conversion Timing" on page 205 for timing details.

Changing Channel or Reference Selection

The MUXn and REFS1:0 bits in the ADMUX Register are single buffered through a temporary register to which the CPU has random access. This ensures that the channels and reference selection only takes place at a safe point during the conversion. The channel and reference selection is continuously updated until a conversion is started. Once the conversion starts, the channel and reference selection is locked to ensure a sufficient sampling time for the ADC. Continuous updating resumes in the last ADC clock cycle before the conversion completes (ADIF in ADCSRA is set). Note that the conversion starts on the following rising ADC clock edge after ADSC is written. The user is thus advised not to write new channel or reference selection values to ADMUX until one ADC clock cycle after ADSC is written.

If Auto Triggering is used, the exact time of the triggering event can be indeterminable. Special care must be taken when updating the ADMUX Register, in order to control which conversion will be affected by the new settings.

If both ADATE and ADEN is written to one, an interrupt event can occur at any time. If the ADMUX Register is changed in this period, the user cannot tell if the next conversion is based on the old or the new settings. ADMUX can be safely updated in the following ways:

1. When ADATE or ADEN is cleared.
2. During conversion, minimum one ADC clock cycle after the trigger event.
3. After a conversion, before the interrupt flag used as trigger source is cleared.

When updating ADMUX in one of these conditions, the new settings will affect the next ADC conversion.

Special care should be taken when changing differential channels. Once a differential channel has been selected, the gain stage may take as much as 125 μ s to stabilize to the new value. Thus conversions should not be started within the first 125 μ s after selecting a new differential channel. Alternatively, conversion results obtained within this period should be discarded.

The same settling time should be observed for the first differential conversion after changing ADC reference (by changing the REFS1:0 bits in ADMUX).

ADC Input Channels

When changing channel selections, the user should observe the following guidelines to ensure that the correct channel is selected:

In Single Conversion mode, always select the channel before starting the conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the conversion to complete before changing the channel selection.

In Free Running mode, always select the channel before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing one to ADSC. However, the simplest method is to wait for the first conversion to complete, and then change the channel selection. Since the next conversion has already started automatically, the next result will reflect the previous channel selection. Subsequent conversions will reflect the new channel selection.

When switching to a differential gain channel, the first conversion result may have a poor accuracy due to the required settling time for the automatic offset cancellation circuitry. The user should preferably disregard the first conversion result.

ADC Voltage Reference

The reference voltage for the ADC (V_{REF}) indicates the conversion range for the ADC. Single ended channels that exceed V_{REF} will result in codes close to 0x3FF. V_{REF} can be selected as either AVCC, internal 2.56V reference, or external AREF pin.

AVCC is connected to the ADC through a passive switch. The internal 2.56V reference is generated from the internal bandgap reference (V_{BG}) through an internal amplifier. In either case, the external AREF pin is directly connected to the ADC, and the reference voltage can be made more immune to noise by connecting a capacitor between the AREF pin and ground. V_{REF} can also be measured at the AREF pin with a high impedant voltmeter. Note that V_{REF} is a high impedant source, and only a capacitive load should be connected in a system.

If the user has a fixed voltage source connected to the AREF pin, the user may not use the other reference voltage options in the application, as they will be shorted to the external voltage. If no external voltage is applied to the AREF pin, the user may switch between AVCC and 2.56V as reference selection. The first ADC conversion result after switching reference voltage source may be inaccurate, and the user is advised to discard this result.





If differential channels are used, the selected reference should not be closer to AVCC than indicated in Table 114 on page 258.

ADC Noise Canceler

The ADC features a noise canceler that enables conversion during sleep mode to reduce noise induced from the CPU core and other I/O peripherals. The noise canceler can be used with ADC Noise Reduction and Idle mode. To make use of this feature, the following procedure should be used:

1. Make sure that the ADC is enabled and is not busy converting. Single Conversion mode must be selected and the ADC conversion complete interrupt must be enabled.
2. Enter ADC Noise Reduction mode (or Idle mode). The ADC will start a conversion once the CPU has been halted.
3. If no other interrupts occur before the ADC conversion completes, the ADC interrupt will wake up the CPU and execute the ADC Conversion Complete interrupt routine. If another interrupt wakes up the CPU before the ADC conversion is complete, that interrupt will be executed, and an ADC Conversion Complete interrupt request will be generated when the ADC conversion completes. The CPU will remain in active mode until a new sleep command is executed.

Note that the ADC will not be automatically turned off when entering other sleep modes than Idle mode and ADC Noise Reduction mode. The user is advised to write zero to ADEN before entering such sleep modes to avoid excessive power consumption. If the ADC is enabled in such sleep modes and the user wants to perform differential conversions, the user is advised to switch the ADC off and on after waking up from sleep to prompt an extended conversion to get a valid result.

Analog Input Circuitry

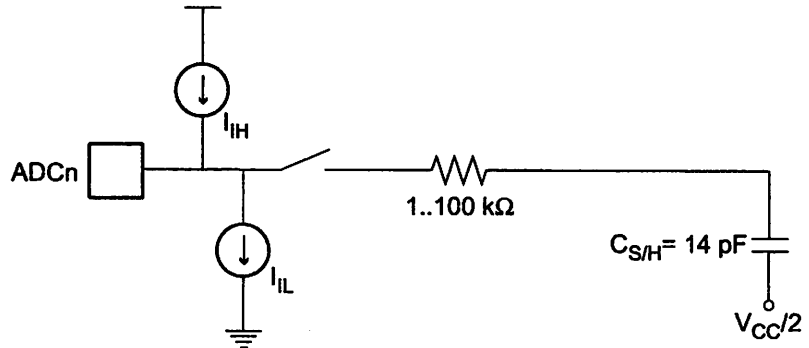
The Analog Input Circuitry for single ended channels is illustrated in Figure 105. An analog source applied to ADC_n is subjected to the pin capacitance and input leakage of that pin, regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 k Ω or less. If such a source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long time the source needs to charge the S/H capacitor, which can vary widely. The user is recommended to only use low impedance sources with slowly varying signals, since this minimizes the required charge transfer to the S/H capacitor.

If differential gain channels are used, the input circuitry looks somewhat different, although source impedances of a few hundred k Ω or less is recommended.

Signal components higher than the Nyquist frequency ($f_{ADC}/2$) should not be present for either kind of channels, to avoid distortion from unpredictable signal convolution. The user is advised to remove high frequency components with a low-pass filter before applying the signals as inputs to the ADC.

Figure 105. Analog Input Circuitry

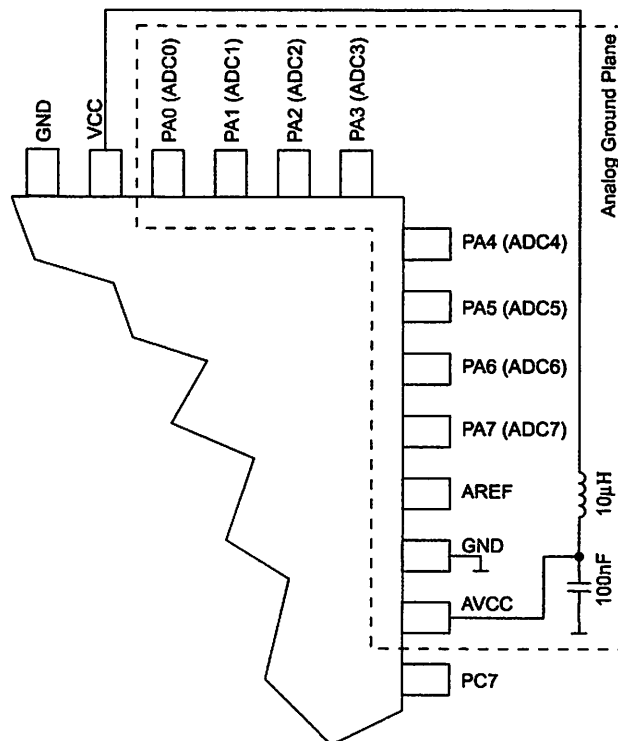


Analog Noise Cancelling Techniques

Digital circuitry inside and outside the device generates EMI which might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

1. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane, and keep them well away from high-speed switching digital tracks.
2. The AVCC pin on the device should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 106.
3. Use the ADC noise canceler function to reduce induced noise from the CPU.
4. If any ADC port pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 106. ADC Power Connections



Offset Compensation Schemes

The gain stage has a built-in offset cancellation circuitry that nulls the offset of differential measurements as much as possible. The remaining offset in the analog path can be measured directly by selecting the same channel for both differential inputs. This offset residue can be then subtracted in software from the measurement results. Using this kind of software based offset correction, offset on any channel can be reduced below one LSB.

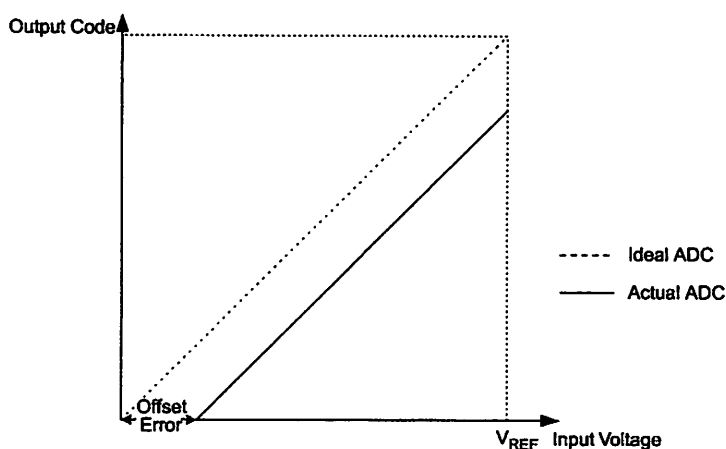
ADC Accuracy Definitions

An n-bit single-ended ADC converts a voltage linearly between GND and V_{REF} in 2^n steps (LSBs). The lowest code is read as 0, and the highest code is read as $2^n - 1$.

