ALAT BANTU NAVIGASI BAGI PENDAKI GUNUNG

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh

Gelar Sarjana Teknik Elektro

Disusun oleh :

Nama : Dhani Pratita

No.Mahasiswa : 04 524 009

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2009

LEMBAR PENGESAHAN PEMBIMBING

ALAT BANTU NAVIGASI BAGI PENDAKI GUNUNG

TUGAS AKHIR

Disusun oleh :

Nama	**:**	**Dhani Pratita**
No.Mahasiswa	**:**	**04 524 009**

Yogyakarta, juli 2009

Pembimbing I	Pembimbing II
Ir. Budi Astuti, MT.	**Wahyudi Budi Pramono, ST.M.Eng.**

LEMBAR PENGESAHAN PENGUJI

ALAT BANTU NAVIGASI BAGI PENDAKI GUNUNG

TUGAS AKHIR

Oleh :

Nama : Dhani Pratita

No. Mahasiswa : 04 524 009

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Agustus 2009

Tim Penguji,

Wahyudi Budi Pramono, ST.M.Eng.
Ketua

Ir. Hj. Budi Astuti, MT.
Anggota I

Medilla K,ST., M.Eng.
Anggota II

Mengetahui,

Ketua Jurusan Teknik Elektro

Universitas Islam Indonesia

YOGYAKARTA

Tito Yuwono, ST,.Msc.

HALAMAN PERSEMBAHAN

Segala Puji bagi Allah SWT, atas Rahmat , Ridho dan Karunia-Nya

Atas kekuatan dan cahaya terang padaku

Semua cobaan dan kesabaran yang Aku syukuri dari-Nya

Segala sesuatu dalam karya yang sederhana ini dapat terselesaikan

dengan selalu beriman dan bertaqwakepada-Nya

terima kasih,

Abahku .... dan Mama ku ....

Atas segala doa, nasehat , cinta dan kasihnya, pengorbanan, dan perhatiannya

Yang telah diberikan takkan terbalas oleh apapun dan sampai kapan pun

Atas didikan, pengalaman, kepribadian dari yang kalian ajarkan...

Aku bisa kuat dan berusaha untuk maju sampai sekarang......

Kakak ku,,,,Adik ku,,,,

Atas keceriaan dan perhatian dan doa-doanya

Luinawati....

Atas segala doa , perhatiannya serta cinta kasihnya..

Seluruh keluarga besarku

Atas segala harapan, doa dan dukungan yang diberikan dalam setiap langkahku

Sahabat-sahabat dan orang-orang terdekat

Atas segala pengalaman , ketulusan , bantuan dan kebersamaan....

MOTTO

Dan (ingatlah juga), tatkala Tuhanmu memaklumkan ; "Sesungguhnya jika kamu bersyukur, pasti kami akan menambah (nikmat) kepadamu, dan jika kamu mengingkari (nikmat-Ku), Maka Sesungguhnya azab-Ku sangat pedih"

(QS. Ibrahim : 7)

" Sungguh bersama kesukaran itu pasti ada kemudahan. Sungguh, oleh karena itu jika kamu telah selesai dari suatu tugas, kerjakan tugas lain dengan sungguh-sungguh. Dan hanya kepada Tuhanmulah kehendaknya kamu memohon dan mengharap "

(QS. Asy-Syarh 5-8)

" Jadilah sabar dan sholat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar "

(Q.S. Al Baqarah ayat 153)

"Sebuah tong penuh dengan pengetahuan belum sama nilainya dengan setetes budi"

(Phytagoras)

KATA PENGANTAR

بسم الله الرحمن الرحيم

assalamu'alaikum warahmatullahi wabarakatuh

Puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya sehingga penulis dapat menyelesaikan Tugas akhir yang berjudul **" Alat Bantu Navigasi Bagi Pendaki Gunung ".** Shalawat dan salam selalu terhaturkan kepada junjungan kita Nabi Muhammad SAW yang menjadi *uswatun hasanah* bagi kita semua hingga akhir zaman.

Penyelesaian Tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana Teknik Elektro pada Fakultas Teknologi Industri Universitas Islam Indonesia. Dengan selesainya Tugas akhir ini, maka seluruh aktivitas studi pada institusi tercinta ini berakhir dan membuka pintu untuk masuk ke tantangan berikutnya. Proses penyelesaian tugas akhir ini merupakan serangkaian integrasi antara teori serta praktek yang telah didapatkan semasa duduk dibangku kuliah. Ilmu dan amal menjadi poin penting akhir dari penyelesaian tugas akhir ini.

Tugas akhir ini merupakan implementasi dari proses perancangan serta pengujian sistem. Aplikasi utama dari sistem ini banyak dimanfaatkan dalam dunia industri, meskipun skala pengerjaan tugas akhir ini hanya sebatas simulasi. Namun, begitu pentingnya fungsi dari perangkat ini, menjadikan penulis mencoba menganalisis sekaligus memodelkan sebuah perangkat yang dapat dimanfaatkan sebagai referensi dalam ilmu pengetahuan dan teknologi.

Penulis menyadari sepenuhnya bahwa isi dari tugas akhir ini masih jauh dari sempurna, namun dengan niat yang tulus dan ikhlas, penulis menerima segala macam kritik dan saran yang membangun demi kelangsungan dan kemajuan ilmu pengetahuan dan teknologi.

Pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang tak terhingga kepada :

1. Bapak Tito Yuwono,.S.T,.M.Sc selaku Ketua Jurusan Teknik Elektro sekaligus selaku dosen pembimbing yang telah membantu dan membagi begitu banyak ilmunya kepada penulis.
2. Ibu Ir. Hj. Budi Astuti, MT. Bapak Wahyudi Budi Pramono, ST.M.Eng. selaku dosen pembimbing yang sangat memberikan banyak inspirasi serta motivasi dan ilmu.
3. Seluruh Dosen dan karyawan FTI-UII yang bersedia dengan sabar membantu dan membagi ilmunya.
4. Ayahanda Winaryanto, Ibunda Titik Fouryati, Mas Sindhu Anggara dan Adek Ryan Ananta yang telah banyak memberikan dorongan, limpahan do'a serta kasih sayangnya yang begitu tulus dan ikhlas.
5. Luina untuk segala cinta kasih yang begitu tulus.
6. Seluruh manusia " *GenKapak* ", Yank Dho, Hendra, Tino, Danis, Seto, Pati, Sinjo, Ipul, Ali semoga kita bisa sukses bersama suatu saat.

7. Teman-teman member, *the 7 alien's & the WeeRDe*. Mari kita taklukan kerasnya dunia.
8. Teman-teman kost " *Bukit Suling* " dan " *Ijo Ceria* " Terima kasih atas segala bantuan, kebahagiaan dan dukungan yang diberikan.
9. Seluruh manusia elektro 04 yang menjadikanku seorang yang bermanfaat buat kalian semua. Terima kasih atas dukungan serta apa yang telah kita ciptakan bersama.
10. Seluruh fasilitas yang membantu nemenin bikin skripsi. Mas Yanche, pelayan Audio, pelayan toko Lima Satu, pelayan Sagan Elektronik, dan penjaga warnet-warnet yang setia membantu.

Akhir kata penulis sampaikan pula harapan semoga Tugas akhir ini dapat memberi manfaat yang cukup berarti khususnya bagi penulis dan bagi pembaca pada umumnya. Semoga Allah SWT senantiasa selalu memberikan rahmat dan hidayah-Nya kepada kita semua. Amiin.

wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, July 2009

Penulis

ABSTRAK

Sistem alat bantu navigasi bagi pendaki gunung ini merupakan sistem yang dirancang untuk memudahkan arah bagi pendaki gunung oleh karena itu maka fungsi dari sistem ini adalah mampu menunjukkan 16 arah mata angin dengan menggunakan CMPS03, sedangkan untuk ketinggiannya digunakan resistor variabel sebagai simulasi dari sensor tekanan. Pada penelitian ini digunakan mikrokontroler AVR ATMegal6 yang merupakan pusat kontrol bekerjanya sistem. Pengujian sistem dilakukan dengan mengubah-ubah arah CMPS03 yang dibandingkan dengan kompas manual. Hasil dari keluaran ditampilkan dengan LCD yang mampu menampilkan arah mata angin beserta derajatnya dan ketinggian ,sedangkan *output* dari speaker memberikan informasi suara berupa arah. Proses perekaman suara menggunakan IC ISD25120 yang mampu merekam suara dalam waktu 120 detik. Sensor CMPS03 menggunakan mode 16 bit sehingga diperoleh tingkat keakurasian 0,0055 derajat/bit. Tingkat kesalahan kompas CMPS03 jika dibandingkan dengan kompas analog didapatkan hasil 0,2%.

Kata kunci : CMPS03, ISD25120, ATMegal6, arah mata angin

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dewasa ini dunia teknologi dan dunia ilmu pengetahuan berkembang dengan sangat cepat dimana perkembangan teknologi mendukung perkembangan ilmu pengetahuan demikian juga perkembangan ilmu pengetahuan mendukung perkembangan teknologi.

Dalam bidang elektronika saat ini, banyak kegunaan dari sistem elektronika baik dari sistem elektronika yang sederhana dengan menggunakan rangkaian analog sampai ke sistem elektronika digital yang menggunakan komponen IC.

Elektronika digital merupakan teknologi yang berkembang sangat cepat. Rangkaian digital selalu digunakan hampir dalam semua produk-produk yang beredar di masyarakat.

Saat ini banyak di jumpai ketika seorang pendaki gunung akan melakukan sebuah pendakian biasanya membawa peralatan navigasi dan keselamatan, pada umumnya para pendaki gunung membawa kompas.

Untuk melihat dan memperkirakan arah pembacaan mata angin sulit sekali , Kompas merupakan salah satu divais yang penting dalam navigasi untuk menentukan arah berdasarkan posisi kutub bumi. Karena itulah sebuah piranti yang memiliki *output* berupa suara dan tampilan LCD yang dapat digunakan dalam kondisi apapun, baik itu malam hari ataupun saat berkabut, sehingga dapat mengurangi resiko tersesatnya pendaki gunung, untuk mengatasi masalah dalam pembacaan yang sulit maka di butuhkan sebuah alat yang mudah dalam pembacaan data, untuk itu di buatlah sebuah alat digital sehingga mudah dalam pembacaan.

Dengan melihat dari berbagai macam fakta keadaan di atas, dengan memanfaatkan mikrokontroler ATMega16, maka peneliti akan membuat alat yang digunakan untuk pembuatan kompas digital dengan menggunakan *ouput* suara dan LCD, di dalam laporan Tugas Akhir penyusun mengambil judul " ALAT BANTU NAVIGASI BAGI PENDAKI GUNUNG ". Sehingga dengan adanya alat ini, diharapkan proses navigasi dapat dilakukan lebih mudah, akurat dan hasil yang presisi sesuai dengan harapan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat diambil suatu rumusan masalah yang akan menjadi pokok pembahasan adalah bagaimana membuat suatu sistem digital yang dapat membantu menunjukkan arah mata angin dan ketinggian bagi pendaki gunung berbasis mikrokontroler ATMega16.

1.3 Batasan Masalah

Dengan adanya batasan masalah, penulis dapat lebih menyederhanakan dan mengarahkan penelitian dan pembuatan sistem agar tidak menyimpang dari apa yang diteliti. Batasan-batasannya adalah sebagai berikut :

1. Penggunaan mikrokontroler ATMega16 sebagai kendali utamanya
2. Penggunaan LCD M1632 dan speaker sebagai *output* dari sistem.
3. Penggunaan modul CMPS03 sebagai sensor arah mata angin.
4. Penggunaan resirtor variabel sebagai simulasi dari sensor tekanan.

1.4 Tujuan Penulisan

Tujuan yang akan dicapai dalam penulisan Tugas Akhir ini adalah sebagai berikut :

a. Sebagai salah satu syarat dalam menyelesaikan studi pada Program Sarjana Teknik Elektro Fakultas Teknik Universitas Islam Indonesia.
b. Merancang piranti digital untuk dapat membantu menunjukkan arah mata angin berbasis mikrokontroler AVR ATMega16.

1.5 Langkah Penelitian

Penyusunan tugas akhir ini mempunyai metoda penelitian yang dapat dijelaskan sebagai berikut :

1. Mendeskripsikan komponen-komponen yang terkandung dalam obyek penelitian.
2. Memahami karakteristik komponen-komponen dan bagian-bagian yang menyusun obyek penelitian.
3. Mengklasifikasikan secara menyeluruh hasil pemahaman sehingga mendapat gambaran deskriptif tentang unsur-unsur yang terkandung dalam obyek penelitian.

4. Menyajikan hasil penelitian yang berupa data sehingga diperoleh gambaran secara jelas tentang model yang dibuat.

1.6 Sistematika Penulisan Laporan.

Dari hasil penelitian yang telah dilaksanakan, sistematika penulisan laporannya adalah sebagai berikut:

BAB I PENDAHULUAN

Bab Pendahuluan berisi tentang Latar Belakang Masalah, Maksud dan Tujuan, Perumusan Masalah, Batasan Masalah, Langkah Penelitian, dan Sistematika Penulisan Laporan.

BAB II STUDI PUSTAKA

Pada bab ini berisi tentang *Literature survey* tentang penelitian sejenis yang telah dilakukan sebelumnya. Analisis, kesimpulan, saran, komentar penelitian sejenis yang telah dilakukan sebelumnya. Penjelasan mengenai kontribusi penelitian yang akan dikerjakan dalam TA

BAB III PERANCANGAN SISTEM

Bagian ini menjelaskan metode-metode perancangan yang digunakan, cara mensimulasikan rancangan dan pengujian sistem yang telah dibuat, pembagian fungsi kerja dalam diagram blok serta berisi lebih terperinci tentang apa yang telah disampaikan pada proposal tugas akhir ini. Penjabaran indikator unjuk kerja sistem : bagaimana validasi atau pengujian sistem akan dilakukan.

BAB IV ANALISA DAN PEMBAHASAN

Bab ini membahas tentang hasil pengujian dan analisis dari sistem yang dibuat dibandingkan dengan dasar teori sistem atau sistem yang lain yang dapat dijadikan sebagai pembanding. Pengujian sistem berdasarkan indikator unjuk kerja yang telah dijelaskan sebelumnya

BAB V PENUTUP

Bagian ini menjelaskan kesimpulan dari TA yang telah selesai dikerjakan berdasarkan analisis dan pembahasan di bab sebelumnya. Saran untuk pengembangan dan penelitian lebih lanjut

BAB II

STUDI PUSTAKA

2.1 Tinjauan Pustaka

Sugiarto indar, 2004 telah melakukan penelitian tentang kompas magnetik dengan *output* LCD. Diagram alir sistem " Kompas Magnetik Digital dengan *Output* LCD " adalah sebagai berikut :

Gambar 2.1 Diagram alir Kompas Magnetik Digital dengan *Output* LCD

Dalam perancangan sistem ini, sistem terdiri atas beberapa bagian penting yaitu sensor navigasi CMPS03 , mikrokontroler ATMega16 sebagai pengolah data, dan sebagai keluaran dari sistem ini berupa LCD.

Proses bekerjanya sistem diawali dengan pembacaan arah mata angin menggunakan CMPS03, data dari sensor CMPS03 tersebut diolah melalui mikrokontroler ATMega16 yang berfungsi sebagai pusat kontrol sistem, dan *output* dari perancangan ini berupa LCD , LCD mampu menampilkan hasil keluaran berupa 4 arah mata angin.