Several parameters describe the deviation from the ideal behavior:

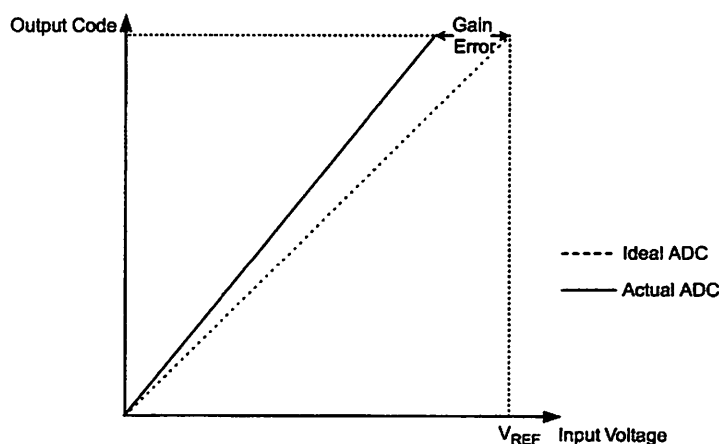
- **Offset:** The deviation of the first transition (0x000 to 0x001) compared to the ideal transition (at 0.5 LSB). Ideal value: 0 LSB.

Figure 107. Offset Error



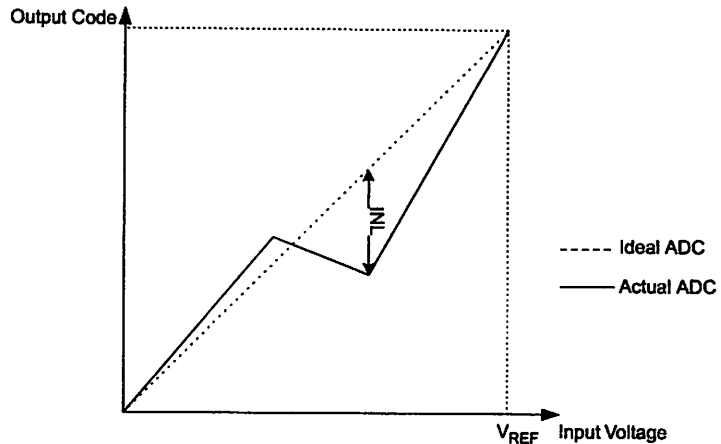
- **Gain Error:** After adjusting for offset, the Gain Error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB.

Figure 108. Gain Error



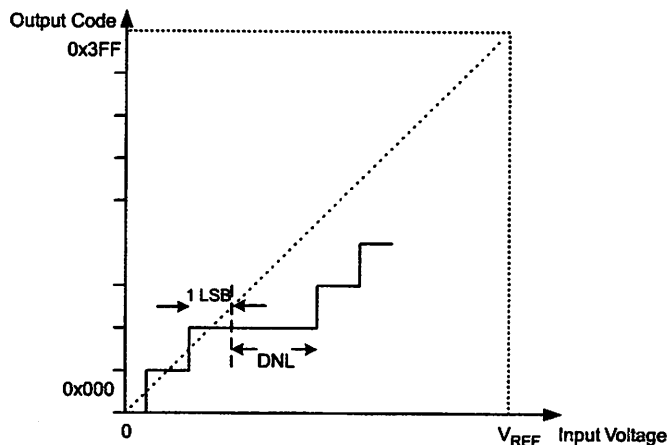
- **Integral Non-linearity (INL):** After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 109. Integral Non-linearity (INL)



- **Differential Non-linearity (DNL):** The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1 LSB). Ideal value: 0 LSB.

Figure 110. Differential Non-linearity (DNL)



- **Quantization Error:** Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ± 0.5 LSB.
- **Absolute Accuracy:** The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of Offset, Gain Error, Differential Error, Non-linearity, and Quantization Error. Ideal value: ± 0.5 LSB.

ADC Conversion Result

After the conversion is complete (ADIF is high), the conversion result can be found in the ADC Result Registers (ADCL, ADCH).

For single ended conversion, the result is

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

where V_{IN} is the voltage on the selected input pin and V_{REF} the selected voltage reference (see Table 84 on page 215 and Table 85 on page 216). 0x000 represents analog ground, and 0x3FF represents the selected reference voltage minus one LSB.

If differential channels are used, the result is

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

where V_{POS} is the voltage on the positive input pin, V_{NEG} the voltage on the negative input pin, $GAIN$ the selected gain factor, and V_{REF} the selected voltage reference. The result is presented in two's complement form, from 0x200 (-512d) through 0x1FF (+511d). Note that if the user wants to perform a quick polarity check of the results, it is sufficient to read the MSB of the result (ADC9 in ADCH). If the bit is one, the result is negative, and if the bit is zero, the result is positive. Figure 111 shows the decoding of the differential input range.

Table 83 shows the resulting output codes if the differential input channel pair (ADCn - ADCm) is selected with a gain of $GAIN$ and a reference voltage of V_{REF} .

Figure 111. Differential Measurement Range

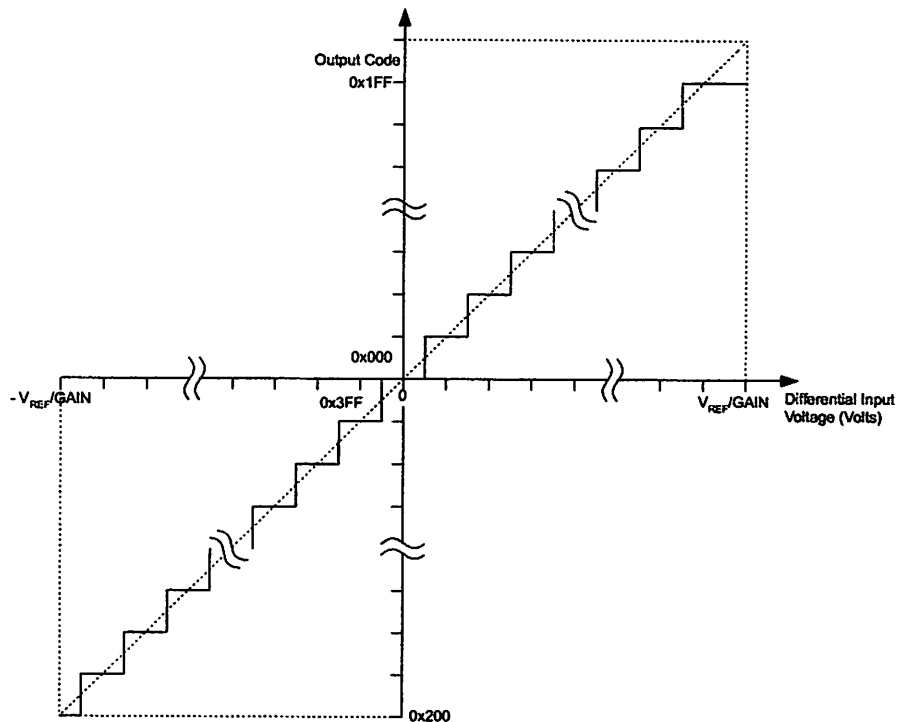


Table 83. Correlation Between Input Voltage and Output Codes

V_{ADCn}	Read Code	Corresponding Decimal Value
$V_{ADCm} + V_{REF}/GAIN$	0x1FF	511
$V_{ADCm} + (511/512) V_{REF}/GAIN$	0x1FF	511
$V_{ADCm} + (510/512) V_{REF}/GAIN$	0x1FE	510
...
$V_{ADCm} + (1/512) V_{REF}/GAIN$	0x001	1
V_{ADCm}	0x000	0
$V_{ADCm} - (1/512) V_{REF}/GAIN$	0x3FF	-1
...
$V_{ADCm} - (511/512) V_{REF}/GAIN$	0x201	-511
$V_{ADCm} - V_{REF}/GAIN$	0x200	-512

Example:

ADMUX = 0xED (ADC3 - ADC2, 10x gain, 2.56V reference, left adjusted result)

Voltage on ADC3 is 300 mV, voltage on ADC2 is 500 mV.

ADCR = $512 * 10 * (300 - 500)/2560 = -400 = 0x270$

ADCL will thus read 0x00, and ADCH will read 0x9C. Writing zero to ADLAR right adjusts the result: ADCL = 0x70, ADCH = 0x02.

ADC Multiplexer Selection Register – ADMUX

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
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 7:6 – REFS1:0: Reference Selection Bits

These bits select the voltage reference for the ADC, as shown in Table 84. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 84. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

• Bit 5 – ADLAR: ADC Left Adjust Result

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately,





regardless of any ongoing conversions. For a complete description of this bit, see “The ADC Data Register – ADCL and ADCH” on page 218.

• **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

The value of these bits selects which combination of analog inputs are connected to the ADC. These bits also select the gain for the differential channels. See Table 85 for details. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set).

Table 85. Input Channel and Gain Selections

MUX4..0	Single Ended Input	Pos Differential Input	Neg Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x

Table 85. Input Channel and Gain Selections (Continued)

MUX4..0	Single Ended Input	Pos Differential Input	Neg Differential Input	Gain
11101		ADC5	ADC2	1x
11110	1.22V (V_{BG})	N/A		
11111	0V (GND)			

ADC Control and Status Register A – ADCSRA

Bit	7	6	5	4	3	2	1	0	ADCSRA
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	
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 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in SFIOR.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.





• **Bits 2:0 – ADPS2:0: ADC Prescaler Select Bits**

These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Table 86. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

The ADC Data Register – ADCL and ADCH

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

When an ADC conversion is complete, the result is found in these two registers. If differential channels are used, the result is presented in two's complement form.

When ADCL is read, the ADC Data Register is not updated until ADCH is read. Consequently, if the result is left adjusted and no more than 8-bit precision is required, it is sufficient to read ADCH. Otherwise, ADCL must be read first, then ADCH.

The ADLAR bit in ADMUX, and the MUXn bits in ADMUX affect the way the result is read from the registers. If ADLAR is set, the result is left adjusted. If ADLAR is cleared (default), the result is right adjusted.

- **ADC9:0: ADC Conversion Result**

These bits represent the result from the conversion, as detailed in “ADC Conversion Result” on page 214.

Special Function IO Register – SFIOR

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:5 – ADTS2:0: ADC Auto Trigger Source**

If ADATE in ADCSRA is written to one, the value of these bits selects which source will trigger an ADC conversion. If ADATE is cleared, the ADTS2:0 settings will have no effect. A conversion will be triggered by the rising edge of the selected interrupt flag. Note that switching from a trigger source that is cleared to a trigger source that is set, will generate a positive edge on the trigger signal. If ADEN in ADCSRA is set, this will start a conversion. Switching to Free Running mode (ADTS[2:0]=0) will not cause a trigger event, even if the ADC Interrupt Flag is set.

Table 87. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

- **Bit 4 – RES: Reserved Bit**

This bit is reserved bit in the ATmega8535, and will always read as zero.

LIQUID CRYSTAL DISPLAY MODULE

M 1 6 3 2

USER MANUAL

Seiko Instruments Inc.

PREFACE

This manual describes technical informations on functions and instructions of M1632 from Seiko Instruments Inc. Please read this instruction manual carefully to understand all the module functions and make the best use of them. Description details may be changed without notice.

Revision Record

<u>Edition</u>	<u>Revision</u>	<u>Date</u>
1	Original	April 1985
2	Completely revised	Jan. 1987

© Seiko Instruments Inc. 1987

Printed in Japan

1. GENERAL

1.1 General

The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view TN LCD panel and a CMOS LCD drive controller built in. The controller has a built-in character generator ROM/RAM, and display data RAM. All the display functions are controlled by instructions and the module can easily be interfaced with an MPU. This makes the module applicable to a wide range of purposes including terminal display units for microcomputers and display units for measuring gages.

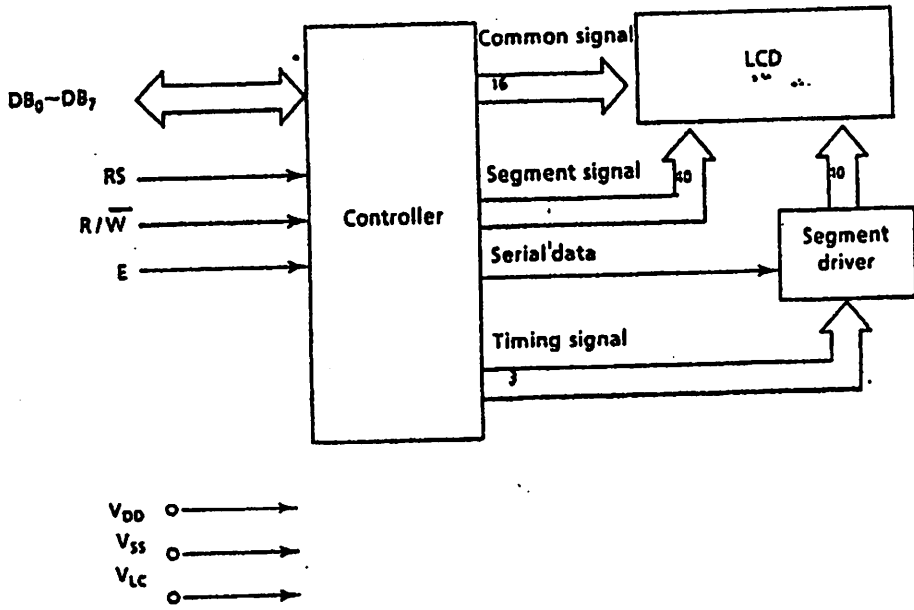
1.2 Features

- 16-character, two-line TN liquid crystal display of 5 x 7 dot matrix + cursor
- Duty ratio: 1/16
- Character generator ROM for 192 character types.
(character font: 5 x 7 dot matrix)
- Character generator RAM for eight character types (program write)
(character font: 5 x 7 dot matrix)
- 80 x 8 bit display data RAM (80 characters maximum)
- Interface with four-bit and eight-bit MPUs possible
- Display data RAM and character generator RAM readable from MPU
- Many instruction functions

Display Clear, Cursor Home, Display ON/OFF, Cursor ON/OFF, Display Character Blink, Cursor Shift, and Display Shift

- Built-in oscillator circuit
- +5 V single power supply
- Built-in automatic reset circuit at power-on
- CMOS process
- Operating temperature range: 0°C to 50°C

1.4 Block Diagram



1.5 Absolute Maximum Ratings

$V_{SS} = 0V$

Item	Symbol	Standard	Unit	Remarks
Power supply voltage	V_{DD}	- 0.3 to + 7.0	V	
	V_{LC}	$V_{DD} - 13.5$ to $V_{DD} + 0.3$	V	
Input voltage	V_{in}	- 0.3 to $V_{DD} + 0.3$,	V	
Operating temperature	T_{opr}	0 to + 50	°C	
Storage temperature	T_{stg}	- 20 to + 60	°C	At 50% RH

1.6 Electrical Characteristics

$V_{DD} = 5V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ C$ to $50^\circ C$

Item		Symbol	Conditions	Standard			Unit
				Min.	Typ.	Max.	
Input voltage	High	V_{IH1}		2.2	-	V_{DD}	V
	Low	V_{IL1}		0	-	0.6	V
Output voltage (TTL)	High	V_{OH1}	$-I_{OH} = 0.205$ mA	2.4	-	-	V
	Low	V_{OL1}	$I_{OL} = 1.2$ mA	-	-	0.4	V
Output voltage (CMOS)	High	V_{OH2}	$-I_{OH} = 0.04$ mA	$0.9V_{DD}$	-	-	V
	Low	V_{OL2}	$I_{OL} = 0.04$ mA	-	-	$0.1V_{DD}$	V
Power supply voltage		V_{DD}		4.75	5.00	5.25	V
		$-V_{LC}$	$V_{DD} = 5V$, $T_A = 25^\circ C$	-	0.25	-	V
Current consumption		I_{DD}		-	2.0	3.0	mA
		I_{LC}	$V_{LC} = 0.25V$	-	-	1.0	mA
Clock oscillation freq.		f_{osc}	Resistance oscillation	190	270	350	kHz

1.7 Optical Characteristics

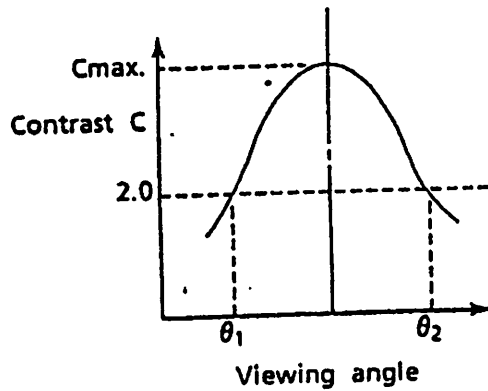
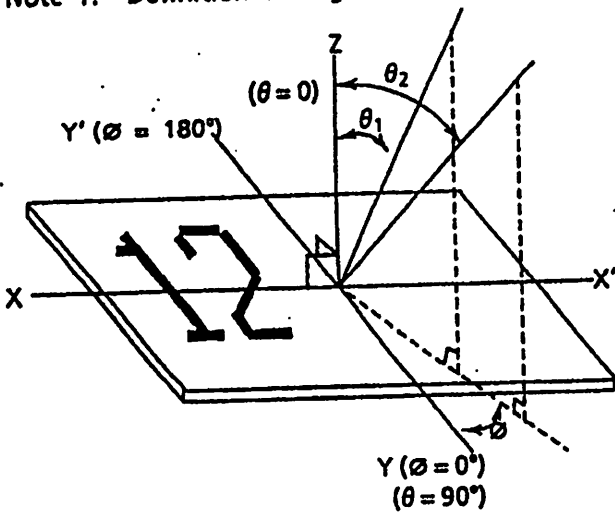
1.7.1 Optical characteristics

Maximum viewing angle: 6 o'clock ($\theta = 0^\circ$)
 $T_A = 25^\circ\text{C}$, $V_{opr} = 4.75\text{ V}$

Item	Symbol	Conditions	Min.	Typ.	Max.	Remarks
Viewing angle	$\theta_2 - \theta_1$	$C \geq 2.0$, $\theta = 0^\circ$	35	-	-	See Notes 1 and 2.
Contrast	C	$\theta = 25^\circ$, $\theta = 0^\circ$	5	8	-	See Note 3.
Rise time	t_{on}	$\theta = 25^\circ$, $\theta = 0^\circ$	-	60 ms	70 ms	See Note 4.
Fall time	t_{off}	$\theta = 25^\circ$, $\theta = 0^\circ$	-	150 ms	170 ms	See Note 4.