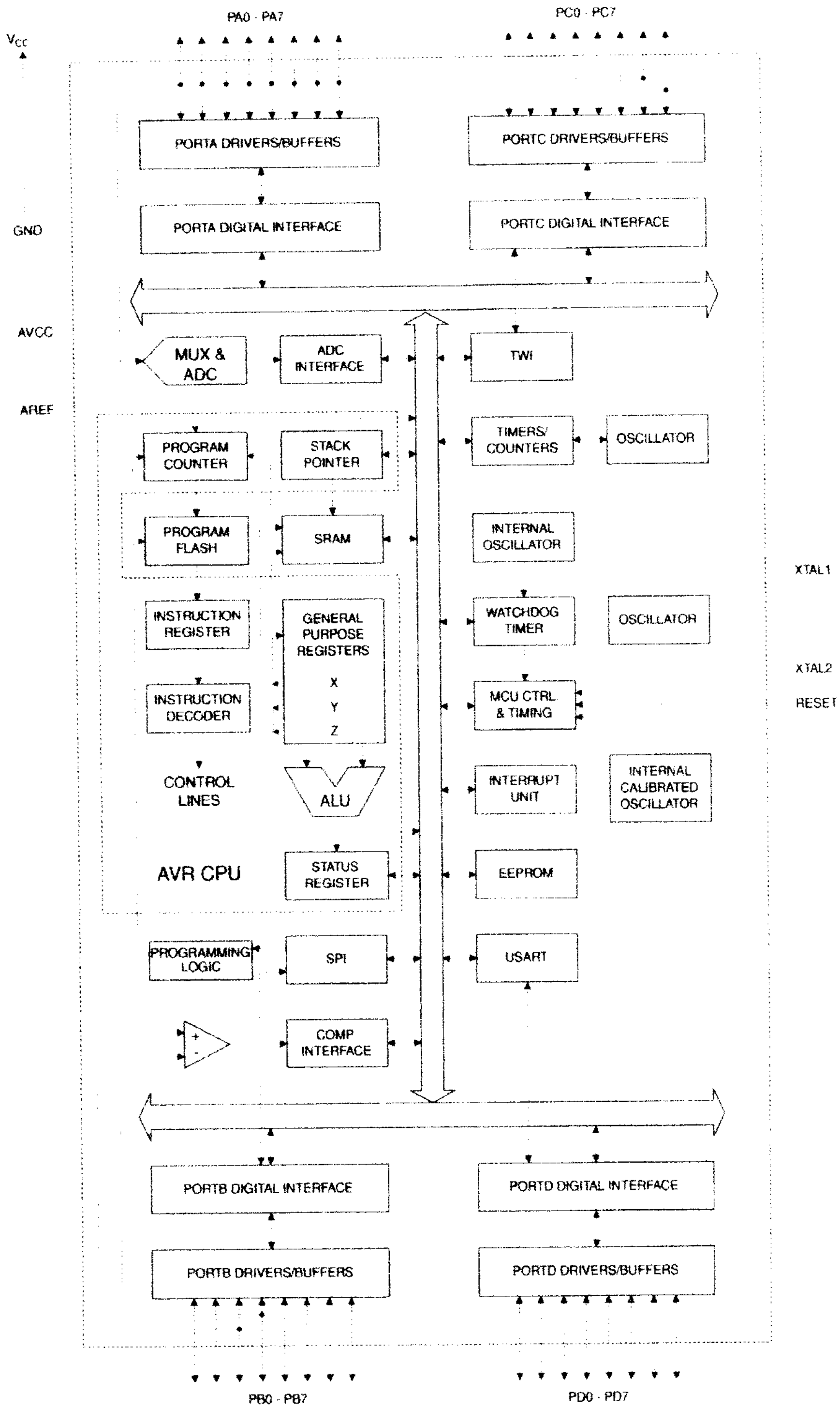
Dalam perancangan sistem yang akan dibuat, sistem terdiri atas beberapa bagian penting yaitu sensor navigasi CMPS03, sensor tekanan, IC suara 25120, dan mikrokontroler ATMega16 sebagai pengolah data, dan sebagai keluaran dari sistem ini terdiri dari LCD dan Speaker.

Proses bekerjanya sistem diawali dengan pendektesian sensor tekanan yang dalam penelitian ini menggunakan resirtor variabel dan pembacaan arah mata angin menggunakan CMPS03, data dari kedua sensor tersebut diolah melalui mikrokontroler ATMega16 yang berfungsi sebagai pusat kontrol sistem. *Output* dari perancangan ini berupa LCD dan IC suara ISD 25120 mengolah keluaran dari mikrokontroler untuk kemudian dihasilkan *output* berupa suara. LCD mampu menampilkan hasil keluaran berupa 16 arah mata angin disertai dengan derajat dan ketinggian.

Penelitian lain yang berkaitan dengan alat bantu navigasi bagi pendaki gunung telah dilakukan Amin husni, 2004 yaitu tentang pemanfaatan MPX 4100 sebagai detektor tekanan dan ketinggian suatu tempat diatas permukaan air laut. Dalam perancangan itu menggunakan sensor tekanan udara, sensor akan mendeteksi tekanan udara disekitar sehingga akan diperoleh tegangan output sensor. Tegangan *output* sensor akan dikirim ke pengkondisi isyarat dan ke ADC. *Output* dari mikro tersebut dikirim ke LCD untuk ditampilkan dalam bentuk angka desimal yaitu nilai ketinggian.

2.2 Mikrokontroler AVR ATMega16

Mikrokontroler AVR memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-*bits word*) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus *clock*.

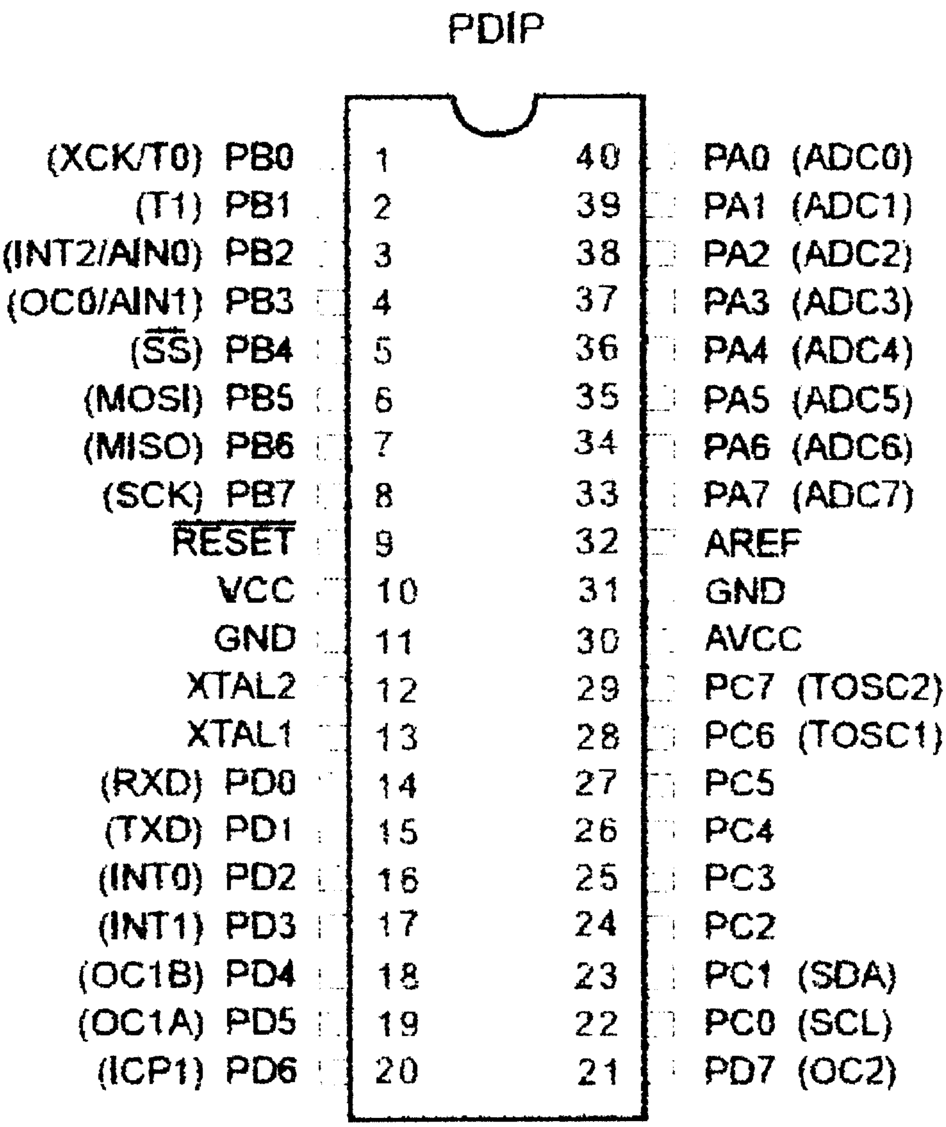


Gambar 2.2 Blok Diagram Fungsional ATMega16

Fitur dasar yang dimiliki Mikrokontroler AVR ATMega16 adalah sebagai berikut :

1. Port I/O 32 jalur (Port A, Port B, Port C, Port D masing-masing 8 bit).
2. ADC 10 bit 8 *channel.*
3. 3 buah *timer/counter.*
4. 32 register dalam CPU.
5. *Watchdog Timer* dengan osilator internal.
6. *Flash* PEROM 16 kb.
7. EEPROM 512 bytes.
8. SRAM 1 Kb.
9. Interupsi Eksternal dan Internal.
10. *Interfacing* dengan komparator analog.
11. Port USART untuk komunikasi serial.

Mikrokontroler AVR ATMega16 terdiri dari 40 Pin, yang konfigurasinya sebagai berkut :



Gambar 2.3 Pin ATMega16

Tabel 2.1 Konfigurasi pin ATMega16 (Lanjutan)

No Pin	Nama	Fungsi
14	PD0 (RXD)	Port D.0 / penerima data serial
15	PD1 (TXD)	Port D.1 / pengirim data serial
16	PD2 (INT0)	Port D.2 / Interupsi eksternal 0
17	PD3 (INT1)	Port D.3 / Interupsi eksternal 1
18	PD4 (OC1B)	Port D.4 / Pembanding *Timer-Counter* 1
19	PD5 (OC1A)	Port D.5 / Pembanding *Timer-Counter*
20	PD6 (ICP1)	Port D.6 / *Timer-Counter* 1 *Input*
21	PD7 (OC2)	Port D.7 / Pembanding *Timer-Counter* 2
22	PC0 (SCL)	Port C.0 / Serial bus clock line
23	PC1 (SDA)	Port C.0 / Serial bus data *input-output*
24 - 27	PC2 – PC5	Port C.0
28	PC6 (TOSC1)	Port C.0 / Timer osilator 1
29	PC7 (TOSC2)	Port C.0 / Timer osilator 2
30	AVCC	Tegangan ADC
31	GND	Sinyal ground ADC
32	AREFF	Tegangan referensi ADC
33 - 40	PA0 (ADC0) – PA7 (ADC7)	Port A.0 – Port A.7 dan input untuk ADC (8 channel : ADC0 – ADC7)

2.3 LCD M1632

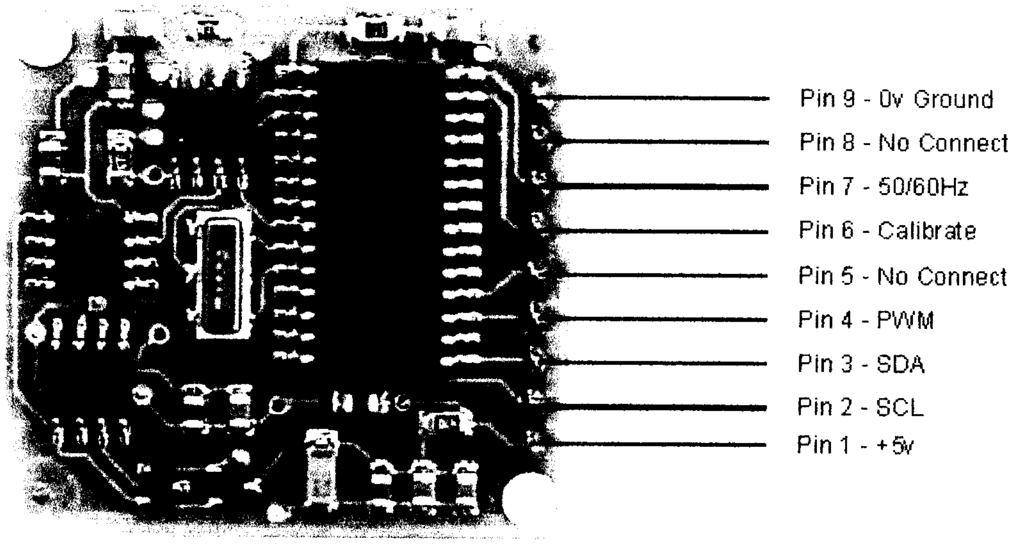
LCD *Display Module M1632* buatan Seiko Instrument Inc. Terdiri dari dua bagian, yang pertama merupakan panel LCD sebagai media penampil informasi dalam bentuk huruf/angka dua baris, masing–masing baris bisa menampung 16 huruf/angka.

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

2.3.1 Tampilan M1632

M1632 mempunyai seperangkat perintah untuk mengatur tata kerjanya, perangkat perintah tersebut meliputi perintah untuk menghapus tampilan, meletakkan kembali cursor pada baris/huruf pertama baris pertama, menghidupkan/mematikan tampilan dan lain sebagainya.

Setelah diberi catu daya, ada beberapa langkah persiapan yang harus dikerjakan dulu agar M1632 bisa dipakai, langkah- langkah tersebut antara lain adalah:



Gambar 2.5 Modul Devantech Magnetic Compass (CMPS03)

Spesifikasi untuk modul CMPS03 – *Devantech Magnetic Compass*, yaitu :

1. Catu daya : +5 VDC,
2. Konsumsi arus : 15 mA,
3. Antarmuka : I2C atau PWM,
4. Akurasi : 3-4 derajat,
5. Resolusi : 0,1 derajat,
6. Waktu konversi : 40ms atau 33,3ms dapat dipilih,
7. Telah dikalibrasi pada daerah dengan sudut inklinasi 67 derajat.

CMPS03 *Magnetic Compass* buatan Devantech Ltd adalah salah satu sensor kompas digital yang berukuran 4 x 4 cm. CMPS03 menggunakan sensor medan magnet Philips KMZ51 yang cukup sensitif untuk mendeteksi medan magnet bumi.

Kompas digital ini hanya memerlukan suplai tegangan sebesar 5V DC, dengan konsumsi arus 15mA. Pada CMPS03, arah mata angin dibagi dalam bentuk derajat yaitu : Utara (0°), Timur (90°), Selatan (180°) dan Barat (270°). Ada dua cara untuk mendapatkan informasi arah dari modul kompas digital ini yaitu dengan membaca sinyal PWM *(Pulse Width Modulation)* pada pin 4 atau dengan membaca data interface I2C pada pin 2 dan 3.

Tabel 2.2 Alokasi *Internal Register* CMPS03

Register	Function
0	Software Revision Number
1	Compass Bearing as a byte, i.e. 0-255 for a full circle
2,3	Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees.
4,5	Internal Test - Sensor1 difference signal - 16 bit signed word
6,7	Internal Test - Sensor2 difference signal - 16 bit signed word
8,9	Internal Test - Calibration value 1 - 16 bit signed word
10,11	Internal Test - Calibration value 2 - 16 bit signed word
12	Unused - Read as Zero
13	Unused - Read as Zero
14	Unused - Read as Undefined
15	Calibrate Command - Write 255 to perform calibration step. See text.

2.5 Rencana Penelitian

Pada rencana penelitian ini dimulai dengan membuat rancangan *hardware* terlebih dahulu, kemudian memilih komponen-komponen yang akan digunakan dalam penelitian ini. Selanjutnya mengumpulkan data yang akan diolah pada sistem ini kemudian membuat program pada sistem. Langkah selanjutnya membandingkan antara hasil pengukuran digital dengan pengukuran analog.

Pada penelitian ini yang akan diukur adalah keluaran dari sensor CMPS03 dari 16 arah mata angin, untuk mendeteksi ketinggian akan disimulasikan dari keluaran potensiometer. Analisa yang dilakukan dari sistem berupa membandingkan data hasil keluaran digital dengan hasil keluaran analog serta *software* yang akan digunakan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan sistem dan hasil analisa yang didapat maka dalam pembuatan *hardware* alat bantu *navigasi* bagi pendaki gunung dapat disimpulkan beberapa hal, yaitu :

1. *Output* suara dari ISD25120 hanya berupa 16 arah mata angin belum dilengkapi dengan derajat dan ketinggian karena keterbatasan dari memori ISD25120.
2. Sensor CMPS03 menggunakan mode 16 bit sehingga diperoleh tingkat keakurasian 0,0055 derajat/bit.
3. *Error* setiap perubahan arah mata angin berkisar antara 0,2° sampai 0,3°.

5.2 Saran

Untuk mengembangkan sistem dimasa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

1. Perlu ditambahkan Sensor Tekanan sehingga mampu mendeteksi ketinggian sebenarnya.
2. Dalam menggunakan CMPS03 sebaiknya dijauhkan dari magnet karena dapat mempengaruhi akurasi dari pengukuran.
3. Perlu digunakan memori ISD yang lebih besar agar dapat memberikan informasi yang lebih banyak.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan sistem dan hasil analisa yang didapat maka dalam pembuatan *hardware* alat bantu *navigasi* bagi pendaki gunung dapat disimpulkan beberapa hal, yaitu :

1. *Output* suara dari ISD25120 hanya berupa 16 arah mata angin belum dilengkapi dengan derajat dan ketinggian karena keterbatasan dari memori ISD25120.
2. Sensor CMPS03 menggunakan mode 16 bit sehingga diperoleh tingkat keakurasian 0,0055 derajat/bit.
3. *Error* setiap perubahan arah mata angin berkisar antara 0,2° sampai 0,3°.

5.2 Saran

Untuk mengembangkan sistem dimasa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

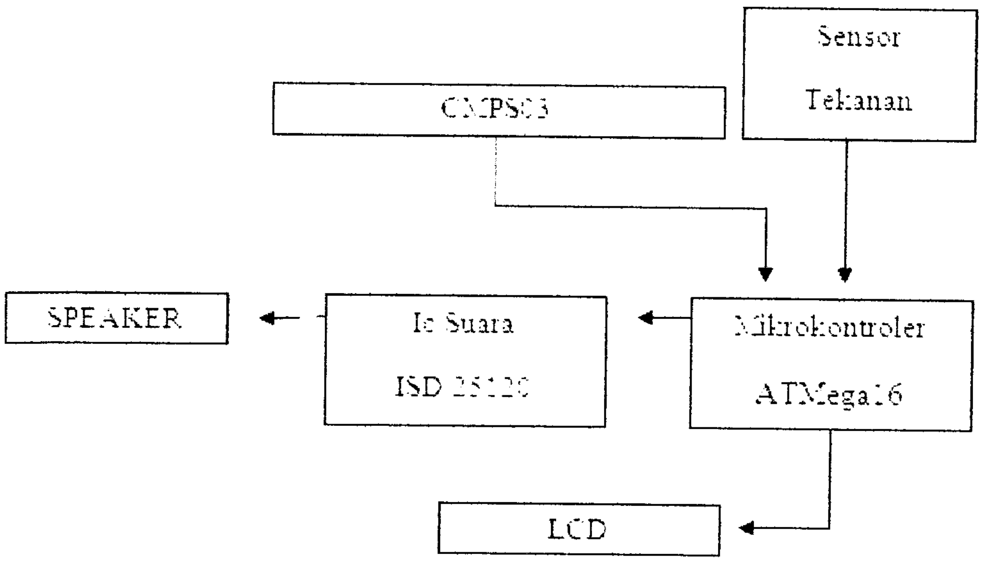
1. Perlu ditambahkan Sensor Tekanan sehingga mampu mendeteksi ketinggian sebenarnya.
2. Dalam menggunakan CMPS03 sebaiknya dijauhkan dari magnet karena dapat mempengaruhi akurasi dari pengukuran.
3. Perlu digunakan memori ISD yang lebih besar agar dapat memberikan informasi yang lebih banyak.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Sistem

Dalam perancangan sistem ini, sistem terdiri atas beberapa bagian penting yaitu sensor navigasi CMPS03, sensor tekanan, IC suara 25120, dan mikrokontroler ATMega16 sebagai pengolah data, dan sebagai keluaran dari sistem ini terdiri dari LCD dan Speaker.



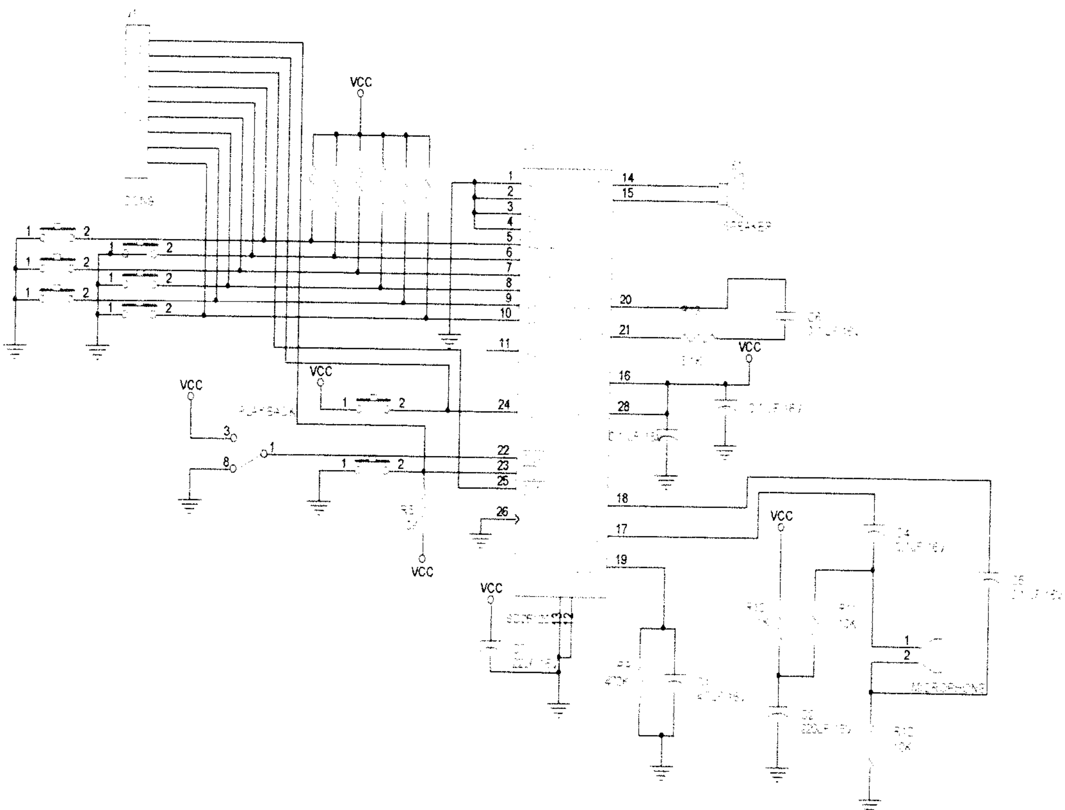
Gambar 3.1 Bagan Alir Sistem Alat Bantu Navigasi Bagi Pendaki Gunung

Proses bekerjanya sistem diawali dengan pendektesian sensor tekanan yang dalam penelitian ini disimulasikan dengan menggunakan resirtor variabel dan pembacaan arah mata angin menggunakan CMPS03, data dari kedua sensor tersebut diolah melalui mikrokontroler ATMega16 yang berfungsi sebagai pusat kontrol sistem. *Output* dari perancangan ini berupa LCD dan IC suara ISD 25120 mengolah keluaran dari mikrokontroler untuk kemudian dihasilkan *output* berupa

3.2.3 Rangkaian IC Suara ISD25120

Gambar 3.3 merupakan rangkaian ISD25120 yang digunakan untuk merekam suara yang akan digunakan pada proses penunjukan arah mata angin. Sistem tersebut mampu merekam 16 arah mata angin.

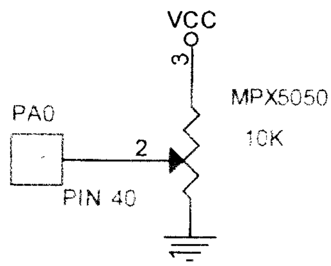
Proses perekaman dimulai dari alamat 0000 sampai 1111. Proses awal perekaman dilakukan dengan menentukan alamat terlebih dahulu kemudian menekan tombol P/R hingga dalam posisi *low*, kemudian menahan tombol CE dalam keadan low diikuti dengan merekam suara, jika terjadi kesalahan dalam perekaman maka tombol PD ditekan untuk menghapus rekaman serta untuk mengembalikan data dari awal. Untuk proses perekaman selanjutnya dilakukan perubahan pada alamat memori.



Gambar 3.3 Rangkaian IC Suara ISD25120

3.2.4 Rangkaian Sensor Tekanan

Pada sistem ini , resistor variabel digunakan sebagai simulasi dari sensor tekanan. Perubahan ketinggian dapat dimisalkan dengan merubah – ubah nilai tahanan dari resistor variabel.



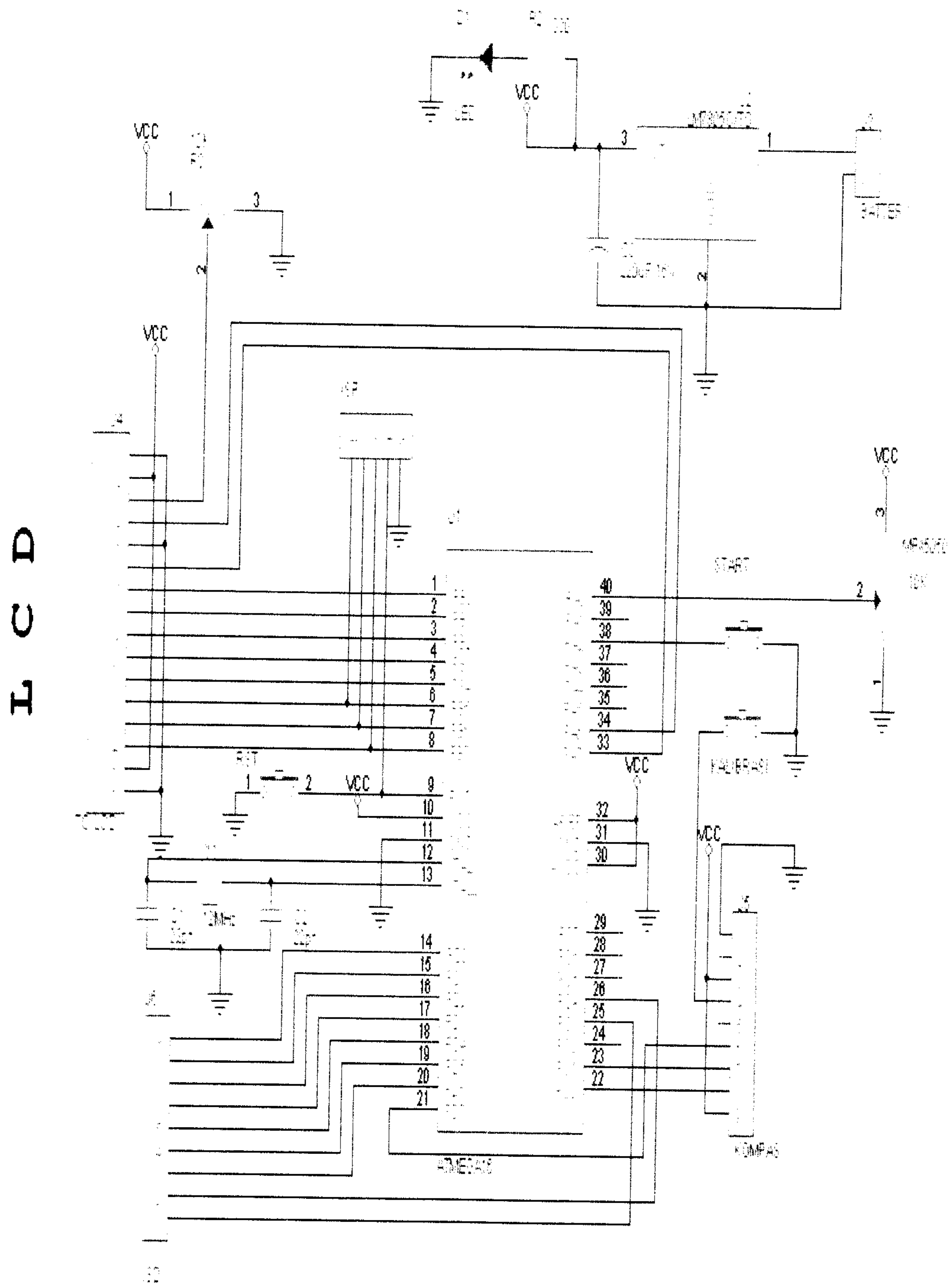
Gambar 3.4 Rangkaian Sensor tekanan

3.2.5 Rangkaian Sistem Minimum ATMega 16

Mikrokontroler ATMega16 memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit (16-bits word) dan sebagian besar instruksi dieksekusi dalam 1 (satu) siklus clock.

Rangkaian osilator pada sistem ini digunakan oleh *mikrokontroler* sebagai sinyal denyut (*clock*). Frekuensi sinyal denyut inilah yang menentukan kecepatan eksekusi yang akan dijalankan. Frekuensi denyut maksimum yang diperbolehkan adalah 33 MHz. Tetapi pada perancangan sistem ini menggunakan XTAL 12 MHz, dan 2 buah kapasitor 22 pF.

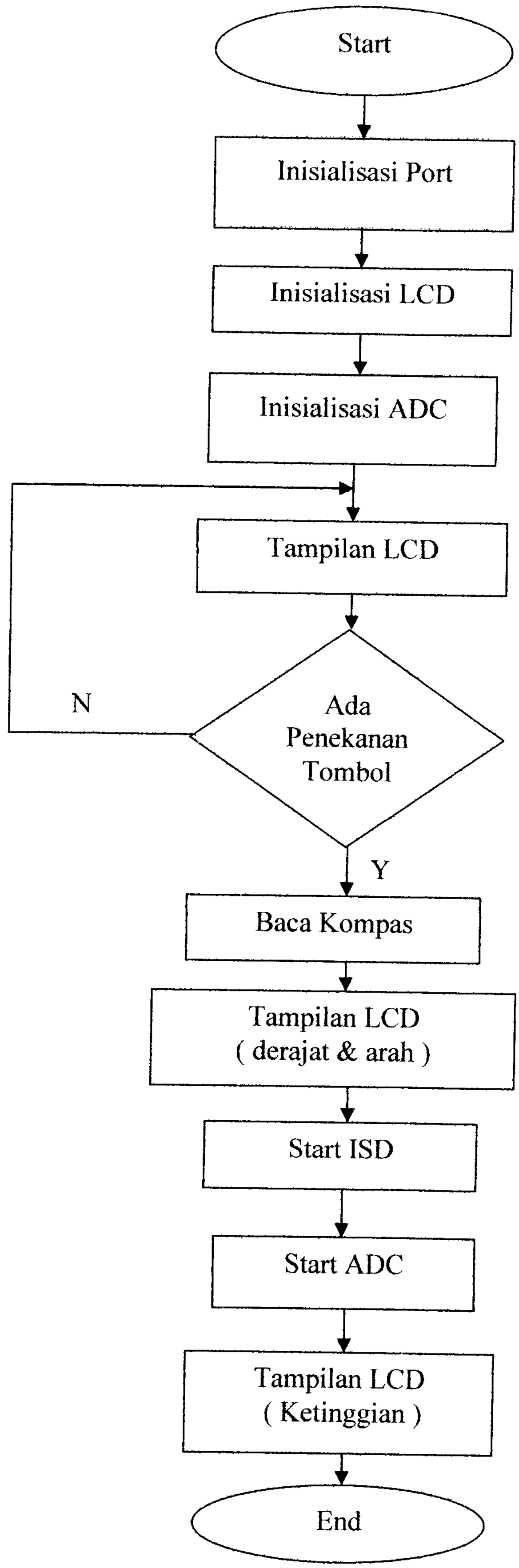
Sedangkan untuk membangkitkan sinyal reset, maka tombol reset langsung dihubungkan dengan ground .