Note 1: Definition of angles θ and θ

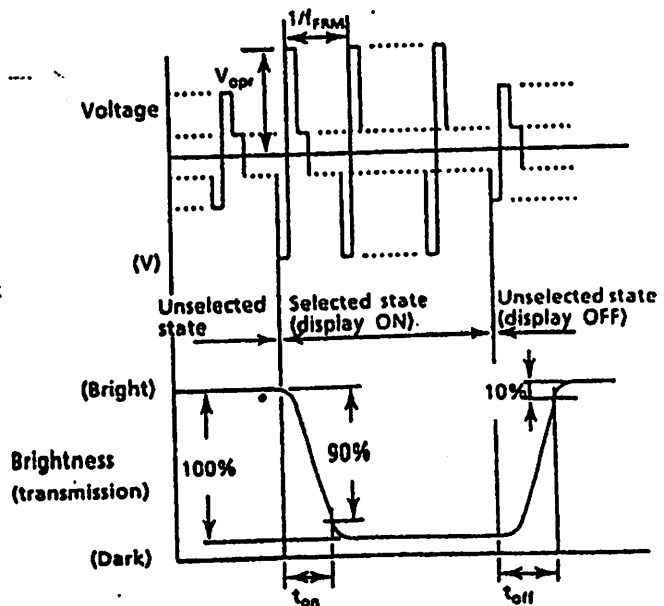
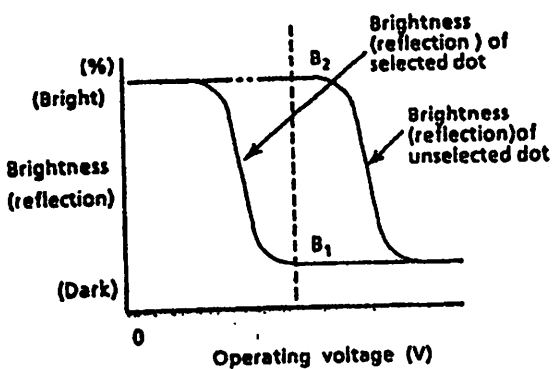
Note 2: Definition of viewing angles θ_1 and θ_2



Note 3: Definition of contrast C

Note 4: Definition of response time

$$C = \frac{\text{Brightness (reflection) of unselected dot (B2)}}{\text{Brightness (reflection) of selected dot (B1)}}$$



V_{opr} : Operating voltage (V)
 f_{FRAM} : Frame frequency (Hz)
 t_{on} : Response time (rise)(ms)
 t_{off} : Response time (fall)(ms)

1.7.2 Recommended operating voltage

The viewing angle and screen contrast of the LCD panel can be varied by changing the liquid crystal operating voltage (V_{opr}), that is V_{LC} .

The optical characteristics is influenced by an ambient temperature. The recommended value of V_{opr} for an ambient temperatures are shown below.

Temperature (°C)	0	10	25	40	50
Voltage V_{opr} (V)	5.00	4.90	4.75	4.60	4.50

$$V_{opr} = V_{DD} - V_{LC}$$

2. OPERATING INSTRUCTIONS

2.1 Terminal Functions

Table 1 Terminal functions

Signal name	No. of terminals	I/O	Destination	Function
DB ₀ to DB ₃	4	I/O	MPU	Tristate bidirectional lower four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. If the interface data is 4 bits, the signals are not used.
DB ₄ to DB ₇	4	I/O	MPU	Tristate bidirectional upper four data buses: Data is read from the module to the MPU or written to the module from the MPU through the buses. DB ₇ is also used as a busy flag.
E	1	Input	MPU	Operation start signal: The signal activates data write or read.
\overline{RW}	1	Input	MPU	Read (R) and Write (W) selection signals 0 : Write 1 : Read
RS	1	Input	MPU	Register selection signals 0 : Instruction register (Write) Busy flag and address counter (Read) 1 : Data register (Write and Read)
V _{LC}	1	-	Power supply	Power supply terminal for driving liquid crystal display: The screen contrast can be varied by changing V _{LC} .
V _{DD}	1	-	Power supply	+ 5 V
V _{SS}	1	-	Power supply	Ground terminal: 0 V

2.2 Basic Operations

2.2.1 Registers

The controller has two kinds of eight-bit registers: the instruction register (IR) and the data register (DR). They are selected by the register select (RS) signal as shown in Table 2.

The IR stores instruction codes such as Display Clear and Cursor Shift, and the address information of display data RAM (DD RAM) and character generator RAM (CG RAM). They can be written from the MPU, but cannot be read to the MPU.

The DR temporarily stores data to be written into DD RAM or CG RAM, or read from DD RAM or CG RAM. When data is written into DD RAM or CG RAM from the MPU, the data in the DR is automatically written into DD RAM or CG RAM by internal operation. However, when data is read from DD RAM or CG RAM, the necessary data address is written into the IR. The specified data is read out to the DR and then the MPU reads it from the DR. After the read operation, the next address is set and DD RAM or CG RAM data at the address is read into the DR for the next read operation.

Table 2 Register selection

RS	R/W	Operation
0	0	IR selection, IR write. Internal operation: Display clear
0	1	Busy flag (DB ₇) and address counter (DB ₀ to DB ₆) read
1	0	DR selection, DR write. Internal operation: DR to DD RAM or CG RAM
1	1	DR selection, DR read. Internal operation: DD RAM or CG RAM to DR

2.2.2 Busy flag (BF)

The flag indicates whether the module is ready to accept the next instruction. As shown in Table 2, the signal is output to DB₇ if RS = 0 and R/W = 1. If the value is 1, the module is working internally and the instruction cannot be accepted. If the value is 0, the next instruction can be written. Therefore, the flag status needs to be checked before executing an instruction. If an instruction is executed without checking the flag status, wait for more than the execution time shown by 2.4 Instruction Outline.

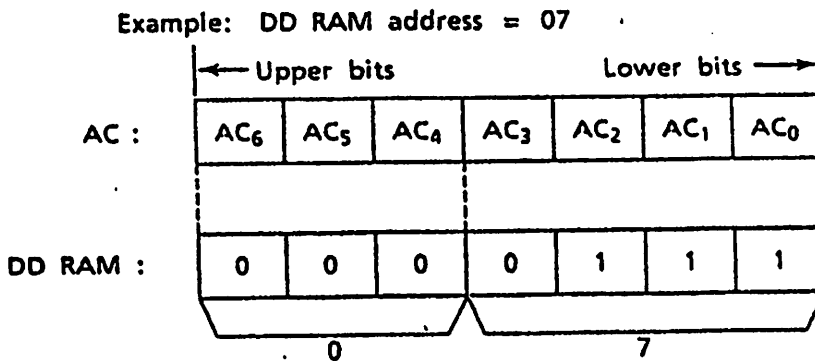
2.2.3 Address counter (AC)

The counter specifies an address when data is written into DD RAM or CG RAM and the data stored in DD RAM or CG RAM is read out. If an Address Set instruction (for DD RAM or CG RAM) is written in the IR, the address information is transferred from the IR to the AC. When display data is written into or read from DD RAM or CG RAM, the AC is automatically incremented or decremented by one according to the Entry Mode Set. The contents of the AC are output to DB₀ to DB₆ as shown in Table 2 if RS = 0 and $\overline{R/W} = 1$.

2.2.4 Display data RAM (DD RAM)

DD RAM has a capacity of up to 80×8 bits and stores display data of 80 eight-bit character codes. Some storage areas of DD RAM which are not used for display can be used as general data RAM.

A DD RAM address to be set in the AC is expressed in hexadecimal form as follows.



00H to 0FH of the DD RAM address is set in the line 1, and 40H to 4FH in the line 2.

Note : The addresses in the digit 16 of line 1 and the digit 1 of line 2 are not consecutive.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DD RAM address
Line 2	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

If the display is shifted, DD RAM address 00H to 27H are displayed in line 1 and 40H to 67H in line 2. The following figures are examples of display shifts.

*Left shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	DD RAM address
Line 2	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	

*Right shift

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Display digit
Line 1	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	DD RAM address
Line 2	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	

2.2.5 Character generator ROM (CG ROM)

Character generator ROM generates 192 types of 5 x 7 dot-matrix character patterns from eight-bit character codes.

Table 3 shows the correspondence between the CG ROM character codes and character patterns.

2.2.6 Character generator RAM (CG RAM)

CG RAM is used to create character patterns freely by programming. Eight types of character patterns can be written.

Table 4 shows the character patterns created from CG RAM addresses and data. To display a created character pattern, the character code in the left column of the table is written into DD RAM corresponding to the display position (digit). The areas not used for display are available as general data RAM.

Table 3 Correspondence between character codes and character patterns

Upper bit 4 bit Lower bit 4 bit	0	2	3	7	8	6	7	1010	1011	1100	1101	1110	1111
x x x x 0000	CG RAM (1)		0	1	2	3	4	5	6	7	8	9	0
x x x x 0001	(2)	1	2	3	4	5	6	7	8	9	0	1	2
x x x x 0010	(3)	3	4	5	6	7	8	9	0	1	2	3	4
x x x x 0011	(4)	4	5	6	7	8	9	0	1	2	3	4	5
x x x x 0100	(5)	5	6	7	8	9	0	1	2	3	4	5	6
x x x x 0101	(6)	6	7	8	9	0	1	2	3	4	5	6	7
x x x x 0110	(7)	7	8	9	0	1	2	3	4	5	6	7	8
x x x x 0111	(8)	8	9	0	1	2	3	4	5	6	7	8	9
x x x x 1000	(1)	9	0	1	2	3	4	5	6	7	8	9	0
x x x x 1001	(2)	0	1	2	3	4	5	6	7	8	9	0	1
x x x x 1010	(3)	1	2	3	4	5	6	7	8	9	0	1	2
x x x x 1011	(4)	2	3	4	5	6	7	8	9	0	1	2	3
x x x x 1100	(5)	3	4	5	6	7	8	9	0	1	2	3	4
x x x x 1101	(6)	4	5	6	7	8	9	0	1	2	3	4	5
x x x x 1110	(7)	5	6	7	8	9	0	1	2	3	4	5	6
x x x x 1111	(8)	6	7	8	9	0	1	2	3	4	5	6	7

Table 4 Relationships between CG RAM addresses (DD RAM) and character codes (CG RAM data) and character patterns (CG RAM data)

Character code (DD RAM data)		CG RAM address				Character pattern (CG RAM data)									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
← Upper bit Lower bit →		← Upper bit		Lower bit →		← Upper bit		Lower bit →							
0 0 0 0 * 0 0 0 0		0 0 0		0 0 0		0 0 0		* * *				Example of character pattern (R)		← Cursor position	
0 0 0 0 * 0 0 0 1		0 0 1		0 1 1		1 0 0		* * *				Example of character pattern (Y)			
0 0 0 0 * 1 1 1 1		1 1 1		1 0 0		1 0 1		* * *							

- Notes:**
- In CG RAM data, 1 corresponds to Selection and 0 to Non-selection on the display.
 - Character code bits 0 to 2 and CG RAM address bits 3 to 5 correspond with each other (three bits, eight types).
 - CG RAM address bits 0 to 2 specify a line position for a character pattern. Line 8 of a character pattern is the cursor position where the logical sum of the cursor and CG RAM data is displayed. Set the data of line 8 to 0 to display the cursor. If the data is changed to 1, one bit lights, regardless of the cursor.