Gambar 3.5 Rangkaian Sistem Minimum ATMega 16

3.3 Perancangan Software

Bagan Alir Program



3.4 Proses Pemrograman pada Mikrokontroler ATMega16

3.4.1 Proses Inisialisasi Program

```
#include<avr/io.h>
#include<avr/interrupt.h>
#include<compat/deprecated.h>
#define LCD_RS 6
#define LCD_CS 7
#define RESET 1
#define START 0
#define EOM 2
#define SCL 0
#define SDA 1
#define nop() __asm__("NOP");
```

3.4.2 Proses Deklarasi Program

```
unsigned char tulisan[]=" KOMPAS DIGITAL ";
unsigned char tulisan1[]="       UII 2009";
unsigned char tulisan2[]="Arah:          ";
unsigned char tulisan3[]="Ketinggian:   m";
```

3.4.3 Proses Fungsi Delay Program

```
void delay(unsigned int nilai_delay)
{
unsigned int i;
unsigned int j;
for(i=0;i<nilai_delay;i++)
        {
        for(j=0;j<=100;j++)
                {
                ;
                }
        }
}
```

3.4.4 Proses Instruksi pada LCD :

```
void WR_INST(int data_)
{
cbi(PORTA,LCD_RS);
PORTB=data_;
sbi(PORTA,LCD_CS);
delay(10);
cbi(PORTA,LCD_CS);
delay(10);
```

3.4.5 Proses akses pembacaan nilai I2C :

```
unsigned char i2c_read(unsigned char address, unsigned char reg)
{
unsigned char read_data = 0;
  TWCR = ((1<<TWINT)|(1<<TWSTA)|(1<<TWEN));
  while(!(TWCR & 0x80));
  TWDR = address;
  TWCR = ((1<<TWINT)|(1<<TWEN));
  while(!(TWCR & 0x80));
  TWDR = reg;
  TWCR = ((1<<TWINT)|(1<<TWEN));
  while(!(TWCR & 0x80));
  TWCR = ((1<<TWINT)|(1<<TWSTA)|(1<<TWEN));
  while(!(TWCR & 0x80));
  TWDR = address+1;
  TWCR = ((1<<TWINT)|(1<<TWEA)|(1<<TWEN));
  while(!(TWCR & 0x80));
  TWCR = ((1<<TWINT)|(1<<TWEN));
  while(!(TWCR & 0x80));
  read_data = TWDR;
  TWCR = ((1<<TWINT)|(1<<TWSTO)|(1<<TWEN));
  return read_data;
}
```

3.4.6 Proses Program Utama :

```
int main(void)
{
unsigned char k;
DDRA=0xc0;
DDRB=0xff;
DDRC=0xff;
DDRD=0xfb;          //1111 1011
PORTA=0xff;
PORTC=0xf3;         //1111 0011
PORTD=0x6;
init_LCD();
ADMUX=0;
ADCSRA=0xca;
SFIOR=0;
//while(1);
for(k=0;k<=15;k++)
{
        WR_DATA(tulisan[k]);
}
WR_INST(0xc0);
for(k=0;k<=15;k++)
{
        WR_DATA(tulisan1[k]);
```

```
}
delay(5000);
WR_INST(1);
for(k=0;k<=15;k++)
{
        WR_DATA(tulisan3[k]);
}
WR_INST(0xc0);
for(k=0;k<=15;k++)
{
        WR_DATA(tulisan2[k]);
}
setup();
sei();
//i2c_transmit(0xc0,2,119);
//delay(100);
while(1)
{
sei();
ADCSRA=0xca;
delay(100);
WR_INST(0x8b);
display_LCD(nilai_ADC*10);
cli();
//delay(7500);
if(bit_is_clear(PINA,3))
{
        delay(250);
        k=i2c_read(0xc0,2);
        hasil16=k*256;
        delay(100);
        k=i2c_read(0xc0,3);
        hasil16=hasil16+k;
        delay(100);
        hasil16=hasil16/450;
        k=PORTD;
        k=(k&7);
        k=(k|(ISD[hasil16]));
//      k=(ISD[hasil16]);
//0x0=utara; 0x8=timur laut; 0x10=timur; 0x18=tenggara;

        0x20=selatan; 0x28=barat daya; 0x30=barat; 0x38=barat

        laut

        PORTD=k;
        ISD_start();
        WR_INST(0xc6);
```

```
        for(k=0;k<=9;k++)
        {
                WR_DATA(utara[k+(hasil16*10)]);
        }
        while(bit_is_set(PIND,2));
        }
}
return 0;
}
```

BAB IV

ANALISA DAN PEMBAHASAN

Pada proses alat bantu navigasi bagi pendaki gunung, terdapat beberapa pengujian yang telah dilakukan yaitu pengujian terhadap sensor kompas magnetik, IC suara ISD25120 , pengujian terhadap simulasi dari sensor tekanan yang menggunakan Potensiometer , dan proses pemrograman pada mikrokontroler ATMega16.

4.1 Pengujian Sensor Kompas Magnetik

Pengujian pada sensor kompas magnetik menggunakan CMPS03 yang dibandingkan dengan kompas analog diperoleh data sebagai berikut :

Tabel 4.1 Pengujian Sensor Kompas Magnetik

ARAH MATA ANGIN	KOMPAS DIGITAL CMPS03 (dalam derajat)	KOMPAS ANALOG (dalam derajat)	KESALAHAN (dalam derajat)
UTARA	0,2	0	0,2
UTARA TIMUR LAUT	22,7	22,5	0,2
TIMUR LAUT	45,2	45	0,2
TIMUR TIMUR LAUT	67,7	67,5	0,2
TMUR	90,2	90	0,2
TIMUR TENGGARA	112,8	112,5	0,3
TENGGARA	135,2	135	0,2

Tabel 4.1 Pengujian Sensor Kompas Magnetik (Lanjutan)

ARAH MATA ANGIN	KOMPAS DIGITAL CMPS03 (dalam derajat)	KOMPAS ANALOG (dalam derajat)	KESALAHAN (dalam derajat)
SELATAN TENGGARA	157,8	157,5	0,3
SELATAN	180,2	180	0,2
SELATAN BARAT DAYA	202,7	202,5	0,2
BARAT DAYA	225,3	225	0,3
BARAT BARAT DAYA	247,7	247,5	0,2
BARAT	270,2	270	0,2
BARAT BARAT LAUT	292,8	292,5	0,3
BARAT LAUT	315,3	315	0,3
UTARA BARAT LAUT	337,7	337,5	0,2

Error hasil pengukuran pada CMPS03 dibandingkan dengan kompas Analog adalah sebagai berikut :

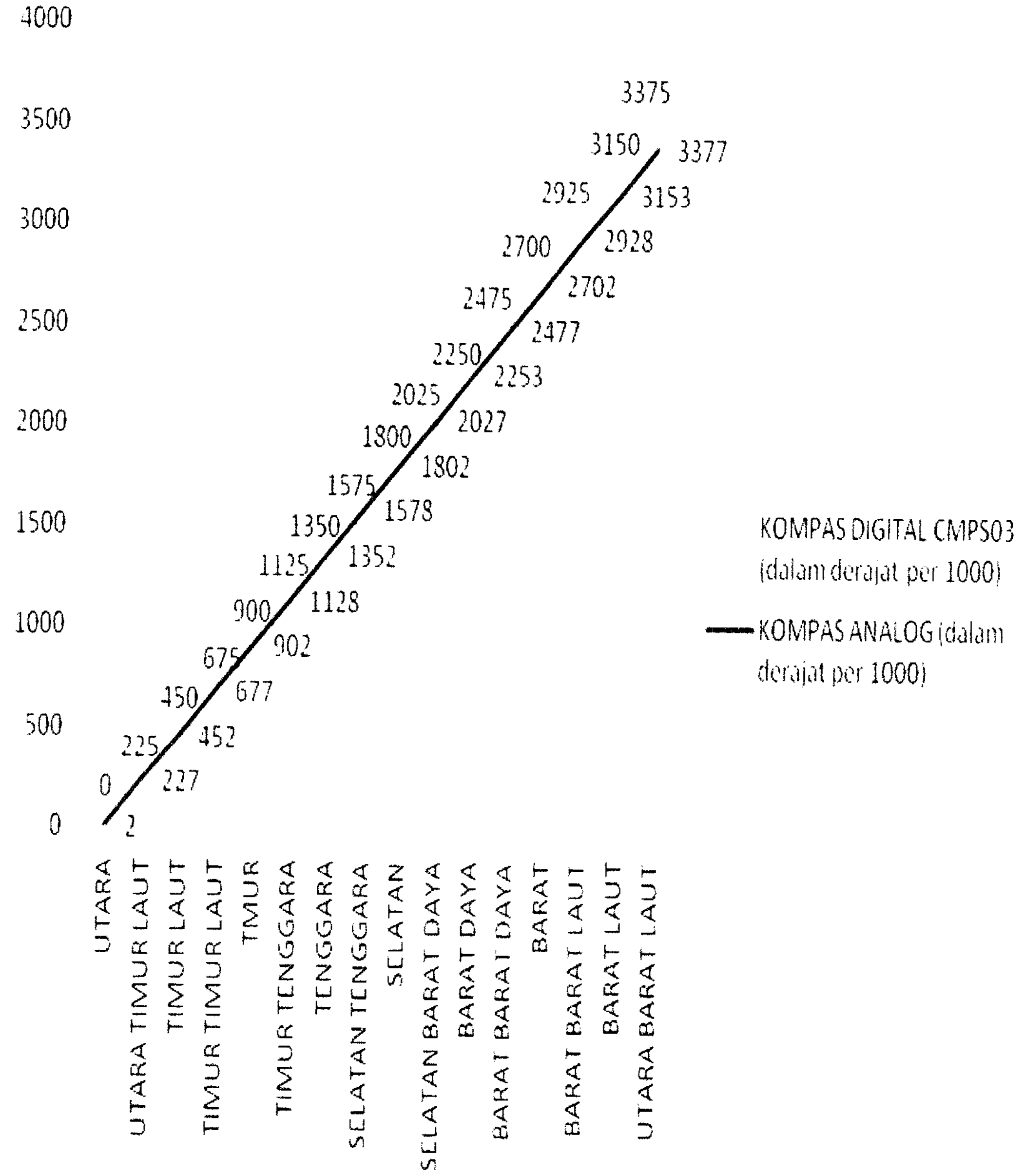
$$n = \frac{\sum kompas\ digital\ CMPS03 - \sum kompas\ ana\log}{\sum kompas\ ana\log} x100\% \qquad (4.1)$$

$$n = \frac{2705{,}5° - 2700°}{2700°} x100\%$$

$$n = \frac{5{,}5°}{2700°} x100\%$$

$$n = 0{,}2\%$$

Perbandingan dari hasil pengukuran antara kompas analog dengan kompas digital CMPS03 terjadi perbedaan. Perbedaan hasil pengukuran ditampilkan pada tabel 4.1 dan gambar 4.1

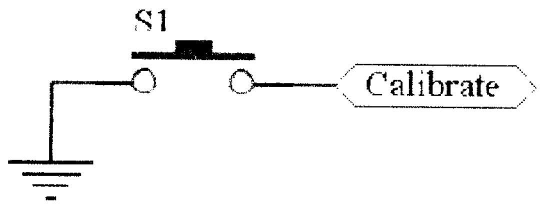


Gambar 4.1 Grafik perbandingan Kompas Analog dengan Kompas Digital CMPS03

Pengukuran dilakukan dengan membandingkan 16 arah mata angin pada kompas analog dengan sensor CPMS03. Cara mengkalibrasi CMPS03 ada dua cara, yaitu dengan metode I2C atau pin (manual). Dalam aplikasi ini dipilih kalibrasi dengan metode pin (manual) karena dinilai lebih mudah dan efisien.

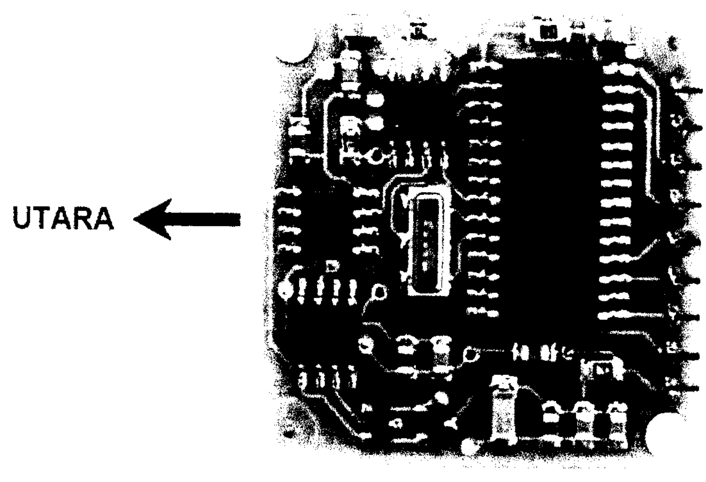
Berikut ini adalah langkah-langkahnya:

1. Gunakan rangkaian *tactile switch* seperti pada Gambar 4.2



Gambar 4.2 Rangkaian *Tactile Switch* untuk Proses *Calibrate*

2. Posisikan orientasi utara dari CMPS03 ke arah utara bumi (gambar 4.3) yang sebenarnya lalu tekan *tactile switch*.



Gambar 4.3 Orientasi CMPS03 yang Menghasilkan Pembacaan Sudut 0°

3. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah timur bumi, lalu tekan *tactile switch.*
4. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah selatan bumi, lalu tekan *tactile switch.*
5. Putar secara perlahan-lahan sampai orientasi utara dari CMPS03 menuju ke arah barat bumi, lalu tekan *tactile switch.*
6. Periksalah apakah kompas telah menampilkan arah yang benar sesuai dengan arah sebenarnya. Jika belum sesuai ulangi lagi mulai langkah 1.
7. Jika penunjukan sudah sesuai dengan arah sebenarnya, maka CMPS03 dinyatakan telah terkalibrasi dengan baik.

Kalibrasi ini hanya dilakukan sekali saja, karena hasil dari pengkalibrasian disimpan dalam EEPROM yang terdapat pada CMPS03. Untuk penggunaan selanjutnya (pada lokasi dengan sudut inklinasi sama), tidak perlu dilakukan kalibrasi ulang.

4.2 Pengujian IC Suara ISD25120

Proses perekaman dimulai dari alamat 0000 sampai 1111. Proses awal perekaman dilakukan dengan menentukan alamat terlebih dahulu kemudian menekan tombol P/R hingga dalam posisi low, kemudian menahan tombol CE dalam keadaan low diikuti dengan merekam suara, jika terjadi kesalahan dalam perekaman maka tombol PD ditekan untuk menghapus rekaman. Untuk proses

Batasan hasil keluaran suara pada IC Suara ISD25120 :

Tabel 4.3 Keluaran Suara Pada IC ISD25120

ARAH MATA ANGIN	DERAJAT
UTARA	0 – 22,4
UTARA TIMUR LAUT	22,5 – 44,9
TIMUR LAUT	45 – 67,4
TIMUR TIMUR LAUT	67,5 – 89,9
TIMUR	90 – 112,4
TIMUR TENGGARA	112,5 – 134,9
TENGARA	135 – 157,4
SELATAN TENGGARA	157,5 – 179,9
SELATAN	180 – 202,4
SELATAN BARAT DAYA	202,5 – 224,9
BARAT DAYA	225 – 247,4
BARAT BARAT DAYA	247,5 – 269,9
BARAT	270 – 292,4
BARAT BARAT LAUT	292,5 – 314,9
BARAT LAUT	315 – 337,4
UTARA BARAT LAUT	337,5 – 359,9

4.3 Pengujian terhadap Simulasi dari Sensor Tekanan yang menggunakan Potensiometer

Dalam simulasi ini digunakan Potensiometer sebagai pengganti dari sensor ketinggian. Potensiometer digunakan hanya untuk indikasi bahwa alat mampu bekerja dan mikrokontroler mampu merespon perubahan tegangan pada Potensiometer, namun Potensiometer tidak dapat merespon perubahan ketinggian lokasi. Berikut ini adalah tabel yang menampilkan hasil dari simulasi sensor tekanan (Potensiometer) pada 3 tempat yang memiliki perbedaan ketinggian :

Table 4.4 Hasil pengukuran pada simulasi ketinggian menggunakan Potensiometer

Lokasi	**Ketinggian (meter)**	**Keluaran (volt)**
Pantai Parangtritis	0	0,08
Lempuyangan	135	0,65
Pos Kaliurang	740	3,54

Dengan menggunakan table 4.4 maka perubahan tegangan untuk tiap meter ketinggiannya adalah sebagai berikut :

- Pada Pantai Parangtritis – Lempuyangan :

  0 meter = 0,08 volt

  135 meter = 0,65 volt

Maka akan di dapat :

135 meter – 0 meter = 0,65 volt – 0,08 volt

135 meter = 0,57 volt

1 meter = 0,004 volt

= 4 miliVolt

- Pada Pantai Parangtritis – Pos Kaliurang :

0 meter = 0,08 volt

740 meter = 3,54 volt

Maka akan di dapat :

740 meter – 0 meter = 3,54 volt – 0,08 volt

740 meter = 3,46volt

1 meter = 0,004 volt

= 4 miliVolt

- Pada Lempuyangan – Pos Kaliurang :

135 meter = 0,65 volt

740 meter = 3,54 volt

Maka akan di dapat :

740 meter – 135 meter = 3,54 volt – 0,65 volt

605 meter = 2,84 volt

1 meter = 0,004 volt

= 4 miliVolt

Jadi perubahan setiap 1 meter ketinggian sebesar 4 miliVolt.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan perancangan sistem dan hasil analisa yang didapat maka dalam pembuatan *hardware* alat bantu *navigasi* bagi pendaki gunung dapat disimpulkan beberapa hal, yaitu :

1. *Output* suara dari ISD25120 hanya berupa 16 arah mata angin belum dilengkapi dengan derajat dan ketinggian karena keterbatasan dari memori ISD25120.
2. Sensor CMPS03 menggunakan mode 16 bit sehingga diperoleh tingkat keakurasian 0,0055 derajat/bit.
3. *Error* setiap perubahan arah mata angin berkisar antara 0,2° sampai 0,3°.