The character pattern column positions correspond to CG RAM data bits 0 to 4 and bit 4 comes to the left end. CG RAM data bits 5 to 7 are not displayed but can be used as general data RAM.

When reading a character pattern from CG RAM, set to 0 all of character code bits 4 to 7. Bits 0 to 2 determine which pattern will be read out. Since bit 3 is not valid, 00H and 08H select the same character.

2.3 Timing Characteristics

2.3.1 Write timing characteristics

$V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ C$ to $50^\circ C$

Item		Symbol	Standard		Unit
			Min.	Max.	
Enable cycle time		t_{cycE}	1000	-	ns
Enable pulse width	High level	PW_{EH}	450	-	ns
Enable rise and fall time		t_{Er}, t_{Ef}	-	25	ns
Setup time	$RS, R\bar{W} - E$	t_{AS}	140	-	ns
Address hold time		t_{AH}	10	-	ns
Data setup time		t_{DSW}	195	-	ns
Data hold time		t_H	10	-	ns

Write operation

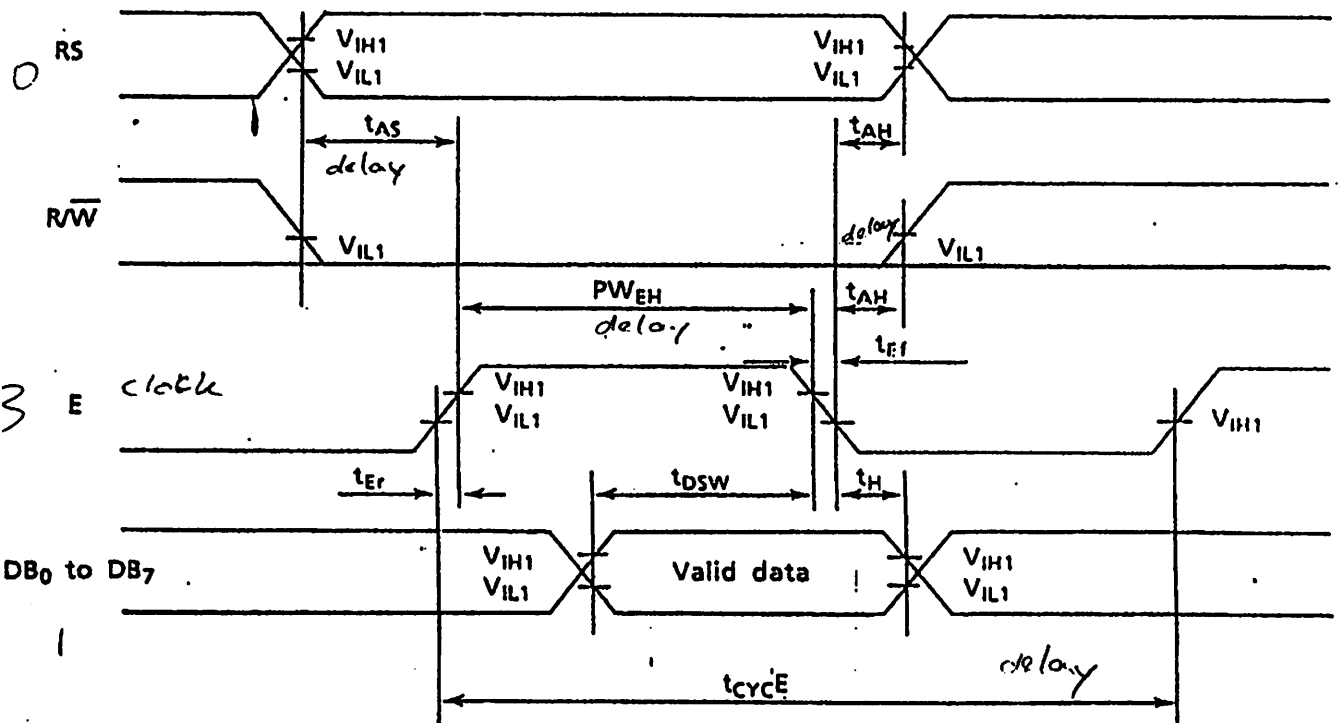


Figure 3 Data write from MPU to module

2.3.2 Read timing characteristics

$V_{DD} = 5.0V \pm 5\%$, $V_{SS} = 0V$, $T_A = 0^\circ\text{C}$ to 50°C

Item	Symbol	Standard		Unit
		Min.	Max.	
Enable cycle time	t_{CYCE}	1000	-	ns
Enable pulse width	High level	PW_{EH}	-	ns
Enable rise and fall time	t_{ER}, t_{EF}	-	25	ns
Setup time	$RS, \overline{R/W}-E$	t_{AS}	-	ns
Address hold time	t_{AH}	10	-	ns
Data delay time	t_{DDR}	-	320	ns
Data hold time	t_{H}	20	-	ns

Read operation

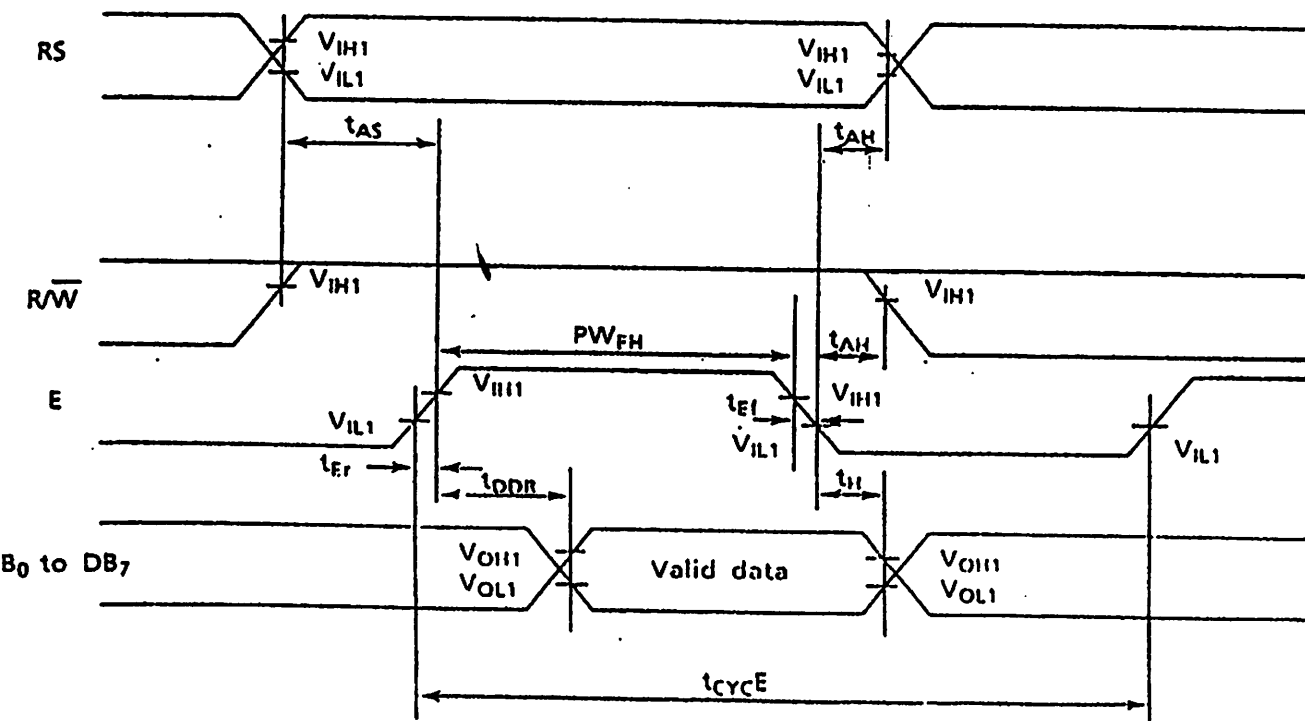


Figure 4 Data read from module to MPU

4 Instruction Outline

Table 5 List of instructions

Instruction	Code											Function	Execution time
	RS	RD	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀			
1) Display clear ✓	0	0	0	0	0	0	0	0	0	0	1	Clears all display and returns cursor to home position (address 0)	1.64 ms
2) Cursor Home ✓	0	0	0	0	0	0	0	0	0	1	•	Returns cursor to home position. Shifted display returns to home position and DD RAM contents do not change.	1.64 ms
3) Entry Mode Set ✓	0	0	0	0	0	0	0	0	1	WD	S ↓ ○	Sets direction of cursor movement and whether display will be shifted when data is written or read	40 μs
4) Display ON / OFF control	0	0	0	0	0	0	0	1	B	C	B	Turns ON/OFF total display (D) and cursor (C), and makes cursor position column start blinking (B)	40 μs
5) Cursor/Display Shift ✓	0	0	0	0	0	0	1	S/C	R/L	•	•	Moves cursor and shifts display without changing DD RAM contents	40 μs
6) Function Set ✓	0	0	0	0	1	DL	1	•	•	•	•	Sets interface data length (DL)	40 μs
7) CG RAM Address Set	0	0	0	1	ACG						Sets CG RAM address to start transmitting or receiving CG RAM data	40 μs	
8) DD RAM Address Set	0	0	1	ADD						Sets DD RAM address to start transmitting or receiving DD RAM data	40 μs		
9) BF/Address Read	0	1	BF	AC						Reads BF indicating module in internal operation and AC contents (used for both CG RAM and DD RAM)	0 μs		
10) Data Write to CG RAM or DD RAM	1	0	Write Data						Writes data into DD RAM or CG RAM	40 μs			
11) Data Read from CG RAM or DD RAM	1	1	Read Data						Reads data from DD RAM or CG RAM	40 μs			

• : Invalid bit

ACG : CG RAM address

ADD : DD RAM address

WD = 1 : Increment

WD = 0 : Decrement

C = 1 : Cursor ON

C = 0 : Cursor OFF

R/L = 1 : Right shift

R/L = 0 : Left shift

S = 1 : Display shift

S = 0 : No display shift

B = 1 : Blink ON

B = 0 : Blink OFF

DL = 1 : 8 bits

DL = 0 : 4 bits

D = 1 : Display ON

D = 0 : Display OFF

S/C = 1 : Display shift

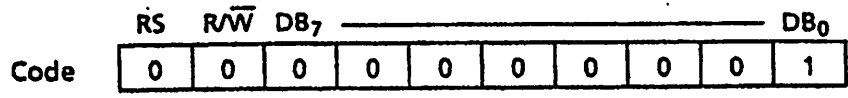
S/C = 0 : Cursor movement

BF = 1 : Internal operation in progress

BF = 0 : Instruction can be accepted

5 Instruction Details

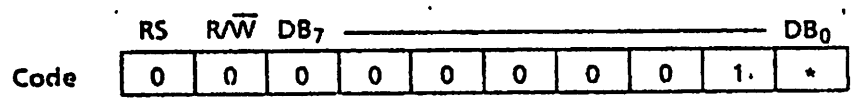
(1) Display Clear



Display Clear clears all display and returns cursor to home position (address 0). Space code 20 (hexadecimal) is written into all the addresses of DD RAM, and DD RAM address 0 is set to the AC. If shifted, the display returns to the original position. After execution of the Display Clear instruction, the entry mode is incremented.

Note: When executing the Display Clear instruction, follow the restrictions listed in Table 6.

(2) Cursor Home



* : Invalid bit

Cursor Home returns cursor to home position (address 0). DD RAM address 0 is set to the AC. The cursor returns to the home position. If shifted, the display returns to the original position. The DD RAM contents do not change. If the cursor or blinking is ON, it returns to the left side.

Note: When executing the Cursor Home instruction, follow the restrictions listed in Table 6.

Table 6 Restrictions on execution of Display Clear and Cursor Home instructions

Conditions of use	Restrictions
When executing the Display Clear or Cursor Home instruction when the display is shifted (after execution of Display Shift instruction)	The Cursor Home instruction should be executed again immediately after the Display Clear or Cursor Home instruction is executed. Do not leave an interval of a multiple of $400/f_{osc}$ second after the first execution. Example: 1.5 ms, 3 ms, 4.5 ms for $f_{osc} = 270$ kHz * f_{osc} : Oscillation frequency
When 23 ₁₁ , 77 ₁₁ , 63 ₁₁ , or 67 ₁₁ is used as a DD RAM address to execute Cursor Home instruction	Before executing the Cursor Home instruction, the data of the four DD RAM addresses given at the left should be read and saved. After execution, write the data again in DD RAM. (This restriction is necessary to prevent the contents of the DD RAM addresses from being destroyed after the Cursor Home instruction has been executed.)

(3) Entry Mode Set

	RS	R/W	DB ₇						DB ₀	
Code	0	0	0	0	0	0	0	1	I/D	S

Entry Mode Set sets the direction of cursor movement and whether display will be shifted.

I/D : The DD RAM address is incremented or decremented by one when a character code is written into or read from DD RAM. This is also true for writing into or reading from CG RAM.

When I/D = 1, the address is incremented by one and the cursor or blink moves to the right.

When I/D = 0, the address is decremented by one and the cursor or blink moves to the left.

S : If S = 1, the entire display is shifted either to the right or left for writing into DD RAM. The cursor position does not change, only the display moves. There is no display shift for reading from DD RAM.

When S = 1 and I/D = 1, the display shifts to the left.

When S = 1 and I/D = 0, the display shifts to the right.

If S = 0, the display does not shift.

(4) Display ON/OFF Control

	RS	R/W	DB ₇						DB ₀	
Code	0	0	0	0	0	0	1	D	C	B

Display ON/OFF Control turns the total display and the cursor ON and OFF, and makes the cursor position start blinking. Cursor ON/OFF and blinking is done at the column indicated by the specified DD RAM address by the AC.

D : When D = 1, the display is turned ON.

When D = 0, the display is turned OFF.

If D = 0 is used, display data remains in DD RAM. Change 0 to 1 to display data.

C : When $C = 1$, the cursor is displayed.

When $C = 0$, the cursor is not displayed.

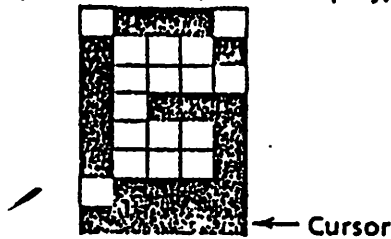
The cursor is displayed in the dot line below the 5 x 7 dot-matrix character fonts. If the cursor is OFF, display data is written into DD RAM in the order specified by I/D.

B : When $B = 1$, the character at the cursor position starts blinking.

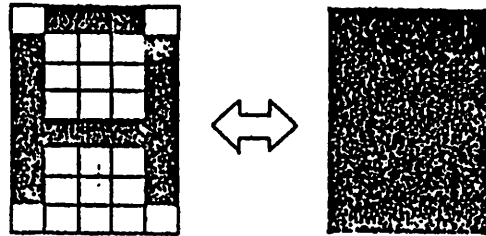
When $B = 0$, it does not blink.

For blinking, all-black dots and the character are switched about every 0.4 seconds. The cursor and blinking can be set at the same time.

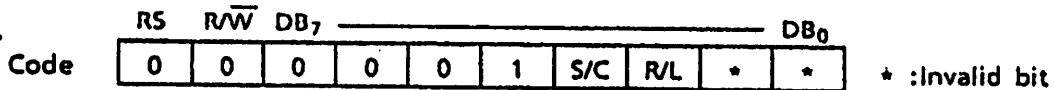
Example: $C = 1$ (cursor display)



$B = 1$ (blinking)



(5) Cursor/Display Shift



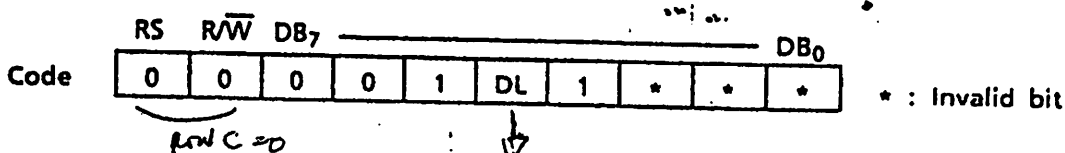
Cursor/Display Shift moves the cursor and shifts the display without changing the DD RAM contents.

The cursor position and the AC contents match. This instruction is available for display correction and retrieval because the cursor position or display can be shifted without writing or reading display data. Since the DD RAM capacity is 40-character and two lines, the cursor is shifted from digit 40 of line 1 to digit 1 of line 2. Displays of lines 1 and 2 are shifted at the same time. Therefore, the display pattern of line 2 is not shifted to line 1.

S/C	R/L	Operation
0	0	The cursor position is shifted to the left (the AC decrements one).
0	1	The cursor position is shifted to the right (the AC increments one).
1	0	The entire display is shifted to the left with the cursor.
1	1	The entire display is shifted to the right with the cursor.

Note: If only display shift is done, the AC contents do not change.

(6) Function Set



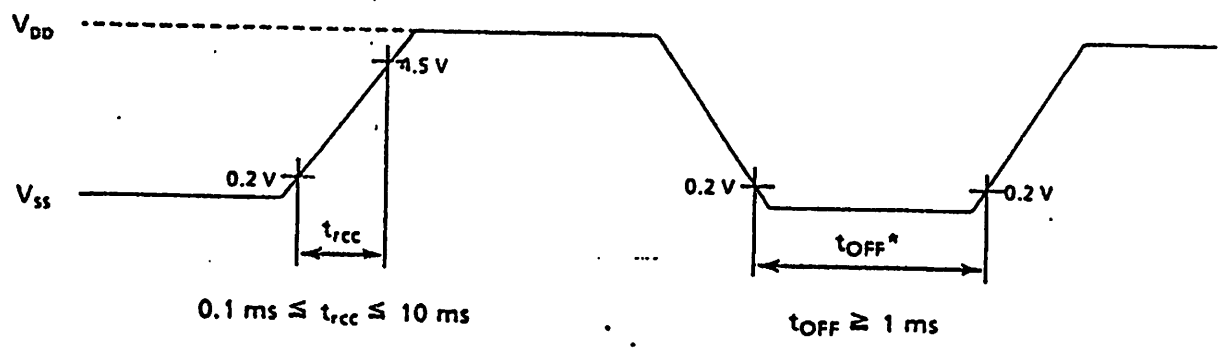
Function Set sets the interface data length.