5.2 Saran

Untuk mengembangkan sistem dimasa yang akan datang maka dapat disarankan beberapa hal sebagai berikut :

1. Perlu ditambahkan Sensor Tekanan sehingga mampu mendeteksi ketinggian sebenarnya.
2. Dalam menggunakan CMPS03 sebaiknya dijauhkan dari magnet karena dapat mempengaruhi akurasi dari pengukuran.
3. Perlu digunakan memori ISD yang lebih besar agar dapat memberikan informasi yang lebih banyak.

DAFTAR PUSTAKA

Atmel Corporation. *ATMega16 Datasheet.* 25 Agustus 2008.

http://labdasar.ee.itb.ac.id/lab/EL3006/0708/sem2/ATMega16.pdf

Hadid T.B. 2004. *CMPS03- Devantech Magnetic Compass.* 28 Juli 2008.

http://pdf1.alldatasheet.com/datasheet-df/view/ETC/CMPS03.html

Nalwan Andi Paulus. 2004. *Panduan Praktis Penggunaan* dan *Antarmuka Modul LCD M1632.* Jakarta: Elek Media Komputindo.

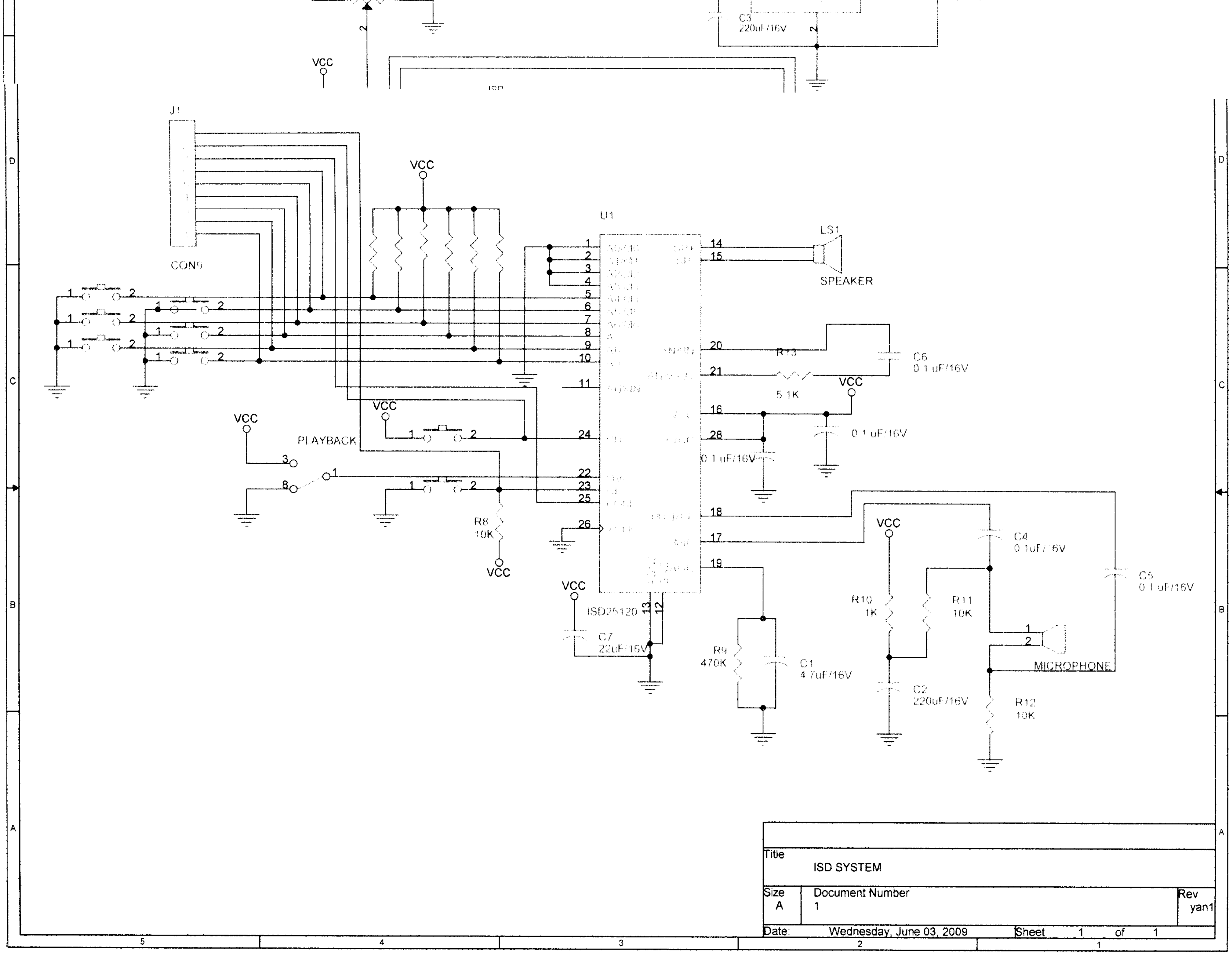
Sugiarto indar. 2004. *Kompas Magnetik dengan output LCD*, 28 Juli 2008.

http://fportfolio.petra.ac.id/user_files/02-002/paper_eccis.pdf

Voice Record/Playback Device ISD2560/75/90/120 25 Agustus 2008.

http://www.digchip.com/datasheets/parts/datasheet/217/ISD2590.php

LAMPIRAN



J1
CON9
VCC
U1
LS1
SPEAKER
R13
5.1K
C6
0.1uF/16V
0.1uF/16V
0.1uF/16V
PLAYBACK
R8
10K
ISD25120
C7
22uF/16V
R9
470K
C1
4.7uF/16V
R10
1K
R11
10K
C2
220uF/16V
R12
10K
C4
0.1uF/16V
C5
0.1uF/16V
MICROPHONE
C3
220uF/16V
Title
ISD SYSTEM
Size
A
Document Number
1
Rev
yan1
Date: Wednesday, June 03, 2009
Sheet 1 of 1

```
ram ini digunakan untuk mengakses CMPS03 dan isd25xx. status: ok

lude<avr/io.h>
lude<avr/interrupt.h>
lude<compat/deprecated.h>
ine LCD_RS 6
ine LCD_CS 7
ine RESET 1
ine START 0
ine EOM 2
ine SCL 0
ine SDA 1
ine nop() __asm__("NOP");

gned char tulisan[]=" KOMPAS DIGITAL ";
gned char tulisan1[]="        UII 2009";
gned char tulisan2[]="Arah:           ";
gned char tulisan3[]="Ketinggian:    m";

gned char utara[]="UTARA           UTARA TIMUR LAUTTIMUR LAUT      TIMUR TIMUR L
IMUR            TIMUR TENGGARA  TENGGARA        SELATAN TENGGARASELATAN         S
 BARAT DAYABARAT DAYA       BARAT BARAT DAYABARAT              BARAT BARAT LAUTBARAT
T       UTARA BARAT LAUT";
gned char ISD[]= {0x0,0x8,0x10,0x18,0x20,0x28,0x30,0x38,0x40,0x48,0x50,0x58,0x60
8,0x70,0x78};
gned int hasil16,nilai_ADC;

i delay(unsigned int nilai_delay)

gned int i;
gned int j;
(i=0;i<nilai_delay;i++)
 {
 for(j=0;j<=100;j++)
      {
      ;
      }
 }



d WR_INST(int data_)

(PORTA,LCD_RS);
TB=data_;
(PORTA,LCD_CS);
ay(10);
(PORTA,LCD_CS);
ay(10);


d WR_DATA(int data_in)

(PORTA,LCD_RS);
TB=data_in;
(PORTA,LCD_CS);
ay(10);
(PORTA,LCD_CS);
ay(10);


d init_LCD()
```

```
NST(0x6);
NST(0xc);
NST(0x3F);
NST(0x1);
NST(0x80);


 display_LCD(unsigned int d)

gned int sisa;
sisa=d%1000;
d=d/1000;
WR_DATA(d+48);
d=sisa;
sisa=d%100;
d=d/100;
WR_DATA(d+48);
d=sisa;
sisa=d%10;
d=d/10;
WR_DATA(d+48);
WR_DATA(sisa+48);


gned char i2c_read(unsigned char address, unsigned char reg)

igned char read_data = 0;

TWCR = ((1<<TWINT)|(1<<TWSTA)|(1<<TWEN));

while(!(TWCR & 0x80));
TWDR = address;

TWCR = ((1<<TWINT)|(1<<TWEN));
while(!(TWCR & 0x80));
TWDR = reg;
TWCR = ((1<<TWINT)|(1<<TWEN));
while(!(TWCR & 0x80));

TWCR = ((1<<TWINT)|(1<<TWSTA)|(1<<TWEN));

while(!(TWCR & 0x80));
TWDR = address+1;
TWCR = ((1<<TWINT)|(1<<TWEA)|(1<<TWEN));

while(!(TWCR & 0x80));
TWCR = ((1<<TWINT)|(1<<TWEN));

while(!(TWCR & 0x80));
read_data = TWDR;
TWCR = ((1<<TWINT)|(1<<TWSTO)|(1<<TWEN));

return read_data;


d i2c_transmit(unsigned char address, unsigned char reg, unsigned char data)

TWCR = ((1<<TWINT)|(1<<TWSTA)|(1<<TWEN));
while(!(TWCR & 0x80));
```

```
WDR = address;

WCR = ((1<<TWINT)|(1<<TWEN));
hile(!(TWCR & 0x80));
WDR = reg;

WCR = ((1<<TWINT)|(1<<TWEN));
hile(!(TWCR & 0x80));
WDR = data;
WCR = ((1<<TWINT)|(1<<TWEN));
hile(!(TWCR & 0x80));
WCR = ((1<<TWINT)|(1<<TWSTO)|(1<<TWEN));



 setup(void)

WBR = 128;
 TWSR=((1<<TWPS1)|(1<<TWPS0));


 ISD_start(void)

PORTD,RESET);
y(30);
PORTD,RESET);
y(30);

PORTD,START);
y(30);
PORTD,START);
y(30);



(ADC_vect)


ai_ADC=ADC;
SRA=(ADCSRA | 0x10);          //ADCSRA=0xe9-> lakukan ini jika memakai metode free
nning



 main(void)

igned char k;

A=0xc0;
B=0xff;
C=0xff;
D=0xfb;        //1111 1011
TA=0xff;
TC=0xf3;       //1111 0011
TD=0x6;
t_LCD();

UX=0;
SRA=0xca;
OR=0;

hile(1);

(k=0;k<=15;k++)

 WR_DATA(tulisan[k]);
```

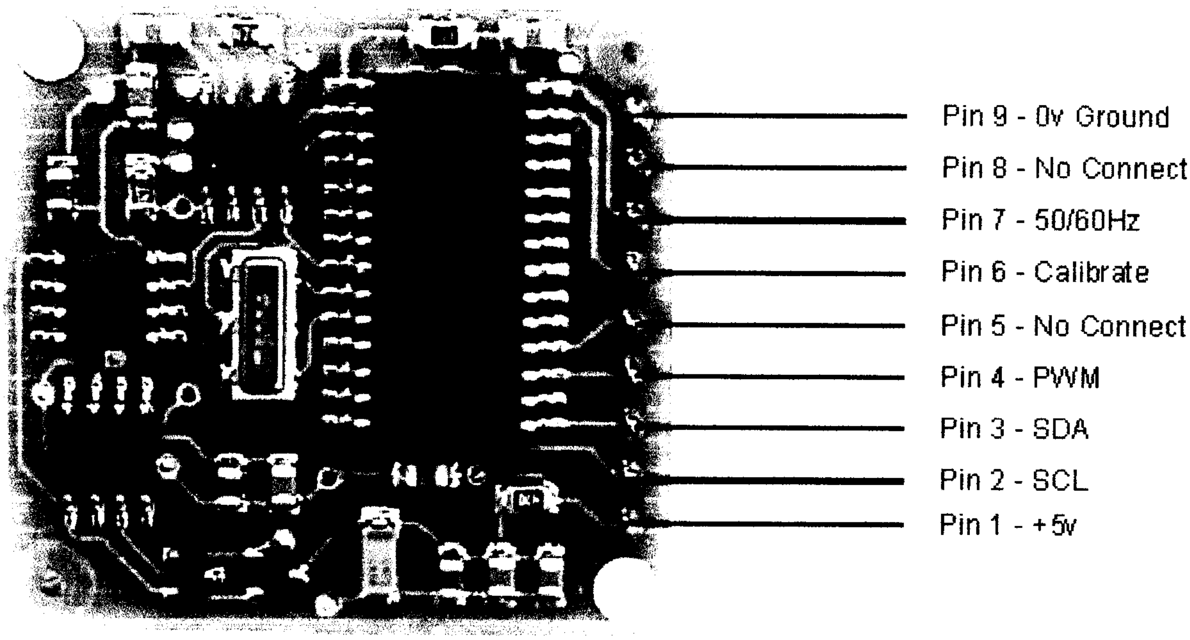
```
NST(0xc0);
k=0;k<=15;k++)

WR_DATA(tulisan1[k]);

y(5000);
p();
);

e(1)


e(bit_is_set(PINA,3));

WR_INST(0x1);
WR_INST(0x80);
delay(250);
k=i2c_read(0xc0,2);
hasil16=k*256;
delay(100);

k=i2c_read(0xc0,3);
hasil16=hasil16+k;
delay(100);

WR_INST(0xc0);
display_LCD(hasil16);

hasil16=hasil16/225;

k=PORTD;
k=(k&7);
k=(k|(ISD[hasil16]));

k=(ISD[hasil16]);               //0x0=utara; 0x8=timur laut; 0x10=timur; 0x18=te
ara; 0x20=selatan; 0x28=barat daya; 0x30=barat; 0x38=barat laut
PORTD=k;
ISD_start();

WR_INST(0x80);
for(k=0;k<=15;k++)
{
    WR_DATA(utara[k+(hasil16*16)]);
}


le(bit_is_set(PIND,2));

delay(5000);
WR_INST(1);
WR_INST(0x80);

(k=0;k<=15;k++)

WR_DATA(tulisan3[k]);



sei();
ADCSRA=0xca;
delay(100);
WR_INST(0x8b);
display_LCD(nilai_ADC*10);
cli();
```

```
rn 0;
```

CMPS03 - Robot Compass Module

This compass module has been specifically designed for use in robots as an aid to navigation. The aim was to produce a unique number to represent the direction the robot is facing. The compass uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earths magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the Earths magnetic field. We have examples of using the Compass module with a wide range of popular controllers.

Connections to the compass module

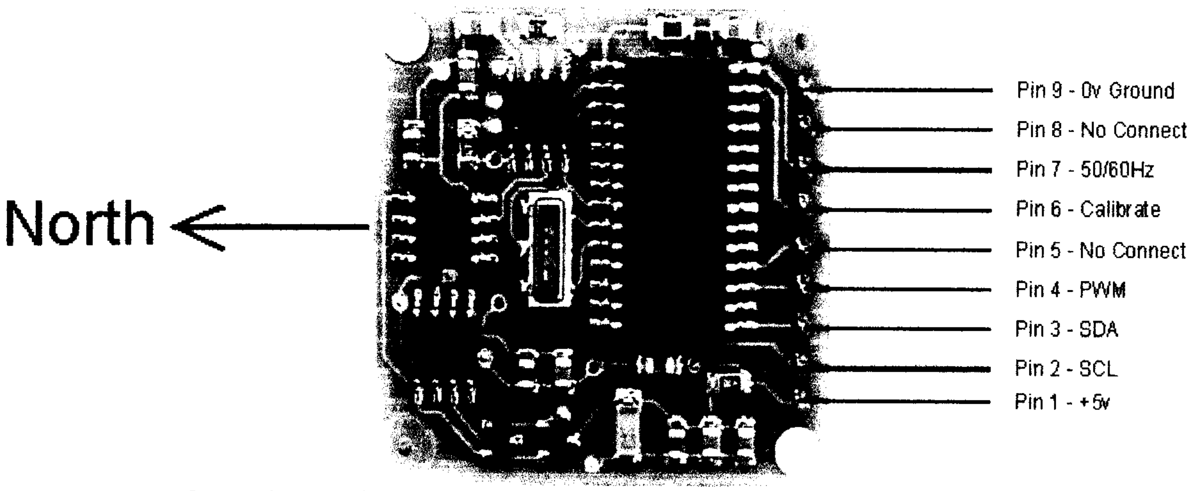


The compass module requires a 5v power supply at a nominal 15mA.
There are two ways of getting the bearing from the module. A PWM signal is available on pin 4, or an I2C interface is provided on pins 2,3.
The PWM signal is a pulse width modulated signal with the positive width of the pulse representing the angle. The pulse width varies from 1mS (0°) to 36.99mS (359.9°) - in other words 100uS/° with a +1mS offset. The signal goes low for 65mS between pulses, so the cycle time is 65mS + the pulse width - ie. 66ms-102ms. The pulse is generated by a 16 bit timer in the processor giving a 1uS resolution, however I would not recommend measuring this to anything better than 0.1° (10uS). Make sure you connect the I2C pins, SCL and SDA, to the 5v supply if you are using the PWM, as there are no pull-up resistors on these pins.
Pin 2,3 are an I2C interface and can be used to get a direct readout of the bearing. If the I2C interface is not used then these pins should be pulled high (to +5v) via a couple of resistors. Around 47k is ok, the values are not at all critical.

Calibrating the CMPS01, CMPS03 Robot Compass Modules

CMPS03 - Calibration procedure is the same as CMPS01 Rev 7.
CMPS01 - As from Software revision 7, the calibration procedure has changed.
Both methods are detailed below

Calibration only needs to be done once - the calibration data is stored in EEPROM on the PIC16F872 chip. You do not need to re-calibrate every time the module is powered up. The module has already been calibrated in our workshop for our inclination, which is 67 degrees. If your location is close to this, you may like to try the compass without re-calibrating at all.



Compass module orientation to produce 0 degrees reading.

Register	**Function**
0	Software Revision Number
1	Compass Bearing as a byte, i.e. 0-255 for a full circle
2,3	Compass Bearing as a word, i.e. 0-3599 for a full circle, representing 0-359.9 degrees.
4,5	Internal Test - Sensor1 difference signal - 16 bit signed word
6,7	Internal Test - Sensor2 difference signal - 16 bit signed word
8,9	Internal Test - Calibration value 1 - 16 bit signed word
10,11	Internal Test - Calibration value 2 - 16 bit signed word
12	Unused - Read as Zero
13	Unused - Read as Zero
14	Calibration Done Flag - Zero in calibrate mode when un-calibrated, 255 otherwise - unused in Rev 7 software & CMPS03
15	Calibrate Command - Write 255 to enter calibrate mode, write zero to exit. See text.

Register 0 is the Software revision number (originally 3, now 7 with new calibration routines and 8 for the CMPS03).

Registers 14 & 15 are used to calibrate the compass. The procedure changed with Rev 7 software. Full calibration information is here

IMPORTANT - The compass module must be kept flat (horizontal and parallel to the earths surface) with the components on top and for the CMPS01 the sensors underneath. Keep the module away from metallic - especially magnetic - objects.

Calibrating Rev 3 Software - Recognized by lack of revision number on CPU chip, or read revision number from register 0

I2C Method

To calibrate the compass using the I2C bus, you only have to write 255 to register 15 and rotate the module very slowly through 360° . Writing zero to register 15 will store the calibration values in the processors internal EEPROM. Readings are taken by the processor at four compass points and these values are used to generate the calibration values. Register 14 reads 255 during normal operation. It reads zero when Calibrate mode is entered and 255 again when the four compass points have been measured. Register 14 will therefore indicate that the four points have been acquired and that zero can be written to register 15 to store the calibration and return to normal operation. It is necessary to rotate the compass very slowly during calibration to avoid missing the required compass points and to keep it horizontal to ensure the calibration figures are accurate.

Pin Method

Pins 5,6 are used to calibrate the compass. The calibrate input (pin 6) has an on-board pull-up resistor and can be left unconnected after calibration. To calibrate the compass you only have to take the calibrate pin low and rotate the module very slowly through 360° . Taking the calibrate pin high will store the calibration values in the processors internal EEPROM. Readings are taken by the processor at four compass points and these values are used to generate the calibration values. The CalDone output pin (pin 5) is high during normal operation. It goes low when the Calibrate pin is pulled low and high again when the four compass points have been measured. The CalDone pin will therefore indicate that the four points have been acquired and that the Calibrate pin can be raised high again. It is necessary to rotate the compass slowly during calibration to avoid missing the required compass points and to keep it horizontal to ensure the calibration figures are accurate.

Calibrating Rev 7 Software - Recognized by revision number label on CMPS01 CPU chip, or read revision number from register 0.
Also applies to **Calibrating the CMPS03 Module**.

Note that pin 5 (CalDone) and register 14 (Calibration Done Flag) are not used with Rev 7 software or the CMPS03. Pin 5 should be left unconnected and register 14 ignored. When calibrating the compass, you must know **exactly** which direction is North, East, South and West. Don't guess at it. Get a magnetic needle compass and check it.

I2C Method

To calibrate using the I2C bus, you only have to write 255 (0xff) to register 15 for each of the four major compass points North, East, South and West. The 255 is cleared internally automatically after each point is calibrated. The compass points can be set in any order, but all four points must be calibrated. For example

1. Set the compass module flat, pointing North. Write 255 to register 15
2. Set the compass module flat, pointing East. Write 255 to register 15
3. Set the compass module flat, pointing South. Write 255 to register 15
4. Set the compass module flat, pointing West. Write 255 to register 15

That's it.

Pin Method

Pin 6 is used to calibrate the compass. The calibrate input (pin 6) has an on-board pull-up resistor and can be left unconnected after calibration. To calibrate the compass you only have to take the calibrate pin low and then high again for each of the four major compass points North, East, South and West. A simple push switch wired from pin6 to 0v (Ground) is OK for this. The compass points can be set in any order, but all four points must be calibrated. For example

1. Set the compass module flat, pointing North. Press and release the switch
2. Set the compass module flat, pointing East. Press and release the switch
3. Set the compass module flat, pointing South. Press and release the switch
4. Set the compass module flat, pointing West. Press and release the switch

That's it.

ISD2560/75/90/120

SINGLE-CHIP, MULTIPLE-MESSAGES,

VOICE RECORD/PLAYBACK DEVICE

60-, 75-, 90-, AND 120-SECOND DURATION

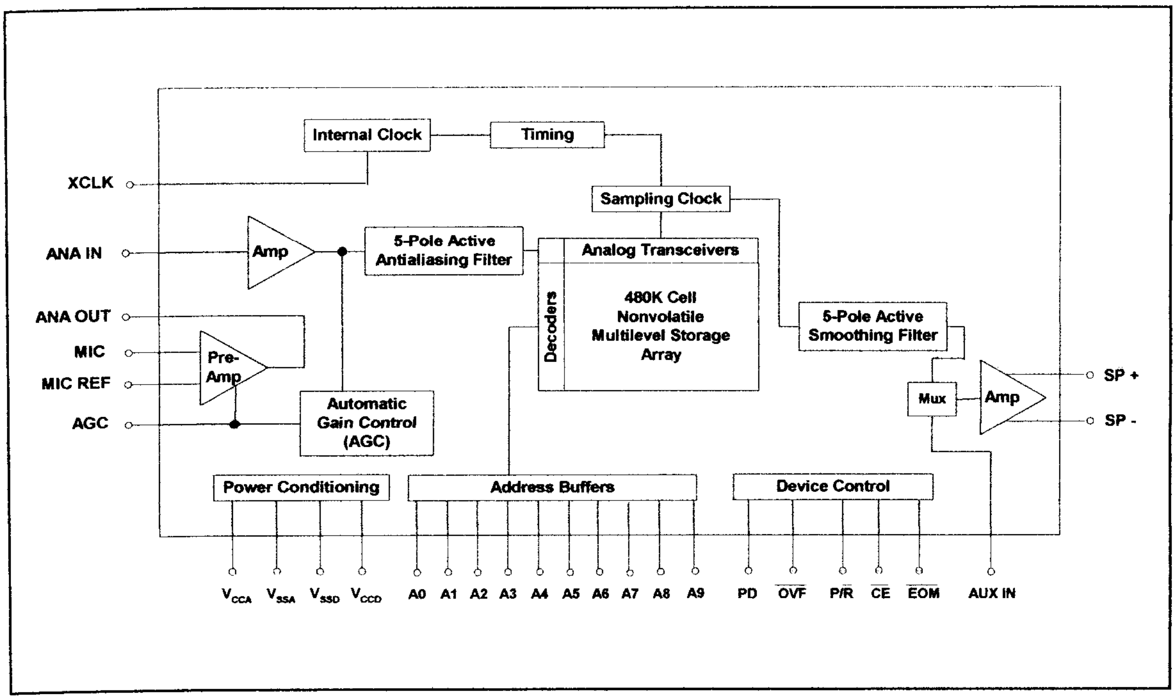
1. GENERAL DESCRIPTION

Winbond's ISD2500 ChipCorder® Series provide high-quality, single-chip, Record/Playback solutions for 60- to 120-second messaging applications. The CMOS devices include an on-chip oscillator, microphone preamplifier, automatic gain control, antialiasing filter, smoothing filter, speaker amplifier, and high density multi-level storage array. In addition, the ISD2500 is microcontroller compatible, allowing complex messaging and addressing to be achieved. Recordings are stored into on-chip nonvolatile memory cells, providing zero-power message storage. This unique, single-chip solution is made possible through Winbond's patented multilevel storage technology. Voice and audio signals are stored directly into memory in their natural form, providing high-quality, solid-state voice reproduction.

2. FEATURES

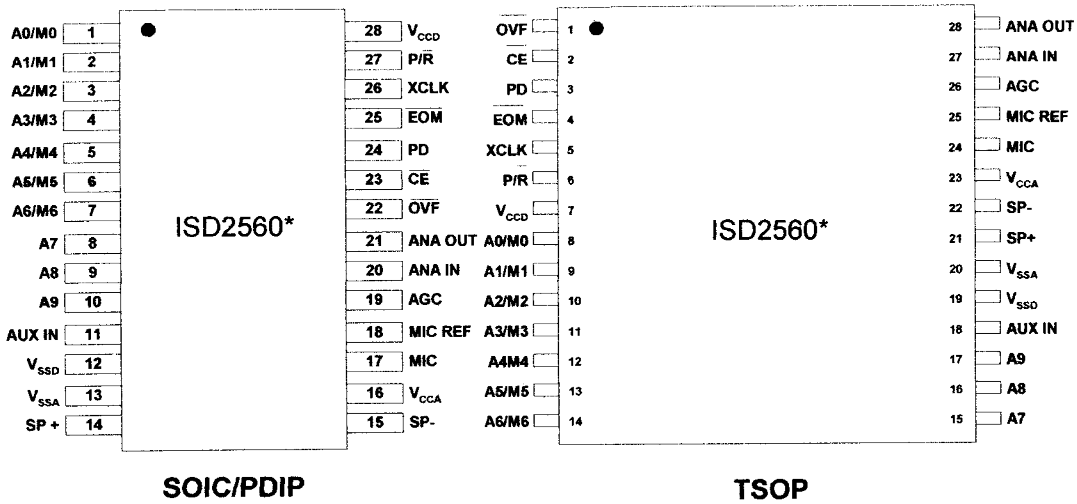
- Easy-to-use single-chip, voice record/playback solution
- High-quality, natural voice/audio reproduction
- Single-chip with duration of 60, 75, 90, or 120 seconds.
- Manual switch or microcontroller compatible
- Playback can be edge- or level-activated
- Directly cascadable for longer durations
- Automatic power-down (push-button mode)
  - Standby current 1 µA (typical)
- Zero-power message storage
  - Eliminates battery backup circuits
- Fully addressable to handle multiple messages
- 100-year message retention (typical)
- 100,000 record cycles (typical)
- On-chip clock source
- Programmer support for play-only applications
- Single +5 volt power supply
- Available in die form, PDIP, SOIC and TSOP packaging
- Temperature = die (0°C to +50°C) and package (0°C to +70°C)

3. BLOCK DIAGRAM



Winbond Electronics Corp.

5. PIN CONFIGURATION



* Same pinouts for ISD2575 / 2590 / 25120 products

6. PIN DESCRIPTION

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
Ax/Mx	1-10/ 1-7	8-17/ 8-14	**Address/Mode Inputs**: The Address/Mode Inputs have two functions depending on the level of the two Most Significant Bits (MSB) of the address pins (A8 and A9). If either or both of the two MSBs are LOW, the inputs are all interpreted as address bits and are used as the start address for the current record or playback cycle. The address pins are inputs only and do not output any internal address information during the operation. Address inputs are latched by the falling edge of $\overline{\text{CE}}$. If both MSBs are HIGH, the Address/Mode inputs are interpreted as Mode bits according to the Operational Mode table on page 12. There are six operational modes (M0...M6) available as indicated in the table. It is possible to use multiple operational modes simultaneously. Operational Modes are sampled on each falling edge of $\overline{\text{CE}}$, and thus Operational Modes and direct addressing are mutually exclusive.
AUX IN	11	18	**Auxiliary Input:** The Auxiliary Input is multiplexed through to the output amplifier and speaker output pins when $\overline{\text{CE}}$ is HIGH, P/$\overline{\text{R}}$ is HIGH, and playback is currently not active or if the device is in playback overflow. When cascading multiple ISD2500 devices, the AUX IN pin is used to connect a playback signal from a following device to the previous output speaker drivers. For noise considerations, it is suggested that the auxiliary input not be driven when the storage array is active.
V_{SSA}, V_{SSD}	13, 12	20, 19	**Ground**: The ISD2500 series of devices utilizes separate analog and digital ground busses. These pins should be connected separately through a low-impedance path to power supply ground.
SP+/SP-	14/15	21/22	**Speaker Outputs**: All devices in the ISD2500 series include an on-chip differential speaker driver, capable of driving 50 mW into 16 Ω from AUX IN (12.2mW from memory). [1] The speaker outputs are held at V_{SSA} levels during record and power down. It is therefore not possible to parallel speaker outputs of multiple ISD2500 devices or the outputs of other speaker drivers. [2] A single-end output may be used (including a coupling capacitor between the SP pin and the speaker). These outputs may be used individually with the output signal taken from either pin. However, the use of single-end output results in a 1 to 4 reduction in its output power.

[1] Connection of speaker outputs in parallel may cause damage to the device.
[2] Never ground or drive an unused speaker output.

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
V_{CCA}, V_{CCD}	16, 28	23, 7	**Supply Voltage**: To minimize noise, the analog and digital circuits in the ISD2500 series devices use separate power busses. These voltage busses are brought out to separate pins and should be tied together as close to the supply as possible. In addition, these supplies should be decoupled as close to the package as possible.
MIC	17	24	**Microphone**: The microphone pin transfers input signal to the on-chip preamplifier. A built-in Automatic Gain Control (AGC) circuit controls the gain of this preamplifier from –15 to 24dB. An external microphone should be AC coupled to this pin via a series capacitor. The capacitor value, together with the internal 10 KΩ resistance on this pin, determines the low-frequency cutoff for the ISD2500 series passband. See Winbond's Application Information for additional information on low-frequency cutoff calculation.
MIC REF	18	25	**Microphone Reference**: The MIC REF input is the inverting input to the microphone preamplifier. This provides a noise-canceling or common-mode rejection input to the device when connected to a differential microphone.
AGC	19	26	**Automatic Gain Control**: The AGC dynamically adjusts the gain of the preamplifier to compensate for the wide range of microphone input levels. The AGC allows the full range of whispers to loud sounds to be recorded with minimal distortion. The "attack" time is determined by the time constant of a 5 KΩ internal resistance and an external capacitor (C2 on the schematic of Figure 5 in section 11) connected from the AGC pin to V_{SSA} analog ground. The "release" time is determined by the time constant of an external resistor (R2) and an external capacitor (C2) connected in parallel between the AGC pin and V_{SSA} analog ground. Nominal values of 470 KΩ and 4.7 µF give satisfactory results in most cases.
ANA IN	20	27	**Analog Input**: The analog input transfers analog signal to the chip for recording. For microphone inputs, the ANA OUT pin should be connected via an external capacitor to the ANA IN pin. This capacitor value, together with the 3.0 KΩ input impedance of ANA IN, is selected to give additional cutoff at the low-frequency end of the voice passband. If the desired input is derived from a source other than a microphone, the signal can be fed, capacitively coupled, into the ANA IN pin directly.
ANA OUT	21	28	**Analog Output**: This pin provides the preamplifier output to the user. The voltage gain of the preamplifier is determined by the voltage level at the AGC pin.