- DL : Interface data length
 - When DL = 1, the data length is set at eight bits (DB7 to DB0).
 - When DL = 0, the data length is set at four bits (DB7 to DB4).
 - The upper four bits are transferred first, then the lower four bits follow.

The Function Set instruction must be executed prior to all other instructions except for Busy Flag/Address Read. If another instruction is executed first, no function instruction except changing the interface data length can be executed.

Remarks: Initialization

The system is automatically initialized at power-on if the following power supply conditions are satisfied.



* t_{off} : Time when power supply is OFF if cut instantaneously or turned ON and OFF repeatedly

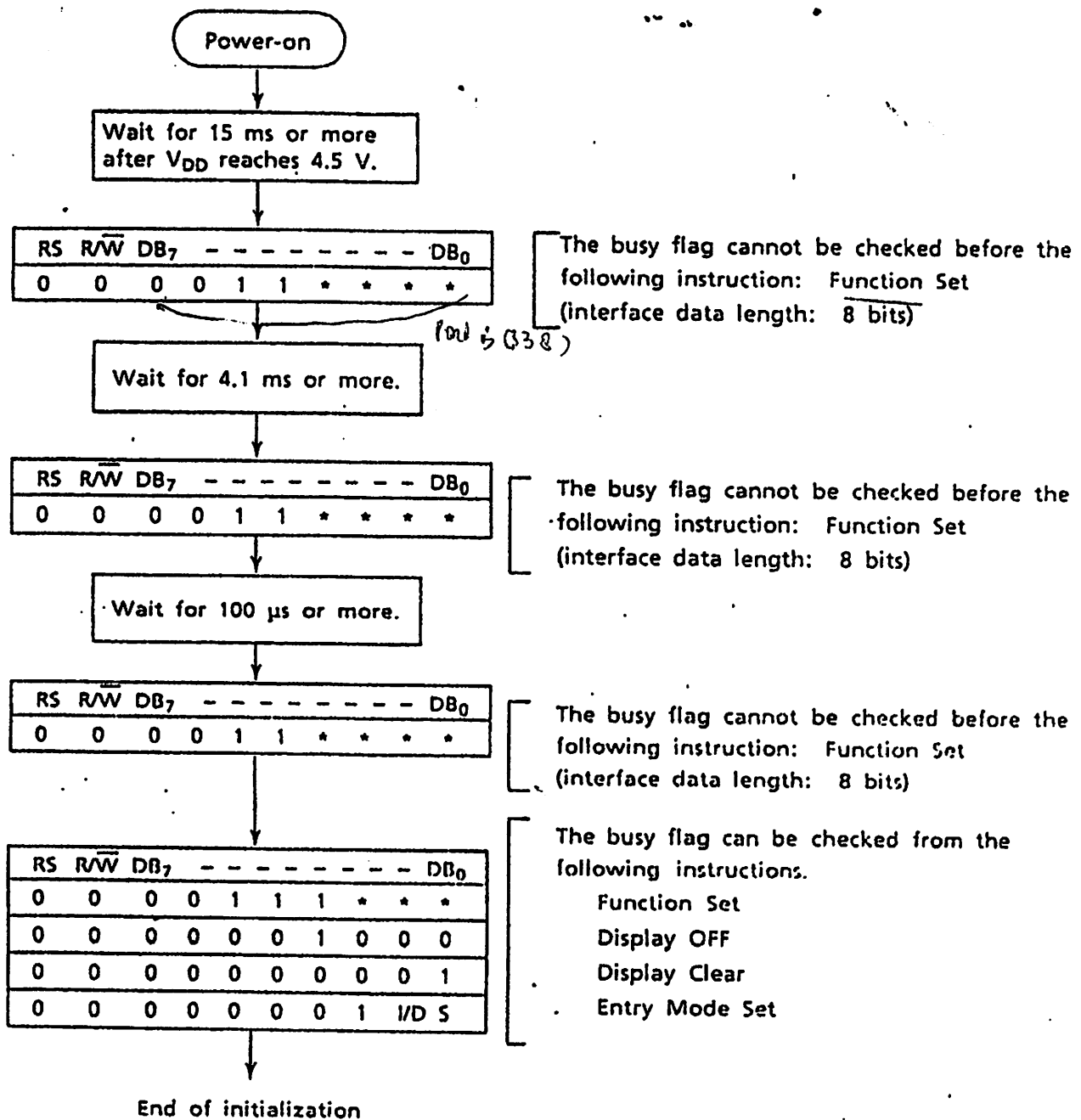
The following instructions are executed for initialization.

- 5 x 7 dot-matrix character font: 1/8 duty
- Display clear
- Function Set DL = 1: Interface data length: 8 bits
- Display ON/OFF Control D = 0: Display OFF
 C = 0: Cursor OFF
 B = 0: Blink OFF
- Entry mode I/O = 1: Increment
 S = 0: No display shift

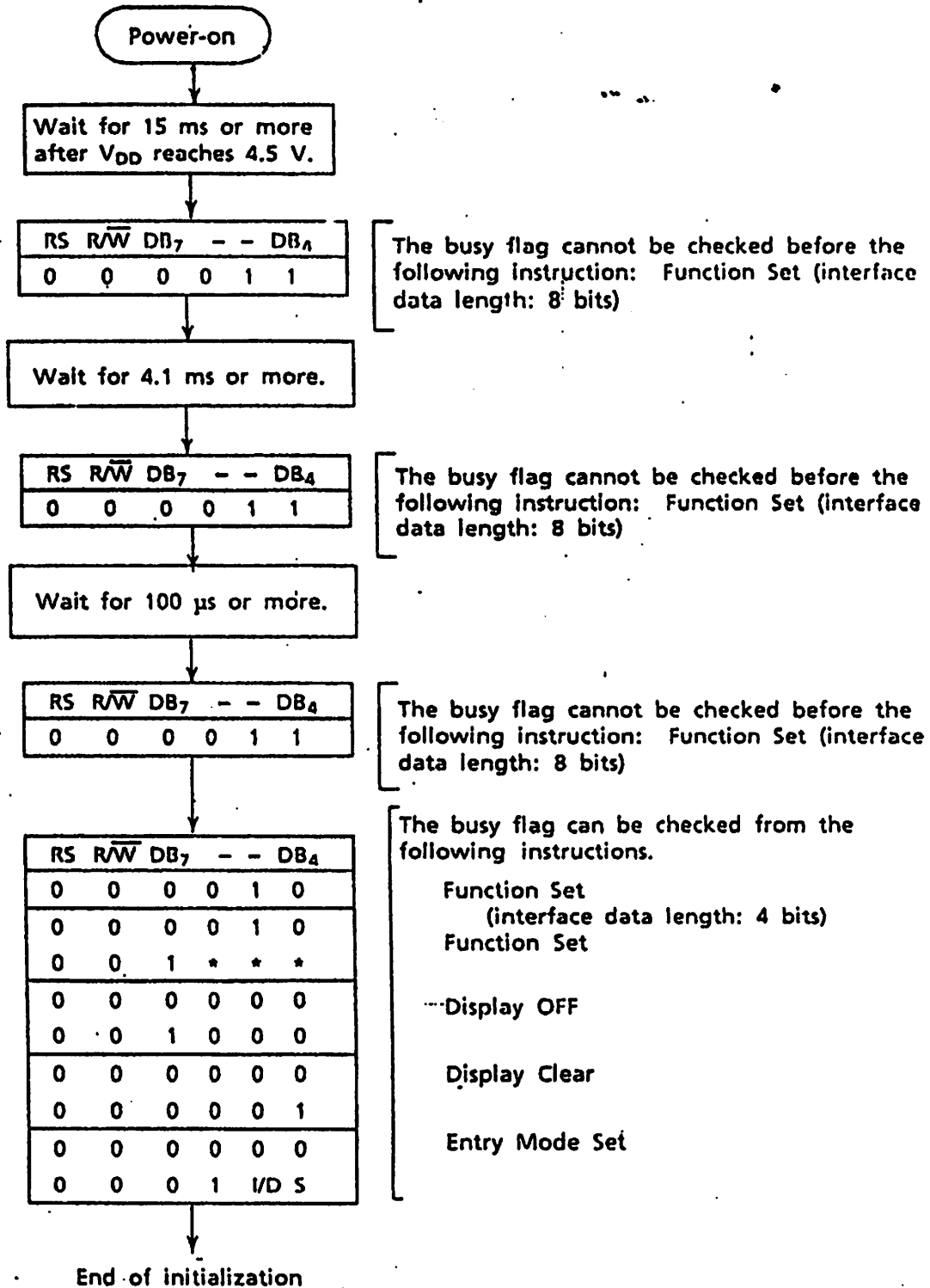
Since the condition is not suitable for the M1632, further function setting is necessary.

If automatic initialization is not executed because the above power supply conditions are not satisfied, use the instruction from next page on.

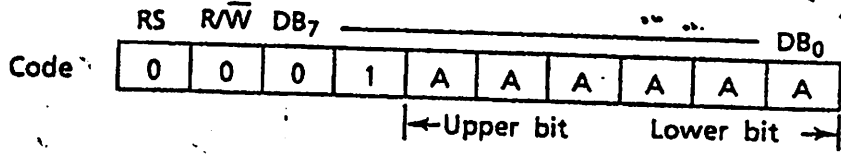
(a) Interface data length : Eight bits



(b) Interface data length: Four bits

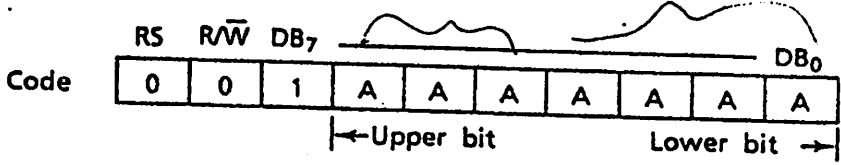


(7) CG RAM Address Set



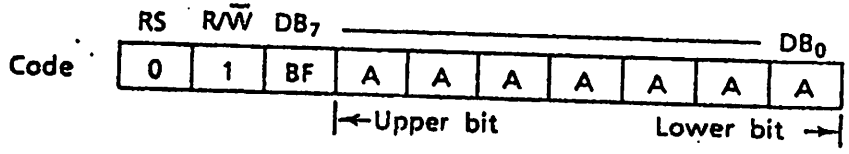
CG RAM addresses expressed as binary AAAAAA are set to the AC. Then data in CG RAM is written from or read to the MPU.

(8) DD RAM Address Set



DD RAM addresses expressed as binary AAAAAAA are set to the AC. Then data in DD RAM is written from or read to the MPU. The addresses used for display in line 1 (AAAAAAA) are 00H to 27H and those for line 2 (AAAAAAA) are 40H to 67H.

9) Busy Flag/Address Read



The BF signal is read out, indicating that the module is working internally because of the previous instruction.

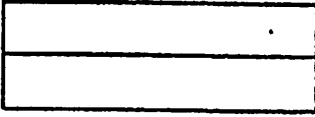

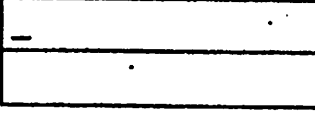

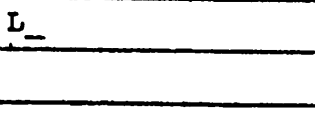
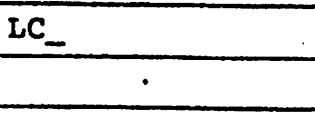


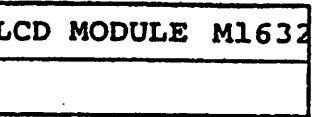
When BF = 1, the module is working internally and the next instruction cannot be accepted until the BF value becomes 0.

When BF = 0, the next instruction can be accepted.

Therefore, make sure that BF = 0 before writing the next instruction. The AC values of binary AAAAAAA are read out at the same time as reading the busy flag. The AC addresses are used for both CG RAM and DD RAM but the address set before execution of the instruction determines which address is to be used.

2.6 Examples of Instruction Use

(1) Interface data length: Eight bits

No.	Instruction	Display	Operation										
1	Power-on <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>/</td> <td>/</td> <td colspan="3">/</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	/	/	/				The built-in reset circuit initializes the module.
RS	R/W	DB ₇	—	DB ₀									
/	/	/											
2	Function Set ✓ <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 1 1</td> <td>*</td> <td>*</td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 1 1 1	*	*		The interface data length is set to 8 bits. The character format becomes 5 x 7 dot-matrix at 1/16 duty cycle.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 1 1 1	*	*									
3	Display ON/OFF Control <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 1 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 1 1 1 0				The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 0 0 1 1 1 0											
4	Entry Mode Set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0 0 1 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	0	0	0 0 0 0 0 1 1 0				The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
RS	R/W	DB ₇	—	DB ₀									
0	0	0 0 0 0 0 1 1 0											
5	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 1 1 0 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 1 1 0 0				L is written. The AC is incremented by one and the cursor shifts to the right.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 1 0 0 1 1 0 0											
6	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0 0 0 1 1</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 1 0 0 0 0 1 1				C is written.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 1 0 0 0 0 1 1											
7													
8	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>—</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 0 1 1 0 0 1 0</td> <td></td> <td></td> </tr> </table>	RS	R/W	DB ₇	—	DB ₀	1	0	0 0 1 1 0 0 1 0				2 is written in digit 16. Cursor disappears.
RS	R/W	DB ₇	—	DB ₀									
1	0	0 0 1 1 0 0 1 0											

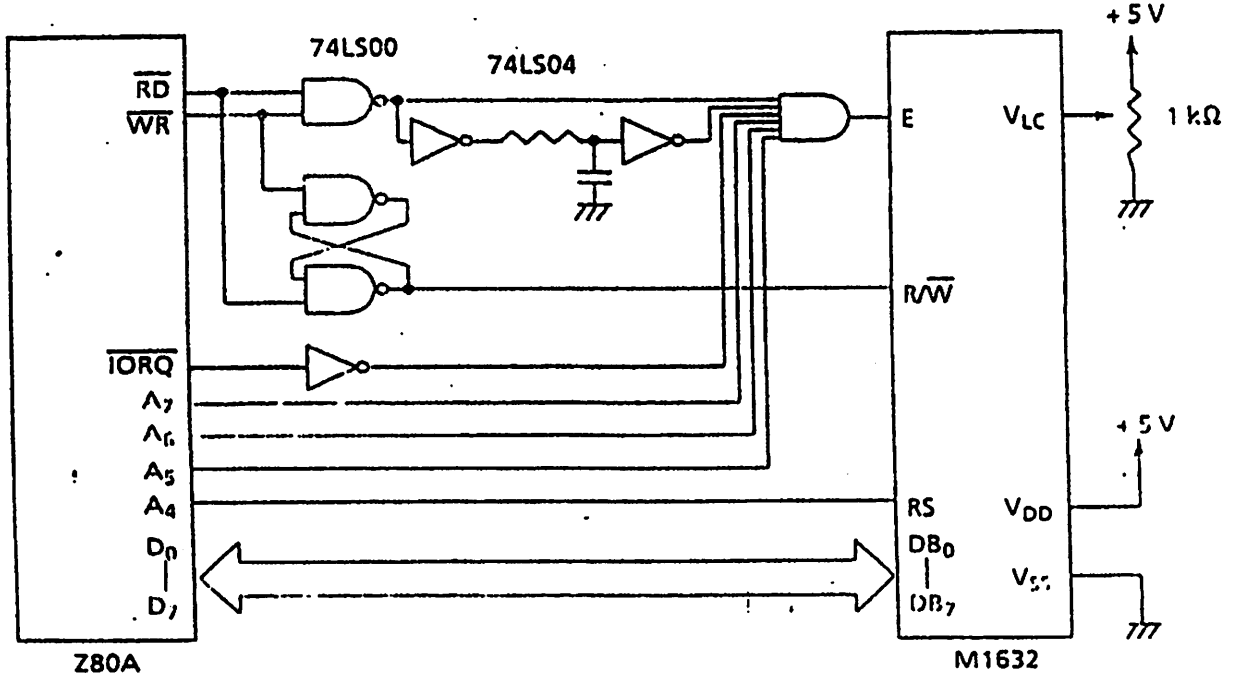
No.	Instruction	Display	Operation																				
9	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	1	1	0	0	0	0	0	0	LCD MODULE M1632 _	The DD RAM address is set so that the cursor appears at digit 1 of line 2.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	1	1	0	0	0	0	0	0														
10	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	0	1	1	0	0	0	1	LCD MODULE M1632 1_	1 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	0	1	1	0	0	0	1														
11	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	0	1	1	0	1	1	0	LCD MODULE M1632 16_	6 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	0	1	1	0	1	1	0														
12																					
13	Write to CG RAM or DD RAM <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	1	0	0	1	0	1	0	0	1	1	LCD MODULE M1632 16DIGITS, 2LINES	5 is written.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
1	0	0	1	0	1	0	0	1	1														
14	DD RAM address set <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	1	0	0	0	0	0	0	0	LCD MODULE M1632 16DIGITS, 2LINES	The cursor returns to the home position.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	1	0	0	0	0	0	0	0														
15	Display clear <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇</td> <td>DB₆</td> <td>DB₅</td> <td>DB₄</td> <td>DB₃</td> <td>DB₂</td> <td>DB₁</td> <td>DB₀</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀	0	0	0	0	0	0	0	0	0	1	_____	All the display disappears and the cursor remains at the home position.
RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀														
0	0	0	0	0	0	0	0	0	1														
16																					

(2) Interface data length: Four bits

No.	Instruction	Display	Operation									
1	<p>Power-on</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	/	/	/		The built-in reset circuit initializes the module.			
RS	R/W	DB ₇ — DB ₄										
/	/	/										
2	<p>Function Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>/</td> <td>/</td> <td>/</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	/	/	/		Four-bit operation mode is set. *Eight-bit operation mode is set by initialization, and the instruction is executed only once.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 1 0										
/	/	/										
3	<p>Function Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 1 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 * * *</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 1 0	0	0	1 * * *		The 4-bit operation mode, 1/16 duty cycle, and 5 x 7 dot-matrix character format are selected. Then 4-bit operation mode starts.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 1 0										
0	0	1 * * *										
4	<p>Display ON/OFF Control</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>1 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	1 1 1 0		The display and cursor are turned ON, but nothing is displayed.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 0 0										
0	0	1 1 1 0										
5	<p>Entry Mode Set</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 0 0 0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0 1 1 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	0	0	0 0 0 0	0	0	0 1 1 0		The address is incremented by one and the cursor shifts to the right in a write operation to internal RAM. The display is not shifted.
RS	R/W	DB ₇ — DB ₄										
0	0	0 0 0 0										
0	0	0 1 1 0										
6	<p>Write to CG RAM or DD RAM.</p> <table border="1"> <tr> <td>RS</td> <td>R/W</td> <td>DB₇ — DB₄</td> </tr> <tr> <td>1</td> <td>0</td> <td>0 1 0 0</td> </tr> <tr> <td>1</td> <td>0</td> <td>1 1 0 0</td> </tr> </table>	RS	R/W	DB ₇ — DB ₄	1	0	0 1 0 0	1	0	1 1 0 0		L is written. the AC is incremented by one and the cursor shifts to the right.
RS	R/W	DB ₇ — DB ₄										
1	0	0 1 0 0										
1	0	1 1 0 0										

2.7 MPU Connection Diagrams

2.7.1 Z80A



2.7.2 Z80A and 8255A