PIN NAME	PIN NO.		FUNCTION
	SOIC/ PDIP	TSOP	
$\overline{OVF}$	22	1	**Overflow**: This signal pulses LOW at the end of memory array, indicating the device has been filled and the message has overflowed. The $\overline{OVF}$ output then follows the $\overline{CE}$ input until a PD pulse has reset the device. This pin can be used to cascade several ISD2500 devices together to increase record/playback durations.
$\overline{CE}$	23	2	**Chip Enable**: The $\overline{CE}$ input pin is taken LOW to enable all playback and record operations. The address pins and playback/record pin (P/$\overline{R}$) are latched by the falling edge of $\overline{CE}$. $\overline{CE}$ has additional functionality in the M6 (Push-Button) Operational Mode as described in the Operational Mode section.
PD	24	3	**Power Down**: When neither record nor playback operation, the PD pin should be pulled HIGH to place the part in standby mode (see I_{SB} specification). When overflow ($\overline{OVF}$) pulses LOW for an overflow condition, PD should be brought HIGH to reset the address pointer back to the beginning of the memory array. The PD pin has additional functionality in the M6 (Push-Button) Operation Mode as described in the Operational Mode section.
$\overline{EOM}$	25	4	**End-Of-Message**: A nonvolatile marker is automatically inserted at the end of each recorded message. It remains there until the message is recorded over. The $\overline{EOM}$ output pulses LOW for a period of T_{EOM} at the end of each message. In addition, the ISD2500 series has an internal V_{CC} detect circuit to maintain message integrity should V_{CC} fall below 3.5V. In this case, $\overline{EOM}$ goes LOW and the device is fixed in Playback-only mode. When the device is configured in Operational Mode M6 (Push-Button Mode), this pin provides an active-HIGH signal, indicating the device is currently recording or playing. This signal can conveniently drive an LED for visual indicator of a record or playback operation in process.

<table>
<tr><th rowspan="2">PIN NAME</th><th colspan="2">PIN NO.</th><th rowspan="2">FUNCTION</th></tr>
<tr><th>SOIC/ PDIP</th><th>TSOP</th></tr>
<tr><td>XCLK</td><td>26</td><td>5</td><td>External Clock: The external clock input has an internal pull-down device. The device is configured at the factory with an internal sampling clock frequency centered to ±1 percent of specification. The frequency is then maintained to a variation of ±2.25 percent over the entire commercial temperature and operating voltage ranges. If greater precision is required, the device can be clocked through the XCLK pin as follows:
<table>
<tr><th>Part Number</th><th>Sample Rate</th><th>Required Clock</th></tr>
<tr><td>ISD2560</td><td>8.0 kHz</td><td>1024 kHz</td></tr>
<tr><td>ISD2575</td><td>6.4 kHz</td><td>819.2 kHz</td></tr>
<tr><td>ISD2590</td><td>5.3 kHz</td><td>682.7 kHz</td></tr>
<tr><td>ISD25120</td><td>4.0 kHz</td><td>512 kHz</td></tr>
</table>
These recommended clock rates should not be varied because the antialiasing and smoothing filters are fixed, and aliasing problems can occur if the sample rate differs from the one recommended. The duty cycle on the input clock is not critical, as the clock is immediately divided by two. If the XCLK is not used, this input must be connected to ground.</td></tr>
<tr><td>P/$\overline{R}$</td><td>27</td><td>6</td><td>Playback/Record: The P/$\overline{R}$ input pin is latched by the falling edge of the $\overline{CE}$ pin. A HIGH level selects a playback cycle while a LOW level selects a record cycle. For a record cycle, the address pins provide the starting address and recording continues until PD or $\overline{CE}$ is pulled HIGH or an overflow is detected (i.e. the chip is full). When a record cycle is terminated by pulling PD or $\overline{CE}$ HIGH, then End-Of-Message ($\overline{EOM}$) marker is stored at the current address in memory. For a playback cycle, the address inputs provide the starting address and the device will play until an $\overline{EOM}$ marker is encountered. The device can continue to pass an $\overline{EOM}$ marker if $\overline{CE}$ is held LOW in address mode, or in an Operational Mode. (See Operational Modes section)</td></tr>
</table>

7. FUNCTIONAL DESCRIPTION

7.1. DETAILED DESCRIPTION

Speech/Sound Quality

The Winbond's ISD2500 series includes devices offered at 4.0, 5.3, 6.4, and 8.0 kHz sampling frequencies, allowing the user a choice of speech quality options. Increasing the duration within a product series decreases the sampling frequency and bandwidth, which affects the sound quality. Please refer to the ISD2560/75/90/120 Product Summary table below to compare the duration, sampling frequency and filter pass band.

The speech samples are stored directly into the on-chip nonvolatile memory without any digitization and compression associated like other solutions. Direct analog storage provides a very true, natural sounding reproduction of voice, music, tones, and sound effects not available with most solid state digital solutions.

Duration

To meet various system requirements, the ISD2560/75/90/120 products offer single-chip solutions at 60, 75, 90, and 120 seconds. Parts may also be cascaded together for longer durations.

TABLE 1: ISD2560/75/90/120 PRODUCT SUMMARY

Part Number	Duration (Seconds)	Input Sample Rate (kHz)	Typical Filter Pass Band * (kHz)
ISD2560	60	8.0	3.4
ISD2575	75	6.4	2.7
ISD2590	90	5.3	2.3
ISD25120	120	4.0	1.7

* 3db roll-off point

EEPROM Storage

One of the benefits of Winbond's ChipCorder® technology is the use of on-chip nonvolatile memory, providing zero-power message storage. The message is retained for up to 100 years typically without power. In addition, the device can be re-recorded typically over 100,000 times.

Microcontroller Interface

In addition to its simplicity and ease of use, the ISD2500 series includes all the interfaces necessary for microcontroller-driven applications. The address and control lines can be interfaced to a microcontroller and manipulated to perform a variety of tasks, including message assembly, message concatenation, predefined fixed message segmentation, and message management.

Programming

The ISD2500 series is also ideal for playback-only applications, where single or multiple messages are referenced through buttons, switches, or a microcontroller. Once the desired message configuration is created, duplicates can easily be generated via a gang programmer.

7.2. Operational Modes

The ISD2500 series is designed with several built-in Operational Modes that provide maximum functionality with minimum external components. These modes are described in details as below. The Operational Modes are accessed via the address pins and mapped beyond the normal message address range. When the two Most Significant Bits (MSB), A8 and A9, are HIGH, the remaining address signals are interpreted as mode bits and not as address bits. Therefore, Operational Modes and direct addressing are not compatible and cannot be used simultaneously.

There are two important considerations for using Operational Modes. First, all operations begin initially at address 0 of its memory. Later operations can begin at other address locations, depending on the Operational Mode(s) chosen. In addition, the address pointer is reset to 0 when the device is changed from record to playback, playback to record (except M6 mode), or when a Power-Down cycle is executed.

Second, Operational Modes are executed when $\overline{CE}$ goes LOW. This Operational Mode remains in effect until the next LOW-going $\overline{CE}$ signal, at which point the current mode(s) are sampled and executed.

TABLE 2: OPERATIONAL MODES

Mode [1]	Function	Typical Use	Jointly Compatible [2]
M0	Message cueing	Fast-forward through messages	M4, M5, M6
M1	Delete $\overline{EOM}$ markers	Position $\overline{EOM}$ marker at the end of the last message	M3, M4, M5, M6
M2	Not applicable	Reserved	N/A
M3	Looping	Continuous playback from Address 0	M1, M5, M6
M4	Consecutive addressing	Record/playback multiple consecutive messages	M0, M1, M5
M5	$\overline{CE}$ level-activated	Allows message pausing	M0, M1, M3, M4
M6	Push-button control	Simplified device interface	M0, M1, M3

[1] Besides mode pin needed to be "1", A8 and A9 pin are also required to be "1" in order to enter into the related operational mode.

[2] Indicates additional Operational Modes which can be used simultaneously with the given mode.

7.2.1. Operational Modes Description

The Operational Modes can be used in conjunction with a microcontroller, or they can be hardwired to provide the desired system operation.

M0 – Message Cueing

Message Cueing allows the user to skip through messages, without knowing the actual physical addresses of each message. Each $\overline{CE}$ LOW pulse causes the internal address pointer to skip to the next message. This mode is used for playback only, and is typically used with the M4 Operational Mode.

M1 – Delete $\overline{EOM}$ Markers

The M1 Operational Mode allows sequentially recorded messages to be combined into a single message with only one $\overline{EOM}$ marker set at the end of the final message. When this Operational Mode is configured, messages recorded sequentially are played back as one continuous message.

M2 – Unused

When Operational Modes are selected, the M2 pin should be LOW.

M3 – Message Looping

The M3 Operational Mode allows for the automatic, continuously repeated playback of the message located at the beginning of the address space. A message can completely fill the ISD2500 device and will loop from beginning to end without $\overline{OVF}$ going LOW.

M4 – Consecutive Addressing

During normal operation, the address pointer will reset when a message is played through an $\overline{EOM}$ marker. The M4 Operational Mode inhibits the address pointer reset on $\overline{EOM}$, allowing messages to be played back consecutively.

M5 - $\overline{CE}$-Level Activated

The default mode for ISD2500 devices is for $\overline{CE}$ to be edge-activated on playback and level-activated on record. The M5 Operational Mode causes the $\overline{CE}$ pin to be interpreted as level-activated as opposed to edge-activated during playback. This is especially useful for terminating playback operations using the $\overline{CE}$ signal. In this mode, $\overline{CE}$ LOW begins a playback cycle, at the beginning of the device memory. The playback cycle continues as long as $\overline{CE}$ is held LOW. When $\overline{CE}$ goes HIGH, playback will immediately end. A new $\overline{CE}$ LOW will restart the message from the beginning unless M4 is also HIGH.

M6 – Push-Button Mode

The ISD2500 series contain a Push-Button Operational Mode. The Push-Button Mode is used primarily in very low-cost applications and is designed to minimize external circuitry and components, thereby reducing system cost. In order to configure the device in Push-Button Operational Mode, the two most significant address bits must be HIGH, and the M6 mode pin must also be HIGH. A device in this mode always powers down at the end of each playback or record cycle after $\overline{CE}$ goes HIGH.

When this operational mode is implemented, three of the pins on the device have alternate functionality as described in the table below.

TABLE 3: ALTERNATE FUNCTIONALITY IN PINS

Pin Name	Alternate Functionality in Push-Button Mode
$\overline{CE}$	Start/Pause Push-Button (LOW pulse-activated)
PD	Stop/Reset Push-Button (HIGH pulse-activated)
$\overline{EOM}$	Active-HIGH Run Indicator

$\overline{CE}$ (START/PAUSE)

In Push-Button Operational Mode, $\overline{CE}$ acts as a LOW-going pulse-activated START/PAUSE signal. If no operation is currently in progress, a LOW-going pulse on this signal will initiate a playback or record cycle according to the level on the P/$\overline{R}$ pin. A subsequent pulse on the $\overline{CE}$ pin, before an $\overline{EOM}$ is reached in playback or an overflow condition occurs, will pause the current operation, and the address counter is not reset. Another $\overline{CE}$ pulse will cause the device to continue the operation from the place where it is paused.

PD (STOP/RESET)

In Push-Button Operational Mode, PD acts as a HIGH-going pulse-activated STOP/RESET signal. When a playback or record cycle is in progress and a HIGH-going pulse is observed on PD, the current cycle is terminated and the address pointer is reset to address 0, the beginning of the message space.

$\overline{EOM}$ (RUN)

In Push-Button Operational Mode, $\overline{EOM}$ becomes an active-HIGH RUN signal which can be used to drive an LED or other external device. It is HIGH whenever a record or playback operation is in progress.

Recording in Push-Button Mode

1. The PD pin should be LOW, usually using a pull-down resistor.

AN.No.1632-711E

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

Edition	Revision	Date
1	Original	April 1985
2	Completely revised	Jan. 1987

ENERAL

.1 General

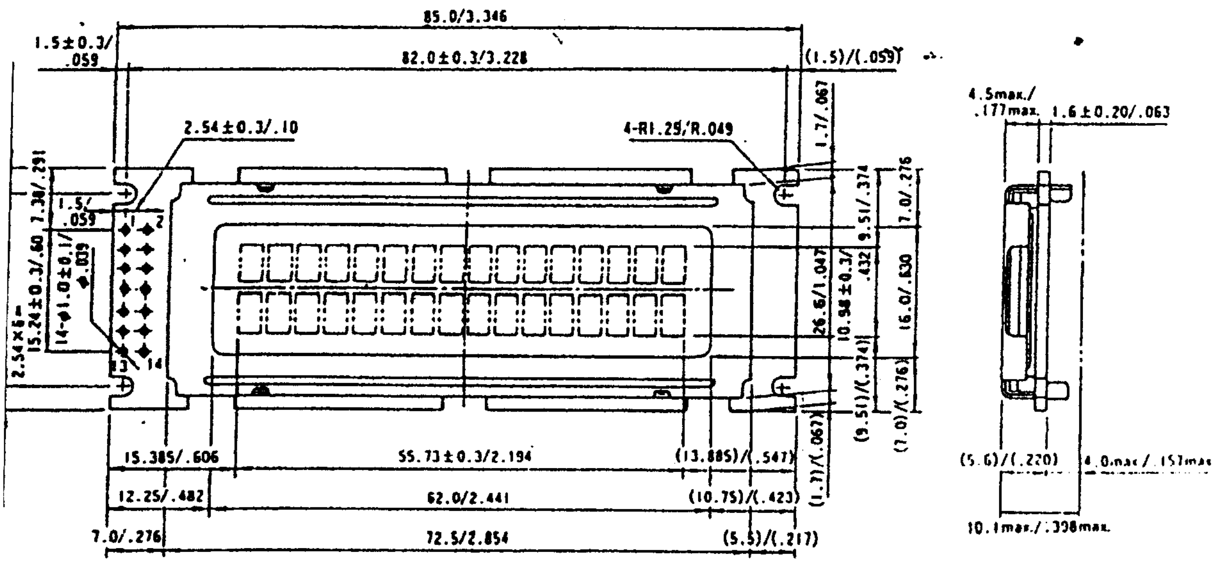
The M1632 is a low-power-consumption dot-matrix liquid crystal display (LCD) module with a high-contrast wide-view TN LCD panel and a CMOS LCD drive controller built in. The controller has a built-in character generator ROM/RAM, and display data RAM. All the display functions are controlled by instructions 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.

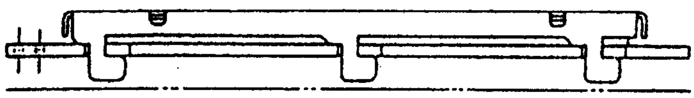
.2 Features

- 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

Dimensions Diagram





Unit : mm/inch

General tolerance : ±0.5 mm

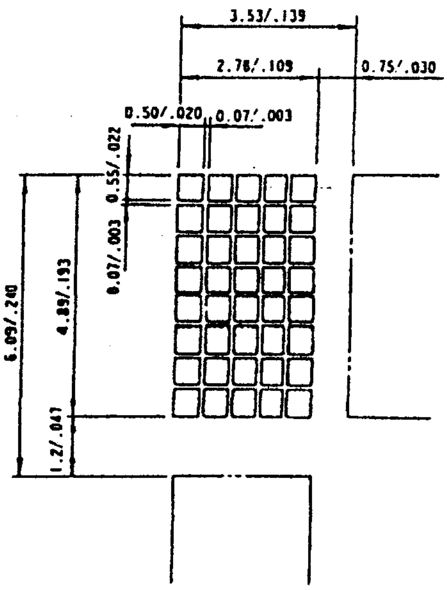
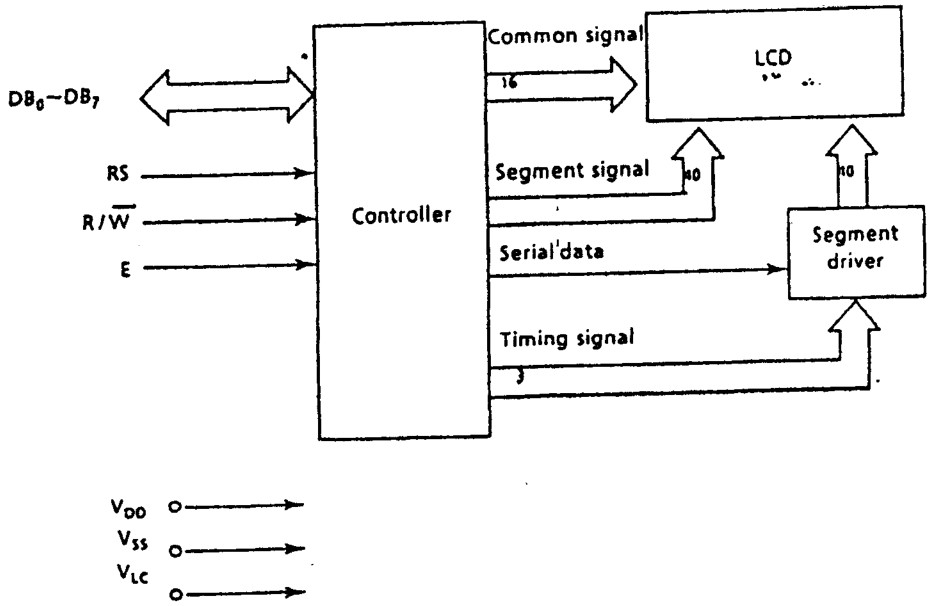


Figure 1 Dimensions diagram

No.	Symbol	Level	Function	
1	Vss	-	Power Supply	0V (GND)
2	Vcc	-		5V ±10%
3	Vee	-		for LCD Drive
4	RS	H/L	H: Data Input L: Instruction Input	
5	R/W	H/L	H:READ L:WRITE	
6	E	H, ⌊	Enable Signal	
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	-	Back Light Supply	4 - 4.2V 50-200mA
16	V- BL	-		0V (GND)

Block Diagram



Absolute Maximum Ratings

V_{SS} = 0 V

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

Electrical Characteristics

V_{DD} = 5 V ± 5%, V_{SS} = 0 V, T_A = 0°C to 50°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} = 5 V, T_A = 25°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

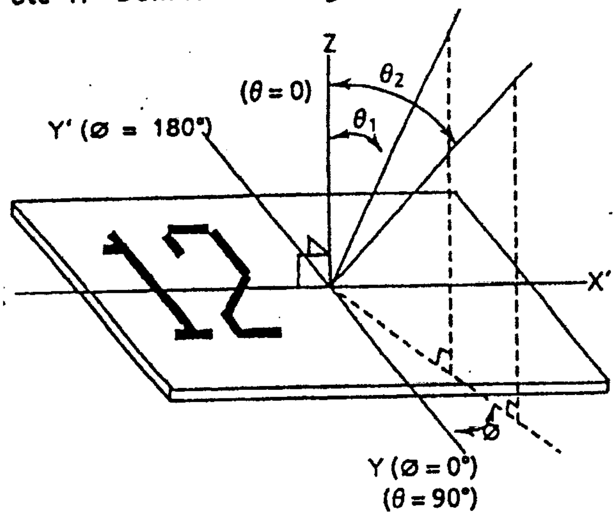
Optical Characteristics

l.7.1 Optical characteristics

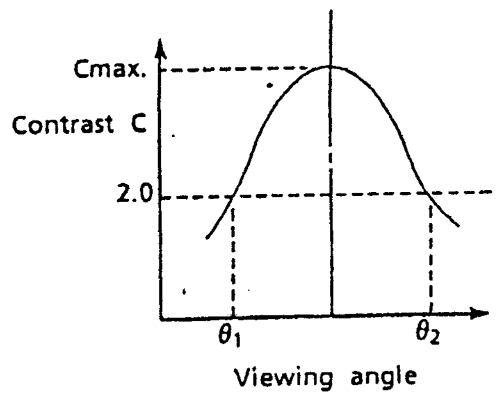
Maximum viewing angle: 6 o'clock (Ø = 0°)
T_A = 25°C, V_{opr} = 4.75 V

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

ote 1: Definition of angles Ø and θ

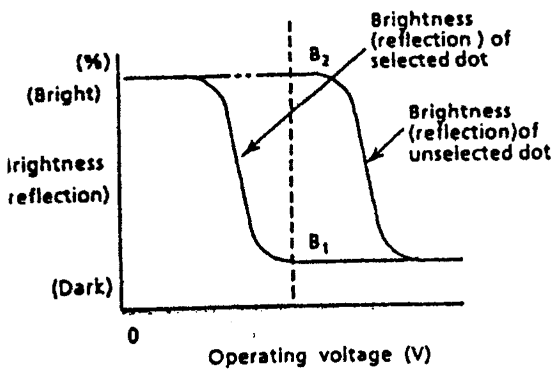


Note 2: Definition of viewing angles θ_1 and θ_2

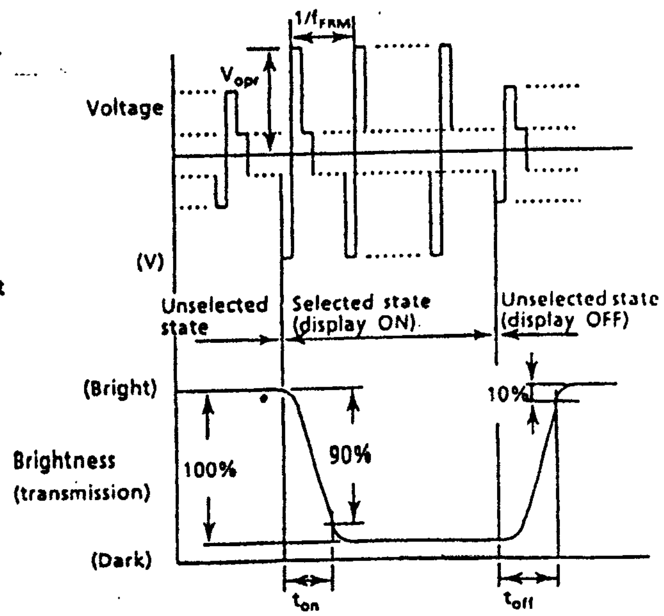


Note 3: Definition of contrast C

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



Note 4: Definition of response time



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

.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 (℃)	0	10	25	40	50
Voltage V_{opr} (V)	5.00	4.90	4.75	4.60	4.50

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

Basic Operations

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	$R/\overline{W}$	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 DB_7 if RS = 0 and $R/\overline{W}$ = 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.

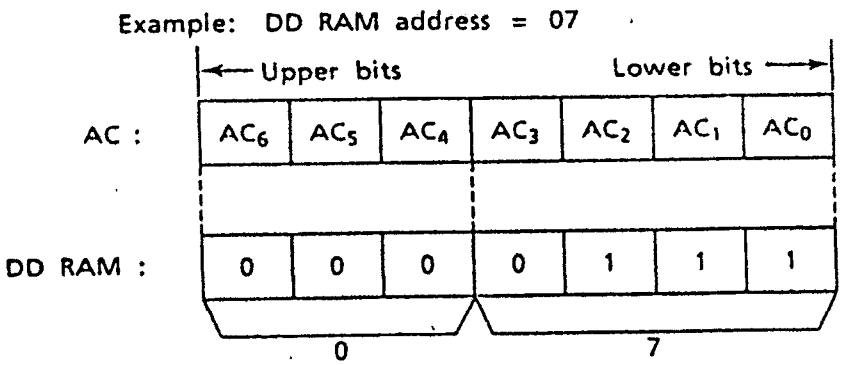
.2.3 Address counter (AC)

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 DB_0 to DB_6 as shown in Table 2 if RS = 0 and $R/\overline{W}$ = 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.



00_H to $0F_H$ of the DD RAM address is set in the line 1, and 40_H to $4F_H$ in the line 2.

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

Table 3 Correspondence between character codes and character patterns

r bit / 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
x0000	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
x0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
x0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
x0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
x0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
x0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
x0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
x0111	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
x x1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
x x1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
x x1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
x x1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
x x1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
x x1101	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
x x1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
x x1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

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, 00_H and 08_H select the same character.

ıres

- performance, Low-power AVR® 8-bit Microcontroller
- nced RISC Architecture
  - 31 Powerful Instructions – Most Single-clock Cycle Execution
  - 2 x 8 General Purpose Working Registers
  - ully Static Operation
  - Jp to 16 MIPS Throughput at 16 MHz
  -)n-chip 2-cycle Multiplier
- Endurance Non-volatile Memory segments
  - 16K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
  - Nrite/Erase Cycles: 10,000 Flash/100,000 EEPROM
  -)ata retention: 20 years at 85°C/100 years at 25°C
  -)ptional Boot Code Section with Independent Lock Bits
  - n-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- ; (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
-)heral 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 in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - 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
- cial 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
- ınd Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- rating Voltages
  - 2.7 - 5.5V for ATmega16L
  - 4.5 - 5.5V for ATmega16
- ed Grades
  - 0 - 8 MHz for ATmega16L
  - 0 - 16 MHz for ATmega16
- /er Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA

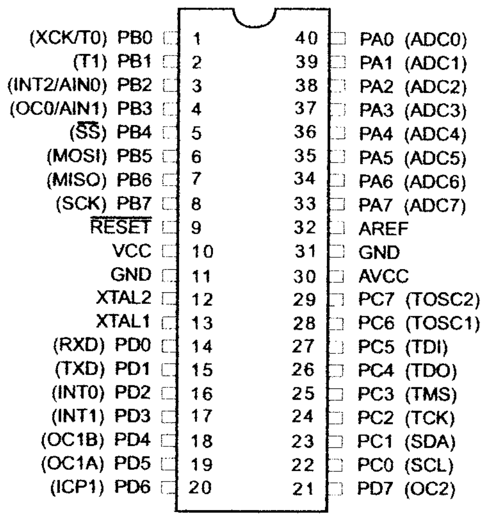
8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

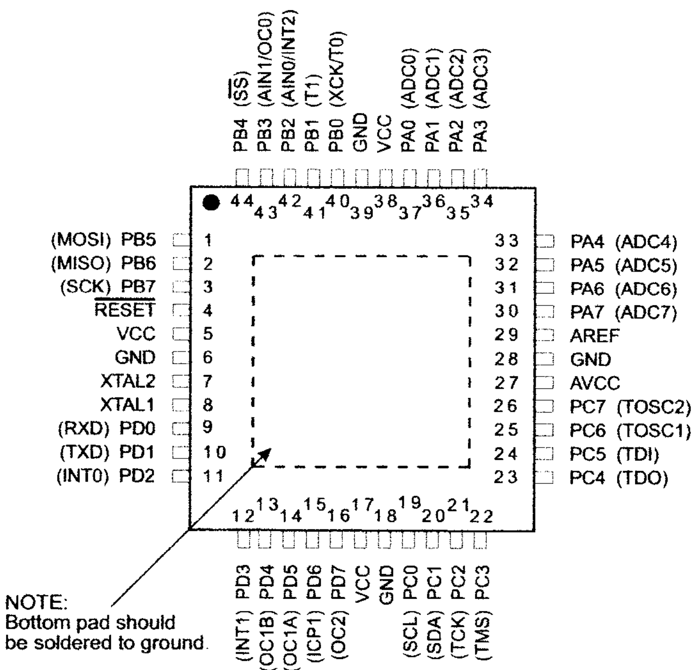
ATmega16
ATmega16L

Note: Not recommended for new designs.

'igurations

Figure 1. Pinout ATmega16





:laimer

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

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

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

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

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

)escriptions

Digital supply voltage.

Ground.

\ (PA7..PA0) Port A serves as the analog inputs to the A/D Converter.

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

(PB7..PB0)

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

Port B also serves the functions of various special features of the ATmega16 as listed on page 58.

(PC7..PC0)

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

Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 61.

) (PD7..PD0)

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

Port D also serves the functions of various special features of the ATmega16 as listed on page 63.

T

Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38. Shorter pulses are not guaranteed to generate a reset.

1

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

2

Output from the inverting Oscillator amplifier.

:

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

:

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

›urces

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

Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

ıt Code nples

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

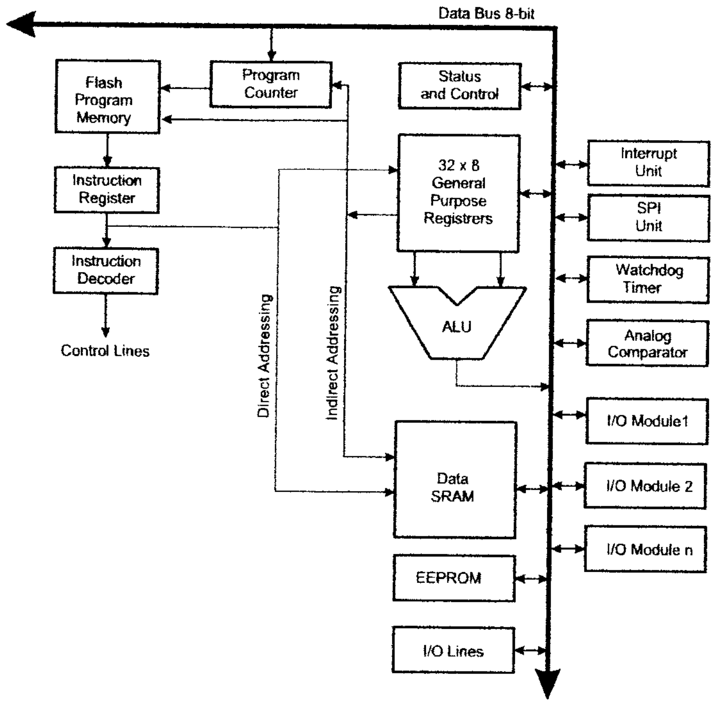
CPU Core

luction

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

tectural /iew

Figure 3. Block Diagram of the AVR MCU Architecture



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

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

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

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

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

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

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

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

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

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

– Arithmetic : Unit

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

ıs Register

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

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

The AVR Status Register – SREG – is defined as:

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

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

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

• **Bit 6 – T: Bit Copy Storage**

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

• **Bit 5 – H: Half Carry Flag**

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

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

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

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

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

• **Bit 2 – N: Negative Flag**

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

• **Bit 1 – Z: Zero Flag**

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

• **Bit 0 – C: Carry Flag**

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

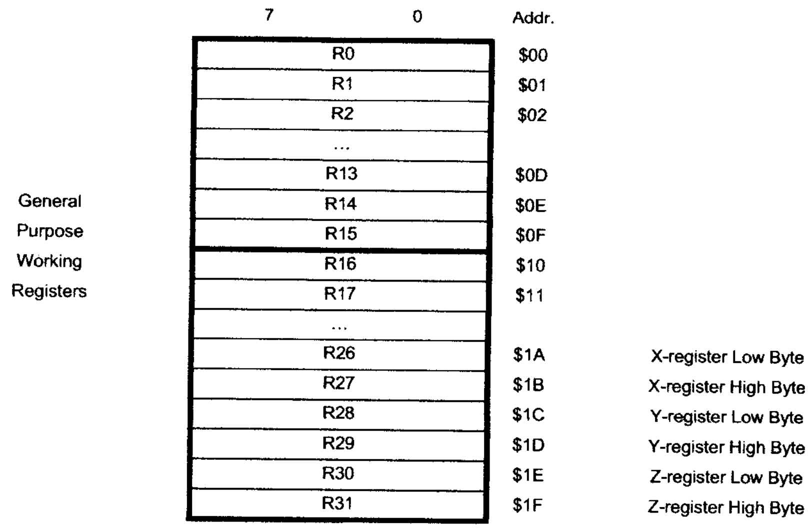
'al Purpose ter File

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

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

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

Figure 4. AVR CPU General Purpose Working Registers



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

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

egister, Y- r and Z-register

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

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

	15	XH		XL	0
X - register	7		0	7	0
	R27 ($1B)			R26 ($1A)	

	15	YH		YL	0
Y - register	7		0	7	0
	R29 ($1D)			R28 ($1C)	

	15	ZH		ZL	0
Z - register	7	0		7	0
	R31 ($1F)			R30 ($1E)	

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

Pointer

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

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

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

Bit	15	14	13	12	11	10	9	8	
	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SPH
	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the General Interrupt Control Register (GICR). Refer to "Interrupts" on page 45 for more information. The Reset Vector can also be moved to the start of the boot Flash section by programming the BOOTRST Fuse, see "Boot Loader Support – Read-While-Write Self-Programming" on page 246.

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

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

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

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

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

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

Assembly Code Example

```
in r16, SREG        ; store SREG value
cli     ; disable interrupts during timed sequence
sbi EECR, EEMWE     ; start EEPROM write
sbi EECR, EEWE
out SREG, r16       ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG;  /* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMWE); /* start EEPROM write */
EECR |= (1<<EEWE);
SREG = cSREG; /* restore SREG value (I-bit) */
```